Implementation of an Implicit-Explicit Scheme for Hybridizable Discontinuous Galerkin

by

Lauren Nicole Kolkman

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Signature redacted

Department of Aeronautics and Astronautics May 24, 2018

Signature redacted

Certified by .................. .................. Jaime Peraire
H.N. Slater Professor of Aeronautics and Astronautics Thesis Supervisor

Signature redacted

Accepted by .......................... Hamsa Balakrishnan
Associate Professor of Aeronautics and Astronautics Chair, Graduate Program Committee
Implementation of an Implicit-Explicit Scheme for
Hybridizable Discontinuous Galerkin

by

Lauren Nicole Kolkman

Submitted to the Department of Aeronautics and Astronautics
on May 25, 2018, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Finite element methods, specifically Hybridizable Discontinuous Galerkin (HDG), are used in many applications. One choice made when implementing HDG for a specific problem is whether time integration should be performed implicitly or explicitly. Both approaches have their advantages but, for some problems, a combination of these methods is a better choice than either on their own. Thus, an implicit-explicit (IMEX) scheme that splits the computational domain into implicit and explicit regions based on the domain geometry is considered in this thesis. This allows for stability throughout the domain and exploits the advantages each scheme has to offer. A study of the convergence and properties of this implementation of the IMEX method is presented along with comparisons to the individual methods.

Thesis Supervisor: Jaime Peraire
Title: H.N. Slater Professor of Aeronautics and Astronautics
Acknowledgments

First of all, I would like to thank my advisor Jaime Peraire for his guidance these past two years. I have learned a lot at MIT and appreciate the opportunity to do so.

Thank you to Ngoc-Cuong Nguyen. I am grateful for all the time you spent teaching me about various topics in this field.

Thank you to Sebastien Terrana. Without you I would never have finished my thesis. Thanks for helping me finish this project and for all the guidance you gave me while writing.

To my labmates in the ACDL, thank you. You have made my time so enjoyable and provided me with an amazing amount of academic assistance.

Thanks to everyone I have met at Sidney Pacific. You all are an incredibly supportive community and I had the best time living and working there.

Thank you to Colin Graber. You have given me a huge amount of support over the years and have helped me to make it through all my academic endeavors.

Finally, thank you to all my friends and family who have supported me over the years. Your help is much appreciated.
Contents

1 Introduction ............................................ 13

2 Mathematical Formulation .......................... 17
   2.1 Governing equations .................................. 17
   2.2 Triangulation ........................................ 18
   2.3 Finite Element Spaces ................................ 19
   2.4 HDG Space Discretization ............................ 20
   2.5 Postprocessing ....................................... 21
   2.6 Time stepping methods ............................... 22

3 Implementation ......................................... 27
   3.1 Square Domain ....................................... 27
      3.1.1 Problem Definition ............................... 27
      3.1.2 Choice of IMEX Scheme .......................... 28
      3.1.3 Mesh ............................................ 28
      3.1.4 Solution ......................................... 29
      3.1.5 Spatial Convergence .............................. 31
      3.1.6 Temporal Convergence ............................ 37
   3.2 L-shaped Mesh ....................................... 40
      3.2.1 Problem Definition ............................... 40
      3.2.2 Choice of IMEX Scheme .......................... 41
      3.2.3 Mesh ............................................ 41
      3.2.4 Implicit-Explicit Regions ........................ 41
3.2.5 Example Solution .......................... 42
3.2.6 Temporal Convergence ................... 44

4 Conclusions ................................. 47

A Butcher Tables ............................. 49
List of Figures

2-1 IMEX mesh description depicting implicit and explicit quantities in the volume as well as on the trace. ............................................ 23

3-1 Full, unrefined mesh for the square domain. ............................... 29
3-2 Solution of the velocity over the explicit domain for \( k = 2 \) and no refinements. ............................................................... 30
3-3 Solution of the velocity over the implicit domain for \( k = 2 \) and no refinements. ............................................................... 30
3-4 Full mesh with implicit and explicit regions. ................................. 31
3-5 Full mesh with one level of refinement. ................................... 32
3-6 Full mesh with two levels of refinement. ................................... 33
3-7 Convergence of position \( u \) for polynomial orders 1 through 4. ....... 34
3-8 Convergence of velocity \( v \) for polynomial orders 1 through 4. ........ 35
3-9 Convergence of gradient \( q \) for polynomial orders 1 through 4. ....... 35
3-10 Convergence of postprocessed position \( u^* \). .............................. 36
3-11 Convergence of postprocessed velocity \( v^* \). .............................. 36
3-12 Change in error of \( u \) for decreasing timesteps on the square domain. 37
3-13 Change in error of \( v \) for decreasing timesteps on the square domain. 37
3-14 Change in error of \( q \) for decreasing timesteps on the square domain. 38
3-15 Change in error of \( u \) for decreasing timesteps on the square domain for the second parameter set. ............................................. 39
3-16 Change in error of \( v \) for decreasing timesteps on the square domain for the second parameter set. ............................................. 39
3-17 Change in error of $q$ for decreasing timesteps on the square domain for
the second parameter set. .................................................. 39
3-18 L-shaped domain base mesh. ............................................. 41
3-19 Example of implicit and explicit regions in the L-shaped domain. .... 42
3-20 Solution in the implicit region of the L-shaped mesh. .................. 43
3-21 Solution in the explicit region of the L-shaped mesh. .................. 43
3-22 Change in error of $u$ for decreasing timesteps on the L-shaped domain. 44
3-23 Change in error of $v$ for decreasing timesteps on the L-shaped domain. 44
# List of Tables

3.1 Given parameters for IMEX solution on the square domain. .......................... 29

3.2 Errors and estimated orders of convergence (e.o.c) for the IMEX test case. ............................................. 34

3.3 Errors and estimated orders of convergence (e.o.c) for the square domain time convergence test. The parameters used are polynomials of order 4, temporal discretization of order 2, no mesh refinements, and $K_{\text{max}} = 0.5$. ................................................................. 38

3.4 Errors and estimated orders of convergence (e.o.c) for the square domain time convergence test. The parameters used are polynomials of order 5, temporal discretization of order 3, no mesh refinements, and $K_{\text{max}} = 1$. ................................................................. 39

3.5 Given parameters for IMEX solution on the L-shaped domain. .................. 42

3.6 Errors and estimated orders of convergence (e.o.c) for the L-domain time convergence test. The parameters used are polynomials of order 2, temporal discretization of order 3, 3 levels of mesh refinements, and $K_{\text{max}} = 0.5$. ................................................................. 45

3.7 Computational time needed for implicit only, IMEX, and explicit only schemes. ......................................................... 45

3.8 Computational time needed for implicit only, IMEX, and explicit only schemes for one level of mesh refinement. ......................... 46
Chapter 1

Introduction

In the field of engineering, partial differential equations (PDE) are regularly used to model physical phenomena of interest. Analytical solutions may exist to these equations, but in many cases finding these solutions is tedious, time intensive, and, in some cases, impossible. In addition, performing physical experiments and creating prototypes to understand the behavior of a system—which is common in fields such as aerodynamics—is both monetarily costly and time consuming. The finite element method (FEM) was introduced to combat these problems by providing a way of computing solutions numerically through spatial discretization.

FEM breaks down the given domain of a problem into small pieces or elements. The combination of all the elements creates the mesh or triangulation. The solution is then computed at discrete locations within this triangulation and assembled to create a full numerical solution. To perform a FEM computation, the PDE must be changed from its given form to one compatible with the discrete triangulation, which is known as the weak formulation. The weak formulation of the PDE is found by multiplying the equation with a test function and integrating by parts. The equation can then be discretized and written in terms of basis functions and their coefficients.

One particular type of FEM is Discontinuous Galerkin (DG). This method uses discontinuous basis functions and test functions. For hyperbolic problems, it converges at a rate of $k + \frac{1}{2}$ in general or $k + 1$ for most regular meshes where $k$ is the polynomial order of the numerical approximation. An advantage of this method
is that it creates sparse matrix systems due to each element only exchanging information with those bordering it. Additionally, the user has the ability to change the polynomial order in each element independently without affecting the other elements and therefore, one is able to solve complex problems efficiently by concentrating high order polynomials in those regions of the domain where the solution is complex.\textsuperscript{9}

The DG method also comes with its disadvantages. One important concern is the number of degrees of freedom needed to solve the system. Neighboring elements have duplicate nodes at the boundary between them. Trouble also arises from time integration, which is a pervasive issue in FEM. Explicit computations cause the timestep size to be dictated by the smallest element in the triangulation, which can result in a very small timestep that is only necessary for a few elements. Implicit integration can result in a very large number of globally coupled degrees of freedom.\textsuperscript{6}

An alternative to DG that helps alleviate the problem of this large global coupling is the Hybridizable Discontinuous Galerkin (HDG) method. HDG significantly reduces the number of degrees of freedom, thus lowering computational cost. A unique property that the HDG method possesses is the guaranteed optimal convergence $(k + 1)$ of the gradient for elliptic problems, while most DG methods ensure only a convergence of order $k$ at most for the gradient. In addition, with optimal convergence comes the ability to post process the numerical solution to obtain superconvergence $(k + 2)$. This can be performed cheaply after the solution is computed to gain an extra spatial order of convergence.\textsuperscript{5}

Time integration in the HDG method can be performed either implicitly or explicitly; both approaches have their advantages and disadvantages. Explicit methods are simplistic in their implementation and require a cheaper cost of computation for each timestep when compared to implicit methods.\textsuperscript{4} This arises from the lack of a need to construct and store large matrices\textsuperscript{11} which causes these methods to be very efficient.\textsuperscript{12} Additionally, no large linear systems need to be inverted. When implemented for HDG, explicit methods will retain the desirable superconvergence property.\textsuperscript{10} However, explicit methods are only conditionally stable. This results in the need for very small timesteps, which can result in an exorbitant number of steps and
a long run time,
particularly when small or distorted elements are used. Conversely, implicit methods can be designed to allow for arbitrarily large timesteps for many applications. This is the case for the wave equation, which exhibits unconditional stability. The tradeoff is that implicit methods store large amounts of data during the computation, causing them to take more time per step.

Because both approaches of time integration have their advantages in certain cases, it can be desirable to combine them into an implicit-explicit (IMEX) method. The fundamental concept of the IMEX method is the splitting of the computational domain into two separate pieces: stiff and nonstiff. This division has flexibility in how it is determined. One example is splitting based on the linearity of the particular problem. Linear terms are computed implicitly while nonlinear terms are computed explicitly. Another way to split the computation is based on the element sizes rather than the equation being solved. Smaller elements, which are stiff, are computed implicitly while large elements, which are nonstiff, are computed explicitly.

It is this spatial splitting that is considered in this thesis. An IMEX HDG implementation is considered for the wave equation. It utilizes the Courant-Friedrichs-Lewy (CFL) condition to determine which elements are explicitly computed by Explicit Runge-Kutta (ERK) or implicitly computed by Diagonally Implicit Runge-Kutta (DIRK). Properties of this method such as the convergence rates of the spatial and temporal discretizations are examined.
Chapter 2

Mathematical Formulation

2.1 Governing equations

The basis of this analysis is the acoustic wave equation. This partial differential equation (PDE) can be written as

\[ \rho \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (A \nabla u) = f, \quad \text{in } \Omega \times (0, T], \]

(2.1)

where \( \Omega \in \mathbb{R}^d \) is the domain of dimension \( d \), \( T \) is the final time, \( f \) is the source term, \( \rho \) is the density, \( A \) is a matrix-valued coefficient, and \( u \) is the scalar displacement.

This second order equation can then be broken down into two first order equations through the introduction of new variables: the velocity \( v \) and the gradient \( q \). These are defined as

\[ v = \frac{\partial u}{\partial t}, \quad \text{in } \Omega \times (0, T], \]

(2.2)

\[ q = \nabla u. \]

(2.3)
Using (2.2) and (2.3), equation (2.1) becomes

\[
\frac{\partial q}{\partial t} - \nabla v = 0, \quad \text{in } \Omega \times (0, T],
\]

\[
\rho \frac{\partial v}{\partial t} - \nabla \cdot \mathbf{A}q = f, \quad \text{in } \Omega \times (0, T].
\]

(2.4)

The following Dirichlet and Neumann boundary conditions are imposed

\[
v = g_D, \quad \text{on } \partial \Omega_D \times (0, T],
\]

\[
\mathbf{A}q \cdot \mathbf{n} = g_N, \quad \text{on } \partial \Omega_N \times (0, T],
\]

(2.5)

where the boundary \( \partial \Omega \) is composed of both the Dirichlet boundary, \( \Omega_D \), and the Neumann boundary, \( \Omega_N \).

\[
\partial \Omega = \partial \Omega_D \cup \partial \Omega_N
\]

(2.6)

Additionally, two initial conditions need to be specified

\[
v(x, t = 0) = v_0(x) \text{ on } \Omega,
\]

\[
q(x, t = 0) = q_0(x) \text{ on } \Omega.
\]

(2.7)

2.2 Triangulation

The domain \( \Omega \) can be discretized with a triangulation denoted \( \mathcal{T}_h \). The triangulation is composed of elements with an individual element denoted as \( K \). The boundary of the triangulation is denoted \( \partial \mathcal{T}_h \) which consists of the collection of element faces on the boundary, \( \partial K \).

Within the triangulation, there are three sets of faces. \( \mathcal{E}^0 \) are the faces in the interior, \( \mathcal{E}^\partial \) are the faces on the boundary, and \( \mathcal{E} \) is the set of all the faces. A single face is denoted \( F \).

Additionally, the mesh is broken into both an explicit and an implicit subdomain. An element is determined to be implicit if it meets the following Courant-Friedrichs-Lewy (CFL) condition

\[
h_{\text{min}} < \frac{\Delta t \cdot k^2 c_{\text{max}}}{K_{\text{max}}},
\]

(2.8)
where $h_{min}$ is the smallest edge of an element, $\Delta t$ is the time step, $c_{max}$ is the maximum wave speed, $K_{max}$ is a constant that depends on the time integration scheme, and $k$ is the polynomial order. The triangulation can then be represented as an explicit portion $\mathcal{T}_h^{ex}$ and an implicit portion $\mathcal{T}_h^{im}$ where

$$\mathcal{T}_h = \mathcal{T}_h^{ex} \cup \mathcal{T}_h^{im}. \quad (2.9)$$

Additionally, the boundary between the implicit and the explicit triangulations can be defined as

$$\Gamma_h^{imex} = \partial \mathcal{T}_h^{ex} \cap \partial \mathcal{T}_h^{im}. \quad (2.10)$$

### 2.3 Finite Element Spaces

Several discontinuous finite element spaces are defined:

$$W_h = \left\{ w \in L^2(\Omega) : w|_K \in W(K), \forall K \in \mathcal{T}_h \right\},$$

$$V_h = \left\{ p \in (L^2(\Omega))^d : p|_K \in V(K), \forall K \in \mathcal{T}_h \right\},$$

where the dimension $d$ here is 2. For the local space, we use polynomials of degree $k$ over the element $K$, $\mathcal{P}_k(K)$, defined as

$$W(K) \times V(K) \equiv \mathcal{P}_k(K) \times (\mathcal{P}_k(K))^d.$$ 

Additionally, a space for the trace is introduced:

$$M_h = \left\{ \mu \in L^2(\mathcal{E}_h) : \mu|_F \in \mathcal{P}_k(F), \forall F \in \mathcal{E}_h \right\}.$$
For the spaces above, the following inner products are defined for functions.

\[
(w, v)_D = \int_D wv, \quad \text{for } w, v \text{ in } L^2(D),
\]

\[
(w, v)_D = \int_D w \cdot v, \quad \text{for } w, v \text{ in } (L^2(D))^d,
\]

\[
\langle \mu, \eta \rangle_F = \int_F \mu \eta, \quad \text{for } \mu, \eta \text{ in } L^2(F).
\]

These inner products can be applied over each element and summed to create the inner product over the entire triangulation:

\[
(w, v)_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} (w, v)_K,
\]

\[
(w, v)_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} (w, v)_K,
\]

\[
\langle \mu, \eta \rangle_{\partial \mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \langle \mu, \eta \rangle_{\partial K}.
\]

### 2.4 HDG Space Discretization

To convert from the strong form presented above to the weak form used for finite element analysis, equations (2.4) are multiplied by test functions \( r \in V(K) \) and \( w \in W(K) \), respectively, and subsequently integrated by parts. This results in the following two equations.

For any element \( K \in \mathcal{T}_h \)

\[
\left( \frac{\partial q}{\partial t}, r \right)_K + (v_h, \nabla \cdot r)_K - \langle \tilde{v}_h, r \cdot \mathbf{n} \rangle_{\partial K} = 0, \quad \forall r \in V(K),
\]

\[
\left( \rho \frac{\partial v_h}{\partial t}, w \right)_K + (Aq_h, \nabla w)_K - \langle A\tilde{q}_h \cdot \mathbf{n}, w \rangle_{\partial K} = (f, w)_K, \quad \forall w \in W(K).
\]

The Hybridizable Discontinuous Galerkin equations are now defined over the entire...
domain. The quantities \((v_h, q_h, \hat{v}_h) \in W_h \times V_h \times M_h\) are searched for such that

\[
\begin{align*}
\left( \frac{\partial q}{\partial t}, r \right)_T + (v_h, \nabla \cdot r)_T - \langle \hat{v}_h, r \cdot n \rangle_{\partial T} &= 0, \\
\langle \rho \frac{\partial v_h}{\partial t}, w \rangle_T + (Aq_h, \nabla w)_T - \langle A\hat{q}_h \cdot n, w \rangle_{\partial T} &= (f, w)_T, \\
\langle A\hat{q}_h \cdot n, \mu \rangle_{\partial T} &= \langle g_N, \mu \rangle_{\partial \Omega_N}, \quad \forall \mu \in M_h.
\end{align*}
\] (2.12)

Finally, \(\hat{q}_h\) is defined as

\[
A\hat{q}_h = Aq_h - \tau(v_h - \hat{v}_h)n, \quad \text{on } \partial T_h. 
\] (2.13)

### 2.5 Postprocessing

It is an HDG feature that an additional order of accuracy—superconvergence—can be gained for the velocity and position fields in the case of optimal convergence of the gradient \(q\). This procedure is performed after the solution is computed. The postprocessed solution can be computed independently at any chosen time step. Solving for the new solution, \(u^*_h \in P_{k+1}(K)\), involves solving the local system

\[
\begin{align*}
(\nabla u^*_h, \nabla w)_K &= (q_h, \nabla w)_K, \\
(u^*_h, 1)_K &= (u_h, 1)_K.
\end{align*}
\] (2.14)

This computation uses the gradient \(q_h\) resulting from the global solve. To solve for the velocity field however, the gradient of the velocity, \(p_h\), needs to be computed in an additional step. This is found using the following equation:

\[
(\nabla p_h, r)_K = -(v_h, \nabla \cdot r)_K + \langle \hat{v}_h, r \cdot n \rangle_{\partial K}, \quad \forall r \in V(K). 
\] (2.15)
The new velocity, $v_h^* \in P_{k+1}(K)$, can then be computed with the system

\[
(\nabla v_h^*, \nabla w)_K = (p_h, \nabla w)_K, \quad \forall w \in P_{k+1}(K),
\]

\[
(v_h^*, 1)_K = (v_h, 1)_K.
\]  

(2.16)

### 2.6 Time stepping methods

To implement the IMEX time integration scheme, the spatial HDG method must be defined for the implicit and explicit regions. Over these two domains the following quantities of interest are defined

\[
v_h^{im} = v_h|_{T_h^{im}}; \quad q_h^{im} = q_h|_{T_h^{im}}; \quad \dot{v}_h^{im} = \dot{v}_h|_{\partial T_h^{im}},
\]

\[
v_h^{ex} = v_h|_{T_h^{ex}}; \quad q_h^{ex} = q_h|_{T_h^{ex}}; \quad \dot{v}_h^{ex} = \dot{v}_h|_{\partial T_h^{ex}\setminus \Gamma_h^{imex}}; \quad \dot{v}_h^{ex}|_{\Gamma_h^{imex}} = \dot{v}_h^{im}|_{\Gamma_h^{imex}}.
\]

Because the mesh is broken into two different regions, a boundary is shared between the explicit and implicit elements. To appropriately define the variables within the HDG method, the quantity of $\dot{v}_h$ on the IMEX boundary, $\Gamma_h^{imex}$, is considered to belong to the implicit triangulation, $T_h^{im}$. This can be seen in Figure 2-1.

Based on these definitions, problem (2.12) can be rewritten for the explicit domain as:

find $(q_h, v_h, \dot{v}_h) \in V_h \times W_h \times M_h$ such that

\[
\left( \frac{\partial q_h^{ex}}{\partial t}, r \right)_{T_h^{ex}} + (v_h^{ex}, \nabla \cdot r)_{T_h^{ex}} - (\dot{v}_h^{ex}, r \cdot n)_{\partial T_h^{ex}\setminus \Gamma_h^{imex}} = (\dot{v}_h^{im}, r \cdot n)_{\Gamma_h^{imex}},
\]

\[
\left( \rho \frac{\partial v_h^{ex}}{\partial t}, w \right)_{T_h^{ex}} - (Aq_h^{ex}, \nabla w)_{T_h^{ex}} - (A\dot{q}_h^{ex}, w)_{\partial T_h^{ex}} = (f, w)_{T_h^{ex}},
\]

\[
(A\dot{q}_h^{ex}, n)_{\partial T_h^{ex}} = (g_N, \mu)_{\partial \Omega_{N_{ex}}},
\]  

for all $(r, w, \mu) \in V(K) \times W(K) \times M(K)$, for all $K$ in $T_h^{ex}$.

(2.17)

Problem (2.12) can be written for the implicit domain as:
Figure 2-1: IMEX mesh description depicting implicit and explicit quantities in the volume as well as on the trace.

find \((q_h, v_h, \dot{v}_h) \in V_h \times W_h \times M_h\) such that

\[
\left( \frac{\partial q_h^{im}}{\partial t}, r \right)_{\Gamma^{im}_h} + (v_h^{im}, \nabla \cdot r)_{\Gamma^{im}_h} - (\dot{v}_h^{im}, r \cdot n)_{\partial \Gamma^{im}_h} = 0, \\
\left( \rho \frac{\partial v_h^{im}}{\partial t}, w \right)_{\Gamma^{im}_h} + (Aq_h^{im}, \nabla w)_{\Gamma^{im}_h} - (A\dot{q}_h^{im} \cdot n, w)_{\partial \Gamma^{im}_h} = (f, w)_{\Gamma^{im}_h}, \\
(A\dot{q}_h^{im} \cdot n, \mu)_{\partial \Gamma^{im}_h} = (Aq_h^{ex} \cdot n, \mu)_{\Gamma^{imex}_h} + (g_N, \mu)_{\partial \Omega_N \cap \partial \Gamma^{im}_h},
\]

for all \((r, w, \mu) \in V(K) \times W(K) \times M(K)\), for all \(K\) in \(\mathcal{T}^{im}_h\).

The problems (2.17) and (2.18) can be solved independently if \(\dot{v}_h^{im}\), or \(A\dot{q}_h^{ex} \cdot n\) respectively, is given. Communication between the two regions occurs only at the boundary \(\Gamma^{imex}\). The explicit domain receives a Dirichlet boundary condition from the implicit domain and the implicit domain receives a Neumann boundary condition from the explicit domain. Based on this observation, an explicit HDG solver may be used for problem (2.17) while an implicit one may be used for problem (2.18), provided
that \( \mathbf{A} \hat{\mathbf{q}}_{\text{ex}} \cdot \mathbf{n} \) and \( \hat{\mathbf{v}}_{\text{h}}^{\text{im}} \) are computed at consistent time stages. This process will be the basis of the IMEX formulation proposed in this thesis.

After discretization, problem (2.12) leads to a set of ordinary differential equations (ODE) that have to be integrated in time. The vector of nodal values for \( u_h \) and \( \mathbf{q}_h \) is denoted \( \mathbf{U} \). Then, problem (2.12) leads to the formulation

\[
\frac{d\mathbf{U}}{dt} = F(\mathbf{U}),
\]

where \( F(\mathbf{U}) \) is the residual over the entire computational domain.

The fundamental concept of the IMEX scheme is the splitting of the residual into two parts: a stiff part of the residual, \( g(\mathbf{U}) \), which is integrated implicitly, and a nonstiff part, \( f(\mathbf{U}) \), which is integrated explicitly

\[
\frac{d\mathbf{U}}{dt} = f(\mathbf{U}) + g(\mathbf{U}).
\]

In this specific IMEX implementation, the separation of the residual is based on the mesh size. The implicit residual is calculated over the implicit region, which corresponds to the smallest elements, and the explicit residual is calculated over the explicit region corresponding to the largest elements. On these regions are defined the implicit and explicit solutions \( \mathbf{U}_{\text{im}} \) and \( \mathbf{U}_{\text{ex}} \) such that

\[
\mathbf{U} = (\mathbf{U}_{\text{ex}}, \mathbf{U}_{\text{im}}),
\]

where \( \mathbf{U}_{\text{ex}} \) corresponds to the nodal values of \( v_h^{\text{ex}} \) and \( \mathbf{q}_h^{\text{ex}} \) and \( \mathbf{U}_{\text{im}} \) corresponds to the nodal values of \( v_h^{\text{im}} \) and \( \mathbf{q}_h^{\text{im}} \). Utilizing this formulation, problems (2.17) and (2.18), after spatial discretization, can be rewritten as the equations

\[
\begin{align*}
\frac{d\mathbf{U}_{\text{ex}}}{dt} &= f(\mathbf{U}_{\text{ex}}, \hat{\mathbf{Q}}_{\text{im}}), \\
\frac{d\mathbf{U}_{\text{im}}}{dt} &= g(\mathbf{U}_{\text{im}}, \hat{\mathbf{Q}}_{\text{ex}}).
\end{align*}
\]

24
where \( \hat{V}^{im} \) and \( \hat{Q}^{ex} \) are the vectors of the nodal values of \( \hat{v}^{im}_h \) on the IMEX boundary and \( \hat{q}^{ex}_h \) on the IMEX boundary.

With this definition established in equation (2.22), the following process for the IMEX method is demonstrated.

\[
\hat{k}_1 = f(U^{ex}_{n,1}, \hat{V}^{im}_{n,1})
\]

for \( i = 1 \) to \( s \)

Solve \( U^{ex}_{n,i} = U^{ex}_{n} + \Delta t \sum_{j=1}^{i} \hat{a}_{i+1,j} \hat{k}_j \)

Solve for \( k_i \) in \( k_i = g(U^{im}_{n,i}, \hat{Q}^{ex}_{n,i}) \), where \( U^{im}_{n,i} = U^{im}_{n} + \Delta t \sum_{j=1}^{i} a_{i,j} k_j \)

Evaluate \( \hat{k}_{i+1} = f(U^{ex}_{n,i}, \hat{V}^{im}_{n,i}) \)

end

\[
U^{ex}_{n+1} = U^{ex}_{n} + \Delta t \sum_{i=1}^{s+1} \hat{b}_j \hat{k}_j \\
U^{im}_{n+1} = U^{im}_{n} + \Delta t \sum_{i=1}^{s} b_j k_j
\]

In this method, the explicit computed residuals are represented by \( \hat{k} \) and the implicit residuals are represented by \( k \). The coefficients \( a, \hat{a}, b, \) and \( \hat{b} \) are found in Appendix A. They refer to the coefficients used for an IMEX scheme composed of given individual explicit and implicit schemes. The method above shows the separate implicit and explicit solves.
Chapter 3

Implementation

3.1 Square Domain

A specific implementation of the IMEX method for the two dimensional wave equation is considered for a unit square domain. This test case has the advantage of having a known exact solution and is useful for testing purposes. Convergence rates can be calculated to determine that the scheme is performing correctly.

3.1.1 Problem Definition

The unit square domain is defined as

\[ Q = (0, 1) \times (0, 1), \quad (3.1) \]

with homogeneous Dirichlet boundary condition

\[ v = 0, \quad \text{on } \partial \Omega_D. \quad (3.2) \]
The exact solution over the domain for $A = I$ and $f = 0$ is known to be

\[ u(x, y, t) = \frac{1}{\sqrt{2\pi}} \sin(\pi x) \sin(\pi y) \sin(\sqrt{2\pi}t), \]

\[ v(x, y, t) = \sin(\pi x) \sin(\pi y) \cos(\sqrt{2\pi}t), \]

\[ q_x(t) = \frac{1}{\sqrt{2}} \cos(\pi x) \sin(\pi y) \sin(\sqrt{2\pi}t), \]

\[ q_y(t) = \frac{1}{\sqrt{2}} \sin(\pi x) \cos(\pi y) \sin(\sqrt{2\pi}t). \]

The exact solutions in (3.3) evaluated at $t = 0$ prescribe the initial conditions for all quantities being solved for.

3.1.2 Choice of IMEX Scheme

The IMEX method chosen for this analysis was 3rd order accurate. It utilized a 3 stage Diagonally Implicit Runge-Kutta (DIRK) implicit integration and a 4 stage Explicit Runge-Kutta (ERK) explicit integration. The necessary coefficients for these parameters can be referenced in Appendix A. Both the implicit and explicit regions have their own Butcher table.

3.1.3 Mesh

The wave equation can be solved over the mesh depicted in Figure 3-1, which was designed to introduce a bias towards the corners of the unit square. This creates a finer mesh structure in the sharp areas of the domain. The variance in element size allows the creation of distinct implicit and explicit regions. Uniform refinement can be applied over the domain to preserve the size variance of the elements.
3.1.4 Solution

An example solution is computed and the results are shown in Figures 3-2 and 3-3. The explicit and implicit domains below can be combined to create the solution over the entire domain. The pertinent parameters are shown in Table 3.2.

<table>
<thead>
<tr>
<th>Solution Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order $k$</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.1: Given parameters for IMEX solution on the square domain.

Figure 3-1: Full, unrefined mesh for the square domain.
Figure 3-2: Solution of the velocity over the explicit domain for $k = 2$ and no refinements.

Figure 3-3: Solution of the velocity over the implicit domain for $k = 2$ of 2 and no refinements.
3.1.5 Spatial Convergence

Test Description

For this test case, the mesh can be broken into an implicit region and an explicit region based on equation 2.8. To determine the spatial convergence of the mesh, uniform refinements were applied over the whole domain. A time step was specifically chosen when computing the implicit and explicit regions to ensure that the IMEX boundary, $\Gamma^{imez}$, would stay in the same location for each consecutive refinement. Figures 3-4, 3-5, and 3-6 show these regions on the base mesh, after one refinement, and after two refinements.

![Figure 3-4: Full mesh with implicit and implicit regions.](image)

A second time step was then chosen for the computation. It was picked using the following expression

$$\Delta t = \frac{T_{end}}{n_{time}},$$

(3.4)

where $T_{end}$ is the chosen end time and $n_{time}$ is the number of time steps to take. For
Figure 3-5: Full mesh with one level of refinement.

the spatial convergence, $T_{end} = 1.1\sqrt{2}$. This was chosen as 1.1 times the period of the solution in order to have nonzero results for the position and gradients.

The error was computed for position, velocity, and gradients. In addition, both the position and velocity were postprocessed. Four mesh sizes were considered corresponding to up to three different levels of refinement. An extra fifth case was run for the first order case.

**Results**

As shown in table 3.2, good convergence results are achieved. The expected order of convergence for $u$, $v$, and $q$ is $k + 1$. This ideal convergence can be observed for all tested cases for each polynomial order $k$. After postprocessing, the expected order of convergence becomes $k + 2$. This is achieved for polynomial orders 1 and 2 for all mesh sizes. The postprocessed values of $u$ exhibit ideal behavior for the first two sets of refinements for $k = 3$ and for the $k = 4$ test set. For $v^*$ evaluated at polynomial
orders 3 and 4, the convergence rates are less than \( k + 2 \), but they do preserve the original \( k + 1 \) rate. This suboptimal convergence is most likely due to the fact that the maximum achievable accuracy is reached for this case.
Table 3.2: Errors and estimated orders of convergence (e.o.c) for the IMEX test case.

<table>
<thead>
<tr>
<th>Order</th>
<th>Mesh</th>
<th>$|u - u_h|$ Error e.o.c</th>
<th>$|v - v_h|$ Error e.o.c</th>
<th>$|q - q_h|$ Error e.o.c</th>
<th>$|u - u_h|$ Error e.o.c</th>
<th>$|v - v_h|$ Error e.o.c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>2.00E-3</td>
<td>1.60E-2</td>
<td>2.09E-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.035</td>
<td>3.00E-4</td>
<td>2.30E-3</td>
<td>2.99E-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0175</td>
<td>4.60E-5</td>
<td>3.81E-4</td>
<td>4.03E-5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0088</td>
<td>8.15E-6</td>
<td>7.94E-5</td>
<td>5.20E-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0044</td>
<td>1.76E-6</td>
<td>1.89E-5</td>
<td>6.59E-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>5.49E-5</td>
<td>3.84E-4</td>
<td>6.66E-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.035</td>
<td>4.59E-6</td>
<td>4.61E-5</td>
<td>3.33E-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0175</td>
<td>5.03E-7</td>
<td>6.21E-6</td>
<td>2.02E-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0088</td>
<td>7.28E-8</td>
<td>8.08E-7</td>
<td>1.22E-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>6.10E-6</td>
<td>4.97E-5</td>
<td>3.31E-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.035</td>
<td>4.98E-7</td>
<td>3.11E-6</td>
<td>6.85E-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0175</td>
<td>2.65E-8</td>
<td>1.93E-7</td>
<td>4.33E-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0088</td>
<td>1.69E-9</td>
<td>1.18E-8</td>
<td>2.71E-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>6.97E-7</td>
<td>4.36E-6</td>
<td>2.00E-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.035</td>
<td>2.09E-8</td>
<td>1.35E-7</td>
<td>2.54E-9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-7: Convergence of position $u$ for polynomial orders 1 through 4.
Figure 3-8: Convergence of velocity $v$ for polynomial orders 1 through 4.

Figure 3-9: Convergence of gradient $q$ for polynomial orders 1 through 4.
Figure 3-10: Convergence of postprocessed position $u^*$. 

Figure 3-11: Convergence of postprocessed velocity $v^*$. 

36
3.1.6 Temporal Convergence

Test Description

After the spatial error convergence was examined, the temporal convergence was computed. For this test, the implicit and explicit regions were not kept the same. Instead, various timesteps were chosen and the condition in equation 2.8 was used to choose the different regions. The timestep started small enough so the entire domain was able to be computed explicitly. It was then increased resulting in an increase in the amount of the domain computed implicitly. The implicit region continued to increase with each increase in timestep until the entire solution was computed implicitly. The resulting numerical solution was then compared to the analytical one.

Results

The time convergence was performed for a polynomial order of $k = 4$ with no mesh refinements and time order of 2. $K_{\text{max}}$ was set to 0.5. This case was chosen because the dominant error was temporal rather than spatial. Five cases were computed with varying numbers of implicit and explicit elements based on the chosen timesteps.

![Figure 3-12: Change in error of $u$ for decreasing timesteps on the square domain.](image1)

![Figure 3-13: Change in error of $v$ for decreasing timesteps on the square domain.](image2)
Figure 3-14: Change in error of $q$ for decreasing timesteps on the square domain.

<table>
<thead>
<tr>
<th>Timestep $\Delta t$</th>
<th>Implicit Elements</th>
<th>Explicit Elements</th>
<th>$|u - u_h|$ Error e.o.c</th>
<th>$|v - v_h|$ Error e.o.c</th>
<th>$|q - q_h|$ Error e.o.c</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00E-2</td>
<td>60</td>
<td>0</td>
<td>2.10E-4</td>
<td>5.70E-4</td>
<td>9.31E-4</td>
</tr>
<tr>
<td>1.00E-2</td>
<td>60</td>
<td>0</td>
<td>5.11E-5</td>
<td>1.55E-4</td>
<td>2.27E-4</td>
</tr>
<tr>
<td>7.00E-3</td>
<td>56</td>
<td>4</td>
<td>2.18E-5</td>
<td>6.84E-5</td>
<td>9.73E-6</td>
</tr>
<tr>
<td>5.00E-3</td>
<td>28</td>
<td>32</td>
<td>1.89E-6</td>
<td>7.20E-6</td>
<td>1.18E-5</td>
</tr>
<tr>
<td>4.00E-3</td>
<td>16</td>
<td>44</td>
<td>7.09E-7</td>
<td>4.49E-6</td>
<td>8.65E-6</td>
</tr>
</tbody>
</table>

Table 3.3: Errors and estimated orders of convergence (e.o.c) for the square domain time convergence test. The parameters used are polynomials of order 4, temporal discretization of order 2, no mesh refinements, and $K_{max} = 0.5$.

As can be see in Table 3.3, the convergence rate is approximately 2 or larger as expected for a time scheme of order 2. This indicates that for a time error dominated problem, the convergence rate of the solution maintains the optimal rate in the IMEX scheme. The only violation of this occurs at the finest timestep size for the velocity $v$ and gradient $q$. This behavior is suspected to be caused by the varying coefficients in the implicit and explicit methods. As the mesh transitions between the two schemes it exhibits more oscillatory convergence rates before coming back to the expected value.

A second case was also examined for a polynomial order of 5 and a 3rd order time scheme. $K_{max}$ was set to 1.
Table 3.4: Errors and estimated orders of convergence (e.o.c) for the square domain time convergence test. The parameters used are polynomials of order 5, temporal discretization of order 3, no mesh refinements, and $K_{\text{max}} = 1$.

Figure 3-15: Change in error of $u$ for decreasing timesteps on the square domain for the second parameter set.

Figure 3-16: Change in error of $v$ for decreasing timesteps on the square domain for the second parameter set.

Figure 3-17: Change in error of $q$ for decreasing timesteps on the square domain for the second parameter set.
The same transitional behavior as observed in the first case appears in this second case. However, if the average convergence rates are examined over all timesteps, the convergence rates are found to be 3.44 for \( u \), 3.31 for \( v \), and 2.96 for \( q \). These demonstrate the expected convergence values for a time scheme of order 3.

### 3.2 L-shaped Mesh

A second test problem is now considered after the verification performed on the square domain. This problem does not have an exact solution but contains more complicated geometry. It is useful to check that the IMEX solver works as expected on a different domain.

#### 3.2.1 Problem Definition

A new, L-shaped domain is defined

\[
\Omega = [(0, 1) \times (0, 0.5)] \cap [(0.5, 1) \times (0.5, 1)],
\]  

(3.5)

with Dirichlet boundary condition

\[
v = 0, \quad \text{on } \partial \Omega_D.
\]  

(3.6)

Initial conditions are prescribed by evaluating the following equations at \( t = 0 \)

\[
\begin{align*}
u(x, y, t) &= \frac{1}{\sqrt{2\pi}} \sin(2\pi x) \sin(2\pi y) \sin(\sqrt{2\pi} t), \\
v(x, y, t) &= \sin(2\pi x) \sin(2\pi y) \cos(\sqrt{2\pi} t), \\
q_x(t) &= \frac{1}{\sqrt{2}} \cos(2\pi x) \sin(2\pi y) \sin(\sqrt{2\pi} t), \\
q_y(t) &= \frac{1}{\sqrt{2}} \sin(2\pi x) \cos(2\pi y) \sin(\sqrt{2\pi} t).
\end{align*}
\]  

(3.7)
3.2.2 Choice of IMEX Scheme

A 3rd order accurate, 3 stage DIRK implicit integration and 4 stage ERK explicit integration IMEX scheme was used. The necessary coefficients can be referenced in Appendix A.

3.2.3 Mesh

The mesh used in this problem was biased towards the inner corner of the L-mesh to create a finer region. Figure 3-18 depicts this bias on the base mesh. Uniform refinement can be performed to create a finer mesh as needed.

![Figure 3-18: L-shaped domain base mesh.](image)

3.2.4 Implicit-Explicit Regions

Like the square mesh, the L-mesh is also divided into implicit and explicit regions based on equation 2.8. With a smaller $\Delta t$, the implicit region is found at the inner corner as shown in Figure 3-19. As the time step increases, this spreads outward to
cover the entire domain.

Figure 3-19: Example of implicit and explicit regions in the L-shaped domain.

3.2.5 Example Solution

The solution can be computed over the explicit and implicit domains. Figure 3-20 shows the implicit region while Figure 3-21 shows the larger explicit region. Because there is no exact solution for this problem, the simulation results can be compared to the implicit reference solution run at a small time step.

<table>
<thead>
<tr>
<th>Solution Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order k</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.5: Given parameters for IMEX solution on the L-shaped domain.
Figure 3-20: Solution in the implicit region of the L-shaped mesh.

Figure 3-21: Solution in the explicit region of the L-shaped mesh.
3.2.6 Temporal Convergence

Test Description

A study of the time convergence was performed over this domain for a 3rd order time scheme, a 3rd order polynomial order, and 3 levels of mesh refinement. The timestep started small enough to compute the solution on the L-mesh entirely explicitly. It was then gradually increased until the entire domain was computed implicitly. The resulting convergence is shown below for 5 test cases of varying amounts of implicit and explicit elements.

Results

As can be seen in Table 3.6, there appears to be a transition of convergence rate in the IMEX region. The convergence rate starts out at approximately 3 as it should for a 3rd order time scheme. It then converges sub-optimally in the IMEX region and returns to the correct optimal convergence rate. Again, this is mostly likely caused by the transition between the coefficients of the two schemes.
<table>
<thead>
<tr>
<th>Timestep ( \Delta t )</th>
<th>Implicit Elements</th>
<th>Explicit Elements</th>
<th>( | u - u_h | ) Error</th>
<th>e.o.c</th>
<th>( | v - v_h | ) Error</th>
<th>e.o.c</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.00E-3</td>
<td>7040</td>
<td>0</td>
<td>5.11E-4</td>
<td>-</td>
<td>1.32E-2</td>
<td>-</td>
</tr>
<tr>
<td>3.00E-3</td>
<td>6528</td>
<td>512</td>
<td>2.24E-4</td>
<td>2.87</td>
<td>5.97E-3</td>
<td>2.77</td>
</tr>
<tr>
<td>1.30E-3</td>
<td>3520</td>
<td>3520</td>
<td>1.13E-4</td>
<td>0.82</td>
<td>3.05E-3</td>
<td>0.80</td>
</tr>
<tr>
<td>9.00E-4</td>
<td>2368</td>
<td>4672</td>
<td>5.85E-5</td>
<td>1.79</td>
<td>1.59E-3</td>
<td>1.77</td>
</tr>
<tr>
<td>4.00E-4</td>
<td>0</td>
<td>7040</td>
<td>4.67E-6</td>
<td>3.12</td>
<td>1.28E-4</td>
<td>3.11</td>
</tr>
</tbody>
</table>

Table 3.6: Errors and estimated orders of convergence (e.o.c) for the L-domain time convergence test. The parameters used are polynomials of order 2, temporal discretization of order 3, 3 levels of mesh refinements, and \( K_{\text{max}} = 0.5 \).

**Computation Time**

A final study was performed to determine the time it took to compute the solution on an implicit only mesh, an IMEX mesh, and an explicit only mesh. This was performed with a polynomial order of 2, time order of 3, \( K_{\text{max}} = 0.5 \), and on the base L-mesh.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Explicit Elements</th>
<th>Implicit Elements</th>
<th>Timestep ( \Delta t )</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>0</td>
<td>110</td>
<td>0.0075</td>
<td>57.13</td>
</tr>
<tr>
<td>IMEX</td>
<td>73</td>
<td>37</td>
<td>0.0075</td>
<td>33.62</td>
</tr>
<tr>
<td>Explicit</td>
<td>110</td>
<td>0</td>
<td>0.0035</td>
<td>33.06</td>
</tr>
</tbody>
</table>

Table 3.7: Computational time needed for implicit only, IMEX, and explicit only schemes.

As seen in Table 3.7, the quickest methods for this L-mesh case are IMEX and explicit. This demonstrates the benefit having a mesh consisting of both implicit and explicit elements rather than just having a purely implicit mesh. The timestep is kept the same but the computational time is decreased by 41%. As the size difference between the smallest and largest elements in the mesh increases, the larger the benefit running an IMEX method will have.

The same test is repeated with one level of mesh refinement. This time, the IMEX method takes the least computational time. The timestep size needed for stability over the explicit mesh increases the number of timesteps sufficiently enough to increase its computational time beyond that of IMEX. The implicit method remains stable but
more computationally expensive than the IMEX mesh with the same timestep. This behavior can be seen in Table 3.8.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Explicit Elements</th>
<th>Implicit Elements</th>
<th>Timestep Δt</th>
<th>CPU Time s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>0</td>
<td>440</td>
<td>0.0033</td>
<td>541.59</td>
</tr>
<tr>
<td>IMEX</td>
<td>308</td>
<td>132</td>
<td>0.0033</td>
<td>283.97</td>
</tr>
<tr>
<td>Explicit</td>
<td>440</td>
<td>0</td>
<td>0.0020</td>
<td>335.48</td>
</tr>
</tbody>
</table>

Table 3.8: Computational time needed for implicit only, IMEX, and explicit only schemes for one level of mesh refinement.
Chapter 4

Conclusions

This thesis examined an HDG implementation of the IMEX integration scheme. Convergence rates for both the spatial and temporal discretizations were examined for a square and L-shaped domain. It was found that spatial convergence for both domains was optimal \((k + 1)\) and the post-processed results of the position and velocity exhibited superconvergence \((k + 2)\). Additionally, in the case of a temporal dominated error, the convergence was found to be optimal in both the fully explicit and implicit cases. The convergence rate was suboptimal during the IMEX transition between these two cases as a result of the varying coefficients of the two schemes.

Future work for this project will be to implement the IMEX method for other problems besides the wave equation. After this verification, it would be interesting to split the computations of the implicit and explicit regions between the CPU and GPU. This could create quicker simulations through parallelization of processes. In the future this could hold many benefits for scientific computing.
Appendix A

Butcher Tables

2 stage, 2nd order DIRK and 3 stage, 2nd order ERK

\[
\begin{array}{c|ccc}
  c & A & \alpha & \alpha \\
  b^T & = & 1 & 1 - \alpha & \alpha \\
  & & 1 - \alpha & \alpha \\
\end{array}
\]

\[
\begin{array}{c|cccc}
  \hat{c} & \hat{A} & \alpha & \alpha & 0 \\
  \hat{b}^T & = & 1 & \delta & 1 - \delta & 0 \\
  & & 0 & 1 - \alpha & \alpha \\
\end{array}
\]

2 stage, 3rd order DIRK and 3 stage, 3rd order ERK (not used in this thesis)

\[
\begin{array}{c|ccc}
  c & A & \alpha & \alpha \\
  b^T & = & 1 - \alpha & 1 - 2\alpha & \alpha \\
  & & \frac{1}{2} & \frac{1}{2} \\
\end{array}
\]
\[
\begin{bmatrix}
\hat{c} \\
\hat{A} \\
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\alpha & \alpha & 0 & 0 \\
1 - \alpha & \alpha - 1 & 2(1 - \alpha) & 0 \\
0 & \frac{1}{2} & \frac{1}{2} \\
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0 \\
1 \\
\end{bmatrix}
\]

3 stage, 3rd order DIRK and 4 stage, 3rd order ERK

\[
\begin{bmatrix}
\hat{c} \\
\hat{A} \\
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0.4358665215 & 0.4358665215 & 0 & 0 \\
0.7179332608 & 0.2820667392 & 0.4358665215 & 0 \\
1 & 1.208496649 & -0.644363171 & 0.4358665215 \\
1.208496649 & -0.644363171 & 0.4358665215 \\
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0.4358665215 & 0.4358665215 & 0 & 0 \\
0.7179332608 & 0.3212788860 & 0.3966543747 & 0 \\
1 & -0.105858296 & 0.5529291479 & 0.5529291479 \\
\end{bmatrix}
\begin{bmatrix}
0 \\
1.208496649 \\
-0.644363171 \\
0.4358665215 \\
\end{bmatrix}
\]
Bibliography


