

Partitioned Implicit Runge-Kutta Timesteppers for Micromagnetics with Eddy
Currents

by

Josh Engwer, A.A., B.S., M.S.

A Dissertation

In

Mathematics

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY

Approved

Katharine R. Long
Chair of Committee

Ram V. Iyer

Victoria E. Howle

Mahdi Sanati

Mark A. Sheridan
Dean of the Graduate School

August, 2018

[Revised September, 2022](#)

[Revised December, 2019](#)

Copyright 2018-2019, 2022, Josh Engwer

ACKNOWLEDGMENTS

I wish to thank my advisor, Katharine Long, for the outstanding mentorship, encouragement, support and Sundance. I also greatly appreciate the feedback from all the committee members.

I am grateful to have been a part of the TTU OnRamps program for the final two years of my doctoral program. I thank Stephanie Hart for providing me this amazing opportunity.

Special thanks goes out to my father for posing interesting thought-provoking questions related to science and mathematics throughout my childhood – not to mention the exposure to chess, basketball, the Apple IIgs computer, the manual transmission, the Algebra textbooks and the Bubble Sort algorithm.

Special thanks also goes to all my past coworkers, bosses and clients from my previous ten-year IT career. All that experience helped me develop enough grit to persevere through grueling coursework, preliminary exams and research.

I am particularly indebted to my mother and father, both former teachers, for greatly influencing my teaching philosophy. The students I taught the first six years here at Texas Tech appreciate it.

In memory of my mother, who wanted nothing more than her only son to obtain a higher education.

TABLE OF CONTENTS

Acknowledgements	ii
Abstract	viii
List of Tables	ix
List of Figures	x
List of Abbreviations	xi
I INTRODUCTION	1
1.1 Micromagnetics Overview	1
1.2 Eddy Currents Overview	2
1.3 Literature Survey on Micromagnetics Coupled with Eddy Currents	2
1.4 Model Derivation	4
1.4.1 Electromagnetic Behaviour in a Solid Material	4
1.4.2 Simplifying Assumptions	4
1.4.3 Eddy Currents Equation (EC)	5
1.4.4 Landau-Lifschitz-Gilbert Equation (LLG)	6
1.4.5 The Coupled PDE System (LLG+EC)	6
II TIMESTEPPERS: CONSTRUCTION & ORDER OF ACCURACY	8
2.1 Model Initial Value Problem	8
2.2 Timestepper Order of Accuracy	8
2.2.1 Local Truncation Error	9
2.2.2 Global Error	9
2.2.3 Order of Accuracy does not Guarantee Convergence	10
2.3 Taylor Series (TS) Timesteppers	10
2.4 Linear Multistep (LM) Timesteppers	11
2.4.1 Adams-Bashforth Timesteppers	12
2.4.2 Adams-Moulton Timesteppers	12
2.4.3 Backward Differentiation Formulas	13
2.5 Explicit Runge-Kutta (ERK) Timesteppers	14

2.6	Implicit Runge-Kutta (IRK) Timesteppers	15
2.7	Runge-Kutta Order Conditions	16
2.7.1	1 st -Order Conditions for 2-stage IRK Timestepper	16
2.7.2	2 nd -Order Conditions for 2-stage IRK Timestepper	17
2.7.3	General RK Order Conditions	18
2.8	Common ERK Timesteppers	19
2.8.1	Forward Euler Timestepper	19
2.8.2	Explicit Midpoint Timestepper	20
2.8.3	Classic RK4 Timestepper	21
2.9	Common IRK Timesteppers	22
2.9.1	Backward Euler Timestepper	22
2.9.2	Trapezoidal Timestepper	23
2.10	Common IRK Timesteppers built from Collocation	24
2.10.1	(Continuous) Collocation	24
2.10.2	Discontinuous Collocation	25
2.10.3	Gauss-Legendre Timesteppers	26
2.10.4	Radau Timesteppers	28
2.10.5	Lobatto Timesteppers	30
2.11	Diagonally Implicit Runge-Kutta (DIRK) Timesteppers	31
2.12	Singly Diagonally Implicit Runge-Kutta (SDIRK) Timesteppers	32
III	TIMESTEPPERS: CONSISTENCY, STABILITY & INVARIANCE	33
3.1	Consistency	33
3.1.1	Consistency w.r.t. Linear Multistep Timesteppers	33
3.1.2	Consistency w.r.t. Runge-Kutta Timesteppers	34
3.2	Zero-Stability	35
3.2.1	Zero-stability w.r.t. Linear Multistep Timesteppers	37
3.2.2	Zero-stability w.r.t. Runge-Kutta Timesteppers	38
3.3	Convergence	38
3.3.1	Convergence w.r.t. Linear Multistep Timesteppers	39

3.3.2	Convergence w.r.t. Runge-Kutta Timesteppers	39
3.4	Stiff ODE Systems	39
3.5	A-Stability	43
3.6	A(α)-Stability	47
3.7	L-Stability	48
3.7.1	DAS containing $\mathbb{C}^{-\text{Re}}$ and more does not imply L-stability	49
3.8	Preservation of Quadratic Invariants	50
3.9	Feature Comparison of Common Timesteppers	54
3.10	Linear Multistep Timesteppers do not Make the Cut	56
3.11	ERK Timesteppers do not Make the Cut	56
3.12	DIRK Timesteppers do not Make the Cut	57
3.13	SDIRK Timesteppers do not Make the Cut	58
3.14	Candidate Timesteppers for the LLG+EC Problem	59
IV	FINITE ELEMENTS	60
4.1	Model PDE: Poisson’s Equation	60
4.2	The Finite Element Method	60
4.2.1	STEP 1: Determine Pre-Element Weak Form	61
4.2.2	STEP 2: Generate Mesh from Domain	62
4.2.3	STEP 3: Select Appropriate Finite Elements	64
4.2.4	STEP 4: Determine Galerkin Weak Form	64
4.2.5	STEP 5: Assemble Finite Element Algebraic System	64
4.2.6	STEP 6: Solve Finite Element Algebraic System	65
4.3	Nodal Finite Elements	66
4.3.1	Lagrange Elements	66
4.3.2	Vector Lagrange Elements	66
4.4	Edge Finite Elements	67
4.4.1	Nédélec (1 st Kind) Elements	67
4.4.2	Nédélec (2 nd Kind) Elements	68
4.4.3	Weak Divergence-Free Property	69

4.4.4	Immunity to Spurious Modes	69
4.5	Software Implementation Remarks	69
4.5.1	A Brief Survey of Finite Element Frameworks	70
4.5.2	Reference Cells	71
4.5.3	Efficient Quadrature on Triangles & Tetrahedra	71
4.5.4	Automatic Jacobian Matrix Computation	71
V	THE COUPLED MICROMAGNETICS PROBLEM	72
5.1	Weak Form	72
5.1.1	Integration-by-Parts for Micromagnetics	72
5.1.2	Pre-Element-Pre-Timestepper Weak Form	73
5.2	Discretization in Space	74
5.2.1	Appropriate Hilbert Spaces	74
5.2.2	Pre-Element-Pre-Timestepper Weak Problem	76
5.2.3	Appropriate Mixed Finite Element	76
5.2.4	Post-Element-Pre-Timestepper Weak Problem	76
5.3	Discretization in Time	76
5.3.1	Partitioned Implicit Runge-Kutta (PIRK) Timesteppers	76
5.3.2	Appropriate PIRK Timestepper	78
5.3.3	Post-Element-Post-Timestepper Weak Form	79
5.3.4	Post-Element-Post-Timestepper Weak Problem	80
5.3.5	Treatment of Inhomogeneous Dirichlet BC's	80
VI	VALIDATION TESTS	81
6.1	Simulation Pseudocode	81
6.2	Remarks on Software Implementation	82
6.2.1	FEniCS	82
6.2.2	NumPy	82
6.2.3	SymPy	83
6.2.4	Automatic Handling of Boundary Conditions	83
6.3	Validation Tests	83

6.3.1	Manufactured Solutions for the Unit Square $\Omega := [0, 1]^2$. . .	85
6.3.2	The Importance of Sufficient Refinement	87
6.3.3	Example Validation Plots with $\mathbf{B}_{ex}(t)$ & $\mathbf{M}_{ex}(t)$	88
6.3.4	Example Validation Plots with $\mathbf{B}_{ex}(\mathbf{x})$ & $\mathbf{M}_{ex}(\mathbf{x})$	90
6.3.5	Example Validation Plots with $\mathbf{B}_{ex}(t; \mathbf{x})$ & $\mathbf{M}_{ex}(t; \mathbf{x})$. . .	92
6.4	Summary of all Validation Tests for Unit Square $\Omega := [0, 1]^2$.	94
VII	SIMULATION ERROR ESTIMATES	95
7.1	A Priori QIP Error Estimate	95
7.2	A Posteriori QIP Error Estimate	98
VIII	CONCLUSION	101
8.1	Future Work	102
8.1.1	Block Preconditioner	102
8.1.2	Validation Tests in 3D on the Unit Cube	104
APPENDIX	116
A	Taylor Series Expansions	116
B	Chain Rules	117
C	Vector Algebra Identities	118
D	Vector Calculus Identities	118
E	Divergence Theorem	118
F	Green's Identities	119
G	Linear Algebra Lemmas	120
H	Tensor Notation	121
I	Normed Linear Spaces	123
J	Inner Product Spaces	123
K	Hilbert Spaces	124
L	Inequalities	124
M	B-Stability	125

ABSTRACT

Our research concerns the numerical solution of the Landau-Lifschitz-Gilbert Equation coupled with the Eddy Currents Equation. We construct a partitioned implicit Runge-Kutta timestepper with one component being L-stable and the other being quadratic invariant-preserving. Mixed (Nédélec, Vector Lagrange) finite elements are employed for spatial discretization. We discuss using the resulting scheme for simulations as well as validation tests and error estimates. Remarks on software implementation are provided.

This dissertation was first revised from late September 2019 until late December 2019 after the discovery that the Lobatto IIIA timestepper is not actually quadratic invariant-preserving (QIP) as previously thought given the encouraging validation plot results. Besides several small typo fixes, grammar corrections, clarifications, relabelings and restylings sprinkled over about twenty-five pages, the remaining revisions in the content, tables and plots mainly involve using Lobatto IIIS as part of the chosen partitioned implicit Runge-Kutta (PIRK) timestepper in place of Lobatto IIIA. Other revisions include adding Cooper's QIP Theorem, removing mention of the Lobatto IIIC*, Radau I & Radau II timesteppers as they did not possess desirable features and space was tight, adding mention of the Radau IIA & IIB timesteppers, removing two terms in the LLG+EC weak form involving a self cross product which will always vanish, and added a few new or missing references to the Bibliography.

In early September 2022, a few more major typos were discovered. In particular, the Trapezoidal timestepper was incorrectly stated as QIP in Table 3.1, and the caption for Table 8.1 incorrectly stated that the first value in each table entry was the number of non-zeros of the Jacobian matrix when instead it represents its matrix size. Both of these typos have now been corrected.

Readers that are interested in the original dissertation and the detailed accompanying errata sheet can click the following URL to access them at the the Texas Tech University Library website: <http://hdl.handle.net/2346/74392>

LIST OF TABLES

1.1	Maxwell’s Equations & Constitutive Equations	4
1.2	Electromagnetic Simplifying Assumptions	5
2.1	Taylor Series Timesteppers	11
2.2	Adams-Bashforth LM Timesteppers	12
2.3	Adams-Moulton LM Timesteppers	13
2.4	Backward Differentiation Formula LM Timesteppers	14
2.5	Runge-Kutta Order Conditions	19
2.6	Gauss-Legendre IRK Timesteppers	28
2.7	Radau IRK Timesteppers	29
2.8	Lobatto IRK Timesteppers	30
2.9	DIRK timesteppers	31
2.10	SDIRK timesteppers	32
3.1	Timestepper Order, Stability & Invariance Summary	55
4.1	(Scalar) Lagrange Elements	66
4.2	Vector Lagrange Elements	66
4.3	First Kind Nédélec Elements	67
4.4	Second Kind Nédélec Elements	68
5.1	Chosen PIRK Timesteppers	78
6.1	CAMR Manufactured Solutions for the Unit Square	85
6.2	CATS Manufactured Solutions for the Unit Square	86
6.3	Validation Tests on the Unit Square	94
8.1	Matrix Size & Percentage of Non-Zeros in Jacobian Matrix	104
H.2	Tensor Form of Vector Algebra/Calculus Operations	122

LIST OF FIGURES

3.1	Solution Plots of Perturbed (TIVP0) w/ “Bad 2-stepper” & BDF2 . . .	36
3.2	DAS Plot of Forward Euler	44
3.3	DAS Plot of GaussLegendre(1)	45
3.4	DAS Plot of Backward Euler	46
3.5	DAS Plots of BDF2 & BDF4	47
3.6	Solution Plots of Curtiss-Hirschfelder Equation w/ TR & BE	49
3.7	DAS Plots of Lobatto IIIC(2) & Radau IIA(2)	49
3.8	DAS Plot of Spijker’s DIRK(2)	50
3.9	Solution Plots to (IVPQ) w/ EM & GL(1)	53
3.10	QIP Error Plots to (IVPQ) w/ EM & GL(1)	53
4.1	Axis-Aligned Meshes for Square Domain	62
4.2	Delaunay Meshes for Square Domain	63
4.3	First-Degree (Scalar) Lagrange Shape Functions	66
4.4	First-Degree First Kind Nédélec Shape Functions	67
4.5	First-Degree Second Kind Nédélec Shape Functions	68
6.1	“Bottoming Out” of CAMR Line Plots	87
6.2	“Bottoming Out” of CATS Line Plots	87
6.3	CAMR Plots of LLG+EC with Time-Dependent Exact Solutions . . .	88
6.4	CATS Plots of LLG+EC with Time-Dependent Exact Solutions . . .	89
6.5	CAMR Plots of LLG+EC with Space-Dependent Exact Solutions . . .	90
6.6	CATS Plots of LLG+EC with Space-Dependent Exact Solutions . . .	91
6.7	CAMR Plots of LLG+EC with Space-Time Exact Solutions	92
6.8	CATS Plots of LLG+EC with Space-Time Exact Solutions	93
8.1	Jacobian Matrix Sparsity Patterns on a Coarse Mesh	103
8.2	Jacobian Matrix Sparsity Patterns on a Fine Mesh	103

LIST OF ABBREVIATIONS

AB r	r -step Adams-Bashforth timestepper
AM r	r -step Adams-Moulton timestepper
BC	Boundary Condition
BDF r	r -step Backward Differentiation Formula timestepper
BE	Backward Euler timestepper
CAMR	Convergence Analysis of Mesh Refinements
CATS	Convergence Analysis of Timestep Sizes
CIZ	Clever Insertion of Zero
CR	Chain Rule (for differentiating functions)
CSI	Cauchy-Schwarz Inequality
DIRK	Diagonally Implicit Runge-Kutta timestepper
EC	Eddy Currents equation
EM	Explicit Midpoint timestepper
ERK	Explicit Runge-Kutta timestepper
FE	Forward Euler timestepper
GE	Global Error (due to timestepper)
GI	Green's Identity (for multiple integral)
GL(s)	s -stage Gauss-Legendre timestepper
IBP	Integration By Parts
IC	Initial Condition
IRK	Implicit Runge-Kutta timestepper
IVP	Initial Value Problem
LLG	Landau-Lifschitz-Gilbert equation
LM r	r -step Linear Multistep timestepper
LTE	Local Truncation Error (due to timestepper)
MIBP	“Micromagnetic” Integration By Parts

MIVP	Model IVP
MVIVP	Model Vector IVP
NNL	Nested Norm Lemma
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PERK	Partitioned Explicit Runge-Kutta timestepper
PIRK	Partitioned Implicit Runge-Kutta timestepper
QFSML	Quadratic Form Symmetric Matrix Lemma
QIL	Quadratic Invariant Lemma
QIP	Quadratic Invariant Preserving (timestepper)
RHS	Right-Hand Side (of equation)
RK(s)	s -stage Runge-Kutta timestepper
SDIRK	Singly Diagonally Implicit Runge-Kutta timestepper
TE	Taylor Expansion (of function)
TI	Triangle Inequality
TIVP0	Test IVP for 0-stability
TIVPA	Test IVP for A-stability
TN	Tensor Notation
TR	Trapezoidal timestepper
TS p	p -derivative Taylor Series timestepper
VA	Vector Algebra identity
VC	Vector Calculus identity
WLOG	Without Loss Of Generality
w.r.t.	with respect to

CHAPTER I INTRODUCTION

1.1 Micromagnetics Overview

Micromagnetics [18, 70] is a subfield of electromagnetics spearheaded in 1935 by Landau and Lifshitz [71, 72] which involves the dynamics of magnetic fields and magnetization fields in ferromagnetic solids. This is essential to the proper function of certain storage devices, sensors and actuators. The prefix “micro” in micromagnetics reflects the fact that this magnetic behaviour occurs at the micrometer scale rather than the nanometer or atomic scales.

The manner in which spinning hard drives store data [61] is as follows: there’s a platter that spins typically at thousands of revolutions per minute, and a read-write arm will change the magnetic field in a tiny portion of the platter by manipulating the magnetic dipoles in that small area. The neighboring dipoles can be arranged in one of two arrangements: one arrangement represents the bit ‘0’, and the other arrangement represents the bit ‘1’. A finite sequence of these bits, therefore, represents a piece of the data that is stored and can be retrieved later as needed. Similar micromagnetic behavior [61] occurs when writing to: a spindle of magnetic tape used in mainframes circa the 1960’s, audio cassette tapes popular in the 1980’s and floppy disks used in the 1980’s. By contrast, CD’s, DVD’s and Blu Ray’s are read and written by a special optical laser.

Sensors and actuators function with the help of a ferromagnet that is magnetostrictive [61]. This means as a sufficiently strong external magnetic field is applied to the solid, it experiences mechanical stress that causes it to either elongate or compress. With a sensor, a stretched ferromagnet blocks a light beam, causing the sensor to trigger. Stretching the solid in an actuator causes an electric circuit to complete and thus activate another portion of the system. Such magnetostrictive components are also found in sonar systems and fuel injection systems.

1.2 Eddy Currents Overview

Consider an ordinary pendulum but instead of a ball suppose a conductive plate is attached at the end [45, 112]. Fix a horseshoe or U-shaped magnet precisely at the lowest point of the pendulum's swing. Starting at a reasonably high point, release the pendulum and observe its subsequent back-and-forth swings. As the plate passes through the magnet, notice that it markedly slows down, eventually coming to a complete stop. This occurs because as the plate enters the magnet's magnetic field \mathbf{H}_{ext} , the magnetic induction field of the plate, \mathbf{B}_{plate} , changes [45, 61, 112]. But as dictated by Lenz's Law, the plate's changing magnetic flux (i.e. $\frac{\partial \mathbf{B}_{plate}}{\partial t} \neq \vec{0}$), induces an electromotive force (emf) which causes the plate's free electrons to move in a swirling motion [45, 61, 112]. These swirling electric currents on the plate's surface are called eddy currents because their motion looks similar to eddys in ocean water.

In everyday devices, the presence of eddy currents are often a nuisance because they cause a loss in power efficiency [112]. Again, it's a loss in efficiency due to Lenz's Law which states the magnetic field due to the eddy currents always opposes the change that produced them. Therefore, to minimize eddy currents, the conducting material either has cuts or interleaving insulation [112]. Of course, there are rare instances where power losses due to eddy currents are actually desirable, say when powering off a high-speed electric saw or braking a subway train [112].

1.3 Literature Survey on Micromagnetics Coupled with Eddy Currents

We are interested in coupling micromagnetic behaviour in the presence of eddy currents. There are many published works on numerical methods for micromagnetics by itself; likewise for eddy currents by itself. Only since the seminal paper by Visintin [124] in 1985 has there been work on numerical methods for the coupling of the two behaviors together.

In terms of the problem's spatial dimension, 1D [47, 63, 86, 108, 111] was sometimes used. As expected, many authors chose 3D [10, 22, 30, 35, 56, 57, 116]. Yanik

et al [127, 128] started in 3D but with their domain choice reduced the problem to one spatial dimension by means of symmetry and a change of coordinates. Monk and Vacus [91] reduced their selected 3D problem to two spatial dimensions due to symmetry.

Regarding spatial discretization of the coupled micromagnetics-eddy-currents model, finite differences [35, 86, 111] are occasionally used. Other papers handle it through Fast Fourier transforms [22, 108, 119, 120]. A hybrid method involving finite elements and boundary elements [56, 57] is also constructed. Most works exclusively used the preferred finite elements [10, 30, 47, 63, 76, 91, 116]. However, they only considered low-order (i.e. zeroth- or first-order) node and edge elements.

For discretization in time, several authors employed a variable-step multistep timestepper [30, 56] while Yanik et al [127, 128] used finite difference time domain. Chang et al [22] reformulated the problem as a differential-algebraic equation and applied an appropriate timestepper. A few others utilized an explicit timestepper [35, 86, 119, 120] – the higher-order ones required extremely tiny timestep sizes such as $\Delta t \approx 10^{-13}$ or smaller which causes the simulation codes to become impractically slow. Monk and Vacus [91] apply an explicit/implicit timestepper. Most papers used the more desirable implicit timestepping schemes [10, 47, 63, 76, 111, 116] of either first- or second-order.

Nearly all the forementioned papers had homogeneous boundary conditions imposed on the equation governing eddy currents in one of the following forms:

$$\mathbf{n} \times \mathbf{E} = \vec{\mathbf{0}}, \quad \mathbf{n} \times \mathbf{H} = \vec{\mathbf{0}}, \quad \mathbf{n} \times (\nabla \times \mathbf{H}) = \vec{\mathbf{0}}, \quad [\mathbf{n} \times \mathbf{H}_{eddy}]_{in}^{out} = \vec{\mathbf{0}} \quad (1.1)$$

The exceptions are Mayergoyz [86] and Serpico [111] who impose inhomogeneous boundary conditions on their 1D eddy currents formulations:

$$\mathbf{H}(z; t)|_{\partial\Omega} = \mathbf{H}_D(t) \quad (1.2)$$

Finally, Slodička and Bañas [116] enforced simplifying assumptions to justify approximating the nonlinear micromagnetics equation with a linear one.

1.4 Model Derivation

1.4.1 Electromagnetic Behaviour in a Solid Material

This model has Maxwell's Equations [112] as the basic foundation of electromagnetic behaviour in a vacuum. However, there's a solid material involved, so the so-called constitutive equations [45] are incorporated with Maxwell's Equations. All together, these equations model electromagnetic behaviour in a solid medium:

Table 1.1: Maxwell's Equations & Constitutive Equations

<u>MAXWELL'S EQUATIONS</u>	<u>CONSTITUTIVE EQUATIONS</u>	<u>INTERPRETATIONS</u>
(MX1) $\nabla \cdot \mathbf{D} = \rho_f$	(CE1) $\mathbf{D} = \epsilon \mathbf{E} + \mathbf{P}$	$\rho_f \equiv$ Free Charge Density $\epsilon \equiv$ Material Permittivity $\mathbf{P} \equiv$ Polarization Field
(MX2) $\nabla \cdot \mathbf{B} = 0$	(CE2) $\mathbf{B} = \mu(\mathbf{H} + \mathbf{M})$	$\mu \equiv$ Material Permeability $\mathbf{M} \equiv$ Magnetization Field
(MX3) $\nabla \times \mathbf{E} = -\mathbf{B}_t$		$\sigma \equiv$ Material Conductivity $\mathbf{E} \equiv$ Electric Field $\mathbf{B} \equiv$ Mag. Inductance
(MX4) $\nabla \times \mathbf{H} = \mathbf{J}_f + \mathbf{D}_t$	(CE3) $\mathbf{J}_f = \sigma \mathbf{E}$	$\mathbf{D} \equiv$ Elec. Displacement $\mathbf{J}_f \equiv$ Free Current Density $\mathbf{H} \equiv$ Magnetic Field

1.4.2 Simplifying Assumptions

This system of equations models a broad spectrum of behaviour. Table 1.2 lists the physical assumptions [2, 60] that will be imposed to simplify the forthcoming model.

Table 1.2: Electromagnetic Simplifying Assumptions

Let $\Omega \subset \mathbb{R}^d$ ($d \in \{2, 3\}$) be a solid medium such that it is/has: conductive ($\mathbf{P} = \vec{\mathbf{0}}$), ferromagnetic ($\mathbf{H}_{exch} \neq \vec{\mathbf{0}}$, $k_{exch} \equiv$ Exchange Stiff.) demagnetizing energy from interacting mag. moments ($\mathbf{H}_{demag} \neq \vec{\mathbf{0}}$) sitting in a vacuum subject to an external magnetic field ($\mathbf{H}_{ext} \neq \vec{\mathbf{0}}$) surface anisotropy ($\mathbf{J}_s \neq \vec{\mathbf{0}}$), no applied voltage ($\rho = 0$) uniaxially anisotropic ($\mathbf{H}_{anis} \neq \vec{\mathbf{0}}$, $k_{anis} \equiv$ Anisotropy, $\hat{\mathbf{a}} \equiv$ Easy Axis) (QSE) quasi-static electromagnetic field ($\ \mathbf{J}_f\ \gg \ \mathbf{D}_t\ $) (SA σ) Constant conductivity of material (σ is constant) (SA μ) Constant permeability of material (μ is constant)

1.4.3 Eddy Currents Equation (EC)

A PDE model for eddy currents results from manipulation of Maxwell's and constitutive equations along with use of the quasi-static/low-frequency approximation (QSE) and vector algebra/calculus identities (see Appendix):

$$\begin{aligned}
 \xrightarrow{MX4} \quad \bar{\nabla} \times \mathbf{H} &= \mathbf{J}_f + \mathbf{D}_t \xrightarrow{QSE} \bar{\nabla} \times \mathbf{H} = \mathbf{J}_f \xrightarrow{CE3} \bar{\nabla} \times \mathbf{H} = \sigma \mathbf{E} \\
 \xrightarrow{\bar{\nabla} \times} \quad \bar{\nabla} \times \bar{\nabla} \times \mathbf{H} &= \bar{\nabla} \times (\sigma \mathbf{E}) \\
 \xrightarrow{VC2} \quad \bar{\nabla} \times \bar{\nabla} \times \mathbf{H} &= \sigma (\bar{\nabla} \times \mathbf{E}) + (\bar{\nabla} \sigma) \times \mathbf{E} \\
 \xrightarrow{SA\sigma} \quad \bar{\nabla} \times \bar{\nabla} \times \mathbf{H} &= \sigma (\bar{\nabla} \times \mathbf{E}) + \vec{\mathbf{0}} \xrightarrow{MX3} \bar{\nabla} \times \bar{\nabla} \times \mathbf{H} = \sigma (-\mathbf{B}_t) \\
 \xrightarrow{CE2} \quad \sigma \mathbf{B}_t + \bar{\nabla} \times \bar{\nabla} \times \left(\frac{1}{\mu} \mathbf{B} - \mathbf{M} \right) &= \vec{\mathbf{0}} \\
 \xrightarrow{SA\mu} \quad \sigma \mathbf{B}_t + \frac{1}{\mu} \bar{\nabla} \times \bar{\nabla} \times \mathbf{B} - \bar{\nabla} \times \bar{\nabla} \times \mathbf{M} &= \vec{\mathbf{0}} \\
 \xrightarrow{VC3} \quad \sigma \mathbf{B}_t + \frac{1}{\mu} \left[\bar{\nabla} (\bar{\nabla} \cdot \mathbf{B}) - \bar{\nabla}^2 \mathbf{B} \right] - \bar{\nabla} \times \bar{\nabla} \times \mathbf{M} &= \vec{\mathbf{0}} \\
 \xrightarrow{MX2} \quad \sigma \mathbf{B}_t - \frac{1}{\mu} \bar{\nabla}^2 \mathbf{B} - \bar{\nabla} \times \bar{\nabla} \times \mathbf{M} &= \vec{\mathbf{0}}
 \end{aligned} \tag{1.3}$$

The resulting constrained PDE model is called the Eddy Currents equation:

$$\begin{cases} \mathbf{B}_t - \frac{1}{\mu\sigma}\nabla^2\mathbf{B} - \frac{1}{\sigma}\nabla \times \nabla \times \mathbf{M} = \vec{\mathbf{0}} & \text{in } \Omega \times [0, T] \\ \nabla \cdot \mathbf{B} = 0 & \text{in } \Omega \times [0, T] \end{cases} \quad (1.4)$$

Maxwell's Equations directly supply the zero divergence of \mathbf{B} , which is interpreted to mean that there are no magnetic monopoles [112].

1.4.4 Landau-Lifschitz-Gilbert Equation (LLG)

Micromagnetic behavior is modeled by the Landau-Lifschitz-Gilbert equation [18, 42, 43, 70, 71, 72]:

$$\begin{cases} \mathbf{M}_t - \frac{\gamma_0}{1+\alpha^2}(\mathbf{M} \times \mathbf{H}_{eff}) - \frac{\gamma_0\alpha}{(1+\alpha^2)M_s}\mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{eff}) = \vec{\mathbf{0}} & \text{for } t > 0 \\ \mathbf{M} = \mathbf{M}_0 & \text{at } t = 0 \end{cases} \quad (1.5)$$

The parameter $\gamma_0 < 0$ represents the gyromagnetic ratio [43, 60, 70, 89] for an electron spin while α is a dimensionless damping parameter [43, 70].

The two cross-product terms are both perpendicular to \mathbf{M} , so that the LLG equation bears a quadratic invariant, namely that the norm of the magnetization field \mathbf{M} should have constant norm for all time:

$$\begin{aligned} \mathbf{M} \times \mathbf{H}_{eff} \perp \mathbf{M} \text{ and } \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{eff}) \perp \mathbf{M} &\stackrel{LLG}{\implies} \mathbf{M}_t \perp \mathbf{M} \\ \therefore \frac{d}{dt} [|\mathbf{M}|^2] = \frac{d}{dt} \left[\frac{1}{2} \mathbf{M} \cdot \mathbf{M} \right] = \mathbf{M} \cdot \mathbf{M}_t &\stackrel{\perp}{=} 0 \\ \therefore |\mathbf{M}(t)| = M_s \quad \forall t \geq 0 & \end{aligned} \quad (1.6)$$

The effective magnetic field \mathbf{H}_{eff} is a convenient expression that allows the LLG Equation to be written more compactly. More importantly, \mathbf{H}_{eff} measures the magnetic field due to the eddy currents in addition to the magnetic field due to everything else (which the magnetic field \mathbf{H} captures by itself.) Hence, the LLG equation is interpreted to mean that the magnetization field \mathbf{M} 'spirals inward' toward the effective magnetic field \mathbf{H}_{eff} over time.

1.4.5 The Coupled PDE System (LLG+EC)

To fully describe the desired micromagnetic behaviour, the Eddy Currents and Landau-Lifschitz-Gilbert equations should be coupled together [60, 89] with homo-

geous Neumann boundary conditions imposed on \mathbf{M} . The resulting model is a system of coupled PDE's: (where $\mathbf{B}, \mathbf{M} \in [C^{(2,1)}(\Omega \times [0, T])]^d$)

$$\left\{ \begin{array}{l}
 \mathbf{B}_t - \frac{1}{\mu\sigma} \overline{\nabla}^2 \mathbf{B} - \frac{1}{\sigma} \overline{\nabla} \times \overline{\nabla} \times \mathbf{M} = \vec{\mathbf{0}} \quad \text{in } \Omega \times [0, T] \\
 \nabla \cdot \mathbf{B} = 0 \quad \text{in } \Omega \times [0, T] \\
 \hline
 \frac{1}{\mu} \mathbf{B} - \mathbf{M} + \frac{2k_{anis}}{M_s} (\mathbf{M} \cdot \hat{\mathbf{a}}) \hat{\mathbf{a}} + \frac{k_{exch}}{M_s^2} \overline{\nabla}^2 \mathbf{M} = \mathbf{H}_{eff} \quad \text{in } \Omega \times [0, T] \\
 \mathbf{M}_t - \frac{\gamma_0}{1+\alpha^2} (\mathbf{M} \times \mathbf{H}_{eff}) - \frac{\gamma_0 \alpha}{(1+\alpha^2) M_s} \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{eff}) = \vec{\mathbf{0}} \quad \text{in } \Omega \times [0, T] \\
 \hline
 \mathbf{B} = \mathbf{B}_D \quad \text{on } \partial\Omega \times [0, T] \\
 \partial\mathbf{M}/\partial n = \vec{\mathbf{0}} \quad \text{on } \partial\Omega \times [0, T] \\
 \hline
 \mathbf{B} = \mathbf{B}_0 \quad \text{at } \overline{\Omega} \times \{t = 0\} \\
 \mathbf{M} = \mathbf{M}_0 \quad \text{at } \overline{\Omega} \times \{t = 0\}
 \end{array} \right. \quad (1.7)$$

We shall assume that the solid conductor's temperature remains perpetually below the Curie temperature. This assumption together with (1.6) means that this coupled system also has a quadratic invariant, this time the magnetization field's norm remains constant for all time everywhere in the solid.

This dissertation shows how to construct a high-order numerical method for the coupled eddy-currents-micromagnetics model with inhomogeneous Dirichlet boundary conditions for the eddy currents equation and homogeneous Neumann boundary conditions for the micromagnetics equation. The method will employ high-order finite elements in order to use domains with complicated geometries, use coarser meshes of the domain, and result in sparse linear systems that can be handled by direct and Krylov solvers. Invoked timesteppers will be high-order implicit schemes to allow coarser timestep sizes while maintaining key stability and invariance properties. Validation tests confirming correct behavior of a FEniCS-based simulation code are shown.

CHAPTER II

TIMESTEPPERS: CONSTRUCTION & ORDER OF ACCURACY

This chapter is a review of standard theory for timestepping, with an eye towards the requirements for our problem.

2.1 Model Initial Value Problem

In order to develop an appropriate timestepper for our micromagnetics problem at hand, a model initial value problem (IVP) shall be assumed along with assumptions about the initial data. Here are the assumed scalar model IVP and vector model IVP respectively [52, 59]:

$$\text{Model IVP (MIVP):} \quad \begin{cases} u'(t) = \Psi(t; u) \\ u(t_0) = u_0 \end{cases} \quad (2.1)$$

$$\text{Model Vector IVP (MVIVP):} \quad \begin{cases} \mathbf{u}'(t) = \mathbf{\Psi}(t; \mathbf{u}) \\ \mathbf{u}(t_0) = \mathbf{u}_0 \end{cases} \quad (2.2)$$

The solution u and right-hand side (RHS) Ψ are at a bare minimum continuously differentiable, but more smoothness may be required for certain classes of timesteppers [77]. Similar requirements are imposed on the components of u and RHS Ψ .

For our particular micromagnetics problem and most real-world problems, the vector model IVP is utilized, but when constructing and analyzing timesteppers, it is sufficient to use the scalar model IVP [77].

2.2 Timestepper Order of Accuracy

One expects that a given timestepper will become more accurate when using smaller and smaller timestep sizes Δt . We would like a measure of “how fast” a timestepper becomes more accurate with decreasing Δt . The desired measure is called order of accuracy.

2.2.1 Local Truncation Error

Order of accuracy can be found by computing the timestepper's local truncation error [52]. Local truncation error is most-commonly expressed as in [59]:

Definition 2.2.1. (Local Truncation Error)

Given the d -dimensional model vector IVP (MVIVP) over time interval $[t_0, T]$ containing timesteps t_1, \dots, t_k and an arbitrary timestepper

$$\mathbf{u}^{(k+1)} = \mathcal{U}_k(\Psi, \Delta t; \mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)})$$

where mapping $\mathcal{U}_k : [C^0([t_0, T])]^d \times \mathbb{R} \times \prod_{j=1}^{k+1} \mathbb{R}^d \rightarrow \mathbb{R}^d$, then the timestepper has order of accuracy $p > 0$, or is p^{th} -order, if its local truncation error

$$\text{LTE} := \mathbf{u}(t_{k+1}) - \mathcal{U}_k(\Psi, \Delta t; \mathbf{u}(t_0), \mathbf{u}(t_1), \dots, \mathbf{u}(t_k)) = \mathcal{O}((\Delta t)^{p+1})$$

The big-oh notation $\mathcal{O}(\cdot)$ is defined with examples in [52, 77].

2.2.2 Global Error

We wish to know the timestepper's total error in the numerical solution at $t = T$:

Definition 2.2.2. (Global Error)

Given the d -dimensional model vector IVP (MVIVP) over time interval $[t_0, T]$ containing all N_T timesteps $t_1, \dots, t_k, t_{k+1}, \dots, t_{N_T} = T$ and an arbitrary timestepper

$$\mathbf{u}^{(k+1)} = \mathcal{U}_k(\Psi, \Delta t; \mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)})$$

then the global error in the timestepper at $t = T$ is

$$\text{GE} := \mathbf{u}(T) - \mathbf{u}^{(N_T)}$$

One can expect that the global error will usually be an order of magnitude higher than the local truncation error, because informally we can estimate:

$$GE \approx \sum_{j=1}^{N_T} \text{LTE} = N_T \cdot \mathcal{O}((\Delta t)^{p+1}) = \frac{T}{\Delta t} \cdot \mathcal{O}((\Delta t)^{p+1}) = \mathcal{O}((\Delta t)^p) \quad (2.3)$$

Rigorously showing this when applying a particular timestepper to a particular IVP is intricate and tedious – see [59, 77] for examples. It is common to use the informal estimate as a “rule of thumb” for the global error; it needs to be tested experimentally on each new class of problem [73, 97, 102].

2.2.3 Order of Accuracy does not Guarantee Convergence

Suppose a p^{th} -order timestepper is applied to a given IVP over a time interval $[0, T]$. Then, one may naively assume that for a chosen Δt that the numerical solution will be “close” to the exact solution; and with decreasing Δt , it is expected to “converge” to the exact solution. Unfortunately, this is not true in general [52, 59, 77]. Certain additional conditions must be satisfied in order to ensure this assumed behavior. This will be explored much further in Chapter 3.

2.3 Taylor Series (TS) Timesteppers

For a sufficiently differentiable solution in time, a timestepper can be developed using the truncated Taylor series of the solution with its derivatives written in terms of derivatives of the IVP’s RHS function [52, 77]:

Definition 2.3.1. (Taylor Series Timesteppers)

The general p -derivative Taylor series timestepper, denoted $\text{TS}p$, is:

$$u^{(k+1)} = u^{(k)} + \sum_{j=1}^p \frac{(\Delta t)^j}{j!} \cdot \left. \frac{d^j u}{dt^j} \right|_{t=t_k} \quad (2.4)$$

where the derivatives of u are to be fully written out in terms of model IVP right-hand side Ψ and its partial derivatives $\Psi_t, \Psi_u, \Psi_{uu}, \Psi_{tu}, \Psi_{ut}, \Psi_{tt}, \dots$ using the 2-Variable Chain Rule in the Appendix.

Proposition 2.3.1. (Taylor Series Timesteppers – Order of Accuracy)

The p -derivative Taylor series timestepper, $\text{TS}p$, is p^{th} -order.

Proof. Referring to the error bounds for Taylor polynomial approximation, it is clear that the local truncation error $\text{LTE}_{\text{TS}p} = \mathcal{O}((\Delta t)^{p+1})$ □

Below are the first three p -derivative Taylor Series timesteppers:

Table 2.1: Taylor Series Timesteppers [52, 77]

TS1 :	$u^{(k+1)} = u^{(k)} + \frac{\Delta t}{1!} \left[\Psi^{(k)} \right]$
TS2 :	$u^{(k+1)} = u^{(k)} + \frac{\Delta t}{1!} \left[\Psi^{(k)} \right] + \frac{(\Delta t)^2}{2!} \left[\Psi_t^{(k)} + \Psi_u^{(k)} \cdot \Psi^{(k)} \right]$
TS3 :	$u^{(k+1)} = u^{(k)} + \frac{\Delta t}{1!} \left[\Psi^{(k)} \right] + \frac{(\Delta t)^2}{2!} \left[\Psi_t^{(k)} + \Psi_u^{(k)} \cdot \Psi^{(k)} \right]$ $+ \frac{(\Delta t)^3}{3!} \left[\Psi_{tu}^{(k)} \cdot \Psi^{(k)} + \left(\Psi_{ut}^{(k)} + \Psi_{uu}^{(k)} \cdot \Psi^{(k)} \right) \cdot \Psi^{(k)} \right]$ $+ \Psi_{tt}^{(k)} + \left(\Psi_t^{(k)} + \Psi_u^{(k)} \cdot \Psi^{(k)} \right) \cdot \Psi_u^{(k)}$

Unfortunately, there are several major problems with this approach. First, partial derivatives of the RHS Ψ must be known a priori [52, 77]. Second, high-order timesteppers are extremely tedious to fully write out due to the Two-Variable Chain Rule. Third, when the model IVP is vector-valued rather than scalar-valued, the derivatives of the solution involve consecutively higher-order tensors which require more memory to store. For these reasons, a better timestepper needs to be employed for our micromagnetics problem.

2.4 Linear Multistep (LM) Timesteppers

Here is our second attempt toward a viable timestepper for our problem.

Let's consider the general r -step linear multistep timestepper [52, 59, 77]:

$$\sum_{j=0}^r a_j u^{(k+j)} = (\Delta t) \sum_{j=0}^r b_j \Psi(t_{k+j}; u^{(k+j)}) \quad (2.5)$$

where the coefficients $a_0, \dots, a_r; b_0, \dots, b_r \in \mathbb{R}$.

This has the advantages over Taylor Series timesteppers in that no explicit computation of derivatives is necessary and for vector IVP's all the terms are vectors, not higher-order tensors.

There are three popular classes of linear multistep timesteppers.

2.4.1 Adams-Bashforth Timesteppers

First up, the Adams-Bashforth class of explicit LM timesteppers [52, 59, 77]:

Definition 2.4.1. (Adams-Bashforth Timesteppers)

Imposing the following requirements on the general LM timestepper (2.6)

$$a_0 = a_1 = \cdots = a_{r-2} := 0, \quad a_{r-1} := -1, \quad a_r := 1 \quad \text{and} \quad b_r := 0$$

results in the r -step Adams-Bashforth timestepper, denoted ABr :

$$u^{(k+r)} - u^{(k+r-1)} = (\Delta t) \sum_{j=0}^{r-1} b_j \Psi(t_{k+j}; u^{(k+j)})$$

where coefficients b_0, \dots, b_{r-1} are chosen to maximize the order of accuracy.

Proposition 2.4.1. (Adams-Bashforth – Order of Accuracy)

The r -step Adams-Bashforth timestepper, ABr , has order r .

Table 2.2: Adams-Bashforth LM Timesteppers [52, 77]

AB1 :	$u^{(k+1)} - u^{(k)}$	$=$	$(\Delta t) \left[\Psi^{(k)} \right]$
AB2 :	$u^{(k+2)} - u^{(k+1)}$	$=$	$(\Delta t) \left[\frac{3}{2} \Psi^{(k+1)} - \frac{1}{2} \Psi^{(k)} \right]$
AB3 :	$u^{(k+3)} - u^{(k+2)}$	$=$	$(\Delta t) \left[\frac{23}{12} \Psi^{(k+2)} - \frac{4}{3} \Psi^{(k+1)} + \frac{5}{12} \Psi^{(k)} \right]$

2.4.2 Adams-Moulton Timesteppers

Next, we have the Adams-Moulton class of implicit LM timesteppers [52, 59, 77]:

Definition 2.4.2. (Adams-Moulton Timesteppers)

Imposing the following requirements on the general LM timestepper (2.6)

$$a_0 = a_1 = \cdots = a_{r-2} := 0, \quad a_{r-1} := -1, \quad a_r := 1$$

results in the r -step Adams-Moulton timestepper, denoted AMr :

$$u^{(k+r)} - u^{(k+r-1)} = (\Delta t) \sum_{j=0}^r b_j \Psi(t_{k+j}; u^{(k+j)})$$

The coefficients b_0, \dots, b_r are chosen to maximize the order of accuracy.

Proposition 2.4.2. (Adams-Moulton – Order of Accuracy)

The r -step Adams-Moulton timestepper, AM r , has order $(r + 1)$.

Table 2.3: Adams-Moulton LM Timesteppers [52, 77]

AM1 :	$u^{(k+1)} - u^{(k)}$	$=$	$(\Delta t) \left[\frac{1}{2} \Psi^{(k+1)} + \frac{1}{2} \Psi^{(k)} \right]$
AM2 :	$u^{(k+2)} - u^{(k+1)}$	$=$	$(\Delta t) \left[\frac{5}{12} \Psi^{(k+2)} + \frac{2}{3} \Psi^{(k+1)} - \frac{1}{12} \Psi^{(k)} \right]$
AM3 :	$u^{(k+3)} - u^{(k+2)}$	$=$	$(\Delta t) \left[\frac{3}{8} \Psi^{(k+3)} + \frac{19}{24} \Psi^{(k+2)} - \frac{5}{24} \Psi^{(k+1)} + \frac{1}{24} \Psi^{(k)} \right]$

2.4.3 Backward Differentiation Formulas

Finally, there are the implicit Backward Differentiation Formulas [52, 59, 77]:

Definition 2.4.3. (Backward Differentiation Formulas)

Imposing the following requirements on the general LM timestepper (2.6)

$$b_0 = b_1 = \dots = b_{r-1} := 0$$

results in the r -step Backward Differentiation Formula, denoted BDF r :

$$\sum_{j=0}^r a_j u^{(k+j)} = (\Delta t) b_r \Psi(t_{k+r}; u^{(k+r)})$$

The coefficients $b_r; a_0, \dots, a_r$ are chosen to maximize the order of accuracy.

Proposition 2.4.3. (Backward Differentiation Formulas – Order of Accuracy)

The r -step Backward Differentiation Formula, BDF r , has order r .

Table 2.4: Backward Differentiation Formula LM Timesteppers [77]

BDF1 :	$u^{(k+1)} - u^{(k)} = (\Delta t)\Psi^{(k+1)}$
BDF2 :	$\frac{3}{2}u^{(k+2)} - 2u^{(k+1)} + \frac{1}{2}u^{(k)} = (\Delta t)\Psi^{(k+2)}$
BDF3 :	$\frac{11}{6}u^{(k+3)} - 3u^{(k+2)} + \frac{3}{2}u^{(k+1)} - \frac{1}{3}u^{(k)} = (\Delta t)\Psi^{(k+3)}$
BDF4 :	$\frac{25}{12}u^{(k+4)} - 4u^{(k+3)} + 3u^{(k+2)} - \frac{4}{3}u^{(k+1)} + \frac{1}{4}u^{(k)} = (\Delta t)\Psi^{(k+4)}$

2.5 Explicit Runge-Kutta (ERK) Timesteppers

Here is another way to construct timesteppers: given current computed solution $u^{(k)}$ at timestep t_k , instead of computing multiple timesteps beyond t_{k+1} each iteration (e.g. t_{k+2}, t_{k+3}, \dots), one can compute intermediate timesteps $t_{k+\alpha_i}$ where $0 \leq \alpha_i \leq 1$ in order to determine $u^{(k+1)}$ in a single step. The i^{th} intermediate timestep is handled in its own stage denoted by K_i . Such timesteppers are known as Runge-Kutta (RK) timesteppers [52, 59, 77].

The most computationally stepwise efficient Runge-Kutta methods are explicit [59, 77] in which each successive stage K_i depends only on previous stages K_1, \dots, K_{i-1} . The resulting code only involves function evaluations at each iteration, which makes it easier for the programmer to write.

Definition 2.5.1. (Explicit Runge-Kutta Timestepper)

TIMESTEPPER FORMALLY WRITTEN OUT: $[t_k := t_0 + k(\Delta t), u^{(k)} := u(t_k)]$

$$\left\{ \begin{array}{l} K_1 = \Psi(t_k; u^{(k)}) \\ K_2 = \Psi(t_k + \alpha_2(\Delta t); u^{(k)} + (\Delta t)[\beta_{21}K_1]) \\ K_3 = \Psi(t_k + \alpha_3(\Delta t); u^{(k)} + (\Delta t)[\beta_{31}K_1 + \beta_{32}K_2]) \\ \vdots \\ K_s = \Psi(t_k + \alpha_s(\Delta t); u^{(k)} + (\Delta t)[\beta_{s1}K_1 + \beta_{s2}K_2 + \dots + \beta_{s,s-1}K_{s-1}]) \\ \hline u^{(k+1)} = u^{(k)} + (\Delta t)[\gamma_1K_1 + \gamma_2K_2 + \dots + \gamma_{s-1}K_{s-1} + \gamma_sK_s] \end{array} \right.$$

$$\begin{array}{r}
 \text{TIMESTEPPER WRITTEN} \\
 \text{AS A} \\
 \text{BUTCHER TABLE:}
 \end{array}
 \begin{array}{c}
 0 \\
 \alpha_2 \\
 \alpha_3 \\
 \vdots \\
 \alpha_s \\
 \hline
 \end{array}
 \begin{array}{cccc}
 & \beta_{21} & & \\
 & \beta_{31} & \beta_{32} & \\
 & \vdots & \vdots & \ddots \\
 & \beta_{s1} & \beta_{s2} & \cdots & \beta_{s,s-1} \\
 \hline
 & \gamma_1 & \gamma_2 & \cdots & \gamma_{s-1} & \gamma_s
 \end{array}$$

The above Butcher table notation [49, 50, 59, 77] (first used by John C. Butcher in [19]) is a concise representation of a Runge-Kutta timestepper.

The matrix $[\beta_{ij}]_{s \times s}$ is strictly lower-triangular in an ERK timestepper.

2.6 Implicit Runge-Kutta (IRK) Timesteppers

A Runge-Kutta timestepper is implicit [59, 77] if there is at least one stage that depends on itself or at least one later stage. This results in having to solve a (possibly) nonlinear system of algebraic equations each iteration, which is harder to program correctly and more computationally expensive per timestep. However, the potential benefits gleaned from that work are nice stability properties that explicit Runge-Kutta timesteppers completely lack – more on this in Chapter 3.

Definition 2.6.1. (Implicit Runge-Kutta Timestepper)

TIMESTEPPER FORMALLY WRITTEN OUT: $[t_k := t_0 + k(\Delta t), \quad u^{(k)} := u(t_k)]$

$$\left\{ \begin{array}{l}
 K_1 = \Psi(t_k + \alpha_1(\Delta t); u^{(k)} + (\Delta t)[\beta_{11}K_1 + \beta_{12}K_2 + \cdots + \beta_{1s}K_s]) \\
 K_2 = \Psi(t_k + \alpha_2(\Delta t); u^{(k)} + (\Delta t)[\beta_{21}K_1 + \beta_{22}K_2 + \cdots + \beta_{2s}K_s]) \\
 K_3 = \Psi(t_k + \alpha_3(\Delta t); u^{(k)} + (\Delta t)[\beta_{31}K_1 + \beta_{32}K_2 + \cdots + \beta_{3s}K_s]) \\
 \vdots \\
 K_s = \Psi(t_k + \alpha_s(\Delta t); u^{(k)} + (\Delta t)[\beta_{s1}K_1 + \beta_{s2}K_2 + \cdots + \beta_{ss}K_s]) \\
 \hline
 u^{(k+1)} = u^{(k)} + (\Delta t)[\gamma_1K_1 + \gamma_2K_2 + \cdots + \gamma_{s-1}K_{s-1} + \gamma_sK_s]
 \end{array} \right.$$

TIMESTEPPER WRITTEN AS A BUTCHER TABLE:	α_1	β_{11}	β_{12}	\cdots	$\beta_{1,s-1}$	β_{1s}
	α_2	β_{21}	β_{22}	\cdots	$\beta_{2,s-1}$	β_{2s}
	α_3	β_{31}	β_{32}	\cdots	$\beta_{3,s-1}$	β_{3s}
	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
	α_s	β_{s1}	β_{s2}	\cdots	$\beta_{s,s-1}$	β_{ss}
		γ_1	γ_2	\cdots	γ_{s-1}	γ_s

The above Butcher table notation [49, 50, 59, 77] (first used by John C. Butcher in [19]) is a concise representation of a Runge-Kutta timestepper.

2.7 Runge-Kutta Order Conditions

As we'll see later on, several low-order Runge-Kutta timesteppers can easily be derived directly from an appropriate quadrature scheme. Such constructions for arbitrary timesteppers are not as straightforward. Instead, they can be constructed by solving an algebraic nonlinear system of equations involving the Butcher table values called the Runge-Kutta (RK) order conditions [59, 77].

In the next two subsections, we will hand-derive the low-order conditions for 2-stage timesteppers. Afterwards, we'll present a table of order conditions that apply to any Runge-Kutta timestepper.

2.7.1 1st-Order Conditions for 2-stage IRK Timestepper

Consider the generic 2-stage IRK timestepper with Ψ being sufficiently smooth:

$$\left\{ \begin{array}{l} K_1 = \Psi(t_k + \alpha_1(\Delta t); u^{(k)} + (\Delta t)[\beta_{11}K_1 + \beta_{12}K_2]) \\ K_2 = \Psi(t_k + \alpha_2(\Delta t); u^{(k)} + (\Delta t)[\beta_{21}K_1 + \beta_{22}K_2]) \\ \hline u^{(k+1)} = u^{(k)} + (\Delta t)[\gamma_1K_1 + \gamma_2K_2] \end{array} \right. \quad (2.6)$$

We rewrite the computed solution at t_{k+1} using a 2-variable Taylor expansion:

$$\begin{aligned} u^{(k+1)} &= u^{(k)} + \gamma_1(\Delta t)K_1 + \gamma_2(\Delta t)K_2 \\ &\stackrel{2TE}{=} u^{(k)} + \gamma_1(\Delta t) [\Psi^{(k)} + \mathcal{O}((\Delta t)^1)] \\ &\quad + \gamma_2(\Delta t) [\Psi^{(k)} + \mathcal{O}((\Delta t)^1)] \end{aligned} \quad (2.7)$$

Collecting like terms results in:

$$u^{(k+1)} = u^{(k)} + (\Delta t)\Psi^{(k)} \cdot [\gamma_1 + \gamma_2] + \mathcal{O}((\Delta t)^2) \quad (2.8)$$

Now we rewrite the exact solution at t_{k+1} using a 1-variable Taylor expansion with the assumption that the computed solution at t_k is exact (i.e. $u^{(k)} = u(t_k)$):

$$\begin{aligned} u(t_{k+1}) &= u(t_k + \Delta t) \\ &\stackrel{1TE}{=} u(t_k) + (\Delta t)u'(t_k) + \mathcal{O}((\Delta t)^2) \\ &\stackrel{MIVP}{=} u^{(k)} + (\Delta t)\Psi^{(k)} + \mathcal{O}((\Delta t)^2) \end{aligned} \quad (2.9)$$

Then we collect terms:

$$u(t_{k+1}) = u^{(k)} + (\Delta t)\Psi^{(k)} \cdot [1] + \mathcal{O}((\Delta t)^2) \quad (2.10)$$

Comparing terms between $u^{(k+1)}$ and $u(t_{k+1})$ yield the 1st-order RK conditions for any 2-stage Runge-Kutta timestepper:

$$\gamma_1 + \gamma_2 = 1 \quad (2.11)$$

2.7.2 2nd-Order Conditions for 2-stage IRK Timestepper

Consider the generic 2-stage IRK timestepper with Ψ being sufficiently smooth:

$$\left\{ \begin{array}{l} K_1 = \Psi(t_k + \alpha_1(\Delta t); u^{(k)} + (\Delta t)[\beta_{11}K_1 + \beta_{12}K_2]) \\ K_2 = \Psi(t_k + \alpha_2(\Delta t); u^{(k)} + (\Delta t)[\beta_{21}K_1 + \beta_{22}K_2]) \\ \hline u^{(k+1)} = u^{(k)} + (\Delta t)[\gamma_1K_1 + \gamma_2K_2] \end{array} \right. \quad (2.12)$$

We rewrite the computed solution at t_{k+1} using a 2-variable Taylor expansion:

$$\begin{aligned} u^{(k+1)} &= u^{(k)} + \gamma_1(\Delta t)K_1 + \gamma_2(\Delta t)K_2 \\ &\stackrel{2TE}{=} u^{(k)} + \gamma_1(\Delta t) \left[\begin{array}{l} \Psi^{(k)} + \alpha_1(\Delta t)\frac{d\Psi^{(k)}}{dt} \\ + (\Delta t)(\beta_{11}K_1 + \beta_{12}K_2)\frac{d\Psi^{(k)}}{du} + \mathcal{O}((\Delta t)^2) \end{array} \right] \\ &\quad + \gamma_2(\Delta t) \left[\begin{array}{l} \Psi^{(k)} + \alpha_2(\Delta t)\frac{d\Psi^{(k)}}{dt} \\ + (\Delta t)(\beta_{21}K_1 + \beta_{22}K_2)\frac{d\Psi^{(k)}}{du} + \mathcal{O}((\Delta t)^2) \end{array} \right] \end{aligned} \quad (2.13)$$

Collecting like terms results in:

$$\begin{aligned}
 u^{(k+1)} &= u^{(k)} + (\Delta t)\Psi^{(k)} \cdot [\gamma_1 + \gamma_2] + (\Delta t)^2 \frac{d\Psi^{(k)}}{dt} \cdot [\gamma_1\alpha_1 + \gamma_2\alpha_2] \\
 &\quad + (\Delta t)^2 \frac{d\Psi^{(k)}}{du} \cdot [(\gamma_1\beta_{11} + \gamma_2\beta_{21})K_1 + (\gamma_1\beta_{12} + \gamma_2\beta_{22})K_2] \\
 &\quad + \mathcal{O}((\Delta t)^3)
 \end{aligned} \tag{2.14}$$

Now we rewrite the exact solution at t_{k+1} using a 1-variable Taylor expansion with the assumption that the computed solution at t_k is exact (i.e. $u^{(k)} = u(t_k)$):

$$\begin{aligned}
 u(t_{k+1}) &= u(t_k + \Delta t) \\
 &\stackrel{1TE}{=} u(t_k) + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2 u''(t_k) + \mathcal{O}((\Delta t)^3) \\
 &\stackrel{MIVP}{=} u^{(k)} + (\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)^2 \frac{d\Psi^{(k)}}{dt} + \mathcal{O}((\Delta t)^3)
 \end{aligned} \tag{2.15}$$

Then we collect terms:

$$u(t_{k+1}) = u^{(k)} + (\Delta t)\Psi^{(k)} \cdot [1] + (\Delta t)^2 \frac{d\Psi^{(k)}}{dt} \cdot \left[\frac{1}{2}\right] + \mathcal{O}((\Delta t)^3) \tag{2.16}$$

Comparing terms between $u^{(k+1)}$ and $u(t_{k+1})$ yield the 2^{nd} -order RK conditions for any 2-stage Runge-Kutta timestepper:

$$\gamma_1 + \gamma_2 = 1, \quad \gamma_1\alpha_1 + \gamma_2\alpha_2 = \frac{1}{2} \tag{2.17}$$

Notice that the 2^{nd} -order RK conditions contain the 1^{st} -order RK condition.

2.7.3 General RK Order Conditions

With an understanding of how the 1^{st} -order and 2^{nd} -order RK conditions arise from a 2-stage Runge-Kutta timestepper, one can generalize these order conditions for an arbitrary s -stage timestepper. Moreover, application of more tedious Taylor expansions yield 3^{rd} -order RK conditions or higher. Table 2.5 lists all the order conditions up to 4^{th} -order [59].

The order conditions are cumulative in the sense that all previous order conditions are automatically included.

Table 2.5: Runge-Kutta Order Conditions [59]

1 st -order:	$\sum_{j=1}^s \gamma_j = 1$
2 nd -order:	$\sum_{j=1}^s \gamma_j = 1, \quad \sum_{j=1}^s \gamma_j \alpha_j = \frac{1}{2}$
3 rd -order:	$\sum_{j=1}^s \gamma_j = 1, \quad \sum_{j=1}^s \gamma_j \alpha_j = \frac{1}{2}, \quad \sum_{i=1}^s \sum_{j=1}^s \gamma_i \beta_{ij} \alpha_j = \frac{1}{6}$
4 th -order:	$\sum_{j=1}^s \gamma_j = 1, \quad \sum_{j=1}^s \gamma_j \alpha_j = \frac{1}{2}, \quad \sum_{i=1}^s \sum_{j=1}^s \gamma_i \beta_{ij} \alpha_j = \frac{1}{6}$ $\sum_{j=1}^s \gamma_j \alpha_j^3 = \frac{1}{4}, \quad \sum_{i=1}^s \sum_{j=1}^s \gamma_i \alpha_i \beta_{ij} \alpha_j = \frac{1}{8},$ $\sum_{i=1}^s \sum_{j=1}^s \gamma_i \beta_{ij} \alpha_j = \frac{1}{12}, \quad \sum_{i=1}^s \sum_{j=1}^s \sum_{k=1}^s \gamma_i \beta_{ij} \beta_{jk} \alpha_k = \frac{1}{24}$

2.8 Common ERK Timesteppers

2.8.1 Forward Euler Timestepper

The simplest ERK timestepper is the Forward Euler (FE) timestepper [52, 59, 77].

We start with the integral form of $u(t_{k+1})$:

$$u(t_{k+1}) = u(t_k) + \int_{t_k}^{t_{k+1}} \Psi(\tau; u(\tau)) d\tau \quad (2.18)$$

We approximate the integral using left-corner rectangular quadrature:

$$\int_{t_k}^{t_{k+1}} \Psi(\tau; u(\tau)) d\tau \approx (\Delta t) \Psi(t_k; u(t_k)) = (\Delta t) \Psi^{(k)} \quad (2.19)$$

Upon applying this quadrature we have the desired timestepper:

$$u^{(k+1)} = u^{(k)} + (\Delta t) \Psi^{(k)} \iff \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \iff \begin{array}{c|c} 0 & \\ \hline & 1 \end{array} \quad (2.20)$$

Going forward, unless it's more confusing than helpful, we leave blank any zero entries in matrix $[\beta_{ij}]_{s \times s}$ as we did above in (2.20) with the rightmost Butcher table.

Proposition 2.8.1. (Forward Euler Timestepper – Order of Accuracy)

Forward Euler is a first-order timestepper.

Proof. Compute Forward Euler's local truncation error (LTE) via a one-variable Taylor expansion (1TE) – see Appendix. Then use the Model IVP (MIVP) to relate the RHS Ψ to u' at timestep $t = t_k$:

$$\begin{aligned}
 \text{LTE}_{FE} &:= u(t_{k+1}) - u^{(k+1)} \quad \text{where} \quad u^{(k)} := u(t_k) \\
 &= u(t_k + \Delta t) - u^{(k+1)} \\
 &\stackrel{FE}{=} u(t_k + \Delta t) - [u^{(k)} + (\Delta t)\Psi^{(k)}] \\
 &\stackrel{1TE}{=} [u(t_k) + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2 u''(\tau_k)] - [u^{(k)} + (\Delta t)\Psi^{(k)}] \\
 &\stackrel{MIVP}{=} [u(t_k) + (\Delta t)\Psi^{(k)} + \frac{1}{2!}(\Delta t)^2 u''(\tau_k)] - [u^{(k)} + (\Delta t)\Psi^{(k)}] \\
 &= \frac{1}{2!}(\Delta t)^2 u''(\tau_k) \quad \text{where} \quad \tau_k \in [t_k, t_{k+1}]
 \end{aligned} \tag{2.21}$$

$$\therefore \text{LTE}_{FE} = \mathcal{O}((\Delta t)^2).$$

□

2.8.2 Explicit Midpoint Timestepper

The Explicit Midpoint (EM) timestepper is among the simplest two-stage Runge-Kutta schemes [59, 77]:

$$\left\{ \begin{array}{l|l} K_1 = \Psi(t_k, u^{(k)}) & 0 \\ K_2 = \Psi(t_k + \frac{1}{2}(\Delta t), u^{(k)} + \frac{1}{2}(\Delta t)K_1) & \iff \frac{1}{2} \quad \frac{1}{2} \\ \hline u^{(k+1)} = u^{(k)} + (\Delta t)K_2 & \left| \begin{array}{l} 0 \quad 1 \end{array} \right. \end{array} \right. \tag{2.22}$$

Its Butcher table values are one solution of the 2^{nd} -order RK conditions.

Its name comes from letting Ψ be solely time-dependent:

$$u^{(k+1)} = u^{(k)} + (\Delta t)\Psi^{(k+1/2)} \tag{2.23}$$

The last term in this simplified version of the timestepper is precisely Midpoint quadrature applied to the integral $\int_{t_k}^{t_{k+1}} \Psi(\tau) d\tau$.

Proposition 2.8.2. (Explicit Midpoint Timestepper – Order of Accuracy)

Explicit Midpoint is a second-order timestepper.

Proof. Compute Explicit Midpoint's local truncation error:

$$\text{LTE}_{EM} := u(t_{k+1}) - u^{(k+1)} \quad \text{where} \quad u^{(k)} := u(t_k) \quad (2.24)$$

starting with one-variable (1TE) & two-variable (2TE) Taylor expansions:

$$\begin{aligned} &= u(t_k + \Delta t) - u^{(k+1)} \\ &\stackrel{EM}{=} u(t_k + \Delta t) - \left[u^{(k)} + (\Delta t)\Psi\left(t_k + \frac{1}{2}(\Delta t); u^{(k)} + \frac{1}{2}(\Delta t)\Psi^{(k)}\right) \right] \\ &\stackrel{2TE}{=} u(t_k + \Delta t) - u^{(k)} \\ &\quad - \left\{ (\Delta t) \left[\Psi^{(k)} + \left[\frac{1}{2}(\Delta t)\right] \Psi_t^{(k)} + \left[\frac{1}{2}(\Delta t)\Psi^{(k)}\right] \Psi_u^{(k)} + \mathcal{O}((\Delta t)^2) \right] \right\} \\ &\stackrel{1TE}{=} u(t_k) + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2 u''(t_k) + \mathcal{O}((\Delta t)^3) \\ &\quad - \left\{ u^{(k)} + (\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)^2 \left[\Psi_t^{(k)} + \Psi_u^{(k)}\Psi^{(k)} \right] + \mathcal{O}((\Delta t)^3) \right\} \end{aligned} \quad (2.25)$$

Now use a two-variable Chain Rule (2CR) and then use the Model IVP (MIVP) to relate the RHS Ψ to u' at timestep $t = t_k$:

$$\begin{aligned} &\stackrel{2CR}{=} u^{(k)} + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2 \left[\Psi_t^{(k)} + \Psi_u^{(k)}u'(t_k) \right] + \mathcal{O}((\Delta t)^3) \\ &\quad - \left\{ u^{(k)} + (\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)^2 \left[\Psi_t^{(k)} + \Psi_u^{(k)}\Psi^{(k)} \right] + \mathcal{O}((\Delta t)^3) \right\} \\ &\stackrel{MIVP}{=} u^{(k)} + (\Delta t)\Psi^{(k)} + \frac{1}{2!}(\Delta t)^2 \left[\Psi_t^{(k)} + \Psi_u^{(k)}\Psi^{(k)} \right] + \mathcal{O}((\Delta t)^3) \\ &\quad - \left\{ u^{(k)} + (\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)^2 \left[\Psi_t^{(k)} + \Psi_u^{(k)}\Psi^{(k)} \right] + \mathcal{O}((\Delta t)^3) \right\} \\ &= \mathcal{O}((\Delta t)^3) \end{aligned} \quad (2.26)$$

$$\therefore \text{LTE}_{EM} = \mathcal{O}((\Delta t)^3).$$

□

2.8.3 Classic RK4 Timestepper

The earliest high-order ERK timestepper is Classic RK4 [52, 59, 77]:

$$\left\{ \begin{array}{l} K_1 = \Psi(t_k, u^{(k)}) \\ K_2 = \Psi\left(t_k + \frac{1}{2}(\Delta t), u^{(k)} + \frac{1}{2}(\Delta t)K_1\right) \\ K_3 = \Psi\left(t_k + \frac{1}{2}(\Delta t), u^{(k)} + \frac{1}{2}(\Delta t)K_2\right) \\ K_4 = \Psi\left(t_k + (\Delta t), u^{(k)} + (\Delta t)K_3\right) \\ \hline u^{(k+1)} = u^{(k)} + (\Delta t) \left[\frac{1}{6}K_1 + \frac{1}{3}K_2 + \frac{1}{3}K_3 + \frac{1}{6}K_4 \right] \end{array} \right. \iff \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ 1 & 0 & 0 & 1 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \quad (2.27)$$

Unlike Forward Euler, Classic RK4 was not originally derived using quadrature. Instead, its Butcher table values were found as one solution of the 4th-order RK conditions. That said, letting Ψ depend only on t reduces Classic RK4 to a simpler form that reveals the underlying quadrature scheme for approximating $\int_{t_k}^{t_{k+1}} \Psi(\tau) d\tau$ is Simpson's Rule [52, 77]:

$$u^{(k+1)} = u^{(k)} + (\Delta t) \left[\frac{1}{6}\Psi^{(k)} + \frac{2}{3}\Psi^{(k+1/2)} + \frac{1}{6}\Psi^{(k+1)} \right] \quad (2.28)$$

Classic RK4 is a 4th-order timestepper [52, 77].

2.9 Common IRK Timesteppers

2.9.1 Backward Euler Timestepper

The simplest IRK timestepper is the Backward Euler (BE) timestepper [52, 59, 77].

We start with the integral form of $u(t_{k+1})$:

$$u(t_{k+1}) = u(t_k) + \int_{t_k}^{t_{k+1}} \Psi(\tau; u(\tau)) d\tau \quad (2.29)$$

We approximate the integral using right-corner rectangular quadrature:

$$\int_{t_k}^{t_{k+1}} \Psi(\tau; u(\tau)) d\tau \approx (\Delta t)\Psi(t_{k+1}; u(t_{k+1})) = (\Delta t)\Psi^{(k+1)} \quad (2.30)$$

Upon applying this quadrature we have the desired timestepper:

$$u^{(k+1)} = u^{(k)} + (\Delta t)\Psi^{(k+1)} \quad \iff \quad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \quad (2.31)$$

Proposition 2.9.1. (Backward Euler Timestepper – Order of Accuracy)

Backward Euler is a first-order timestepper.

Proof. Compute Backward Euler's local truncation error:

$$\text{LTE}_{BE} := u(t_{k+1}) - u^{(k+1)} \quad \text{where} \quad u^{(k)} := u(t_k) \quad (2.32)$$

via two applications of one-variable Taylor expansion (1TE). Use the Model IVP (MIVP) to relate the RHS Ψ to u' at timestep $t = t_{k+1}$:

$$\begin{aligned}
 &= u(t_k + \Delta t) - u^{(k+1)} \\
 &\stackrel{BE}{=} u(t_k + \Delta t) - [u^{(k)} + (\Delta t)\Psi^{(k+1)}] \\
 &\stackrel{1TE}{=} [u(t_k) + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(\tau_k)] - [u^{(k)} + (\Delta t)\Psi^{(k+1)}] \\
 &\stackrel{MIVP}{=} [u(t_k) + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(\tau_k)] - [u^{(k)} + (\Delta t)u'(t_{k+1})] \quad (2.33) \\
 &\stackrel{t_{k+1}}{=} [(\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(\tau_k)] - [(\Delta t)u'(t_k + \Delta t)] \\
 &\stackrel{1TE}{=} [(\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(\tau_k)] - [(\Delta t)u'(t_k) + (\Delta t)^2u''(\xi_k)] \\
 &= \frac{1}{2!}(\Delta t)^2u''(\tau_k) - (\Delta t)^2u''(\xi_k) \quad \text{where} \quad \tau_k, \xi_k \in [t_k, t_{k+1}]
 \end{aligned}$$

$$\therefore \text{LTE}_{BE} = \mathcal{O}((\Delta t)^2).$$

□

2.9.2 Trapezoidal Timestepper

Another basic IRK timestepper is the Trapezoidal (TR) timestepper [52, 59, 77].

Integral approximation is employed via trapezoidal quadrature:

$$\int_{t_k}^{t_{k+1}} \Psi(\tau; u(\tau)) d\tau \approx (\Delta t) \left[\frac{1}{2}\Psi^{(k)} + \frac{1}{2}\Psi^{(k+1)} \right] \quad (2.34)$$

Upon applying this quadrature we have the desired timestepper:

$$u^{(k+1)} = u^{(k)} + (\Delta t) \left[\frac{1}{2}\Psi^{(k)} + \frac{1}{2}\Psi^{(k+1)} \right] \iff \begin{array}{c|cc} 0 & & \\ \hline 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad (2.35)$$

Proposition 2.9.2. (Trapezoidal Timestepper – Order of Accuracy)

Trapezoidal is a second-order timestepper.

Proof. Compute Trapezoidal's local truncation error starting with an one-variable Taylor expansion (1TE) of $u(t_k + \Delta t)$ and using the Model IVP (MIVP) to relate the RHS Ψ to u' at timesteps $t = t_k$ & $t = t_{k+1}$:

$$\begin{aligned}
 \text{LTE}_{TR} &:= u(t_{k+1}) - u^{(k+1)} \quad \text{where} \quad u^{(k)} := u(t_k) \\
 &= u(t_k + \Delta t) - u^{(k+1)} \\
 &\stackrel{TR}{=} u(t_k + \Delta t) - \left[u^{(k)} + \frac{1}{2}(\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)\Psi^{(k+1)} \right] \\
 &\stackrel{1TE}{=} \left[u(t_k) + (\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(t_k) + \frac{1}{3!}(\Delta t)^3u'''(\tau_k) \right] \\
 &\quad - \left[u^{(k)} + \frac{1}{2}(\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)\Psi^{(k+1)} \right] \\
 &= \left[(\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(t_k) + \frac{1}{3!}(\Delta t)^3u'''(\tau_k) \right] \\
 &\quad - \left[\frac{1}{2}(\Delta t)\Psi^{(k)} + \frac{1}{2}(\Delta t)\Psi^{(k+1)} \right] \\
 &\stackrel{MIVP}{=} \left[(\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(t_k) + \frac{1}{3!}(\Delta t)^3u'''(\tau_k) \right] \\
 &\quad - \left[\frac{1}{2}(\Delta t)u'(t_k) + \frac{1}{2}(\Delta t)u'(t_k + \Delta t) \right] \\
 &= \left[\frac{1}{2}(\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(t_k) + \frac{1}{3!}(\Delta t)^3u'''(\tau_k) \right] \\
 &\quad - \left[\frac{1}{2}(\Delta t)u'(t_k + \Delta t) \right]
 \end{aligned} \tag{2.36}$$

Now perform a single-variable Taylor expansion of $u'(t_k + \Delta t)$ and simplify:

$$\begin{aligned}
 &\stackrel{1TE}{=} \left[\frac{1}{2}(\Delta t)u'(t_k) + \frac{1}{2!}(\Delta t)^2u''(t_k) + \frac{1}{3!}(\Delta t)^3u'''(\tau_k) \right] \\
 &\quad - \left[\frac{1}{2}(\Delta t)u'(t_k) + \frac{1}{2}(\Delta t)^2u''(t_k) + \frac{1}{4}(\Delta t)^3u'''(\xi_k) \right] \\
 &= \frac{1}{3!}(\Delta t)^3u'''(\tau_k) - \frac{1}{4}(\Delta t)^3u'''(\xi_k) \quad \text{where} \quad \tau_k, \xi_k \in [t_k, t_{k+1}]
 \end{aligned} \tag{2.37}$$

$$\therefore \text{LTE}_{TR} = \mathcal{O}((\Delta t)^3).$$

□

2.10 Common IRK Timesteppers built from Collocation

2.10.1 (Continuous) Collocation

The construction of higher-order IRK timesteppers by hand merely from the Runge-Kutta order conditions is extremely painful at best. It can be automated with Mathematica[®] [84] or Maple[®] [83], but computation times can be long. Fortunately, the notion of collocation allows one to easily construct such an IRK timestepper using only a selected set of quadrature points and then building a polynomial that acts as a continuous solution to (MIVP) at the quadrature points [50, 59]:

Definition 2.10.1. Let the collocation points $c_1, \dots, c_s \in [0, 1]$ be distinct. Then the s -degree collocation polynomial $p(t) := \sum_{j=0}^s p_j \cdot (t - t_k)^j$ satisfies:

$$\begin{cases} p(t_k) & = & u^{(k)} \\ p'(t_k + c_j \Delta t) & = & \Psi(t_k + c_j \Delta t; p(t_k + c_j \Delta t)) \quad \text{for } j = 1, \dots, s \end{cases}$$

Computing $p(t_{k+1})$ generates the corresponding collocation timestepper.

Collocation always generates an IRK timestepper [50, 59]:

Lemma 2.10.1. (Collocation-to-IRK Lemma)

Every collocation timestepper is an IRK timestepper.

Proof. See [59]. □

However, the converse is not true – not every IRK timestepper can be constructed via collocation. See [59] for an example.

2.10.2 Discontinuous Collocation

Relaxing the polynomial degree and continuity conditions a bit in Continuous Collocation allows entire classes of IRK timesteppers to be generated that would not otherwise [50]:

Definition 2.10.2. Let the collocation points $c_2, \dots, c_{s-1} \in [0, 1]$ be distinct and let $d_1, d_s \in \mathbb{R}$. Then the $(s - 2)$ -degree discontinuous collocation polynomial $p(t) := \sum_{j=0}^{s-2} p_j \cdot (t - t_k)^j$ satisfies:

$$\begin{cases} p(t_k) & = & u^{(k)} - d_1(\Delta t) [p'(t_k) - \Psi(t_k; p(t_k))] \\ p'(t_k + c_j \Delta t) & = & \Psi(t_k + c_j \Delta t; p(t_k + c_j \Delta t)) \quad \text{for } j = 2, \dots, (s - 1) \\ p(t_{k+1}) & = & u^{(k+1)} + d_s(\Delta t) [p'(t_{k+1}) - \Psi(t_{k+1}; p(t_{k+1}))] \end{cases}$$

$p(t_{k+1})$ generates the corresponding discontinuous collocation timestepper.

2.10.3 Gauss-Legendre Timesteppers

The m -point Gauss-Legendre quadrature scheme [33, 52, 123] (first described in [39]) is:

$$\int_{-1}^1 f(x) dx \approx \sum_{j=1}^m w_j f(x_j) \quad (2.38)$$

where the weights $w_1, \dots, w_m \in \mathbb{R}$ and points $x_1, \dots, x_m \in (-1, 1)$ are chosen so that the scheme exactly integrates polynomials of degree $(2m - 1)$ and lower.

The Gauss-Legendre family of IRK timesteppers [50, 59] are constructed using collocation on Gauss-Legendre quadrature points on the interval $[0, 1]$, excluding both endpoints. The s -stage timestepper, $\text{GL}(s)$, has order $2s$, which bears the highest order per stage of all Runge-Kutta timesteppers [59]. $\text{GL}(1)$ is also known as Implicit Midpoint.

GL(2) was first constructed in 1955 [51] and was the first application of collocation: Let polynomial $p(t) := p_0 + p_1(t - t_k) + p_2(t - t_k)^2$ satisfy:

$$\begin{cases} p(t_k) &= u^{(k)} \\ p'(t_k + c_1\Delta t) &= \Psi(t_k + c_1\Delta t; p(t_k + c_1\Delta t)) \equiv \Psi^{(k+c_1)} \\ p'(t_k + c_2\Delta t) &= \Psi(t_k + c_2\Delta t; p(t_k + c_2\Delta t)) \equiv \Psi^{(k+c_2)} \end{cases} \quad (2.39)$$

where $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ and $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$

are the 2-point Gauss-Legendre quadrature pts on $[0, 1]$. Then:

$$\begin{cases} p_0 &= u^{(k)} \\ p_1 + (2c_1\Delta t)p_2 &= \Psi^{(k+c_1)} \\ p_1 + (2c_2\Delta t)p_2 &= \Psi^{(k+c_2)} \end{cases} \implies \begin{cases} p_1 &= \frac{c_2}{c_2-c_1}\Psi^{(k+c_1)} - \frac{c_1}{c_2-c_1}\Psi^{(k+c_2)} \\ (\Delta t)p_2 &= \frac{-1/2}{c_2-c_1}\Psi^{(k+c_1)} + \frac{1/2}{c_2-c_1}\Psi^{(k+c_2)} \end{cases} \quad (2.40)$$

With the p_0, p_1, p_2 expressed in terms of $u^{(k)}, \Psi^{(k+c_1)}, \Psi^{(k+c_2)}$, we compute:

$$\begin{aligned} u^{(k+1)} &= p(t_k + \Delta t) = p_0 + (\Delta t)p_1 + (\Delta t)^2p_2 \\ &= u^{(k)} + (\Delta t) \left[\frac{1}{2}\Psi^{(k+c_1)} + \frac{1}{2}\Psi^{(k+c_2)} \right] \\ &\implies \gamma_1 = \frac{1}{2} \text{ and } \gamma_2 = \frac{1}{2} \end{aligned} \quad (2.41)$$

We evaluate p at the collocation points to determine the matrix entries β_{ij} :

$$\begin{aligned} p(t_k + c_1\Delta t) &= p_0 + (c_1\Delta t)p_1 + (c_1\Delta t)^2p_2 \\ &= u^{(k)} + (\Delta t) \left[\frac{2c_1c_2 - c_1^2}{2(c_2 - c_1)}\Psi^{(k+c_1)} + \frac{-c_1^2}{2(c_2 - c_1)}\Psi^{(k+c_2)} \right] \\ &= u^{(k)} + (\Delta t) \left[\frac{1}{4}\Psi^{(k+c_1)} + \left(\frac{1}{4} - \frac{\sqrt{3}}{6} \right) \Psi^{(k+c_2)} \right] \\ &\implies \beta_{11} = \frac{1}{4} \text{ and } \beta_{12} = \frac{1}{4} - \frac{\sqrt{3}}{6} \end{aligned} \quad (2.42)$$

Similarly, evaluating $p(t_k + c_2\Delta t)$ yields $\beta_{21} = \frac{1}{4} + \frac{\sqrt{3}}{6}$ and $\beta_{22} = \frac{1}{4}$.

$$\begin{array}{l} \therefore \text{Collocation produces} \\ \text{Butcher table} \end{array} \quad \begin{array}{c|cc|cc} \alpha_1 & \beta_{11} & \beta_{12} & \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \alpha_2 & \beta_{21} & \beta_{22} & \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \gamma_1 & \gamma_2 & & \frac{1}{2} & \frac{1}{2} \end{array}$$

Below are the 1-stage thru 3-stage Gauss-Legendre timesteppers:

Table 2.6: Gauss-Legendre IRK Timesteppers [50, 59]

TIMESTEPPER:	BUTCHER TABLE:			
GL(1)	$\frac{1}{2}$		$\frac{1}{2}$	
			1	
GL(2)	$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$	
	$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	
			$\frac{1}{2}$	$\frac{1}{2}$
GL(3)	$\frac{1}{2} - \frac{\sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{2}{9} - \frac{\sqrt{15}}{15}$	$\frac{5}{36} - \frac{\sqrt{15}}{30}$
	$\frac{1}{2}$	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$
	$\frac{1}{2} + \frac{\sqrt{15}}{10}$	$\frac{5}{36} + \frac{\sqrt{15}}{30}$	$\frac{2}{9} + \frac{\sqrt{15}}{15}$	$\frac{5}{36}$
			$\frac{5}{18}$	$\frac{4}{9}$

2.10.4 Radau Timesteppers

There are several IRK timesteppers based on Radau quadrature points. Quadrature on an interval is called RadauI quadrature if the left boundary of the interval is one of the quadrature points but the right boundary is not – the vice-versa case is known as RadauII quadrature.

The m -point RadauI quadrature scheme [33, 52, 123] (first described in [101]) is:

$$\int_{-1}^1 f(x) dx \approx \omega_{-1}f(-1) + \sum_{j=2}^m w_j f(x_j) \quad (2.43)$$

where the weights $\omega_{-1}; w_2, \dots, w_m \in \mathbb{R}$ and points $x_2, \dots, x_m \in (-1, 1)$ are chosen so that the scheme exactly integrates polynomials of degree $(2m - 2)$ and lower.

The m -point RadauII quadrature scheme [33, 52, 123] (first described in [101]) is:

$$\int_{-1}^1 f(x) dx \approx \omega_1 f(1) + \sum_{j=2}^m w_j f(x_j) \quad (2.44)$$

where the scheme exactly integrates up to $(2m - 2)$ -degree polynomials.

Several classes of timesteppers can be constructed [50] using collocation on RadauI or RadauII quadrature points on the interval $[0, 1]$. An s -stage Radau timestepper has order $(2s - 1)$. Below are some Radau timesteppers:

Table 2.7: Radau IRK Timesteppers [37, 40]

TIME-STEP- PER:	BUTCHER TABLE:	TIME-STEP- PER:	BUTCHER TABLE:		
Radau IA(2)	$ \begin{array}{c cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array} $	Radau IA(3)	$ \begin{array}{c ccc} 0 & \frac{1}{9} & \frac{-20-20\sqrt{6}}{360} & \frac{-20+20\sqrt{6}}{360} \\ \frac{6-\sqrt{6}}{10} & \frac{1}{9} & \frac{88+7\sqrt{6}}{360} & \frac{88-43\sqrt{6}}{360} \\ \frac{6+\sqrt{6}}{10} & \frac{1}{9} & \frac{88+43\sqrt{6}}{360} & \frac{88-7\sqrt{6}}{360} \\ \hline & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36} \end{array} $		
Radau IB(2)	$ \begin{array}{c cc} 0 & \frac{1}{8} & -\frac{1}{8} \\ \frac{2}{3} & \frac{7}{24} & \frac{3}{8} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array} $	Radau IB(3)	$ \begin{array}{c ccc} 0 & \frac{1}{18} & \frac{-1-\sqrt{6}}{36} & \frac{-1+\sqrt{6}}{36} \\ \frac{6-\sqrt{6}}{10} & \frac{52+3\sqrt{6}}{450} & \frac{16+\sqrt{6}}{72} & \frac{472-217\sqrt{6}}{1800} \\ \frac{6+\sqrt{6}}{10} & \frac{52-3\sqrt{6}}{450} & \frac{472+217\sqrt{6}}{1800} & \frac{16-\sqrt{6}}{72} \\ \hline & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36} \end{array} $		
Radau IIA(2)	$ \begin{array}{c cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} $	Radau IIA(3)	$ \begin{array}{c ccc} \frac{4-\sqrt{6}}{10} & \frac{440-35\sqrt{6}}{1800} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{440+35\sqrt{6}}{1800} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array} $		
Radau IIB(2)	$ \begin{array}{c cc} \frac{1}{3} & \frac{3}{8} & -\frac{1}{24} \\ 1 & \frac{7}{8} & \frac{1}{8} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} $	Radau IIB(3)	$ \begin{array}{c ccc} \frac{4-\sqrt{6}}{10} & \frac{16-\sqrt{6}}{72} & \frac{328-167\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{450} \\ \frac{4+\sqrt{6}}{10} & \frac{328+167\sqrt{6}}{1800} & \frac{16+\sqrt{6}}{72} & \frac{-2-3\sqrt{6}}{450} \\ 1 & \frac{85-10\sqrt{6}}{180} & \frac{85+10\sqrt{6}}{180} & \frac{1}{18} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array} $		

Other variants exist, such as Radau I and Radau II [20], but are omitted here.

2.10.5 Lobatto Timesteppers

Quadrature on an interval is called Lobatto quadrature (or LobattoIII quadrature) if both boundaries of the interval are quadrature points.

The m -point Lobatto quadrature scheme [33, 52, 123] (first seen in [79]) is:

$$\int_{-1}^1 f(x) dx \approx \omega_{-1}f(-1) + \omega_1f(1) + \sum_{j=3}^m w_j f(x_j) \quad (2.45)$$

where the scheme exactly integrates up to $(2m - 3)$ -degree polynomials.

Several classes of s -stage Lobatto timesteppers can be constructed using collocation on Lobatto quadrature points [50] on interval $[0, 1]$ and having order $(2s - 2)$.

Table 2.8: Lobatto IRK Timesteppers [19, 21, 24, 37, 40]

TIMESTEPPER:	BUTCHER TABLE:				TIMESTEPPER:	BUTCHER TABLE:			
Lobatto IIIA(2)	0	0	0		Lobatto IIIB(2)	0	$\frac{1}{2}$	0	
	1	$\frac{1}{2}$	$\frac{1}{2}$			1	$\frac{1}{2}$	0	
		$\frac{1}{2}$	$\frac{1}{2}$				$\frac{1}{2}$	$\frac{1}{2}$	
Lobatto IIIA(3)	0	0	0	0	Lobatto IIIB(3)	0	$\frac{1}{6}$	$-\frac{1}{6}$	0
	$\frac{1}{2}$	$\frac{5}{24}$	$\frac{1}{3}$	$-\frac{1}{24}$		$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{3}$	0
	1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$		1	$\frac{1}{6}$	$\frac{5}{6}$	0
		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$			$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
Lobatto IIIC(2)	0	$\frac{1}{2}$	$-\frac{1}{2}$		Lobatto IIIS(2)	0	$\frac{1}{4}$	0	
	1	$\frac{1}{2}$	$\frac{1}{2}$			1	$\frac{1}{2}$	$\frac{1}{4}$	
		$\frac{1}{2}$	$\frac{1}{2}$				$\frac{1}{2}$	$\frac{1}{2}$	
Lobatto IIIC(3)	0	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{6}$	Lobatto IIIS(3)	0	$\frac{1}{12}$	$-\frac{1}{12}$	0
	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{5}{12}$	$-\frac{1}{12}$		$\frac{1}{2}$	$\frac{3}{16}$	$\frac{1}{3}$	$-\frac{1}{48}$
	1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$		1	$\frac{1}{6}$	$\frac{3}{4}$	$\frac{1}{12}$
		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$			$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Other variants exist, such as Lobatto IIIC* [19, 37], but are omitted.

2.11 Diagonally Implicit Runge-Kutta (DIRK) Timesteppers

An s -stage IRK timestepper with a lower-triangular stage matrix $[\beta_{ij}]_{s \times s}$ is called a diagonally implicit Runge-Kutta (DIRK) timestepper [3, 49, 66]:

Definition 2.11.1. (Diagonally Implicit Runge-Kutta Timestepper)

$$\begin{array}{l}
 \text{TIMESTEPPER WRITTEN} \\
 \text{AS A} \\
 \text{BUTCHER TABLE:}
 \end{array}
 \begin{array}{c}
 \alpha_1 \\
 \alpha_2 \\
 \vdots \\
 \alpha_s
 \end{array}
 \left|
 \begin{array}{cccc}
 \beta_{11} & & & \\
 \beta_{21} & \beta_{22} & & \\
 \vdots & \vdots & \ddots & \\
 \beta_{s1} & \beta_{s2} & \cdots & \beta_{ss}
 \end{array}
 \right.
 \begin{array}{c}
 \gamma_1 \\
 \gamma_2 \\
 \cdots \\
 \gamma_s
 \end{array}$$

DIRK timesteppers are more efficient per timestep than same-stage fully IRK timesteppers because each Newton system can be solved one stage at a time [3, 66].

Table 2.9: DIRK timesteppers by Spijker [68], Scherer [109] and Miller [88]

TIMESTEPPER:	BUTCHER TABLE:	ORDER:
Spijker's DIRK(2)	$ \begin{array}{c cc} \frac{1}{2} & & \frac{1}{2} \\ \frac{3}{2} & -\frac{1}{2} & 2 \\ \hline & -\frac{1}{2} & \frac{3}{2} \end{array} $	1
Scherer's DIRK(2)	$ \begin{array}{c cc} \frac{1}{3} & & \frac{1}{3} \\ \frac{7}{12} & \frac{1}{3} & \frac{1}{4} \\ \hline & \frac{1}{3} & \frac{2}{3} \end{array} $	2
Miller's DIRK(2)	$ \begin{array}{c cc} \frac{1}{3} & & \frac{1}{3} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} $	2
Miller's DIRK(3)	$ \begin{array}{c ccc} 1 & & & 1 \\ \frac{1}{3} & -\frac{1}{12} & \frac{5}{12} & \\ 1 & 0 & \frac{3}{4} & \frac{1}{4} \\ \hline & 0 & \frac{3}{4} & \frac{1}{4} \end{array} $	2

Consult [66] for a comprehensive list of DIRK timesteppers from 1971-2016.

2.12 Singly Diagonally Implicit Runge-Kutta (SDIRK) Timesteppers

DIRK timesteppers with identical main diagonal entries are known as singly diagonally implicit Runge-Kutta (SDIRK) timesteppers [3, 49, 66]:

Definition 2.12.1. (Singly Diagonally Implicit Runge-Kutta Timestepper)

$$\begin{array}{c}
 \text{TIMESTEPPER WRITTEN} \\
 \text{AS A} \\
 \text{BUTCHER TABLE:}
 \end{array}
 \begin{array}{c|ccc}
 \alpha_1 & \beta & & \\
 \alpha_2 & \beta_{21} & \beta & \\
 \vdots & \vdots & \vdots & \ddots \\
 \alpha_s & \beta_{s1} & \beta_{s2} & \cdots & \beta \\
 \hline
 & \gamma_1 & \gamma_2 & \cdots & \gamma_s
 \end{array}$$

SDIRK timesteppers are nice in that for each timestep, the Jacobian matrix can be LU-factored once and reused every Newton iteration [3, 49].

Table 2.10: SDIRK timesteppers by Geng [40] and Crouzeix [3]

TIME-STEPPER:	BUTCHER TABLE:	ORDER:
Geng's SDIRK(2)	$ \begin{array}{c cc} \frac{1}{4} & \frac{1}{4} & \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} $	2
Crouzeix's SDIRK(2)	$ \begin{array}{c cc} \frac{1}{2} + \frac{1}{2\sqrt{3}} & \frac{1}{2} + \frac{1}{2\sqrt{3}} & \\ \frac{1}{2} - \frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{2} + \frac{1}{2\sqrt{3}} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} $	2
Crouzeix's SDIRK(3)	$ \begin{array}{c ccc} \rho & \rho & & \\ \rho_+ & \rho_- & \rho & \\ 1 & \rho_1 & \rho_2 & \rho \\ \hline & \rho_1 & \rho_2 & \rho \end{array} \left\{ \begin{array}{l} \rho \in (\frac{1}{6}, \frac{1}{2}) \\ \rho^3 - 3\rho^2 + \frac{3}{2}\rho - \frac{1}{6} = 0 \\ \rho_+ := \frac{1}{2}(1 + \rho), \rho_- := \frac{1}{2}(1 - \rho) \\ \rho_1 := -(6\rho^2 - 16\rho + 1)/4 \\ \rho_2 := (6\rho^2 - 20\rho + 5)/4 \end{array} \right. $	3

Consult [66] for a comprehensive list of SDIRK timesteppers from 1974-2016.

CHAPTER III
TIMESTEPPERS: CONSISTENCY, STABILITY & INVARIANCE

While minimization of the global error in the computed solution is the goal when selecting a timestepper and timestep size, some problems possess an invariant quantity or qualitative behavior that we expect to be preserved when applying a timestepper. Moreover, we desire to use timesteppers that are “reliable” even when using very coarse timestep sizes. This chapter details the properties that a timestepper must have in order to achieve some of these expectations.

3.1 Consistency

As a basic measure of a reasonable timestepper, when the timestep size Δt shrinks to zero, we expect its local truncation error size to decay to zero faster than Δt :

Definition 3.1.1. A timestepper is consistent [77] if its LTE is $o(\Delta t)$. That is,

$$\left| \frac{\text{LTE}}{\Delta t} \right| \rightarrow 0 \text{ as } \Delta t \rightarrow 0$$

Naturally, non-consistent timesteppers should never be used at all.

3.1.1 Consistency w.r.t. Linear Multistep Timesteppers

Recall the general r -step linear multistep timestepper for the model IVP (MIVP):

$$\text{LM}r : \quad \sum_{j=0}^r a_j u^{(k+j)} = (\Delta t) \sum_{j=0}^r b_j \Psi^{(k+j)} \quad (3.1)$$

Assuming that u and Ψ are sufficiently smooth, that the computed solution is exact for the previous timesteps, and that $a_r = 1$ to ensure unique coefficients, we

compute its local truncation error via a one-variable Taylor expansion (1TE):

$$\begin{aligned}
 \text{LTE}_{LMr} &:= u(t_k + r\Delta t) - u^{(k+r)} \\
 &\stackrel{LMr}{=} u(t_k + r\Delta t) - \left[\sum_{j=0}^r (\Delta t) b_j \Psi^{(k+j)} - \sum_{j=0}^{r-1} a_j u^{(k+j)} \right] \\
 &\stackrel{MIVP}{=} u(t_k + r\Delta t) - \left[\sum_{j=0}^r (\Delta t) b_j u'(t_k + j\Delta t) - \sum_{j=0}^{r-1} a_j u(t_k + j\Delta t) \right] \\
 &\stackrel{1TE}{=} u(t_k) + r(\Delta t)u'(t_k) + \frac{1}{2!}(r\Delta t)^2 u''(\tau_{k,-1}) \\
 &\quad - (\Delta t) \sum_{j=0}^r b_j [u'(t_k) + j(\Delta t)u''(\tau_{k,j})] \\
 &\quad + \sum_{j=0}^{r-1} a_j \left[u(t_k) + j(\Delta t)u'(t_k) + \frac{1}{2!}(j\Delta t)^2 u''(\tau_{k,j+r+1}) \right] \\
 &= \left[1 + \sum_{j=0}^{r-1} a_j \right] u(t_k) + (\Delta t) \left[r + \sum_{j=0}^{r-1} j a_j - \sum_{j=0}^r b_j \right] u'(t_k) \\
 &\quad + (\Delta t)^2 \left[\frac{1}{2!} r u''(\tau_{r,-1}) - \sum_{j=0}^r j b_j u''(\tau_{k,j}) + \sum_{j=0}^{r-1} \frac{1}{2!} j^2 a_j u''(\tau_{k,j+r+1}) \right] \\
 &\stackrel{a_r}{=} \left[\sum_{j=0}^r a_j \right] u(t_k) + (\Delta t) \left[\sum_{j=0}^r j a_j - \sum_{j=0}^r b_j \right] u'(t_k) \\
 &\quad + (\Delta t)^2 \left[\frac{1}{2!} r u''(\tau_{r,-1}) - \sum_{j=0}^r j b_j u''(\tau_{k,j}) + \sum_{j=0}^{r-1} \frac{1}{2!} j^2 a_j u''(\tau_{k,j+r+1}) \right]
 \end{aligned} \tag{3.2}$$

For LMr , it is consistent precisely when the $u(t_k)$ and $u_t(t_k)$ coefficients of LTE_{LMr} are both zero:

$$\sum_{j=0}^r a_j = 0 \quad \text{and} \quad \sum_{j=0}^r j a_j = \sum_{j=0}^r b_j \tag{3.3}$$

In fact, the consistency requirements can be stated more concisely [77] by defining the following useful polynomials:

$$\rho_{LMr}(\zeta) := \sum_{j=0}^r a_j \zeta^j \quad \text{and} \quad \sigma_{LMr}(\zeta) := \sum_{j=0}^r b_j \zeta^j \quad \text{where} \quad \zeta \in \mathbb{C} \tag{3.4}$$

In terms of these polynomials, LMr is consistent if these two conditions hold:

$$\rho_{LMr}(1) = 0 \quad \text{and} \quad \rho'_{LMr}(1) = \sigma_{LMr}(1) \tag{3.5}$$

3.1.2 Consistency w.r.t. Runge-Kutta Timesteppers

Since the s -stage Runge-Kutta (RK(s)) timestepper is of the form:

$$u^{(k+1)} = u^{(k)} + (\Delta t) \sum_{i=1}^s \gamma_i K_i \quad \text{where} \quad K_i := \Psi \left(t_{k+\alpha_i}; u^{(k)} + (\Delta t) \sum_{j=1}^s \beta_{ij} K_j \right) \tag{3.6}$$

It can be viewed as a 1-step linear multistep timestepper with

$$\rho_{RK(s)}(\zeta) := \zeta - 1 \quad \text{and} \quad \sigma_{RK(s)}(\zeta) := \sum_{j=1}^s \gamma_j \zeta^j \quad (3.7)$$

Clearly, $\rho_{RK(s)}(1) = 0$. Hence, a Runge-Kutta timestepper is consistent if

$$\sum_{j=1}^s \gamma_j = 1 \quad (3.8)$$

Notice that consistency coincides with the 1st-order RK condition.

3.2 Zero-Stability

In practice, a timestepper may not necessarily work well at all for certain systems. In fact, some timesteppers (even if they're high-order) cannot handle the simplest differential equation of all [77]:

$$\text{(TIVP0):} \quad \begin{cases} u'(t) = 0 \\ u(0) = 0 \end{cases} \quad (3.9)$$

The exact solution of (TIVP0) is obviously $u(t) = 0$.

Definition 3.2.1. An r -step timestepper is zero-stable if $\forall \Delta t > 0$, applying it to (TIVP0) with starting timestep data

$$u^{(0)} = 0, \quad u^{(1)} = u^{(2)} = \dots = u^{(r-1)} = \epsilon > 0$$

results in the computed solution remaining uniformly bounded.

This means with a zero-stable timestepper applied to (TIVP0), even if the starting timestep data is perturbed from zero due to floating-point arithmetic, the computed solution's magnitude will never blow up to infinity. Timesteppers that are not zero-stable should never be used [59], period.

As an example, consider the following two timesteppers – the “bad” one we fabricated, the other one being the 2-step Backward Differentiation Formula [77]:

$$\begin{aligned} \text{”Bad” :} \quad & u^{(k+2)} - 5u^{(k+1)} + 6u^{(k)} = (\Delta t)\Psi^{(k)} \\ \text{BDF2 :} \quad & \frac{3}{2}u^{(k+2)} - 2u^{(k+1)} + \frac{1}{2}u^{(k)} = (\Delta t)\Psi^{(k+2)} \end{aligned} \quad (3.10)$$

with starting timestep data: $u^{(0)} = 0, \quad u^{(1)} = \epsilon := 10^{-15}$

Proposition 3.2.1. BDF2 is zero-stable while “bad” timestepper (3.10) is not.

Proof. Solve the timesteppers’ associated linear difference equations:

<p>Assume $u_{BAD}^{(k)} = \zeta^k$</p> $\zeta^2 - 5\zeta + 6 = 0$ $\zeta \in \{5, 6\}$ $u_{BAD}^{(k)} = c_{11} \cdot 5^k + c_{12} \cdot 6^k$ $u_{BAD}^{(k)} = 6^k \epsilon \left[1 - \left(\frac{5}{6}\right)^k \right]$ <p>$\therefore u_{BAD}^{(k)} \rightarrow \infty$ as $k \rightarrow \infty$</p> <p>$\therefore u_{BAD}^{(k)}$ blows up</p> <p>\therefore BAD is not 0-stable</p>	<p>Assume $u_{BDF2}^{(k)} = \zeta^k$</p> $\frac{3}{2}\zeta^2 - 2\zeta + \frac{1}{2} = 0$ $\zeta \in \left\{ \frac{1}{3}, 1 \right\}$ $u_{BDF2}^{(k)} = c_{21} \cdot \left(\frac{1}{3}\right)^k + c_{22} \cdot 1^k$ $u_{BDF2}^{(k)} = \frac{3}{2}\epsilon \left[1 - \left(\frac{1}{3}\right)^k \right]$ <p>$\therefore u_{BDF2}^{(k)} \rightarrow \frac{3}{2}\epsilon$ as $k \rightarrow \infty$</p> <p>$\therefore u_{BDF2}^{(k)}$ remains uniformly bounded</p> <p>\therefore BDF2 is 0-stable</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Note that Δt doesn’t show up in $u_{BAD}^{(k)}$ or $u_{BDF2}^{(k)}$ – so reducing Δt doesn’t help.

□

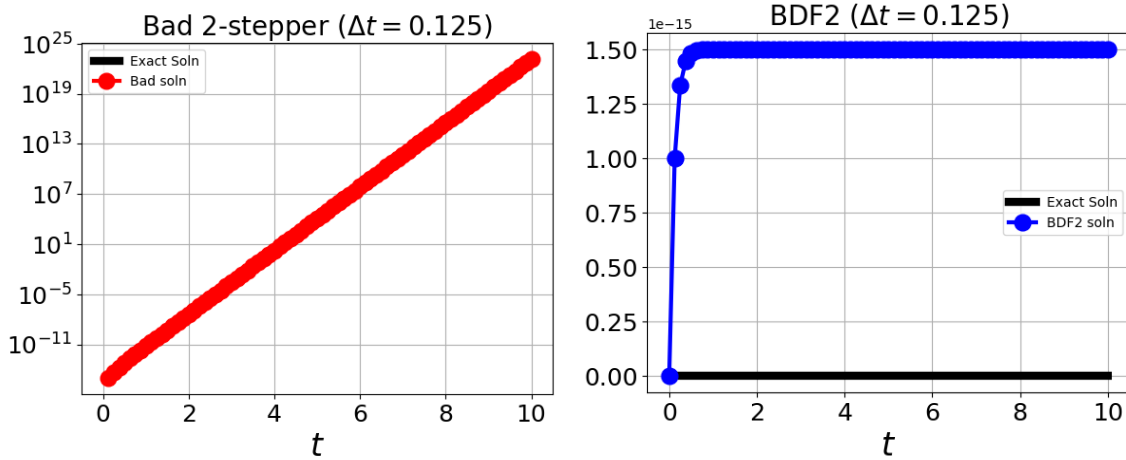


Figure 3.1: Solution Plots of (TIVP0) with $u_{BAD}^{(k)}$ (left) and $u_{BDF2}^{(k)}$ (right) and $u^{(1)} := 10^{-15}$. The axes are log-lin scale for $u_{BAD}^{(k)}$ and lin-lin scale for $u_{BDF2}^{(k)}$.

3.2.1 Zero-stability w.r.t. Linear Multistep Timesteppers

Recall the general r -step linear multistep timestepper for the model IVP (MIVP):

$$\text{LM}r : \quad \sum_{j=0}^r a_j u^{(k+j)} = (\Delta t) \sum_{j=0}^r b_j \Psi^{(k+j)} \quad (3.12)$$

Applying $\text{LM}r$ to (TIVP0) results in a homogeneous linear difference equation:

$$\sum_{j=0}^r a_j u^{(k+j)} = 0 \quad (3.13)$$

Theory [77] dictates to assume the difference equation has a solution of the form

$$u^{(k)} = \zeta^k \quad (3.14)$$

Substituting this into the equation and simplifying via division by ζ^k yields the polynomial equation

$$0 = \sum_{j=0}^r a_j \zeta^j := \rho_{\text{LM}r}(\zeta) \quad (3.15)$$

Now, since the exact solution to (TIVP0), $u(t) = 0$, is trivially bounded, we expect the computed solution $u^{(k)}$ to also be bounded for all timesteps t_k . To ensure that the difference equation has bounded solutions, the polynomial $\rho_{\text{LM}r}(\zeta)$ must satisfy the following condition [59, 77]:

Definition 3.2.2. A polynomial $\rho(\zeta)$ satisfies the root condition provided the following two conditions are true:

- (1) All roots of ρ are in the closed unit disc $\overline{\mathbb{D}} := \{z \in \mathbb{C} : |z| \leq 1\}$
- (2) All repeated roots of ρ are in the open unit disc $\mathbb{D} := \{z \in \mathbb{C} : |z| < 1\}$

Armed with this definition, we can now succinctly state the zero-stability requirement for $\text{LM}r$:

Theorem 3.2.1. An r -step linear multistep timestepper $\text{LM}r$ is zero-stable if and only if its polynomial $\rho_{\text{LM}r}(\zeta)$ satisfies the root condition.

Adams-Bashforth and Adams-Moulton timesteppers are all zero-stable due to their polynomial ρ clearly satisfying the root condition:

$$\begin{aligned}\rho_{ABr}(\zeta) &= \zeta^{r-1}(\zeta - 1) \\ \rho_{AMr}(\zeta) &= \zeta^{r-1}(\zeta - 1)\end{aligned}\tag{3.16}$$

Unfortunately, BDF r is zero-stable [59, 77] only for $1 \leq r \leq 6$.

3.2.2 Zero-stability w.r.t. Runge-Kutta Timesteppers

Unlike linear multistep timesteppers, Runge-Kutta timesteppers have the good fortune of guaranteed zero-stability:

Theorem 3.2.2. All Runge-Kutta timesteppers are zero-stable.

Proof. Recall the s -stage Runge-Kutta (RK(s)) timestepper:

$$u^{(k+1)} = u^{(k)} + (\Delta t) \sum_{i=1}^s \gamma_i K_i, \quad K_i := \Psi \left(t_{k+\alpha_i}; u^{(k)} + (\Delta t) \sum_{j=1}^s \beta_{ij} K_j \right)\tag{3.17}$$

Upon applying RK(s) to (TIVP0) with a perturbed IC:

$$\begin{cases} u'(t) &= 0 \\ u(0) &= \epsilon \end{cases}\tag{3.18}$$

we get (since $\Psi = 0$ for all t):

$$\begin{cases} u^{(0)} &= \epsilon \\ u^{(k+1)} &= u^{(k)} \end{cases}\tag{3.19}$$

It trivially follows that $u^{(k)} = \epsilon$ for all timesteps t_k which implies that $u^{(k)}$ is uniformly bounded. Therefore, RK(s) is zero-stable.

□

3.3 Convergence

Naturally, we expect the timestepper to be convergent in the sense that the error in the computed solution improves as the timestep size shrinks. However, it is possible

that it may not converge for certain problems or certain initial conditions. For the idea of convergence to apply to the broadest set of “reasonable” problems and initial conditions, we employ the following definition [77]:

Definition 3.3.1. An r -step timestepper is convergent if applying it to (MIVP) subject to the conditions

- (1) RHS $\Psi(t; u)$ is Lipschitz continuous in u
- (2) Initial condition u_0 is chosen so (MIVP) has a unique solution for $t \in [t_0, T]$
- (3) The timestepper’s starting values $u^{(0)}, u^{(1)}, \dots, u^{(r-1)} \rightarrow u_0$ as $\Delta t \rightarrow 0$

results in $|u(T) - u^{(N_T)}| \rightarrow 0$ as $\Delta t \rightarrow 0$.

3.3.1 Convergence w.r.t. Linear Multistep Timesteppers

In order for linear multistep timesteppers to be convergent, they must be both consistent and zero-stable [77]:

Theorem 3.3.1. A linear multistep timestepper is convergent if and only if it is consistent and zero-stable.

The proof can be found in [31].

3.3.2 Convergence w.r.t. Runge-Kutta Timesteppers

A lesser requirement holds for Runge-Kutta timesteppers:

Theorem 3.3.2. A Runge-Kutta timestepper is convergent if and only if it is consistent.

See §6.3.4 of [77] for a proof.

3.4 Stiff ODE Systems

Here, we introduce notions of stability stronger than zero-stability that may be necessary for a given problem.

Consider the following 2×2 linear system:

$$(\text{Stiff-IVP}): \quad \begin{cases} \mathbf{u}'(t) = A\mathbf{u}, & A := \begin{bmatrix} -51 & 49 \\ 49 & -51 \end{bmatrix} \\ \mathbf{u}(0) = \mathbf{u}_0, & \mathbf{u}_0 := \begin{bmatrix} 2 \\ 0 \end{bmatrix} \end{cases} \quad (3.20)$$

Then, the exact solution is

$$\mathbf{u}(t) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} e^{-2t} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{-100t} \quad (3.21)$$

Moreover, the square matrix A is diagonalizable:

$$(\text{DIAG}) \quad A = V\Lambda V^{-1}, \quad V = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} -100 & 0 \\ 0 & -2 \end{bmatrix} \quad (3.22)$$

Of course, V^{-1} is the inverse of matrix V :

$$(\text{INV}) \quad VV^{-1} = V^{-1}V = I, \quad V^{-1} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.23)$$

Clearly, $\mathbf{u}(t) \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ as $t \rightarrow \infty$.

However, some convergent timesteppers may not exhibit a similar asymptotic behavior in the computed solution $\mathbf{u}^{(k)}$ if the timestep size Δt is too large.

Case in point, consider solving (Stiff-IVP) using Forward Euler. Then:

$$\begin{aligned} \mathbf{u}^{(k+1)} &\stackrel{FE}{=} \mathbf{u}^{(k)} + (\Delta t)\Psi^{(k)} \\ &= \mathbf{u}^{(k)} + \Delta t (A\mathbf{u}^{(k)}) \\ &= [I + (\Delta t)A]\mathbf{u}^{(k)} \\ &\stackrel{IND}{=} [I + (\Delta t)A]^{k+1}\mathbf{u}^{(0)} \\ &\stackrel{DIAG}{=} [I + (\Delta t)V\Lambda V^{-1}]^{k+1}\mathbf{u}^{(0)} \\ &\stackrel{INV}{=} [VIV^{-1} + (\Delta t)V\Lambda V^{-1}]^{k+1}\mathbf{u}^{(0)} \\ &= V[I + (\Delta t)\Lambda]^{k+1}V^{-1}\mathbf{u}^{(0)} \end{aligned} \quad (3.24)$$

Since $\mathbf{u}^{(0)}$, V and V^{-1} are all constant, it follows that the computed solution approaches the zero vector provided successive powers of the diagonal matrix, $I + (\Delta t)\Lambda$, approach the zero matrix – this requirement is satisfied if all its diagonal entries are less than one in magnitude:

$$\begin{aligned}
 \mathbf{u}^{(k)} \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} &\iff \|I + (\Delta t)\Lambda\| < 1 \\
 &\iff \left\| \begin{bmatrix} (1 - 100\Delta t) & 0 \\ 0 & (1 - 2\Delta t) \end{bmatrix} \right\| < 1 \\
 &\iff |1 - 100\Delta t| < 1 \quad \text{and} \quad |1 - 2\Delta t| < 1 \\
 &\iff \Delta t < \frac{1}{50}
 \end{aligned} \tag{3.25}$$

Hence, Δt must be smaller than 0.02 to assure that the computed solution $\mathbf{u}^{(k)}$ exhibits the same asymptotic decay to the zero vector as the exact solution $\mathbf{u}(t)$. A larger timestep size, say $\Delta t = 0.1$, will cause the computed solution to blow-up in magnitude (i.e. $\|\mathbf{u}^{(k)}\|_2 \rightarrow \infty$), which is not faithful to the decaying behavior of the exact solution.

Now let's solve (Stiff-IVP) using GL(1). Then:

$$\begin{aligned}
 \mathbf{u}^{(k+1)} &\stackrel{GL(1)}{=} \mathbf{u}^{(k)} + (\Delta t)\mathbf{K}_1 \\
 &= \mathbf{u}^{(k)} + (\Delta t) \left[I - \frac{1}{2}(\Delta t)A \right]^{-1} A\mathbf{u}^{(k)} \\
 &= \left[I + (\Delta t) \left(I - \frac{1}{2}(\Delta t)A \right)^{-1} A \right] \mathbf{u}^{(k)} \\
 &\stackrel{IND}{=} \left[I + (\Delta t) \left(I - \frac{1}{2}(\Delta t)A \right)^{-1} A \right]^{k+1} \mathbf{u}^{(0)} \\
 &\stackrel{DIAG}{=} \left[VV^{-1} + (\Delta t) \left(VV^{-1} - \frac{1}{2}(\Delta t)A \right)^{-1} A \right]^{k+1} \mathbf{u}^{(0)} \\
 &\stackrel{INV}{=} \left[VV^{-1} + (\Delta t) \left(VV^{-1} - \frac{1}{2}(\Delta t)V\Lambda V^{-1} \right)^{-1} V\Lambda V^{-1} \right]^{k+1} \mathbf{u}^{(0)} \\
 &= V \left[I + (\Delta t) \left(I - \frac{1}{2}(\Delta t)\Lambda \right)^{-1} \Lambda \right]^{k+1} V^{-1}\mathbf{u}^{(0)}
 \end{aligned} \tag{3.26}$$

Since $\mathbf{u}^{(0)}$, V and V^{-1} are all constant, following the argument used in (3.25),

$$\begin{aligned}
 \mathbf{u}^{(k)} \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} &\iff \|I + (\Delta t) (I - \frac{1}{2}(\Delta t)\Lambda)^{-1} \Lambda\| < 1 \\
 &\iff \left\| \begin{bmatrix} \frac{1-50\Delta t}{1+50\Delta t} & 0 \\ 0 & \frac{1-\Delta t}{1+\Delta t} \end{bmatrix} \right\| < 1 \\
 &\iff \left| \frac{1-50\Delta t}{1+50\Delta t} \right| < 1 \quad \text{and} \quad \left| \frac{1-\Delta t}{1+\Delta t} \right| < 1 \\
 &\iff \Delta t \in (0, \infty)
 \end{aligned} \tag{3.27}$$

So GL(1) imposes no restriction on Δt . In other words, the computed solution $\mathbf{u}^{(k)}$ will always converge to the zero vector regardless of how large Δt is. Now of course, a large Δt may result in substantial loss of accuracy, but the qualitative asymptotic behavior will always be correct.

What forced restrictions on Δt for Forward Euler but not GL(1) is the stiffness of this particular ODE system [52, 59]:

Definition 3.4.1. An ODE system is stiff if certain timesteppers require a substantially tiny timestep size Δt in order for the computed solution error to not blow up in magnitude.

This definition does not provide a quantitative measure of stiffness by which a given system can be judged. Here is such a measure for ODE systems [52, 59, 77]:

Definition 3.4.2. The stiffness ratio of a linear ODE system $\mathbf{u}'(t) = A\mathbf{u}$ is the ratio of the magnitudes of the largest and smallest eigenvalues of the matrix A .

The stiffness ratio of a non-linear ODE system $\mathbf{u}'(t) = \Psi(t; \mathbf{u})$ is the ratio of the magnitudes of the largest and smallest eigenvalues of the Jacobian matrix $\frac{\partial \Psi}{\partial \mathbf{u}}$ associated with the system.

Stiff linear ODE systems have large stiffness ratios. However, large stiffness ratios for non-linear systems may not necessarily indicate that they're actually stiff – stiffness correlates only when the linearized system is a sufficiently accurate approximation. Stiff systems arise from dynamical processes whose rates of change vary on markedly different time scales [52, 59, 77].

3.5 A-Stability

Consider the following linear IVP:

$$(TIVPA): \quad \begin{cases} u'(t) = \lambda u, & \operatorname{Re}(\lambda) < 0 \\ u(0) = 1 \end{cases} \quad (3.28)$$

The exact solution to (TIVPA) is $u(t) = e^{\lambda t}$ which decays to zero as $t \rightarrow \infty$. Hence, (TIVPA) bears similar asymptotic behavior as (Stiff-IVP), but is simpler to analyze since the solution is a scalar rather than a vector. We saw in the previous section that the eigenvalues of the matrix A in a stiff linear IVP play a central role – here in (TIVPA), λ is the sole eigenvalue of the degenerate 1×1 matrix λ . Since eigenvalues can be complex, we allow λ to be possibly complex in (TIVPA).

Armed with this “test” linear IVP, we can now define a notion of stability [59, 77] stronger than zero-stability:

Definition 3.5.1. A timestepper’s stability function is the function $S(z)$ on the complex plane such that upon applying the timestepper to (TIVPA),

$$u^{(k+1)} = [S(\lambda\Delta t)]^{k+1} u^{(0)}, \quad \text{where } \lambda \in \mathbb{C}$$

Definition 3.5.2. A timestepper is absolutely stable if when applied to (TIVPA) with a given Δt and $\lambda \in \mathbb{C}$, $|u_{k+1}| \rightarrow 0$ as $t \rightarrow \infty$ for all initial values u_0 . This is satisfied whenever its stability function $S(z)$ satisfies $|S(\lambda\Delta t)| < 1$.

The domain of absolute stability (DAS) of a timestepper is the region in the complex plane where it is absolutely stable, namely:

$$\Omega^{stab} := \{z = \lambda(\Delta t) \in \mathbb{C} : |S(z)| < 1\}$$

A timestepper is A-stable if its DAS contains the left-half plane of \mathbb{C} :

$$\Omega^{stab} \supseteq \mathbb{C}^{-\operatorname{Re}}, \quad \text{where} \quad \mathbb{C}^{-\operatorname{Re}} := \{z \in \mathbb{C} : \operatorname{Re}(z) < 0\}$$

Let's determine the stability function for Forward Euler by applying it to (TIVPA):

$$\begin{aligned}
 u^{(k+1)} &\stackrel{FE}{=} u^{(k)} + (\Delta t)\Psi^{(k)} \\
 &\stackrel{TIVPA}{=} u^{(k)} + \Delta t (\lambda u^{(k)}) \\
 &= [1 + \lambda\Delta t]u^{(k)} \\
 &\stackrel{IND}{=} [1 + z]^{k+1}u^{(0)} \quad \text{where } z := \lambda\Delta t
 \end{aligned}
 \tag{3.29}$$

Therefore, the stability function for Forward Euler is

$$S_{FE}(z) = 1 + z \tag{3.30}$$

Next, let's determine its domain of absolute stability:

$$\begin{aligned}
 \Omega_{FE}^{stab} &= \{z \in \mathbb{C} : |S_{FE}(z)| < 1\} \\
 &= \{z \in \mathbb{C} : |1 + z| < 1\} \\
 &= \{z \in \mathbb{C} : z \text{ is in open disk centered at } -1, \text{ radius } 1\} \\
 &\not\supseteq \mathbb{C}^{-\text{Re}}
 \end{aligned}
 \tag{3.31}$$

Hence, Forward Euler is not A-stable, and this is precisely why it failed for large enough timestep size Δt when applied to (Stiff-IVP):

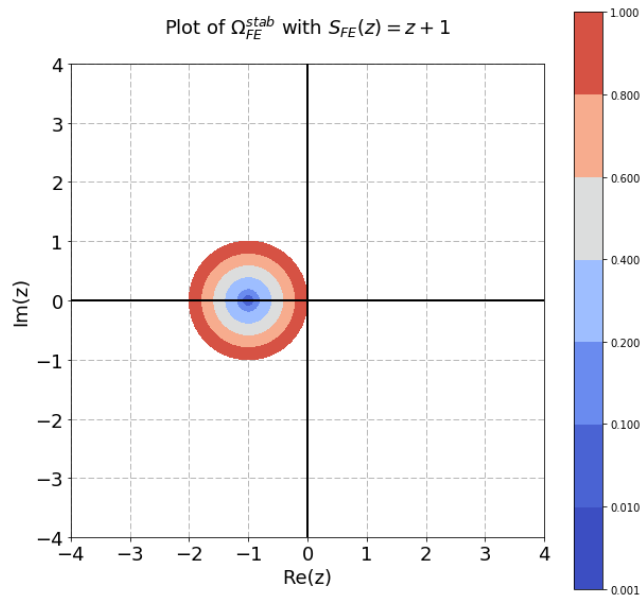


Figure 3.2: DAS Plot of Forward Euler

Let's determine the stability function for GL(1):

$$\begin{aligned}
 u^{(k+1)} &\stackrel{GL(1)}{=} u^{(k)} + (\Delta t)K_1 \\
 &\stackrel{TIVPA}{=} u^{(k)} + (\Delta t) \left[\frac{\lambda}{1 - \frac{1}{2}\lambda\Delta t} \right] u^{(k)} \\
 &= \left[\frac{1 + \frac{1}{2}\lambda\Delta t}{1 - \frac{1}{2}\lambda\Delta t} \right] u^{(k)} \\
 &\stackrel{IND}{=} \left[\frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right]^{k+1} u^{(0)} \quad \text{where } z := \lambda\Delta t
 \end{aligned} \tag{3.32}$$

Therefore, the stability function for GL(1) is

$$S_{GL(1)}(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \tag{3.33}$$

Next, let's determine its domain of absolute stability:

$$\begin{aligned}
 \Omega_{GL(1)}^{stab} &= \{z \in \mathbb{C} : |S_{GL(1)}(z)| < 1\} \\
 &= \{z \in \mathbb{C} : \left| \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right| < 1\} \\
 &= \{z \in \mathbb{C} : |2 + z| < |2 - z|\} \\
 &= \{z \in \mathbb{C} : z \text{ is closer to } -2 \text{ than to } +2\} \\
 &= \mathbb{C}^{-\text{Re}}
 \end{aligned} \tag{3.34}$$

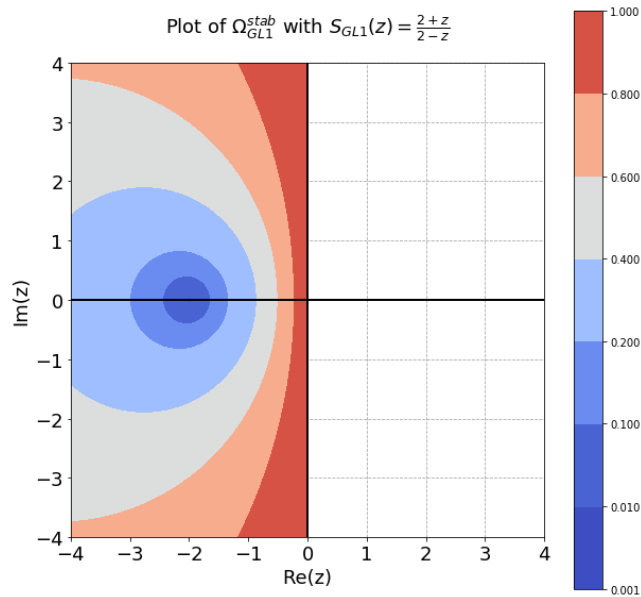


Figure 3.3: DAS Plot of GaussLegendre(1)

Hence, GL(1) is A-stable, and this is precisely why it always converges, even for large timestep size Δt , when applied to (Stiff-IVP).

Finally, we determine the stability function for Backward Euler:

$$\begin{aligned}
 u^{(k+1)} &\stackrel{BE}{=} u^{(k)} + (\Delta t)\Psi^{(k+1)} \\
 &\stackrel{TIVPA}{=} u^{(k)} + (\Delta t) [\lambda u^{(k+1)}] \\
 &= \left[\frac{1}{1-\lambda\Delta t} \right] u^{(k)} \\
 &\stackrel{IND}{=} \left[\frac{1}{1-z} \right]^{k+1} u^{(0)} \quad \text{where } z := \lambda\Delta t
 \end{aligned} \tag{3.35}$$

Therefore, the stability function for Backward Euler is $S_{BE}(z) = \frac{1}{1-z}$

Next, let's determine its domain of absolute stability:

$$\begin{aligned}
 \Omega_{BE}^{stab} &= \{z \in \mathbb{C} : |S_{BE}(z)| < 1\} \\
 &= \{z \in \mathbb{C} : \left| \frac{1}{1-z} \right| < 1\} = \{z \in \mathbb{C} : |1-z| > 1\} \\
 &= \{z \in \mathbb{C} : z \text{ is outside open disk centered at } +1, \text{ radius } 1\} \\
 &\supset \mathbb{C}^{-\text{Re}}
 \end{aligned} \tag{3.36}$$

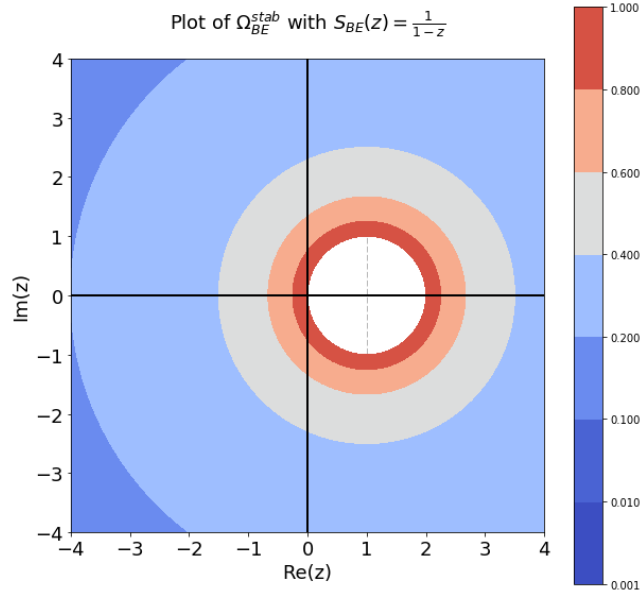


Figure 3.4: DAS Plot of Backward Euler

Hence, BE is A-stable.

Discussion of A-stability of linear multistep timesteppers is omitted because, out of the three linear multistep classes discussed (ABr , AMr , $BDFr$), only BDF1 and BDF2 are A-stable [59, 77].

3.6 $A(\alpha)$ -Stability

Some stiff ODE systems have its eigenvalues on or very close to the negative real axis. Such systems are considered mildly stiff. With mildly stiff systems, A-stable timesteppers may be considered overkill. Timesteppers whose domain of absolute stability includes not the entire left-half complex plane, but solely an infinite sector containing the origin and the entire negative real axis, are perfectly suitable for mildly stiff systems [77]:

Definition 3.6.1. A timestepper is $A(\alpha)$ -stable if its DAS contains the set

$$\mathbb{C}_\alpha^{-\text{Re}} := \{re^{i\theta} \in \mathbb{C} : r \in [0, \infty) \text{ and } \theta \in (\pi - \alpha, \pi + \alpha)\}$$

for some $\alpha \in [0, \pi/2]$.

An A-stable timestepper is always $A(\alpha)$ -stable with $\alpha = \pi/2$.

BDF3 through BDF6 are not A-stable but $A(\alpha)$ -stable [77], and therefore they are preferred for mildly stiff systems due to their high order and easier computation than A-stable IRK timesteppers.

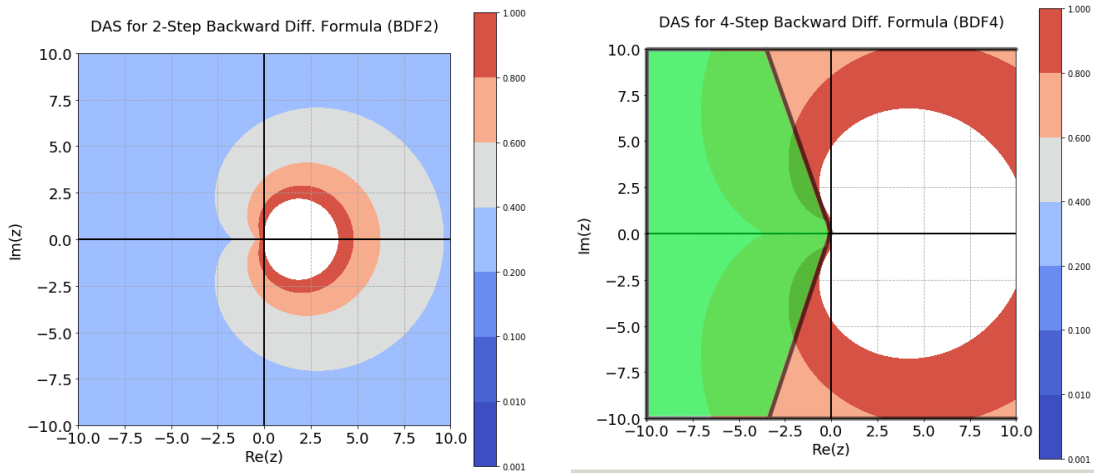


Figure 3.5: DAS Plots of BDF2 (left) and BDF4 (right). The green translucent region represents $\mathbb{C}_\alpha^{-\text{Re}}$ for BDF4 [77] with $\alpha \approx 73^\circ$.

3.7 L-Stability

For some stiff problems, A-stability is not good enough. An even stronger notion of stability has its place [77]:

Definition 3.7.1. A timestepper is L-stable if it is A-stable and $\lim_{z \rightarrow \infty} |S(z)| = 0$.

Geometrically, L-stable timesteppers have the point $z = \infty$, which is the “north pole” on the Riemann sphere [106], in the interior of its DAS whereas A-stable timesteppers have $z = \infty$ merely on the boundary [77].

It’s no surprise that Forward Euler is not L-stable since it’s not A-stable and

$$\lim_{z \rightarrow \infty} |S_{FE}(z)| = \lim_{z \rightarrow \infty} |1 + z| = \infty \quad (3.37)$$

GL(1) is A-stable, but not L-stable since

$$\lim_{z \rightarrow \infty} |S_{GL(1)}(z)| = \lim_{z \rightarrow \infty} \left| \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right| = 1 \quad (3.38)$$

Backward Euler is L-stable since it’s A-stable and

$$\lim_{z \rightarrow \infty} |S_{BE}(z)| = \lim_{z \rightarrow \infty} \left| \frac{1}{1 - z} \right| = 0 \quad (3.39)$$

For stiff ODE systems with a rapid transient, an L-stable timestepper effectively damps out the transient after the first iteration [52, 77] whereas an A-stable timestepper forces the computed solution to honor the transient and, as a result, oscillate about the true solution for all timesteps.

To illustrate this effect L-stability has on a computed solution versus A-stability, consider the following stiff IVP, known as the Curtiss-Hirschfelder equation [49]:

$$\begin{cases} u' + 50u = 50 \cos t \\ u(0) = 0 \end{cases} \quad (3.40)$$

The exact solution is

$$u(t) = -\frac{2500}{2501}e^{-50t} + \frac{2500}{2501} \cos t + \frac{50}{2501} \sin t \quad (3.41)$$

Note that the exponential term is the rapid transient in this case.

Just as with A-stability, the only L-stable linear multistep timesteppers are BDF1 (which is Backward Euler) and BDF2 [77].

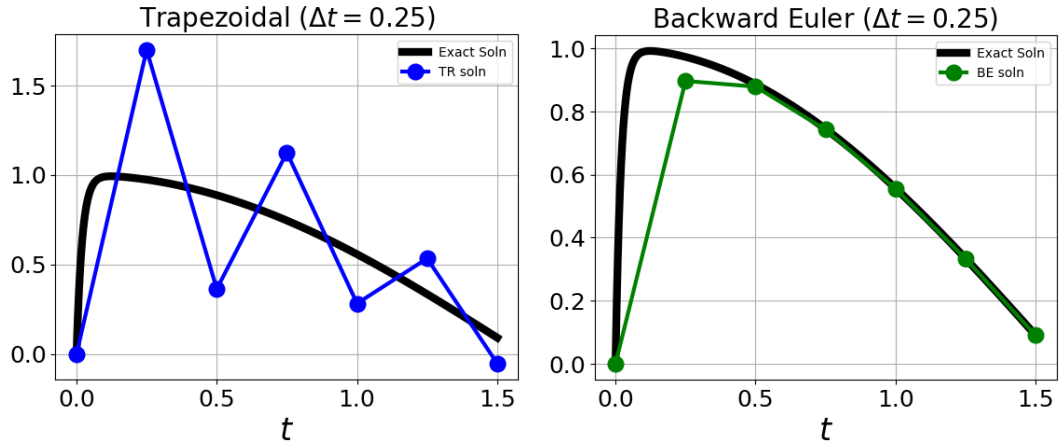


Figure 3.6: Solution Plots of Curtiss-Hirschfelder Equation with Trapezoidal (left) and Backward Euler (right). Notice how the Trapezoidal solution oscillates about the exact solution.

3.7.1 DAS containing $\mathbb{C}^{-\text{Re}}$ and more does not imply L-stability

The DAS plots of many L-stable IRK timesteppers contain the entire left-half complex plane $\mathbb{C}^{-\text{Re}}$ and then some more:

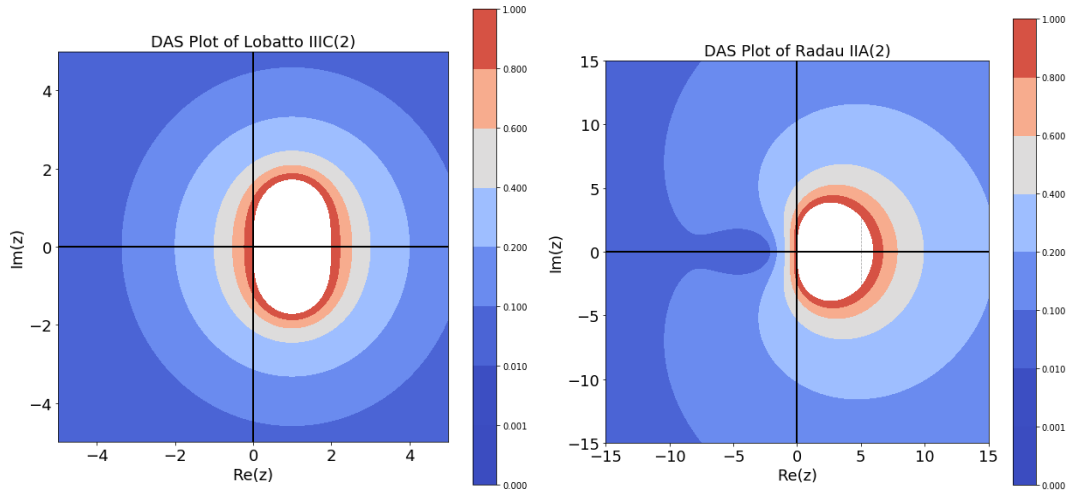


Figure 3.7: DAS Plots of Lobatto IIIC(2) (left) and Radau IIA(2) (right)

However, the assumption that this behavior implies L-stability is refuted with Spijker’s DIRK(2) serving as a counterexample:

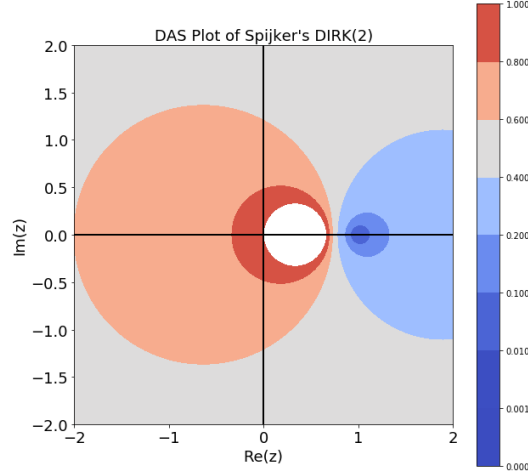


Figure 3.8: DAS Plot of Spijker's DIRK(2)

The issue lies with its stability function:

$$\text{Spijker's DIRK(2): } S(z) = \frac{z-1}{2z-1} \implies \lim_{z \rightarrow \infty} |S(z)| = \frac{1}{2} \neq 0 \quad (3.42)$$

3.8 Preservation of Quadratic Invariants

ODE systems in some applications [59] preserve quadratic invariants:

$$\text{(IVPQ) : } \begin{cases} \mathbf{u}'(t) = A\mathbf{u} \\ \mathbf{u}(t_0) = \mathbf{u}_0 \end{cases}, \quad A := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{u} := \begin{bmatrix} u \\ v \end{bmatrix}, \quad \mathbf{u}_0 := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (3.43)$$

Proposition 3.8.1. The following expression is a quadratic invariant for (IVPQ):

$$Q(\mathbf{u}) := \mathbf{u}^T I \mathbf{u}, \quad \text{where } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Proof. It suffices to show that $Q(\mathbf{u})$ never changes in time:

$$\begin{aligned} \frac{d}{dt} [Q(\mathbf{u}(t))] &= \frac{d}{dt} [\mathbf{u}^T I \mathbf{u}] \\ &\stackrel{IVPQ}{=} \frac{d}{dt} [u^2 + v^2] \\ &\stackrel{1CR}{=} 2u \cdot u' + 2v \cdot v' \\ &\stackrel{IVPQ}{=} 2u \cdot (-v) + 2v \cdot (u) \\ &= 0 \end{aligned} \quad (3.44)$$

□

Quadratic invariants are key qualitative geometric properties of systems which leads to the question of which timesteppers, if any, preserve them:

Definition 3.8.1. A timestepper is quadratic invariant preserving (QIP) if when applied to an ODE system that has a quadratic invariant, the computed value of that invariant remains the same, in exact arithmetic, for all time. In floating-point arithmetic, the quadratic invariant error is on the order of machine epsilon, ϵ_{mach} .

Timesteppers lacking A-stability tend to lack preservation of quadratic invariants:

Proposition 3.8.2. Forward Euler is not QIP.

Proof. It suffices to show Forward Euler fails to honor quadratic invariant in (IVPQ):

$$\begin{aligned}
 Q(\mathbf{u}^{(k+1)}) &\stackrel{IVPQ}{=} [\mathbf{u}^{(k+1)}]^T I \mathbf{u}^{(k+1)} \\
 &\stackrel{FE}{=} [\mathbf{u}^{(k)} + (\Delta t) \boldsymbol{\Psi}^{(k)}]^T I [\mathbf{u}^{(k)} + (\Delta t) \boldsymbol{\Psi}^{(k)}] \\
 &\stackrel{IVPQ}{=} [\mathbf{u}^{(k)}]^T [I + (\Delta t)A]^T I [I + (\Delta t)A] \mathbf{u}^{(k)} \\
 &= [\mathbf{u}^{(k)}]^T [I + (\Delta t)(A + A^T) + (\Delta t)^2 A^T A] \mathbf{u}^{(k)} \\
 &= [\mathbf{u}^{(k)}]^T [I + (\Delta t)^2] \mathbf{u}^{(k)} \\
 &\neq Q(\mathbf{u}^{(k)}) \quad \text{for } \Delta t \neq 0
 \end{aligned} \tag{3.45}$$

□

L-stable timesteppers tend to not be QIP:

Proposition 3.8.3. Backward Euler is not QIP.

Proof. It suffices to show Backward Euler fails to honor quadratic invariant in (IVPQ):

$$\begin{aligned}
 Q(\mathbf{u}^{(k+1)}) &\stackrel{IVPQ}{=} [\mathbf{u}^{(k+1)}]^T I \mathbf{u}^{(k+1)} \\
 &\stackrel{BE}{=} [\mathbf{u}^{(k)} + (\Delta t) \boldsymbol{\Psi}^{(k+1)}]^T I [\mathbf{u}^{(k)} + (\Delta t) \boldsymbol{\Psi}^{(k+1)}] \\
 &\stackrel{IVPQ}{=} [\mathbf{u}^{(k)}]^T [I - (\Delta t)A]^{-T} I [I - (\Delta t)A]^{-1} \mathbf{u}^{(k)} \\
 &= \frac{1}{[1 + (\Delta t)^2]^2} [\mathbf{u}^{(k)}]^T \begin{bmatrix} 1 + (\Delta t)^2 & 0 \\ 0 & 1 + (\Delta t)^2 \end{bmatrix} \mathbf{u}^{(k)} \\
 &= \frac{1}{1 + (\Delta t)^2} [\mathbf{u}^{(k)}]^T I \mathbf{u}^{(k)} \\
 &\neq Q(\mathbf{u}^{(k)}) \quad \text{for } \Delta t \neq 0
 \end{aligned} \tag{3.46}$$

□

However, some A-stable timesteppers are QIP [50, 59]:

Proposition 3.8.4. GaussLegendre(s) timesteppers are QIP.

Proof. (from §IV.2.1 of [50])

WLOG, let Model Vector IVP (MVIVP) have quadratic invariant $Q(\mathbf{u}) = \mathbf{u}^T S \mathbf{u}$ where matrix S is symmetric due to QFSML (Lemma G.1 in Appendix).

Let $p(t)$ be the s -degree collocation polynomial of GL(s) with its collocation points c_1, \dots, c_s identical to the quadrature points of s -point Gauss quadrature. Then:

$$(*1) \quad p(t_k) = \mathbf{u}^{(k)}, \quad p(t_k + \Delta t) = \mathbf{u}^{(k+1)}$$

$$(*2) \quad \frac{d}{dt} [Q(p)] = 2p^T S p' \text{ which is a } (2s - 1)\text{-degree polynomial}$$

$$(*3) \quad \begin{array}{l} s\text{-point Gauss quadrature exactly integrates} \\ \int_{t_k}^{t_k + \Delta t} [p(\tau)]^T S p'(\tau) d\tau = \sum_{j=1}^s w_j [p(t_k + c_j(\Delta t))]^T S p'(t_k + c_j(\Delta t)) \end{array}$$

$$(*4) \quad [u(t)]^T S \Psi(t; \mathbf{u}) = 0 \quad \forall(t, \mathbf{u}) \text{ due to QIL (Lemma G.2 in Appendix)}$$

Applying (*1) – (*4) and relating the RHS Ψ to p' via the Model Vector IVP,

$$\begin{aligned} & Q(\mathbf{u}^{(k+1)}) - Q(\mathbf{u}^{(k)}) \\ \stackrel{(*1)}{=} & Q(p(t_{k+1})) - Q(p(t_k)) \\ = & \int_{t_k}^{t_k + \Delta t} \frac{d}{d\tau} Q(p(\tau)) d\tau \\ \stackrel{(*2)}{=} & 2 \int_{t_k}^{t_k + \Delta t} [p(\tau)]^T S p'(\tau) d\tau \\ \stackrel{(*3)}{=} & 2 \sum_{j=1}^s w_j [p(t_k + c_j(\Delta t))]^T S p'(t_k + c_j(\Delta t)) \\ \stackrel{MVIVP}{=} & 2 \sum_{j=1}^s w_j [p(t_k + c_j(\Delta t))]^T S \Psi(t_k + c_j(\Delta t)) \\ \stackrel{(*4)}{=} & 0 \end{aligned} \tag{3.47}$$

\therefore Since timestep t_k was arbitrary, $Q(\mathbf{u}^{(k+1)}) - Q(\mathbf{u}^{(k)}) = 0$ for all timesteps.

\therefore GL(s) preserves $Q(\mathbf{u})$.

\therefore GL(s) is QIP. □

To help visualize the striking difference between a non-QIP timestepper, say Explicit Midpoint, and a QIP timestepper, say Implicit Midpoint, the two plots below compare resulting qualitative behaviors of the solution to the system (IVPQ) with the unit circle as the quadratic invariant:

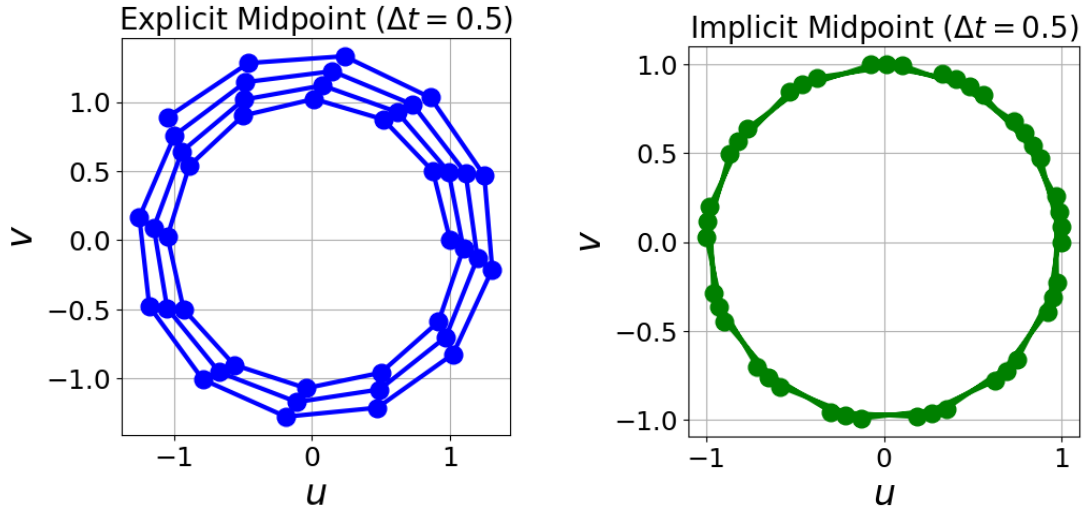


Figure 3.9: Solution Plots to (IVPQ) with EM (left) & GL(1) (right)

To the naked eye, the QIP computed solution appears to stay on the unit circle, but one cannot be certain until the magnitude error is visualized:

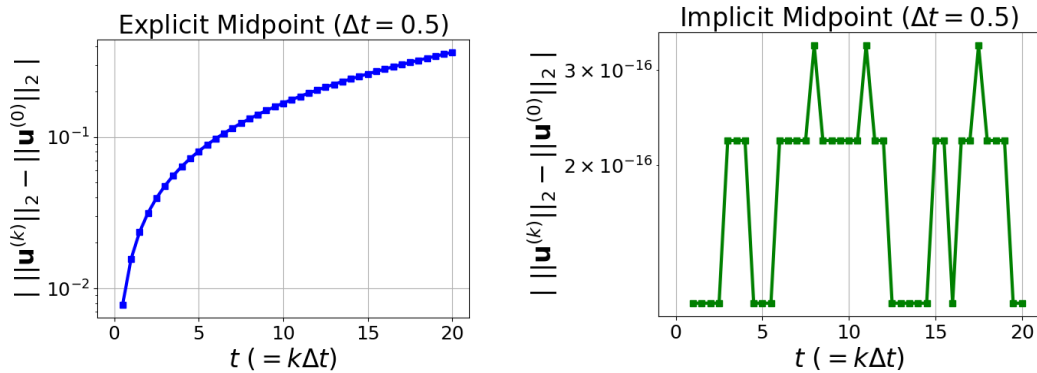


Figure 3.10: QIP Error Plots to (IVPQ) with EM (left) & GL(1) (right). Note the axes are log-lin for EM and lin-lin for GL(1). The error remains near ϵ_{mach} for the QIP timestepper (right) while the non-QIP timestepper's error (left) grows.

Given a timestepper's Butcher table, one can easily determine whether it is QIP or not via the following theorem [28, 50]:

Theorem 3.8.1. (Cooper's QIP Theorem)

An s -stage Runge-Kutta timestepper is QIP if its Butcher table satisfies:

$$\gamma_i \beta_{ij} + \gamma_j \beta_{ji} = \gamma_i \gamma_j \quad \forall i, j = 1, \dots, s$$

A proof can be found in [50].

3.9 Feature Comparison of Common Timesteppers

With a vast catalog of timesteppers to choose from, it's instructive to summarize their orders, stability and invariance properties into a table, which is on the next page. All timesteppers shown in the table are zero-stable.

Table 3.1: Timestepper Order, Stability & Invariance Summary

TIMESTEPPER:	ORDER:	A(α)-stable?	A-stable?	L-stable?	QIP?
AB r	r	No	No	No	No
AM r	$r + 1$	No	No	No	No
BDF r , $1 \leq r \leq 2$	r	Yes	Yes	Yes	No
BDF r , $3 \leq r \leq 6$	r	Yes	No	No	No
Forward Euler	1	No	No	No	No
Explicit Midpoint	2	No	No	No	No
Classic RK4	4	No	No	No	No
Backward Euler	1	Yes	Yes	Yes	No
Trapezoidal	2	Yes	Yes	No	No
Lobatto IIIA(s)	$2s - 2$	Yes	Yes	No	No
Lobatto IIIB(s)	$2s - 2$	Yes	Yes	No	No
Lobatto IIIC(s)	$2s - 2$	Yes	Yes	Yes	No
Lobatto IIIS(s)	$2s - 2$	Yes	Yes	No	Yes
Radau IA(s)	$2s - 1$	Yes	Yes	Yes	No
Radau IB(s)	$2s - 1$	Yes	Yes	No	Yes
Radau IIA(s)	$2s - 1$	Yes	Yes	Yes	No
Radau IIB(s)	$2s - 1$	Yes	Yes	No	Yes
GaussLegendre(s)	$2s$	Yes	Yes	No	Yes
Spijker DIRK(2)	1	Yes	Yes	No	No
Miller DIRK(2)	2	Yes	Yes	Yes	No
Miller DIRK(3)	2	Yes	Yes	Yes	No
Geng SDIRK(2)	2	Yes	Yes	No	Yes
Crouzeix SDIRK(2)	2	Yes	Yes	Yes	No
Crouzeix SDIRK(3)	3	Yes	Yes	Yes	No

3.10 Linear Multistep Timesteppers do not Make the Cut

Linear multistep timesteppers will never be employed in the micromagnetics problem because their maximum order is only slightly above the stage count. This limitation is expressed by Dahlquist's 1st Barrier [31, 59]:

Theorem 3.10.1. (Dahlquist's Order Barrier for LM Timesteppers)

An r -step explicit LM timestepper has order r at best.
 An r -step implicit LM timestepper $\begin{cases} \text{has order } (r + 1) \text{ at best} & , \text{ if } r \text{ is odd} \\ \text{has order } (r + 2) \text{ at best} & , \text{ if } r \text{ is even} \end{cases}$

Even worse, they are nearly universally not A-stable with the exception of low-order BDF1 (which is Backward Euler) and BDF2. This observation is generalized in Dahlquist's 2nd Barrier [32, 52, 59, 77]:

Theorem 3.10.2. (Dahlquist's A-Stability Barrier for LM Timesteppers)

No explicit linear multistep timesteppers are A-stable.
 A-stable implicit linear multistep timesteppers are 2nd-order at best.

Going from BDF2 to BDF3 drops stability down to $A(\alpha)$ -stability whose DAS incrementally contracts in BDF4, BDF5 and BDF6 [77]. Worse still, very high order BDF timesteppers lack zero-stability, rendering them utterly useless [59, 77]. Finally, none of them preserve quadratic invariants. Therefore, we must choose among the Runge-Kutta timesteppers.

3.11 ERK Timesteppers do not Make the Cut

In addition, explicit Runge-Kutta timesteppers are avoided altogether due to their guaranteed lack of A-stability. This guarantee holds because an ERK timestepper's stability function $S(z)$ is a polynomial [59, 77] whose modulus blows up for an ever-increasing modulus of z :

$$\text{If } S(z) \text{ is a polynomial, then } |S(z)| \rightarrow \infty \text{ as } \text{Re}(z) \rightarrow -\infty \quad (3.48)$$

Therefore, an ERK timestepper's DAS cannot contain the left-half complex plane.

On the other hand, an IRK timestepper's stability function $S(z)$ will be a rational function [59, 77] in z . Hence, if the zeros of $S(z)$ are in the left-half complex plane and its poles are all in the right-half complex plane, then the Maximum Modulus Theorem [106] can be utilized to conclude that the DAS contains $\mathbb{C}^{-\text{Re}}$ and, hence, the timestepper is A-stable. Besides A-stability, some IRK timesteppers preserve quadratic invariants while none of the ERK timesteppers do. Hence, an appropriate IRK timestepper will be sought out for the coupled micromagnetics problem. We pick this discussion up later in Chapter 5 after a survey of spatial discretization using finite elements.

3.12 DIRK Timesteppers do not Make the Cut

Even though DIRK timesteppers are implicit, they are only slightly so. While they are easier to implement and their Newton systems are simpler to solve, a price is paid for these benefits. This causes some undesirable realities to arise.

To start, some DIRK timesteppers suffer from order reduction in that the order is actually less than the stage count. Examples shown earlier are Spijker's DIRK(2) [68] and Miller's DIRK(3) [88], both of which have orders one less than their stage count.

An order barrier when weights γ_j are all positive was found in 1980 [48, 49]:

Theorem 3.12.1. (Hairer's Barrier for Positively-weighted DIRK Timesteppers)

A DIRK timestepper with all positive weights γ_j has order 6 at best.

This is bad news for DIRK timesteppers, but unfortunately the news gets even worse. Going from a fully IRK timestepper to a DIRK timestepper places many restrictions on attainable order and stability as this section will point out.

A non-confluent [49] Runge-Kutta timestepper has its $\alpha_1, \dots, \alpha_s$ all distinct. Explicit Midpoint and Radau IA(3), are non-confluent whereas Classic RK4 is confluent. A second order barrier was found for non-confluent DIRK timesteppers in 1993 [5]:

Theorem 3.12.2. (Al-Rabeh’s Barrier for Non-confluent DIRK Timesteppers)

$$\left(\begin{array}{l} \text{A non-confluent DIRK}(s) \text{ timestepper} \\ \text{with non-zero weights } \gamma_1, \dots, \gamma_s \end{array} \right) \begin{array}{l} \text{has order } s + 1 \text{ at best, for } s \leq 5 \\ \text{has order } s \text{ at best, for } s \geq 6 \end{array}$$

A third barrier by Hairer concerns B-stability (see Appendix M) [48, 49]:

Theorem 3.12.3. (Hairer’s Barrier for B-stable DIRK Timesteppers)

A B-stable DIRK timestepper has order 4 at best.

Going forward, we will never mention B-stability again as L-stability is a sufficiently strong notion of stability for our needs.

Notice that Geng’s SDIRK(2) is the only DIRK timestepper shown earlier that is QIP [40]. With some of these order barriers affecting stability, perhaps constructing DIRK timesteppers that are QIP but not necessarily A-stable or L-stable will allow higher orders than the stage count. Alas, that hope is extinguished thanks to an order barrier by Geng in 1993 [40]:

Theorem 3.12.4. (Geng’s Barrier for QIP DIRK Timesteppers)

A QIP DIRK timestepper has order 4 at best.

Finally, if these order barriers were not bad enough, a comprehensive list of DIRK timesteppers [66, 14] reveals that all L-stable DIRK(s) timesteppers have order s at best, so it does not look promising to find L-stable DIRK(s) timesteppers of order $(s + 1)$ or higher.

3.13 SDIRK Timesteppers do not Make the Cut

The situation is dire for SDIRK timesteppers as the four DIRK barriers apply to SDIRK timesteppers as well. Unfortunately, there is an additional order barrier specifically for SDIRK timesteppers once again discovered by Hairer in 1980 [48, 49]:

Theorem 3.13.1. (Hairer’s Barrier for Positive-weighted SDIRK Timesteppers)

A SDIRK timestepper with all positive weights γ_j has order 4 at best.

Notice the reduction in order (from 6 to 4) going from positively-weighted DIRK to positively-weighted SDIRK. This is not surprising since requiring the main diagonal entries to all be equal necessarily reduces possible solutions to the Runge-Kutta order conditions.

Finally, a comprehensive list of SDIRK timesteppers [66, 14] reveals that all L-stable SDIRK(s) timesteppers have order s at best, so it does not look promising to find L-stable SDIRK(s) timesteppers of order $(s + 1)$ or higher.

3.14 Candidate Timesteppers for the LLG+EC Problem

To summarize, we should not use LM, ERK, DIRK, or SDIRK timesteppers in the coupled micromagnetics problem. Our remaining choice is to use an IRK timestepper that is either fully implicit or at least more fully implicit than DIRK and SDIRK timesteppers. From Table 3.1, the viable choices are:

- Lobatto IIIC(s)
- Lobatto IIIS(s)
- Radau IA(s)
- Radau IB(s)
- Radau IIA(s)
- Radau IIB(s)
- GaussLegendre(s)

More on our eventual choice of timesteppers in Chapter 5.

CHAPTER IV

FINITE ELEMENTS

4.1 Model PDE: Poisson's Equation

The finite element method is an involved algorithm, so it behooves us to demonstrate its utility by applying it to the simplest time-independent 2^{nd} -order scalar PDE in two spatial dimensions: Poisson's Equation. The technique is similar when applied to three spatial dimensions and even one spatial dimension. The overview of the Finite Element Method and accompanying notation more-or-less follows that of [44, 74]. Rigorous treatment of finite element methods can be found in [16, 90].

Going forward, the domain $\Omega \in \mathbb{R}^d$ ($d \in \{2, 3\}$) is assumed to be a bounded and connected open set with piecewise-smooth boundary $\partial\Omega$ and outward normal \mathbf{n} .

For this chapter only, the boundary is assumed to be partitioned

$$\partial\Omega := \Gamma_D \cup \Gamma_N \quad \text{such that} \quad \Gamma_D \cap \Gamma_N = \emptyset \quad (4.1)$$

where Γ_D prescribes homogeneous Dirichlet BC's and Γ_N bears homogeneous Neumann BC's.

Poisson's Equation arises in many problems such as electrostatics and steady-state heat transport. Along with the BC's, it is:

$$\begin{cases} -\nabla^2 u = f & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_D \\ \mathbf{n} \cdot \nabla u = 0 & \text{on } \Gamma_N \end{cases} \quad (4.2)$$

4.2 The Finite Element Method

Since the Finite Element Method is a long process consisting of many sub-tasks, we carefully go over the algorithm step-by-step as it applies to Poisson's Equation.

4.2.1 STEP 1: Determine Pre-Element Weak Form

Multiply both sides of Poisson's Equation by a test function \bar{u} which is zero on the Dirichlet boundary Γ_D , and then integrate both sides over the domain:

$$-\int_{\Omega} \bar{u} \nabla^2 u \, d\mathbf{x} = \int_{\Omega} f \bar{u} \, d\mathbf{x} \quad (4.3)$$

Perform Integration-by-Parts (IBP) on the integral containing the Laplacian:

$$-\oint_{\partial\Omega} [\bar{u} \bar{\nabla} u] \cdot \mathbf{n} \, dS + \int_{\Omega} \bar{\nabla} \bar{u} \cdot \bar{\nabla} u \, d\mathbf{x} = \int_{\Omega} f \bar{u} \, d\mathbf{x} \quad (4.4)$$

Split the boundary integral into its partitioned boundary parts:

$$-\oint_{\partial\Omega} [\bar{u} \bar{\nabla} u] \cdot \mathbf{n} \, dS = -\int_{\Gamma_D} [\bar{u} \bar{\nabla} u] \cdot \mathbf{n} \, dS - \int_{\Gamma_N} [\bar{u} \bar{\nabla} u] \cdot \mathbf{n} \, dS \quad (4.5)$$

The boundary integral over Γ_D is zero since $\bar{u} = 0$ on Γ_D :

$$\int_{\Gamma_D} [\bar{u} \bar{\nabla} u] \cdot \mathbf{n} \, dS = \int_{\Gamma_D} [0 \cdot \bar{\nabla} u] \cdot \mathbf{n} \, dS = \int_{\Gamma_D} 0 \, dS = 0 \quad (4.6)$$

The boundary integral over Γ_N is zero due to the homogeneous Neumann BC:

$$\int_{\Gamma_N} [\bar{u} \bar{\nabla} u] \cdot \mathbf{n} \, dS \stackrel{VA3}{=} \int_{\Gamma_N} \bar{u} \cdot [\mathbf{n} \cdot \bar{\nabla} u] \, dS = \int_{\Gamma_N} \bar{u} \cdot 0 \, dS = 0 \quad (4.7)$$

The boundary integrals vanish and we are left with the pre-element weak form:

$$\int_{\Omega} \bar{\nabla} \bar{u} \cdot \bar{\nabla} u \, d\mathbf{x} = \int_{\Omega} f \bar{u} \, d\mathbf{x} \quad (4.8)$$

However, we need to specify from which function spaces do u , \bar{u} and f belong to ensure these two integrals are well-defined:

$$\begin{aligned} f \in L^2(\Omega) &:= \{g : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} g^2 \, d\mathbf{x} < \infty\} \\ H^1(\Omega) &:= \{v \in L^2(\Omega) : \bar{\nabla} v \in [L^2(\Omega)]^d\} \\ u, \bar{u} \in H_0^1(\Omega) &:= \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\} \end{aligned} \quad (4.9)$$

We can verify that the integrals are well-defined via the Cauchy-Schwarz and Triangle Inequalities:

$$\begin{aligned} \left| \int_{\Omega} \bar{u}_x u_x \, d\mathbf{x} \right| &\stackrel{CSI}{\leq} \left(\int_{\Omega} (\bar{u}_x)^2 \, d\mathbf{x} \right)^{1/2} \left(\int_{\Omega} (u_x)^2 \, d\mathbf{x} \right)^{1/2} \stackrel{H^1}{<} \infty \cdot \infty = \infty \\ \left| \int_{\Omega} \bar{u}_y u_y \, d\mathbf{x} \right| &\stackrel{CSI}{\leq} \left(\int_{\Omega} (\bar{u}_y)^2 \, d\mathbf{x} \right)^{1/2} \left(\int_{\Omega} (u_y)^2 \, d\mathbf{x} \right)^{1/2} \stackrel{H^1}{<} \infty \cdot \infty = \infty \end{aligned} \quad (4.10)$$

$$\begin{aligned}
 \left| \int_{\Omega} \nabla \bar{u} \cdot \nabla u \, d\mathbf{x} \right| &= \left| \int_{\Omega} \bar{u}_x u_x + \bar{u}_y u_y \, d\mathbf{x} \right| \\
 &\stackrel{TI}{\leq} \left| \int_{\Omega} \bar{u}_x u_x \, d\mathbf{x} \right| + \left| \int_{\Omega} \bar{u}_y u_y \, d\mathbf{x} \right| \\
 &\stackrel{(4.9)}{<} \infty + \infty \\
 &= \infty
 \end{aligned} \tag{4.11}$$

$$\left| \int_{\Omega} f \bar{u} \, d\mathbf{x} \right| \stackrel{CSI}{\leq} \left(\int_{\Omega} f^2 \, d\mathbf{x} \right)^{1/2} \left(\int_{\Omega} \bar{u}^2 \, d\mathbf{x} \right)^{1/2} \stackrel{L^2}{<} \infty \cdot \infty = \infty \tag{4.12}$$

Now we can properly state the pre-element weak form of Poisson's Equation:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } \int_{\Omega} \nabla \bar{u} \cdot \nabla u \, d\mathbf{x} = \int_{\Omega} f \bar{u} \, d\mathbf{x} \quad \forall \bar{u} \in H_0^1(\Omega) \tag{4.13}$$

4.2.2 STEP 2: Generate Mesh from Domain

The next major step is to discretize the domain Ω into a mesh, denoted Ω_h . For our purposes, we will exclusively use triangles in 2D and tetrahedra in 3D. Each triangle or tetrahedron is called a cell of the mesh. For a rectangular domain, axis-aligned triangulation [44, 74] works wells:

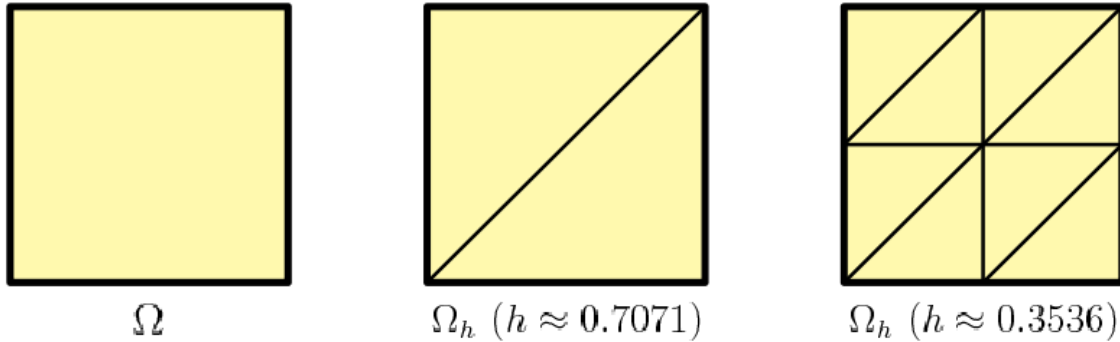


Figure 4.1: Square Domain (left) and two Axis-Aligned Meshes (middle & right)

Otherwise, Delaunay triangulation [34, 36, 44, 74] is preferred:

FEniCS [80] can perform axis-aligned triangulation while Delaunay triangulation can be achieved using, for example, CUBIT [12], GMSH [41], NETGEN [110] or TRIANGLE [113, 114].

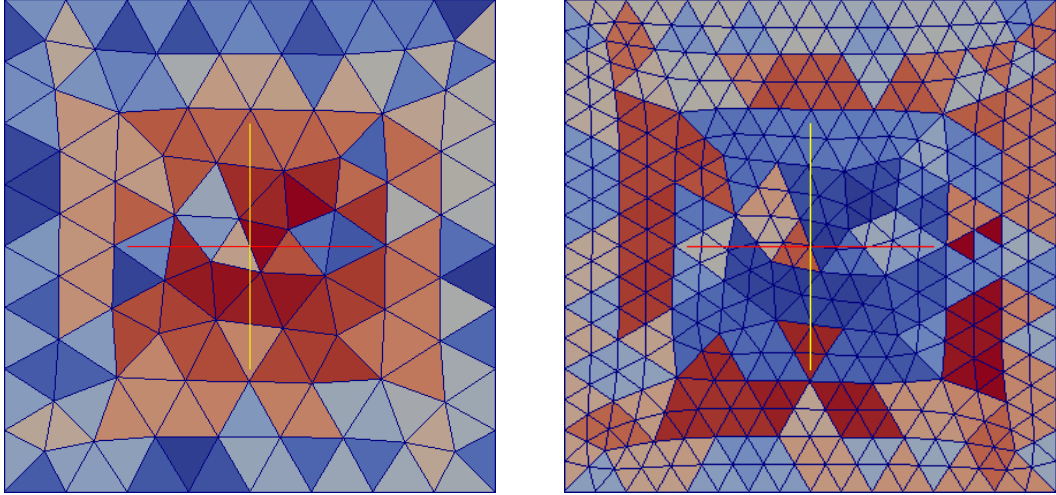


Figure 4.2: Coarse (left) & Fine (right) Delaunay Meshes for Square Domain

In either case, the resulting mesh will not contain any “skinny” triangles [36, 74] in which one of the angles is near 0° or 180° . The absence of “skinny” triangles ensures that finite element error estimates [74] are not too loose. In addition, every pair of adjacent triangles intersect at a common vertex or a common edge.

Similar tetrahedralization schemes [36] are used in 3D to ensure that every pair of tetrahedra intersect at a common vertex or common edge or common face. Moreover, care must be taken to ensure a tetrahedralization contains no “skinny” tetrahedra, “flat” tetrahedra and “slivers” [36]. Axis-aligned tetrahedralization for box-shaped domains is handled in FEniCS [80] and Delaunay tetrahedralization is done with, for instance, CUBIT [12], GMSH [41], NETGEN [110] or TetGen [115].

Finer meshes with smaller cells will improve accuracy in the computed solution. Mesh refinement is measured by the cell diameter [44, 74]:

Definition 4.2.1. Given a domain Ω and a corresponding mesh, the cell diameter $h > 0$ of the mesh is defined as follows:

$$h := (\text{Maximum edge length of a cell over all cells in the mesh})$$

The degree of mesh refinement is indicated by smaller values of h .

The mesh of Ω is denoted Ω_h .

4.2.3 STEP 3: Select Appropriate Finite Elements

With the mesh generated, we can now select a finite-dimensional subspace of $H^1(\Omega)$, denoted V_h , that is a space of continuous piecewise polynomials up to some degree defined on the mesh Ω_h .

Moreover, define space $V_{h,0} \subset V_h$ as follows:

$$V_{h,0} := \{v \in V_h : v = 0 \text{ on } \partial\Omega_h\} \quad (4.14)$$

The basis functions of $V_{h,0}$, denoted φ_j , on the N_V nodes of the mesh interior (which for quadratic or higher degree polynomials will require additional points besides the vertices of the cells) called shape functions, to represent an approximation u_h , called the trial function, of the exact solution of the weak form on a mesh Ω_h :

$$u_h = \sum_{j=1}^{N_V} c_j \varphi_j \quad (4.15)$$

4.2.4 STEP 4: Determine Galerkin Weak Form

Equipped with these finite-dimensional spaces, we can rewrite the pre-element weak form as the post-element weak form:

$$\text{Find } u_h \in V_{h,0} \text{ such that } \int_{\Omega} \bar{\nabla} \bar{u}_h \cdot \bar{\nabla} u_h \, d\mathbf{x} = \int_{\Omega} f \bar{u}_h \, d\mathbf{x} \quad \forall \bar{u}_h \in V_{h,0} \quad (4.16)$$

Replacing u_h with its basis expansion and choosing the basis functions for the test functions, we arrive at the Galerkin weak form [44, 52, 74]:

$$\begin{aligned} &\text{Find } c_1, \dots, c_{N_V} \text{ such that} \\ &\int_{\Omega} \bar{\nabla} \varphi_i \cdot \bar{\nabla} \left(\sum_{j=1}^{N_V} c_j \varphi_j \right) \, d\mathbf{x} = \int_{\Omega} f \varphi_i \, d\mathbf{x}, \quad \text{for } i = 1, \dots, N_V \end{aligned} \quad (4.17)$$

4.2.5 STEP 5: Assemble Finite Element Algebraic System

Harnessing the linearity of finite summation, gradients, dot products, and integrals, we can rewrite the Galerkin weak form into a more convenient form:

$$\begin{aligned} &\text{Find coefficients } c_1, \dots, c_{N_V} \text{ such that} \\ &\sum_{j=1}^{N_V} c_j \left(\int_{\Omega} \bar{\nabla} \varphi_i \cdot \bar{\nabla} \varphi_j \, d\mathbf{x} \right) = \int_{\Omega} f \varphi_i \, d\mathbf{x}, \quad \text{for } i = 1, \dots, N_V \end{aligned} \quad (4.18)$$

Let the stiffness matrix be $M_{ij} := \int_{\Omega} \bar{\nabla} \varphi_i \cdot \bar{\nabla} \varphi_j \, d\mathbf{x}$,

Let the load vector be $f_i := \int_{\Omega} f \varphi_i \, d\mathbf{x}$.

Then the discretized Galerkin weak form [44, 52, 74] is an algebraic system of equations:

$$\text{Find } c_1, \dots, c_{N_V} \text{ such that } \sum_{j=1}^{N_V} M_{ij} c_j = f_i, \quad \text{for } i = 1, \dots, N_V \quad (4.19)$$

which expressed in matrix-form vector is the $N_V \times N_V$ linear system:

$$\text{Find coefficient vector } \mathbf{c} \text{ such that } M\mathbf{c} = \mathbf{f} \quad (4.20)$$

For a nonlinear PDE, the discretized weak form becomes a nonlinear system.

4.2.6 STEP 6: Solve Finite Element Algebraic System

Since the test functions were chosen to be precisely the shape functions, the stiffness matrix M is symmetric positive definite. Furthermore, the locality of the shape functions implies that M is also a sparse matrix. Therefore, the finite element linear system can be efficiently solved via the Conjugate Gradient iterative solver [52, 64, 104, 122] with incomplete Cholesky [52, 104] or multigrid [17, 52, 104] preconditioning.

In general, however, we cannot expect M to be symmetric positive definite. In such cases, we would solve the linear system using GMRES [64, 104, 122] with an appropriate preconditioner [80, 104].

For a nonlinear system, we employ a Newton solver [52, 64, 65, 121, 122].

4.3 Nodal Finite Elements

4.3.1 Lagrange Elements

Lagrange elements [44, 74] mandate well-defined first derivatives:

Table 4.1: (Scalar) Lagrange Elements

(Scalar) Lagrange Elements of degree k	
Approximates Functions in	$H^1(\Omega) := \{u \in L^2(\Omega) : \overline{\nabla}u \in [L^2(\Omega)]^d\}$
$H^1(\Omega)$ norm	$\ u\ _{H^1(\Omega)}^2 := \ u\ _{L^2(\Omega)}^2 + \ \overline{\nabla}u\ _{[L^2(\Omega)]^d}^2$
Subspace of $H^1(\Omega)$	$H_0^1(\Omega) := \{u \in H^1(\Omega) : u = 0 \text{ on } \partial\Omega\}$
Inter-Element Property	C^0 -continuity
L^2 Convergence Rate	$\mathcal{O}(h^{k+1})$

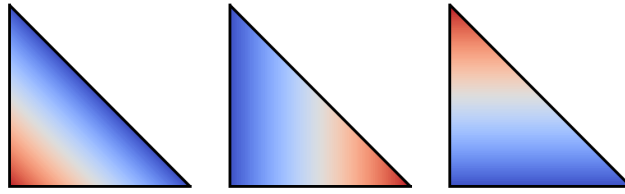


Figure 4.3: First-Degree (Scalar) Lagrange Shape Functions

4.3.2 Vector Lagrange Elements

Vector Lagrange elements [80] are component-wise Lagrange elements:

Table 4.2: Vector Lagrange Elements

Vector Lagrange Elements of degree k	
Approximates Functions in	$[H^1(\Omega)]^d := \{\mathbf{M} \in [L^2(\Omega)]^d : \overline{\nabla}\mathbf{M} \in [L^2(\Omega)]^{d \times d}\}$
$[H^1(\Omega)]^d$ norm	$\ \mathbf{M}\ _{[H^1(\Omega)]^d}^2 := \ \mathbf{M}\ _{[L^2(\Omega)]^d}^2 + \ \overline{\nabla}\mathbf{M}\ _{[L^2(\Omega)]^{d \times d}}^2$
Subspace of $[H^1(\Omega)]^d$	$[H_0^1(\Omega)]^d := \{\mathbf{M} \in [H^1(\Omega)]^d : \mathbf{M} = \vec{\mathbf{0}} \text{ on } \partial\Omega\}$
Inter-Element Property	Component-wise C^0 -continuity
L^2 Convergence Rate	$\mathcal{O}(h^{k+1})$

4.4 Edge Finite Elements

4.4.1 Nédélec (1st Kind) Elements

First kind Nédélec elements [13, 62, 80, 90] debuted in Jean-Claude Nédélec’s seminal paper [94] in 1980. Note, however, that 1st-degree first kind Nédélec elements [74, 103] were first discovered by Hassler Whitney [126] in 1957 and, hence, are sometimes called Whitney elements.

Table 4.3: First Kind Nédélec Elements

Nédélec (1 st Kind) Elements of degree k	
Approximates Functions in	$H(\text{curl}; \Omega) := \{\mathbf{B} \in [L^2(\Omega)]^3 : \nabla \times \mathbf{B} \in [L^2(\Omega)]^3\}$
$H(\text{curl}; \Omega)$ norm	$\ \mathbf{B}\ _{H(\text{curl})}^2 := \ \mathbf{B}\ _{[L^2(\Omega)]^3}^2 + \ \nabla \times \mathbf{B}\ _{[L^2(\Omega)]^3}^2$
Subspace of $H(\text{curl})$	$H_0(\text{curl}; \Omega) := \{\mathbf{B} \in H(\text{curl}; \Omega) : \hat{\mathbf{n}} \times \mathbf{B} = \vec{\mathbf{0}} \text{ on } \partial\Omega\}$
Subset of $H(\text{curl})$	$H_D(\text{curl}; \Omega) := \{\mathbf{B} \in H(\text{curl}; \Omega) : \mathbf{B} = \mathbf{B}_D \text{ on } \partial\Omega\}$
Inter-Element Property	Tangential continuity along edges
Mesh Property	Weakly divergence-free
L^2 Convergence Rate	$\mathcal{O}(h^k)$

Here are the visual shape functions for 1st-degree first kind Nédélec [23]:

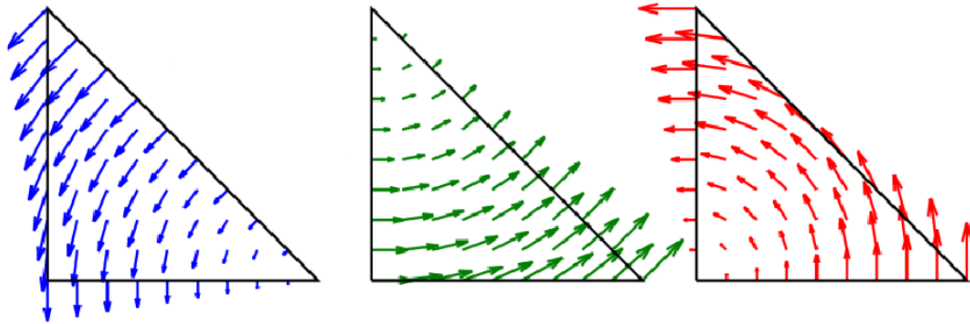


Figure 4.4: First-Degree First Kind Nédélec Shape Functions

Note that higher-order bases can be constructed either in an interpolary or hierarchical manner. Consult §8.6 & §8.7 of [62] for examples and references of interpolary and hierarchical basis constructions, respectively.

4.4.2 Nédélec (2nd Kind) Elements

Second kind Nédélec elements [80, 90] were unveiled in Jean-Claude Nédélec’s second paper [95] in 1986. The 1st-degree second kind Nédélec elements were first discovered by Gerrit Mur and Adrianus de Hoop [93] in 1985.

Table 4.4: Second Kind Nédélec Elements

Nédélec Elements (2 nd Kind) of degree k	
Approximates Functions in	$H(\text{curl}; \Omega) := \{\mathbf{B} \in [L^2(\Omega)]^3 : \nabla \times \mathbf{B} \in [L^2(\Omega)]^3\}$
$H(\text{curl}; \Omega)$ norm	$\ \mathbf{B}\ _{H(\text{curl})}^2 := \ \mathbf{B}\ _{[L^2(\Omega)]^3}^2 + \ \nabla \times \mathbf{B}\ _{[L^2(\Omega)]^3}^2$
Subspace of $H(\text{curl})$	$H_0(\text{curl}; \Omega) := \{\mathbf{B} \in H(\text{curl}; \Omega) : \hat{\mathbf{n}} \times \mathbf{B} = \vec{\mathbf{0}} \text{ on } \partial\Omega\}$
Subset of $H(\text{curl})$	$H_D(\text{curl}; \Omega) := \{\mathbf{B} \in H(\text{curl}; \Omega) : \mathbf{B} = \mathbf{B}_D \text{ on } \partial\Omega\}$
Inter-Element Property	Tangential continuity along edges
Mesh Property	Weakly divergence-free
L^2 Convergence Rate	$\mathcal{O}(h^{k+1})$

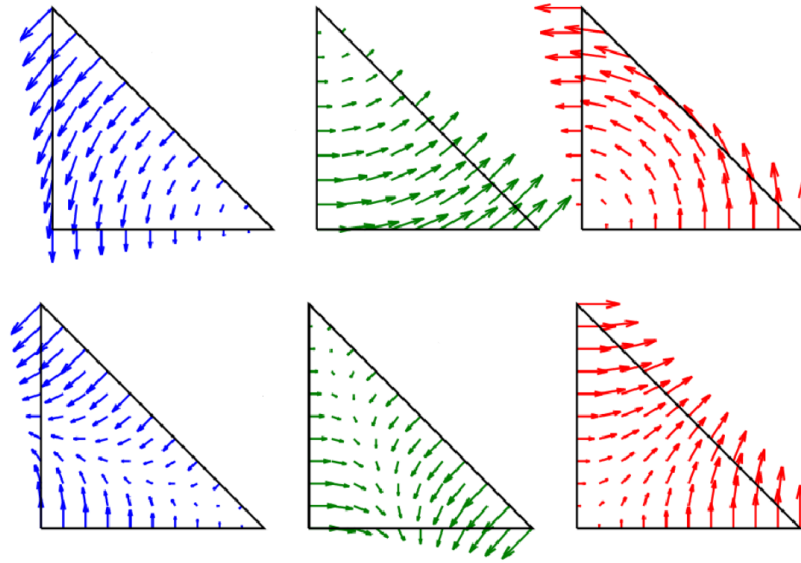


Figure 4.5: First-Degree Second Kind Nédélec Shape Functions. Notice that the first-degree first kind Nédélec shape functions are included.

4.4.3 Weak Divergence-Free Property

All Nédélec elements are divergence-free in a weak sense [13, 69]:

Proposition 4.4.1. (Weak Divergence-Free Property)

Let $\Omega \subset \mathbb{R}^d$ be a piecewise-smooth bounded domain with $d \in \{2, 3\}$.

Let vector field $\mathbf{B}(\mathbf{x}) \in H(\text{curl}; \Omega)$. Then:

$$\int_{\Omega} \mathbf{B} \cdot \nabla v \, d\mathbf{x} = 0 \quad \forall v(\mathbf{x}) \in H^1(\Omega)$$

Replacing Ω with its mesh Ω_h is called the Discrete Compactness Property.

1st-degree first kind Nédélec elements are not only weakly divergence-free but strongly divergence-free in that the divergence of their shape functions are exactly zero over the entire domain [13, 62, 74, 90, 94].

4.4.4 Immunity to Spurious Modes

Using Vector Lagrange elements for certain problems in electromagnetics will cause undesirable spurious modes [62, 90, 103] to occur. Spurious modes are nonphysical numerical solutions that are never found experimentally. What's worse for our problem, imposing Vector Lagrange elements for \mathbf{B} spawns spurious modes that have large non-zero divergences which contradicts the mandate that \mathbf{B} remain divergence-free. Nédélec elements of either kind do not have this issue because they are weakly divergence-free and tangentially continuous along cell edges [62, 90, 103].

4.5 Software Implementation Remarks

For simple problems with a simple coarse mesh and simple data such as a polynomial for the RHS f and the boundary conditions, performing the corresponding Finite Element Method by hand is feasible. However, in general the mesh will have a complicated geometry and the PDE data will not be simple polynomials. In such cases, programming the Finite Element Method in software will be absolutely necessary [44, 74].

4.5.1 A Brief Survey of Finite Element Frameworks

Writing a flexible FEM program from scratch is extremely daunting, but fortunately there are several general-purpose industry-grade FEM frameworks available. We absolutely require a framework that implements both kinds of edge elements of arbitrary degree; likewise, the framework should implement or allow programming of high-order IRK timesteppers.

Several of the open-source offerings lack comprehensive support for edge elements: Sundance [81] completely lacks edge elements, GetFEM++ [100] has only 1st-degree first kind Nédélec, Elmer [82] and FreeFem++ [53] support only up to 2nd-degree first kind Nédélec, and deal.II [8] supports arbitrary-degree first kind Nédélec but not second kind Nédélec. Other open source FEM frameworks exist but either they lack edge elements, they are not general-purpose, or their latest version is at least five years old.

On the other hand, commercial options have very intuitive graphical user interfaces and implementations of several ERK timesteppers, but they tend to lack high-order IRK timesteppers. ANSYS[®] [6] supports only Backward Euler and Newmark; however, while Newmark is 2nd-order, it is tailored specifically for problems in structural dynamics [96]. COMSOL[®] just has Backward Euler and implicit linear multistep BDF2 through BDF5, but Braxton Bragg implemented GaussLegendre(1) [15] – programming higher-order GaussLegendre(*s*) timesteppers in COMSOL[®] is not obvious. MSC Nastran[®] [92] and MATLAB[®] [85] along with other proprietary FEM solutions do not even support edge elements as they mainly focus on structural mechanics, heat transfer and/or fluid dynamics.

This leaves us with the FEniCS [80] open source finite element framework. It supports arbitrary-degree Nédélec elements of both kinds. Furthermore, IRK timesteppers can be programmed in Python with minimal boilerplate code. Therefore, we will use FEniCS for our simulation codes and validation tests. Details on FEniCS usage will be deferred to Chapter 6.

4.5.2 Reference Cells

In order to conserve computer memory, every cell in the mesh should be mapped to a fixed reference cell via a linear transformation. This allows only a single group of shape functions to be constructed. Once the coefficients of the shape functions are determined, the inverse transformation is applied to glean the approximate FEM solution for the original cell [44, 74].

4.5.3 Efficient Quadrature on Triangles & Tetrahedra

In general, integrals involving the RHS f will be non-elementary, meaning the antiderivative cannot be represented in a finite closed-form. Therefore, an appropriate quadrature rule [44, 74] must be applied, the most efficient of which are Gauss quadrature rules. However, tensor-product Gauss quadrature schemes require more quadrature points than is necessary. Instead, one should utilize efficient quadrature schemes specifically designed for triangles and tetrahedra that use far fewer quadrature points and, hence, require less memory and computation [62, 118, 129]. FEniCS provides efficient quadrature schemes on triangles and tetrahedra up to 6th degree [80].

4.5.4 Automatic Jacobian Matrix Computation

When solving a nonlinear system of algebraic equations via a Newton solver [52, 64, 65, 121, 122], the Jacobian matrix must be provided each iteration. Performing the computation by hand is undesirable as many problems lead to a Jacobian matrix whose entries are long and tedious. Fortunately, FEniCS employs Automatic Differentiation [4, 80] to compute the Jacobian matrix for each Newton iteration. Sundance has a similar capability [81].

CHAPTER V
THE COUPLED MICROMAGNETICS PROBLEM

5.1 Weak Form

5.1.1 Integration-by-Parts for Micromagnetics

We wish to generalize integration-by-parts for the micromagnetics problem's weak form by utilizing tensor notation (Table H.2) and Green's identities (see Appendix):

Proposition 5.1.1. (Micromagnetic Integration-by-Parts – MIBP)

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with $d \in \{2, 3\}$ and outward unit normal \mathbf{n} .

Let vector field $\mathbf{A}(\mathbf{x}) \in [C(\bar{\Omega})]^d \cap [C^2(\Omega)]^d$.

Let vector field $\mathbf{B}(\mathbf{x}) \in [C(\bar{\Omega})]^d \cap [C^1(\Omega)]^d$. Then:

$$\begin{aligned}
 (MIBP1) \quad \int_{\Omega} \mathbf{B} \cdot (\mathbf{A} \times \bar{\nabla}^2 \mathbf{A}) \, d\mathbf{x} &= \oint_{\partial\Omega} \mathbf{B} \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS \\
 &\quad - \int_{\Omega} \bar{\nabla}(\mathbf{B} \times \mathbf{A}) : \bar{\nabla} \mathbf{A} \, d\mathbf{x} \\
 (MIBP2) \quad \int_{\Omega} \mathbf{B} \cdot [\mathbf{A} \times (\mathbf{A} \times \bar{\nabla}^2 \mathbf{A})] \, d\mathbf{x} &= \oint_{\partial\Omega} (\mathbf{B} \times \mathbf{A}) \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS \\
 &\quad - \int_{\Omega} \bar{\nabla}[(\mathbf{B} \times \mathbf{A}) \times \mathbf{A}] : \bar{\nabla} \mathbf{A} \, d\mathbf{x}
 \end{aligned}$$

Proof. Let $\mathbf{A}(\mathbf{x}) = (A_1, A_2, A_3)$ and $\mathbf{B}(\mathbf{x}) = (B_1, B_2, B_3)$. Then:

$$\begin{aligned}
 (MIBP1) \quad \int_{\Omega} \mathbf{B} \cdot (\mathbf{A} \times \bar{\nabla}^2 \mathbf{A}) \, d\mathbf{x} &\stackrel{TN3}{=} \int_{\Omega} \epsilon_{ijk} B_i A_j \nabla^2 A_k \, d\mathbf{x} \\
 &\stackrel{GI1}{=} \oint_{\partial\Omega} \epsilon_{ijk} B_i A_j \frac{\partial A_k}{\partial n} \, dS - \int_{\Omega} \bar{\nabla}(\epsilon_{ijk} B_i A_j) \cdot \bar{\nabla} A_k \, d\mathbf{x} \\
 &\stackrel{TN3}{=} \oint_{\partial\Omega} \mathbf{B} \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS - \int_{\Omega} \bar{\nabla}(\mathbf{B} \times \mathbf{A})_k \cdot \bar{\nabla}(\mathbf{A})_k \, d\mathbf{x} \\
 &\stackrel{TN4}{=} \oint_{\partial\Omega} \mathbf{B} \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS - \int_{\Omega} \bar{\nabla}(\mathbf{B} \times \mathbf{A}) : \bar{\nabla} \mathbf{A} \, d\mathbf{x}
 \end{aligned} \tag{5.1}$$

(MIBP2)

$$\begin{aligned}
& \int_{\Omega} \mathbf{B} \cdot \left[\mathbf{A} \times (\mathbf{A} \times \bar{\nabla}^2 \mathbf{A}) \right] d\mathbf{x} \\
\stackrel{VA7}{=} & \int_{\Omega} \mathbf{B} \cdot \left[(\mathbf{A} \cdot \bar{\nabla}^2 \mathbf{A}) \mathbf{A} - (\mathbf{A} \cdot \mathbf{A}) \bar{\nabla}^2 \mathbf{A} \right] d\mathbf{x} \\
\stackrel{VA1}{=} & \int_{\Omega} (\mathbf{B} \cdot \mathbf{A}) (\mathbf{A} \cdot \bar{\nabla}^2 \mathbf{A}) - (\mathbf{A} \cdot \mathbf{A}) (\mathbf{B} \cdot \bar{\nabla}^2 \mathbf{A}) d\mathbf{x} \\
\stackrel{VA8}{=} & \int_{\Omega} (\mathbf{B} \times \mathbf{A}) \cdot (\mathbf{A} \times \bar{\nabla}^2 \mathbf{A}) d\mathbf{x} \stackrel{TN1}{=} \int_{\Omega} (\mathbf{B} \times \mathbf{A})_i (\mathbf{A} \times \bar{\nabla}^2 \mathbf{A})_i d\mathbf{x} \\
\stackrel{TN2}{=} & \int_{\Omega} \epsilon_{jki} B_j A_k \epsilon_{mni} A_m \nabla^2 A_n d\mathbf{x} \\
\stackrel{GI1}{=} & \int_{\partial\Omega} \epsilon_{jki} B_j A_k \epsilon_{mni} A_m \frac{\partial A_n}{\partial n} dS - \int_{\Omega} \bar{\nabla} \cdot (\epsilon_{mni} \epsilon_{jki} B_j A_k A_m) \cdot \bar{\nabla} A_n d\mathbf{x} \\
\stackrel{TN2}{=} & \int_{\partial\Omega} (\mathbf{B} \times \mathbf{A})_i \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right)_i dS - \int_{\Omega} \bar{\nabla} [\epsilon_{mni} (\mathbf{B} \times \mathbf{A})_i A_m] \cdot \bar{\nabla} A_n d\mathbf{x} \\
\stackrel{TN\epsilon}{=} & \int_{\partial\Omega} (\mathbf{B} \times \mathbf{A}) \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS - \int_{\Omega} \bar{\nabla} [\epsilon_{imn} (\mathbf{B} \times \mathbf{A})_i A_m] \cdot \bar{\nabla} A_n d\mathbf{x} \\
\stackrel{TN2}{=} & \int_{\partial\Omega} (\mathbf{B} \times \mathbf{A}) \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS - \int_{\Omega} \bar{\nabla} [(\mathbf{B} \times \mathbf{A}) \times \mathbf{A}]_n \cdot \bar{\nabla} A_n d\mathbf{x} \\
\stackrel{TN4}{=} & \int_{\partial\Omega} (\mathbf{B} \times \mathbf{A}) \cdot \left(\mathbf{A} \times \frac{\partial \mathbf{A}}{\partial \mathbf{n}} \right) dS - \int_{\Omega} \bar{\bar{\nabla}} [(\mathbf{B} \times \mathbf{A}) \times \mathbf{A}] : \bar{\bar{\nabla}} \mathbf{A} d\mathbf{x}
\end{aligned} \tag{5.2}$$

□

5.1.2 Pre-Element-Pre-Timestepper Weak Form

Upon dot-producting both sides of the Eddy Currents PDE by test field $\bar{\mathbf{B}}$, dot-producting both sides of the LLG PDE by test field $\bar{\mathbf{M}}$, add the two resulting equations together and then applying vector algebra/calculus identities, Green's identities (see Appendix) and micromagnetic integration-by-parts, we arrive at the so-called pre-element-pre-timestepper weak form of the micromagnetic problem which is

$N(\mathbf{B}, \mathbf{M}; \overline{\mathbf{B}}, \overline{\mathbf{M}}) = 0$, where:

$$\begin{aligned}
 N(\mathbf{B}, \mathbf{M}; \overline{\mathbf{B}}, \overline{\mathbf{M}}) &:= - \int_{\Omega} \overline{\mathbf{B}} \cdot \mathbf{B}_t \, d\mathbf{x} + c_1 \int_{\Omega} \overline{\mathbf{B}} \cdot \mathbf{F} \, d\mathbf{x} \\
 &\quad - c_2 \int_{\Omega} (\overline{\nabla} \times \overline{\mathbf{B}}) \cdot (\overline{\nabla} \times \mathbf{B}) \, d\mathbf{x} \\
 &\quad + c_1 \int_{\Omega} (\overline{\nabla} \times \overline{\mathbf{B}}) \cdot (\overline{\nabla} \times \mathbf{M}) \, d\mathbf{x} \\
 &\quad - \int_{\Omega} \overline{\mathbf{M}} \cdot \mathbf{M}_t \, d\mathbf{x} + c_5 \int_{\Omega} \overline{\mathbf{M}} \cdot \mathbf{Q} \, d\mathbf{x} \\
 &\quad - c_6 \int_{\Omega} \overline{\overline{\nabla}} [\overline{\mathbf{M}} \times \mathbf{M}] : k_4 \overline{\overline{\nabla}} \mathbf{M} \, d\mathbf{x} \\
 &\quad - c_7 \int_{\Omega} \overline{\overline{\nabla}} [(\overline{\mathbf{M}} \times \mathbf{M}) \times \mathbf{M}] : k_4 \overline{\overline{\nabla}} \mathbf{M} \, d\mathbf{x} \\
 &\quad + c_6 \int_{\Omega} \overline{\mathbf{M}} \cdot [\mathbf{M} \times k_3 (\mathbf{M} \cdot \hat{\mathbf{a}}) \hat{\mathbf{a}}] \, d\mathbf{x} \\
 &\quad + c_7 \int_{\Omega} \overline{\mathbf{M}} \cdot [\mathbf{M} \times (\mathbf{M} \times k_3 (\mathbf{M} \cdot \hat{\mathbf{a}}) \hat{\mathbf{a}})] \, d\mathbf{x} \\
 &\quad + c_6 \int_{\Omega} \overline{\mathbf{M}} \cdot [\mathbf{M} \times k_2 \mathbf{B}] \, d\mathbf{x} \\
 &\quad + c_7 \int_{\Omega} \overline{\mathbf{M}} \cdot [\mathbf{M} \times (\mathbf{M} \times k_2 \mathbf{B})] \, d\mathbf{x}
 \end{aligned} \tag{5.3}$$

Here, constants and parameters are lumped into symbols c_j 's and k_j 's for readability, better fit to the page, and far easier programmability:

$$c_1 = \frac{1}{\sigma}, \quad c_2 = \frac{1}{\mu\sigma}, \quad c_5 = \frac{1}{(1 + \alpha^2)M_s}, \quad c_6 = \frac{\gamma_0}{1 + \alpha^2}, \quad c_7 = \frac{\gamma_0\alpha}{(1 + \alpha^2)M_s} \tag{5.4}$$

$$k_2 = \frac{1}{\mu}, \quad k_3 = \frac{2k_{anis}}{M_s}, \quad k_4 = \frac{k_{exch}}{M_s^2} \tag{5.5}$$

5.2 Discretization in Space

5.2.1 Appropriate Hilbert Spaces

Since the weak form contains integrals involving $\overline{\nabla} \times \mathbf{B}$, $\overline{\nabla} \times \overline{\mathbf{B}}$, $\overline{\overline{\nabla}} \mathbf{M}$, $\overline{\overline{\nabla}} \overline{\mathbf{M}}$ and contains no boundary integrals, the following Hilbert spaces [13, 74, 90] are paramount:

Proposition 5.2.1. (Hilbert Spaces & Subsets for Coupled Micromagnetics Problem)

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with $d \in \{2, 3\}$ and outward unit normal \mathbf{n} .

Then, in order for the weak form of the coupled micromagnetics problem to be well-defined and devoid of non-zero boundary integrals, we require:

- Solution $\mathbf{B} \in H_*(\text{curl}; \Omega) := \{\mathbf{H} \in H(\text{curl}; \Omega) : \mathbf{H} = \mathbf{B}_D \text{ on } \partial\Omega\}$
- Test field $\overline{\mathbf{B}} \in H_0(\text{curl}; \Omega) := \{\mathbf{H} \in H(\text{curl}; \Omega) : \mathbf{n} \times \mathbf{H} = \vec{\mathbf{0}} \text{ on } \partial\Omega\}$

- Solution $\mathbf{M} \in [H^1(\Omega)]^d := \{\mathbf{H} \in [L^2(\Omega)]^d : \overline{\nabla} \mathbf{H} \in [L^2(\Omega)]^{d \times d}\}$, where $d \in \{2, 3\}$
- Test field $\overline{\mathbf{M}} \in [H_0^1(\Omega)]^d := \{\mathbf{H} \in [H^1(\Omega)]^d : \mathbf{H} = \vec{\mathbf{0}} \text{ on } \partial\Omega\}$

Recall the Hilbert space [78, 105] $L^2(\Omega) := \{u : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} u^2 \, d\mathbf{x} < \infty\}$.

When handling terms involving both \mathbf{B} & \mathbf{M} , the following fact helps:

Lemma 5.2.1. (A Tensor Gradient-conforming Vector Field is also Curl-conforming)

$$[H^1(\Omega)]^3 \subset H(\text{curl}; \Omega).$$

Proof. $\|\cdot\|$ is the 2-norm on \mathbb{R}^3 . Also, some inequalities from the Appendix are used.

Let $\mathbf{F}(\mathbf{x}) = (M, N, P) \in [H^1(\Omega)]^3$, then:

$$\begin{cases} M(\mathbf{x}) \in H^1(\Omega) \implies \int_{\Omega} \|\overline{\nabla} M\|^2 \, d\mathbf{x} < \infty \\ N(\mathbf{x}) \in H^1(\Omega) \implies \int_{\Omega} \|\overline{\nabla} N\|^2 \, d\mathbf{x} < \infty \\ P(\mathbf{x}) \in H^1(\Omega) \implies \int_{\Omega} \|\overline{\nabla} P\|^2 \, d\mathbf{x} < \infty \end{cases} \quad (5.6)$$

$$\implies \begin{cases} \int_{\Omega} (M_x)^2 \, d\mathbf{x} < \infty, \int_{\Omega} (M_y)^2 \, d\mathbf{x} < \infty, \int_{\Omega} (M_z)^2 \, d\mathbf{x} < \infty \\ \int_{\Omega} (N_x)^2 \, d\mathbf{x} < \infty, \int_{\Omega} (N_y)^2 \, d\mathbf{x} < \infty, \int_{\Omega} (N_z)^2 \, d\mathbf{x} < \infty \\ \int_{\Omega} (P_x)^2 \, d\mathbf{x} < \infty, \int_{\Omega} (P_y)^2 \, d\mathbf{x} < \infty, \int_{\Omega} (P_z)^2 \, d\mathbf{x} < \infty \end{cases} \quad (5.7)$$

Now, $\overline{\nabla} \times \mathbf{F} = (P_y - N_z, M_z - P_x, N_x - M_y)$.

$$\begin{aligned} \therefore \int_{\Omega} |P_y - N_z|^2 \, d\mathbf{x} &= \int_{\Omega} |P_y^2 - 2P_y N_z + N_z^2| \, d\mathbf{x} \\ &\stackrel{TI}{\leq} \int_{\Omega} |P_y^2| + 2|P_y N_z| + |N_z^2| \, d\mathbf{x} \\ &= \int_{\Omega} P_y^2 \, d\mathbf{x} + 2 \int_{\Omega} |P_y N_z| \, d\mathbf{x} + \int_{\Omega} N_z^2 \, d\mathbf{x} \\ &\stackrel{CS}{\leq} \int_{\Omega} P_y^2 \, d\mathbf{x} + 2 \left(\int_{\Omega} P_y^2 \, d\mathbf{x} \right)^{1/2} \left(\int_{\Omega} N_z^2 \, d\mathbf{x} \right)^{1/2} + \int_{\Omega} N_z^2 \, d\mathbf{x} \\ &< \infty \end{aligned} \quad (5.8)$$

Similarly, $\int_{\Omega} |M_z - P_x|^2 \, d\mathbf{x} < \infty$ and $\int_{\Omega} |N_x - M_y|^2 \, d\mathbf{x} < \infty$.

$$\therefore \int_{\Omega} \|\overline{\nabla} \times \mathbf{F}\|^2 \, d\mathbf{x} < \infty \implies \overline{\nabla} \times \mathbf{F} \in [L^2(\Omega)]^3 \implies \mathbf{F} \in H(\text{curl}; \Omega)$$

□

5.2.2 Pre-Element-Pre-Timestepper Weak Problem

With the necessary Hilbert Spaces and their relevant subspaces and subsets all identified, we can now state the pre-element-pre-timestepper weak problem:

$$\begin{aligned}
 &\text{Find } (\mathbf{B}(t; \cdot), \mathbf{M}(t; \cdot)) \in H_*(\text{curl}; \Omega) \times H^1(\Omega) \\
 &\quad \text{such that } N(\mathbf{B}(t; \cdot), \mathbf{M}(t; \cdot); \overline{\mathbf{B}}, \overline{\mathbf{M}}) = 0 \\
 &\quad \forall (\overline{\mathbf{B}}, \overline{\mathbf{M}}) \in H_0(\text{curl}; \Omega) \times H_0^1(\Omega) \quad \forall t > 0.
 \end{aligned} \tag{5.9}$$

5.2.3 Appropriate Mixed Finite Element

In terms of finite elements, the coupled micromagnetics problem requires a mixed element where \mathbf{B}_h uses Nedelec(1st kind) or Nedelec(2nd kind) elements and \mathbf{M}_h uses Vector Lagrange elements. This choice follows from the required Hilbert spaces stated in Proposition 5.2.1.

5.2.4 Post-Element-Pre-Timestepper Weak Problem

Armed with the appropriate mixed finite element, here is the post-element-pre-timestepper weak problem:

$$\begin{aligned}
 &\text{Find } (\mathbf{B}_h(t; \cdot), \mathbf{M}_h(t; \cdot)) \in H_*(\text{curl}; \Omega_h) \times H^1(\Omega_h) \\
 &\quad \text{such that } N(\mathbf{B}_h(t; \cdot), \mathbf{M}_h(t; \cdot); \overline{\mathbf{B}}, \overline{\mathbf{M}}) = 0 \\
 &\quad \forall (\overline{\mathbf{B}}, \overline{\mathbf{M}}) \in H_0(\text{curl}; \Omega_h) \times H_0^1(\Omega_h) \quad \forall t > 0.
 \end{aligned} \tag{5.10}$$

5.3 Discretization in Time

5.3.1 Partitioned Implicit Runge-Kutta (PIRK) Timesteppers

Sometimes, using the same timestepper for each solution variable of a time-dependent system is not good enough as each variable may require a timestepper with different stability and/or invariance properties. This is precisely the conundrum we face with the coupled micromagnetics problem. A fix to this predicament is to employ a partitioned Runge-Kutta timestepper [50] (first described in [55]):

Definition 5.3.1. (Partitioned Runge-Kutta Timesteppers)

Given the following ODE system:

$$(PRK - IVP) : \begin{cases} \mathbf{B}_t(t) = \mathbf{F}(t; \mathbf{B}, \mathbf{M}) \\ \mathbf{M}_t(t) = \mathbf{Q}(t; \mathbf{B}, \mathbf{M}) \\ \mathbf{B}(t_0) = \mathbf{B}_0 \\ \mathbf{M}(t_0) = \mathbf{M}_0 \end{cases}$$

Then the corresponding s -stage partitioned Runge-Kutta timestepper is:

$$\begin{array}{c|cccc|c|cccc} \alpha_1^{[\mathbf{B}]} & \beta_{11}^{[\mathbf{B}]} & \beta_{12}^{[\mathbf{B}]} & \cdots & \beta_{1s}^{[\mathbf{B}]} & \alpha_1^{[\mathbf{M}]} & \beta_{11}^{[\mathbf{M}]} & \beta_{12}^{[\mathbf{M}]} & \cdots & \beta_{1s}^{[\mathbf{M}]} \\ \alpha_2^{[\mathbf{B}]} & \beta_{21}^{[\mathbf{B}]} & \beta_{22}^{[\mathbf{B}]} & \cdots & \beta_{2s}^{[\mathbf{B}]} & \alpha_2^{[\mathbf{M}]} & \beta_{21}^{[\mathbf{M}]} & \beta_{22}^{[\mathbf{M}]} & \cdots & \beta_{2s}^{[\mathbf{M}]} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_s^{[\mathbf{B}]} & \beta_{s1}^{[\mathbf{B}]} & \beta_{s2}^{[\mathbf{B}]} & \cdots & \beta_{ss}^{[\mathbf{B}]} & \alpha_s^{[\mathbf{M}]} & \beta_{s1}^{[\mathbf{M}]} & \beta_{s2}^{[\mathbf{M}]} & \cdots & \beta_{ss}^{[\mathbf{M}]} \\ \hline & \gamma_1^{[\mathbf{B}]} & \gamma_2^{[\mathbf{B}]} & \cdots & \gamma_s^{[\mathbf{B}]} & & \gamma_1^{[\mathbf{M}]} & \gamma_2^{[\mathbf{M}]} & \cdots & \gamma_s^{[\mathbf{M}]} \end{array}$$

where the left & right Butcher tables correspond to IRK timesteppers exclusively used to time-discretize \mathbf{B}_t & \mathbf{M}_t respectively.

- If both timesteppers are explicit, then the entire timestepper is called a partitioned explicit Runge-Kutta (PERK) timestepper.
- If both timesteppers are implicit, then the entire timestepper is called a partitioned implicit Runge-Kutta (PIRK) timestepper.

Proposition 5.3.1. (Order of partitioned Runge-Kutta timestepper)

Let the two timesteppers comprising a partitioned Runge-Kutta timestepper have orders p & q respectively. Then, the effective order of the entire partitioned Runge-Kutta timestepper is $\min\{p, q\}$.

Proof. (From [50]) Apply the timestepper to the following decoupled ODE system:

$$(PRK - IVP) : \begin{cases} \mathbf{B}_t(t) = \mathbf{F}(t; \mathbf{B}) \\ \mathbf{M}_t(t) = \mathbf{Q}(t; \mathbf{M}) \\ \mathbf{B}(t_0) = \mathbf{B}_0 \\ \mathbf{M}(t_0) = \mathbf{M}_0 \end{cases} \quad (5.11)$$

□

5.3.2 Appropriate PIRK Timestepper

For the coupled micromagnetics problem, it's essential that such a partitioned Runge-Kutta timestepper be implicit. The timestepper for \mathbf{B} ideally should be L-stable. The timestepper for \mathbf{M} should be QIP due to the quadratic invariant involving \mathbf{M} . Only IRK timesteppers can possibly be either L-stable or QIP at high order. Ideally, the α_j 's for each chosen timestepper should be identical. Fortunately, we can achieve all this by using certain Lobatto family timesteppers:

Proposition 5.3.2. (Appropriate PIRK Timestepper for coupled problem)

For the coupled problem, construct an s -stage PIRK timestepper such that:

- The timestepper for \mathbf{B} is LobattoIIIC(s).
- The timestepper for \mathbf{M} is LobattoIIIS(s).

The resulting PIRK timestepper will be called LobattoIII[C|S](s).

From Table 3.1 & Proposition 5.3.1, LobattoIII[C|S](s) has order $(2s - 2)$.

Table 5.1: Chosen PIRK Timesteppers

TIMESTEPPER:	BUTCHER TABLE:							
Lobatto III[C S](2)	0				0			
	$\frac{1}{2}$	$-\frac{1}{2}$			$\frac{1}{4}$	0		
	1	$\frac{1}{2}$	$\frac{1}{2}$		1	$\frac{1}{2}$	$\frac{1}{4}$	
		$\frac{1}{2}$	$\frac{1}{2}$			$\frac{1}{2}$	$\frac{1}{2}$	
Lobatto III[C S](3)	0	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{6}$	0	$\frac{1}{12}$	$-\frac{1}{12}$	0
	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{5}{12}$	$-\frac{1}{12}$	$\frac{1}{2}$	$\frac{3}{16}$	$\frac{1}{3}$	$-\frac{1}{48}$
	1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$	1	$\frac{1}{6}$	$\frac{3}{4}$	$\frac{1}{12}$
		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Similar Radau PIRK timesteppers can be constructed (e.g. Radau I[A|B](s) & Radau II[A|B](s)) but are omitted here as the Lobatto ones were found first.

5.3.3 Post-Element-Post-Timestepper Weak Form

Finally, armed with the appropriate mixed finite element and s -stage PIRK timestepper, we arrive at the final weak form $\sum_{j=1}^s N_j = 0$, where:

$$\begin{aligned}
 N_j = & - \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{B}}^{[j]} \cdot \mathbf{K}_{\mathbf{B},h}^{[j]} d\mathbf{x} + c_1 \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{B}}^{[j]} \cdot \mathbf{F}_h^{(k+\alpha_j)} d\mathbf{x} \\
 & - c_2 \int_{\Omega_h} (\overline{\nabla} \times \overline{\mathbf{K}}_{\mathbf{B}}^{[j]}) \cdot (\overline{\nabla} \times \mathbf{B}_h^{(\omega_j)}) d\mathbf{x} \\
 & + c_1 \int_{\Omega_h} (\overline{\nabla} \times \overline{\mathbf{K}}_{\mathbf{B}}^{[j]}) \cdot (\overline{\nabla} \times \mathbf{M}_h^{(\omega_j)}) d\mathbf{x} \\
 & - \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \cdot \mathbf{K}_{\mathbf{M},h}^{[j]} d\mathbf{x} + c_5 \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \cdot \mathbf{Q}_h^{(k+\alpha_j)} d\mathbf{x} \\
 & - c_6 \int_{\Omega_h} \overline{\overline{\nabla}} \left[\overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \times \mathbf{M}_h^{(\omega_j)} \right] : k_4 \overline{\overline{\nabla}} \mathbf{M}_h^{(\omega_j)} d\mathbf{x} \\
 & - c_7 \int_{\Omega_h} \overline{\overline{\nabla}} \left[\left(\overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \times \mathbf{M}_h^{(\omega_j)} \right) \times \mathbf{M}_h^{(\omega_j)} \right] : k_4 \overline{\overline{\nabla}} \mathbf{M}_h^{(\omega_j)} d\mathbf{x} \\
 & + c_6 \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \cdot \left[\mathbf{M}_h^{(\omega_j)} \times k_3 (\mathbf{M}_h^{(\omega_j)} \cdot \widehat{\mathbf{a}}) \widehat{\mathbf{a}} \right] d\mathbf{x} \\
 & + c_7 \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \cdot \left[\mathbf{M}_h^{(\omega_j)} \times \left(\mathbf{M}_h^{(\omega_j)} \times k_3 (\mathbf{M}_h^{(\omega_j)} \cdot \widehat{\mathbf{a}}) \widehat{\mathbf{a}} \right) \right] d\mathbf{x} \\
 & + c_6 \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \cdot \left[\mathbf{M}_h^{(\omega_j)} \times k_2 \mathbf{B}_h^{(\omega_j)} \right] d\mathbf{x} \\
 & + c_7 \int_{\Omega_h} \overline{\mathbf{K}}_{\mathbf{M}}^{[j]} \cdot \left[\mathbf{M}_h^{(\omega_j)} \times \left(\mathbf{M}_h^{(\omega_j)} \times k_2 \mathbf{B}_h^{(\omega_j)} \right) \right] d\mathbf{x}
 \end{aligned} \tag{5.12}$$

Recall the symbols c_j 's and k_j 's were defined in subsection 5.1.2. Moreover, some special notation is employed for readability and ease-of-writing:

Definition 5.3.2. (Timestep Notation for \mathbf{F}_h & \mathbf{Q}_h)

Suppose an s -stage PIRK timestepper is used for the micromagnetics problem.

Then, the final weak form will contain the following quantities:

$$\begin{aligned}
 \mathbf{F}_h^{(k+\alpha_j)} & := \mathbf{F}_h(t_k + \alpha_j(\Delta t); \mathbf{x}) \\
 \mathbf{Q}_h^{(k+\alpha_j)} & := \mathbf{Q}_h(t_k + \alpha_j(\Delta t); \mathbf{x})
 \end{aligned}$$

where t_k is the current timestep, Δt is the timestep size, the α_j 's come from the timestepper's Butcher table, and index $j \in \{1, 2, \dots, s\}$.

Definition 5.3.3. (Warp Notation for \mathbf{B}_h & \mathbf{M}_h)

Suppose an s -stage PIRK timestepper is used for the micromagnetics problem.

Then, the final weak form will contain the following quantities:

$$\begin{aligned}\mathbf{B}_h^{(k+\omega_j)} &:= \mathbf{B}_h^{(k)} + (\Delta t) \left(\beta_{j,1}^{[\mathbf{B}]} \mathbf{K}_{\mathbf{B}}^{[1]} + \cdots + \beta_{j,s}^{[\mathbf{B}]} \mathbf{K}_{\mathbf{B}}^{[s]} \right) \\ \mathbf{M}_h^{(k+\omega_j)} &:= \mathbf{M}_h^{(k)} + (\Delta t) \left(\beta_{j,1}^{[\mathbf{M}]} \mathbf{K}_{\mathbf{M}}^{[1]} + \cdots + \beta_{j,s}^{[\mathbf{M}]} \mathbf{K}_{\mathbf{M}}^{[s]} \right)\end{aligned}$$

where $\beta_{j,k}^{[\mathbf{B}]}$ & $\beta_{j,k}^{[\mathbf{M}]}$ are the (j, k) -entries of the PIRK Butcher table matrices for \mathbf{B} & \mathbf{M} respectively, Δt is the timestep size, and the index $j \in \{1, 2, \dots, s\}$.

5.3.4 Post-Element-Post-Timestepper Weak Problem

With the final weak form, we can now state the full weak formulation of the micromagnetics problem when using an s -stage PIRK timestepper:

$$\begin{aligned}\text{Find } (\mathbf{K}_{\mathbf{B},h}^{[1]}(t; \cdot), \mathbf{K}_{\mathbf{M},h}^{[1]}(t; \cdot), \dots, \mathbf{K}_{\mathbf{B},h}^{[s]}(t; \cdot), \mathbf{K}_{\mathbf{M},h}^{[s]}(t; \cdot)) &\in \prod_{j=1}^s [H_*(\text{curl}; \Omega_h) \times H^1(\Omega_h)] \\ \text{such that } \sum_{j=1}^s N_j &= 0 \tag{5.13} \\ \forall (\bar{\mathbf{K}}_{\mathbf{B}}^{[1]}, \bar{\mathbf{K}}_{\mathbf{M}}^{[1]}, \dots, \bar{\mathbf{K}}_{\mathbf{B}}^{[s]}, \bar{\mathbf{K}}_{\mathbf{M}}^{[s]}) &\in \prod_{j=1}^s [H_0(\text{curl}; \Omega_h) \times H_0^1(\Omega_h)] \quad \forall t > 0.\end{aligned}$$

5.3.5 Treatment of Inhomogeneous Dirichlet BC's

In contrast to other methods such as Dirichlet Splitting, the inhomogeneous Dirichlet boundary conditions on \mathbf{B} are handled by the timestepper by first taking the time-derivative on both sides and then applying the s -stage timestepper:

$$\begin{aligned}\mathbf{B} &= \mathbf{B}_D \quad \text{on } \partial\Omega \\ \xrightarrow{\partial/\partial t} \mathbf{B}_t &= \frac{\partial \mathbf{B}_D}{\partial t} \quad \text{on } \partial\Omega \\ \xrightarrow{\text{PIRK}} \mathbf{K}_{\mathbf{B}}^{[j]} &= \frac{\partial}{\partial t} \left[\mathbf{B}_D^{(k+\alpha_j)} \right] \quad \text{on } \partial\Omega \quad \text{for } j \in \{1, 2, \dots, s\}\end{aligned} \tag{5.14}$$

CHAPTER VI
VALIDATION TESTS

The core algorithm is shown below independent of programming language:

6.1 Simulation Pseudocode

<p>Data: Mesh Ω_h; timestep size Δt; final timestep T</p> <p>Data: Physical constants $M_s, \gamma_0, \alpha, \sigma, \mu_0, k_{anis}, k_{exch}, \hat{\mathbf{a}}$</p> <p>Data: Dirichlet boundary condition (BC) $\mathbf{B}_D(t; \mathbf{x})$</p> <p>Data: Initial conditions (IC's) $\mathbf{B}_0(\mathbf{x})$ & $\mathbf{M}_0(\mathbf{x})$</p> <p>Data: s-stage [L-stable, QIP]-PIRK timestepper</p> <hr/> <p>Initialize computed soln's with their IC's: $(\mathbf{B}_h^{(0)}, \mathbf{M}_h^{(0)}) \leftarrow (\mathbf{B}_0, \mathbf{M}_0)$;</p> <p>Initialize timestep counter $k \leftarrow 0$;</p> <p>Initialize timestep $t \leftarrow 0$;</p> <p>while $t < T$ do</p> <div style="border-left: 1px solid black; border-right: 1px solid black; padding-left: 10px;"> <p>for $j \in \{1, 2, \dots, s\}$ do</p> <div style="border-left: 1px solid black; border-right: 1px solid black; padding-left: 10px;"> <p>Set Dirichlet BC for j^{th} stage $\mathbf{K}_{\mathbf{B},h}^{[j]}$ to $\frac{\partial}{\partial t} [\mathbf{B}_D^{(k+\alpha_j)}]$;</p> </div> <p>end</p> <p>Solve weak form $\sum_{j=1}^s N_j = 0$ for $(\mathbf{K}_{\mathbf{B},h}^{[1]}, \mathbf{K}_{\mathbf{M},h}^{[1]}, \dots, \mathbf{K}_{\mathbf{B},h}^{[s]}, \mathbf{K}_{\mathbf{M},h}^{[s]})$;</p> <p>$\mathbf{B}_h^{(k+1)} \leftarrow \mathbf{B}_h^{(k)} + (\Delta t)[\gamma_1^{[\mathbf{B}]} \mathbf{K}_{\mathbf{B},h}^{[1]} + \dots + \gamma_s^{[\mathbf{B}]} \mathbf{K}_{\mathbf{B},h}^{[s]}]$;</p> <p>$\mathbf{M}_h^{(k+1)} \leftarrow \mathbf{M}_h^{(k)} + (\Delta t)[\gamma_1^{[\mathbf{M}]} \mathbf{K}_{\mathbf{M},h}^{[1]} + \dots + \gamma_s^{[\mathbf{M}]} \mathbf{K}_{\mathbf{M},h}^{[s]}]$;</p> <p>(Compute any post-processing functionals);</p> <p>(Write out visualization profiles to file);</p> <p>$k \leftarrow k + 1$;</p> <p>$t \leftarrow t + \Delta t$;</p> </div> <p>end</p> <hr/> <p>Result: Computed solutions $\mathbf{B}_h(t)$ & $\mathbf{M}_h(t)$ for $t \in \{0, \Delta t, 2\Delta t, \dots, T\}$</p>

Algorithm 1: Pseudocode for LLG+EC simulation

6.2 Remarks on Software Implementation

6.2.1 FEniCS

The simulation code is written in Python and uses the FEniCS library [80] for all finite element functionality:

- Basic mesh generation
- Axis-aligned triangulation/tetrahedralization of mesh
- Assortment of common finite elements & function spaces
- Efficient quadrature schemes on triangles & tetrahedra
- Assembly of weak forms into (non)linear systems
- Access to (non)linear solvers provided by system's linear algebra backend (e.g. PETSc [7], Eigen, Trilinos, ...)
- Computation of post-processing functionals
- Exportation of computations to file for visualization tools (e.g. ParaView)

6.2.2 NumPy

The NumPy [98] library provides the core scientific computation capabilities that FEniCS utilizes. Occasionally, it is necessary to directly call NumPy functions to compute:

- Averages: `numpy.mean()`
- Maximums: `numpy.max()`
- Norms of vectors: `numpy.linalg.norm()`

6.2.3 SymPy

In order to automatically treat boundary conditions as well as painlessly perform validation tests and sanity checks, we use SymPy [87] to perform symbolic computation. Moreover, the `sympy.vector` module is used to compute divergences and curls of vector fields.

6.2.4 Automatic Handling of Boundary Conditions

Symbolic manipulation via SymPy is necessary to impose the appropriate Dirichlet boundary conditions for each timestepper stage without doing so manually:

$$\frac{\partial}{\partial t} \left[\mathbf{B}_D^{(k+\alpha_j)} \right] \iff \text{sympy.diff}(\text{BD.subs}([(t, t+(\text{alpha_j*dt}))]), t) \quad (6.1)$$

6.3 Validation Tests

Just because the code executes error-free does not guarantee that the program is actually working. To assure that there are no logic errors, it is vital to perform validation tests on the code. Since our problem involves PDE's, we will use a particular kind of validation called the Method of Manufactured Solutions [73, 97, 102, 107]:

1. Pick a domain $\Omega \subset \mathbb{R}^d$ ($d \in \{2, 3\}$) with a piecewise smooth boundary $\partial\Omega$.
2. Replace the RHS zero vectors of the micromagnetics problem's two PDE's with arbitrary vector fields $\mathbf{F}(t; \mathbf{x})$ and $\mathbf{Q}(t; \mathbf{x})$:

$$\left\{ \begin{array}{l} \mathbf{B}_t - \frac{1}{\mu\sigma} \bar{\nabla}^2 \mathbf{B} - \frac{1}{\sigma} \bar{\nabla} \times \bar{\nabla} \times \mathbf{M} = \mathbf{F} \quad \text{in } \Omega \times [0, T] \\ \nabla \cdot \mathbf{B} = 0 \quad \text{in } \Omega \times [0, T] \\ \hline \frac{1}{\mu} \mathbf{B} - \mathbf{M} + \frac{2k_{anis}}{M_s} (\mathbf{M} \cdot \hat{\mathbf{a}}) \hat{\mathbf{a}} + \frac{k_{exch}}{M_s^2} \bar{\nabla}^2 \mathbf{M} = \mathbf{H}_{eff} \quad \text{in } \Omega \times [0, T] \\ \mathbf{M}_t - \frac{\gamma_0}{1+\alpha^2} (\mathbf{M} \times \mathbf{H}_{eff}) - \frac{\gamma_0 \alpha}{(1+\alpha^2)M_s} \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{eff}) = \mathbf{Q} \quad \text{in } \Omega \times [0, T] \\ \hline \mathbf{B} = \mathbf{B}_D \quad \text{on } \partial\Omega \times [0, T] \\ \partial\mathbf{M}/\partial n = \vec{\mathbf{0}} \quad \text{on } \partial\Omega \times [0, T] \\ \hline \mathbf{B} = \mathbf{B}_0 \quad \text{at } \bar{\Omega} \times \{t = 0\} \\ \mathbf{M} = \mathbf{M}_0 \quad \text{at } \bar{\Omega} \times \{t = 0\} \end{array} \right. \quad (6.2)$$

3. Manufacture an exact solution for \mathbf{B} in $[H_D(\text{curl}; \Omega)]^d$, denoted \mathbf{B}_{ex} , such that it is uniformly divergence-free in space and time.
4. Manufacture an exact solution for \mathbf{M} in $[H^1(\Omega)]^d$, denoted \mathbf{M}_{ex} , such that it satisfies the homogeneous Neumann BC on $\partial\Omega$ as well as being uniformly constant-norm in space and time.
5. Set saturation constant $M_s = \|\mathbf{M}\|$.
6. Choose values for the physical constants $\mu, \sigma, k_{anis}, k_{exch}, \gamma_0, \alpha, \hat{\mathbf{a}}$.
7. Plug these manufactured solutions into the two PDE's in order to determine the RHS's \mathbf{F} and \mathbf{Q} .
8. Evaluate \mathbf{B}_{ex} on the boundary $\partial\Omega$, resulting in the BC \mathbf{B}_D .
9. Evaluate \mathbf{B}_{ex} and \mathbf{M}_{ex} at time $t = 0$, resulting in the IC's \mathbf{B}_0 and \mathbf{M}_0 .
10. Using all the problem data (i.e. $\mathbf{F}, \mathbf{Q}, \mathbf{B}_D, \mathbf{B}_0, \mathbf{M}_0, M_s, \mu, \sigma, k_{anis}, k_{exch}, \gamma_0, \alpha, \hat{\mathbf{a}}$), run the simulation code to determine the computed solutions \mathbf{B}_h and \mathbf{M}_h .
11. Compute the solution errors $\|\mathbf{B}_h - \mathbf{B}_{ex}\|_{[L^2(\Omega)]^d}$ and $\|\mathbf{M}_h - \mathbf{M}_{ex}\|_{[L^2(\Omega)]^d}$.
12. Repeat the simulation run with either a smaller (refined) cell diameter h or timestep size Δt .
 - Refining h each run is called a CAMR validation test.
 - CAMR is short for “Convergence Analysis of Mesh Refinements.”
 - Refining Δt each run is called a CATS validation test.
 - CATS is short for “Convergence Analysis of Timestep Sizes.”
13. Post-process the results: plot, check convergence rates, etc...

6.3.1 Manufactured Solutions for the Unit Square $\Omega := [0, 1]^2$

It is key to run validation tests with manufactured solutions that may not be fully space- and time-dependent to ensure that the simulation code is robust. Here is a set of manufactured solutions for CAMR validation tests:

Table 6.1: CAMR Manufactured Solutions for the Unit Square

Solution Pairs: $\left(\mathbf{M}_{ex}(\cdot) := \begin{bmatrix} \cos \theta(\cdot) \\ -\sin \theta(\cdot) \end{bmatrix}, p(x, y) := xy(1-x)(1-y) \right)$	
$\mathbf{B}_{ex}(t) := \begin{bmatrix} \sin(t) \\ -\cos(t) \end{bmatrix}$	$\theta(t) := t$
$\mathbf{B}_{ex}(t) := \begin{bmatrix} t/100 \\ -t/100 \end{bmatrix}$	$\theta(\mathbf{x}) := 10 \cos(\sqrt{3} \cdot p(x, y))$
$\mathbf{B}_{ex}(t) := \begin{bmatrix} t/100 \\ -t/100 \end{bmatrix}$	$\theta(t; \mathbf{x}) := (t/100) + 10 \cos(\sqrt{3} \cdot p(x, y))$
$\mathbf{B}_{ex}(\mathbf{x}) := \begin{bmatrix} 10 \sin(10\sqrt{7} \cdot y) \\ 10 \cos(10\sqrt{5} \cdot x) \end{bmatrix}$	$\theta(t) := t/100$
$\mathbf{B}_{ex}(\mathbf{x}) := \begin{bmatrix} 10 \sin(10\sqrt{7} \cdot y) \\ 10 \cos(10\sqrt{5} \cdot x) \end{bmatrix}$	$\theta(\mathbf{x}) := 10 \cos(\sqrt{3} \cdot p(x, y))$
$\mathbf{B}_{ex}(\mathbf{x}) := \begin{bmatrix} 10 \sin(10\sqrt{7} \cdot y) \\ 10 \cos(10\sqrt{5} \cdot x) \end{bmatrix}$	$\theta(t; \mathbf{x}) := (t/100) + 10 \cos(\sqrt{3} \cdot p(x, y))$
$\mathbf{B}_{ex}(t; \mathbf{x}) := \begin{bmatrix} (t/100) + 10 \sin(10\sqrt{7} \cdot y) \\ -(t/100) + 10 \cos(10\sqrt{5} \cdot x) \end{bmatrix}$	$\theta(t) := t/100$
$\mathbf{B}_{ex}(t; \mathbf{x}) := \begin{bmatrix} (t/100) + 10 \sin(10\sqrt{7} \cdot y) \\ -(t/100) + 10 \cos(10\sqrt{5} \cdot x) \end{bmatrix}$	$\theta(\mathbf{x}) := 10 \cos(\sqrt{3} \cdot p(x, y))$
$\mathbf{B}_{ex}(t; \mathbf{x}) := \begin{bmatrix} (t/100) + 10 \sin(10\sqrt{7} \cdot y) \\ -(t/100) + 10 \cos(10\sqrt{5} \cdot x) \end{bmatrix}$	$\theta(t; \mathbf{x}) := (t/100) + 10 \cos(\sqrt{3} \cdot p(x, y))$

Below is a corresponding set of manufactured solutions for CATS validation tests:

Table 6.2: CATS Manufactured Solutions for the Unit Square

Solution Pairs: $\left(\mathbf{M}_{ex}(\cdot) := \begin{bmatrix} \cos \theta(\cdot) \\ -\sin \theta(\cdot) \end{bmatrix}, p(x, y) := xy(1-x)(1-y) \right)$	
$\mathbf{B}_{ex}(t) := \begin{bmatrix} 10 \sin(10\sqrt{7} \cdot t) \\ 10 \cos(10\sqrt{5} \cdot t) \end{bmatrix}$	$\theta(t) := t$
$\mathbf{B}_{ex}(t) := \begin{bmatrix} 10 \sin(10\sqrt{7} \cdot t) \\ 10 \cos(10\sqrt{5} \cdot t) \end{bmatrix}$	$\theta(\mathbf{x}) := [p(x, y)]^4$
$\mathbf{B}_{ex}(t) := \begin{bmatrix} 10 \sin(10\sqrt{7} \cdot t) \\ 10 \cos(10\sqrt{5} \cdot t) \end{bmatrix}$	$\theta(t; \mathbf{x}) := \sin(\sqrt{3} \cdot t) + [p(x, y)]^{12}$
$\mathbf{B}_{ex}(\mathbf{x}) := \begin{bmatrix} y \\ -x \end{bmatrix}$	$\theta(t) := t$
$\mathbf{B}_{ex}(\mathbf{x}) := \begin{bmatrix} y \\ -x \end{bmatrix}$	$\theta(\mathbf{x}) := [p(x, y)]^4$
$\mathbf{B}_{ex}(\mathbf{x}) := \begin{bmatrix} y \\ -x \end{bmatrix}$	$\theta(t; \mathbf{x}) := \sin(\sqrt{3} \cdot t) + [p(x, y)]^{12}$
$\mathbf{B}_{ex}(t; \mathbf{x}) := \begin{bmatrix} y + 10 \sin(10\sqrt{7} \cdot t) \\ -x + 10 \cos(10\sqrt{5} \cdot t) \end{bmatrix}$	$\theta(t) := t$
$\mathbf{B}_{ex}(t; \mathbf{x}) := \begin{bmatrix} y + 10 \sin(10\sqrt{7} \cdot t) \\ -x + 10 \cos(10\sqrt{5} \cdot t) \end{bmatrix}$	$\theta(\mathbf{x}) := [p(x, y)]^4$
$\mathbf{B}_{ex}(t; \mathbf{x}) := \begin{bmatrix} y + 10 \sin(10\sqrt{7} \cdot t) \\ -x + 10 \cos(10\sqrt{5} \cdot t) \end{bmatrix}$	$\theta(t; \mathbf{x}) := \sin(\sqrt{3} \cdot t) + [p(x, y)]^{12}$

6.3.2 The Importance of Sufficient Refinement

In general, when running validation tests, using too coarse of a fixed timestep size Δt may cause the resulting CAMR lineplot to “bottom out” as shown here with the blue and orange lineplots:

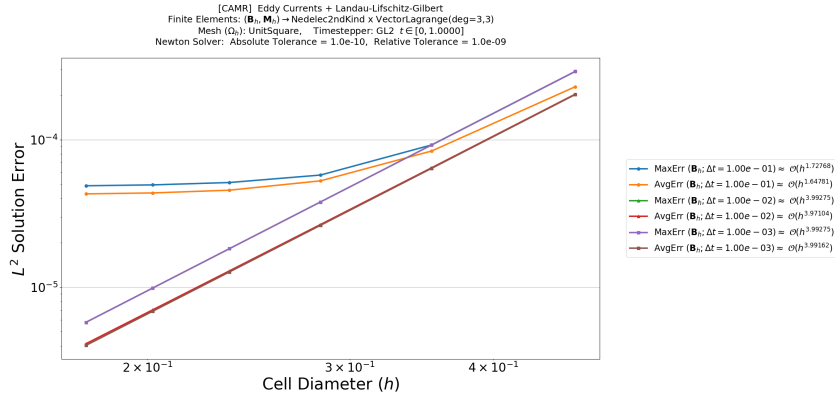


Figure 6.1: “Bottoming Out” of CAMR Line Plots (blue & orange lines)

What caused this is the fact that the solution error due to the fixed timestep size ($\Delta t = 0.1$) is at least an order of magnitude larger than the error due to the current cell diameter. Fortunately, this issue vanishes upon refinement ($\Delta t = 0.01$.)

A coarse fixed cell diameter h may result in “bottoming out” for CATS plots:

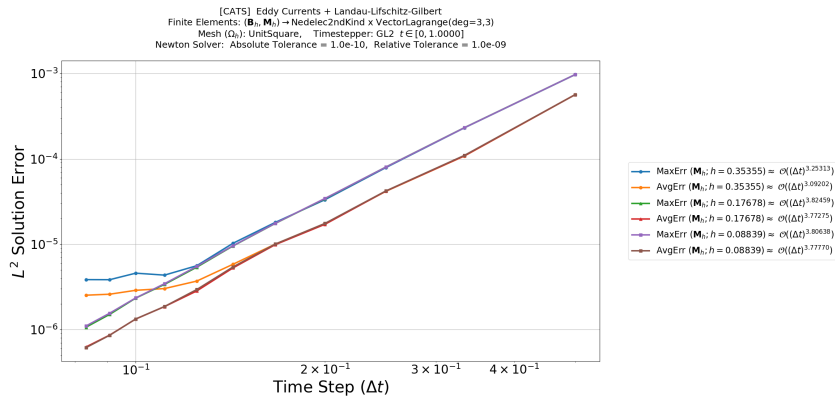


Figure 6.2: “Bottoming Out” of CATS Line Plots (blue & orange lines)

What caused this is the fact that the solution error due to the fixed cell diameter ($h \approx 0.3536$) is at least an order of magnitude larger than the error due to the current

timestep size. The cure is to use a finer mesh (e.g. $h \approx 0.1768$.)

The example validation plots in the next three subsections use sufficiently refined timestep sizes and cell diameters to prevent such undesirable behavior.

6.3.3 Example Validation Plots with $\mathbf{B}_{ex}(t)$ & $\mathbf{M}_{ex}(t)$

Since the exact solutions solely depend on time and not on space, refining the cell diameter h neither improves nor worsens the error in the computed solutions. Hence, their CAMR lineplots are flat, even with the coarsest timestep size of $\Delta t = 0.5$:

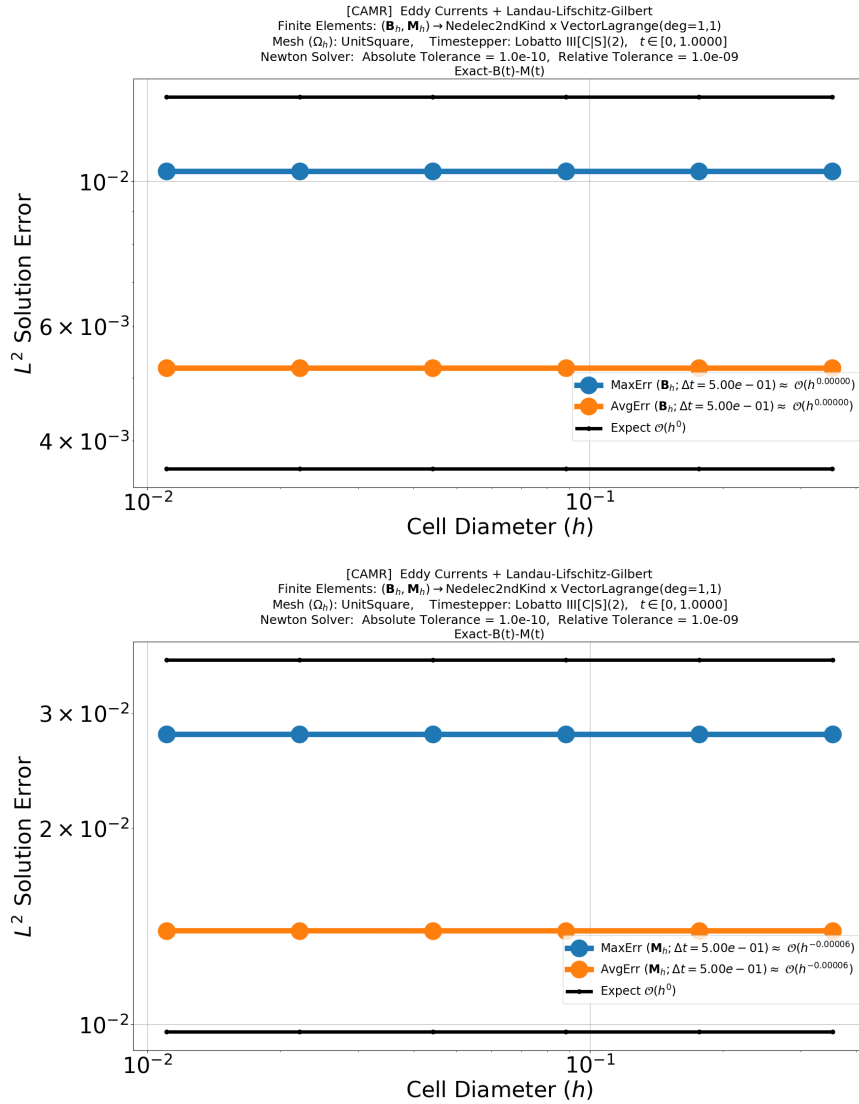


Figure 6.3: CAMR Plots of LLG+EC with Time-Dependent Exact Solutions

On the otherhand, refining the timestep size Δt should improve the solution errors nearly at the expected convergence rate of the timestepper. Therefore, their CATS lineplots should closely track the expected convergence lines (thick black lines):

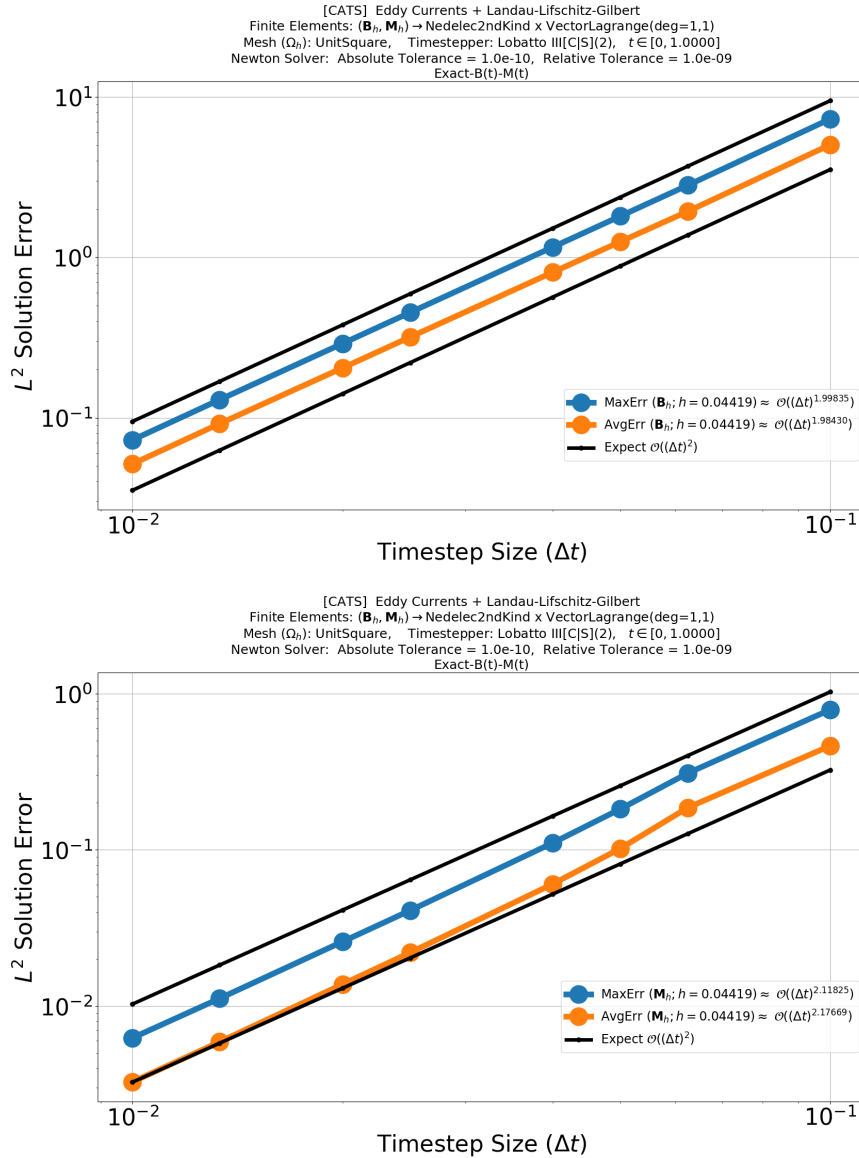


Figure 6.4: CATS Plots of LLG+EC with Time-Dependent Exact Solutions

6.3.4 Example Validation Plots with $\mathbf{B}_{ex}(\mathbf{x})$ & $\mathbf{M}_{ex}(\mathbf{x})$

With solely space-dependent exact solutions, refining the cell diameter h should improve the solution errors nearly at the expected convergence rate of the corresponding finite element. Therefore, their CAMR lineplots should closely track the expected convergence lines (thick black lines):

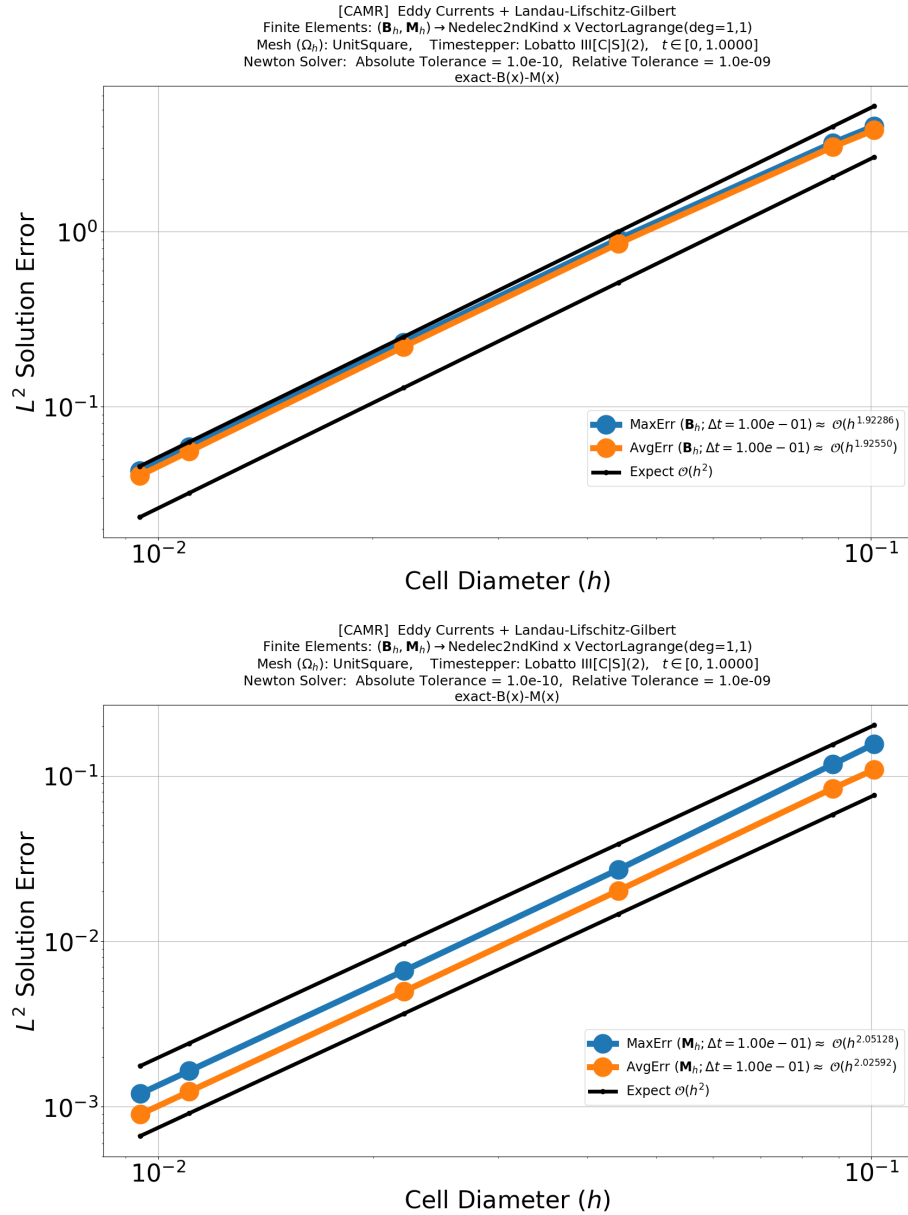


Figure 6.5: CAMR Plots of LLG+EC with Space-Dependent Exact Solutions

The lack of time dependence implies that their CATS line plots should be flat, even with the coarsest mesh bearing cell diameter $h \approx 0.7071$:

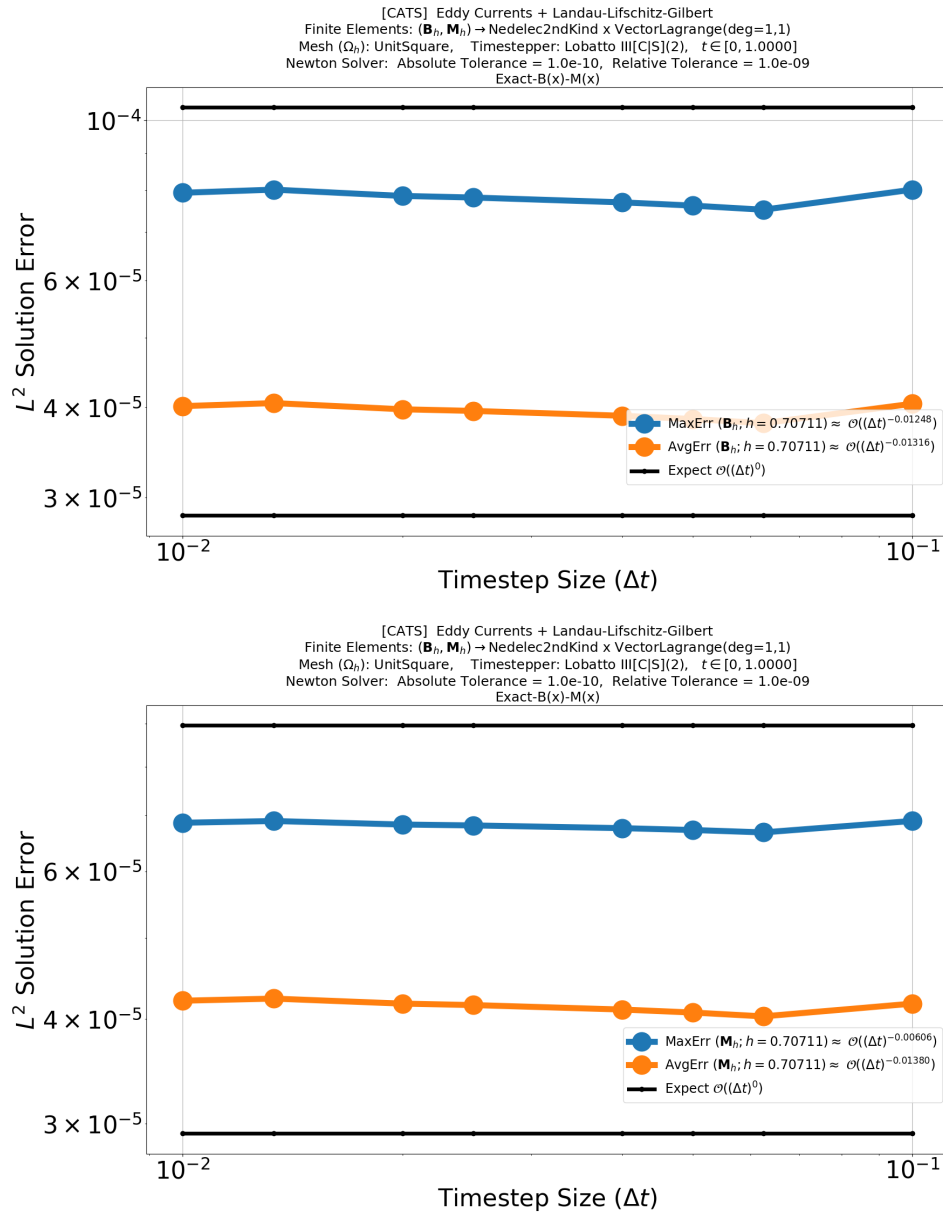


Figure 6.6: CATS Plots of LLG+EC with Space-Dependent Exact Solutions

6.3.5 Example Validation Plots with $\mathbf{B}_{ex}(t; \mathbf{x})$ & $\mathbf{M}_{ex}(t; \mathbf{x})$

With exact solutions bearing full space and time dependence, the resulting CAMR and CATS lineplots should track expected convergence rates for the corresponding finite element and timestepper, respectively:

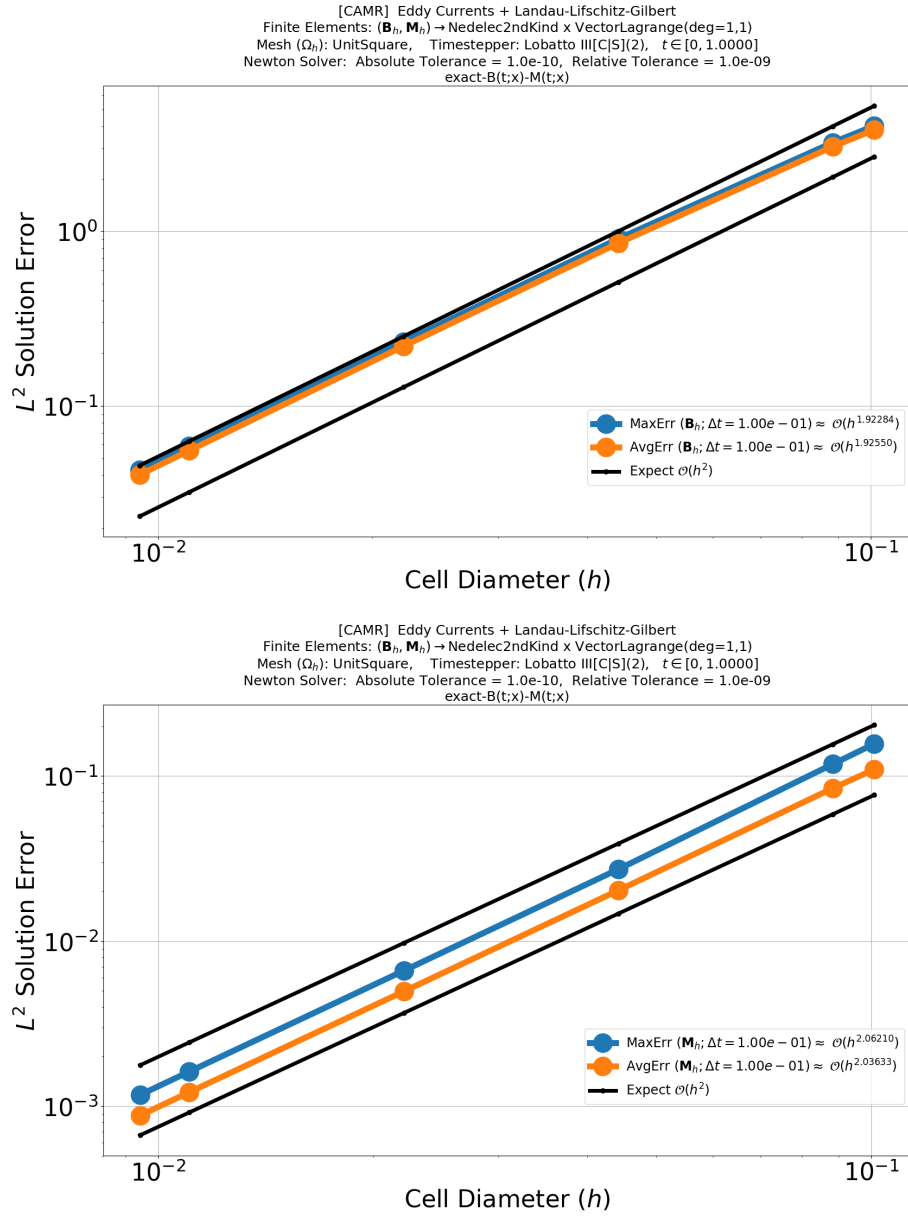


Figure 6.7: CAMR Plots of LLG+EC with Space-Time Exact Solutions

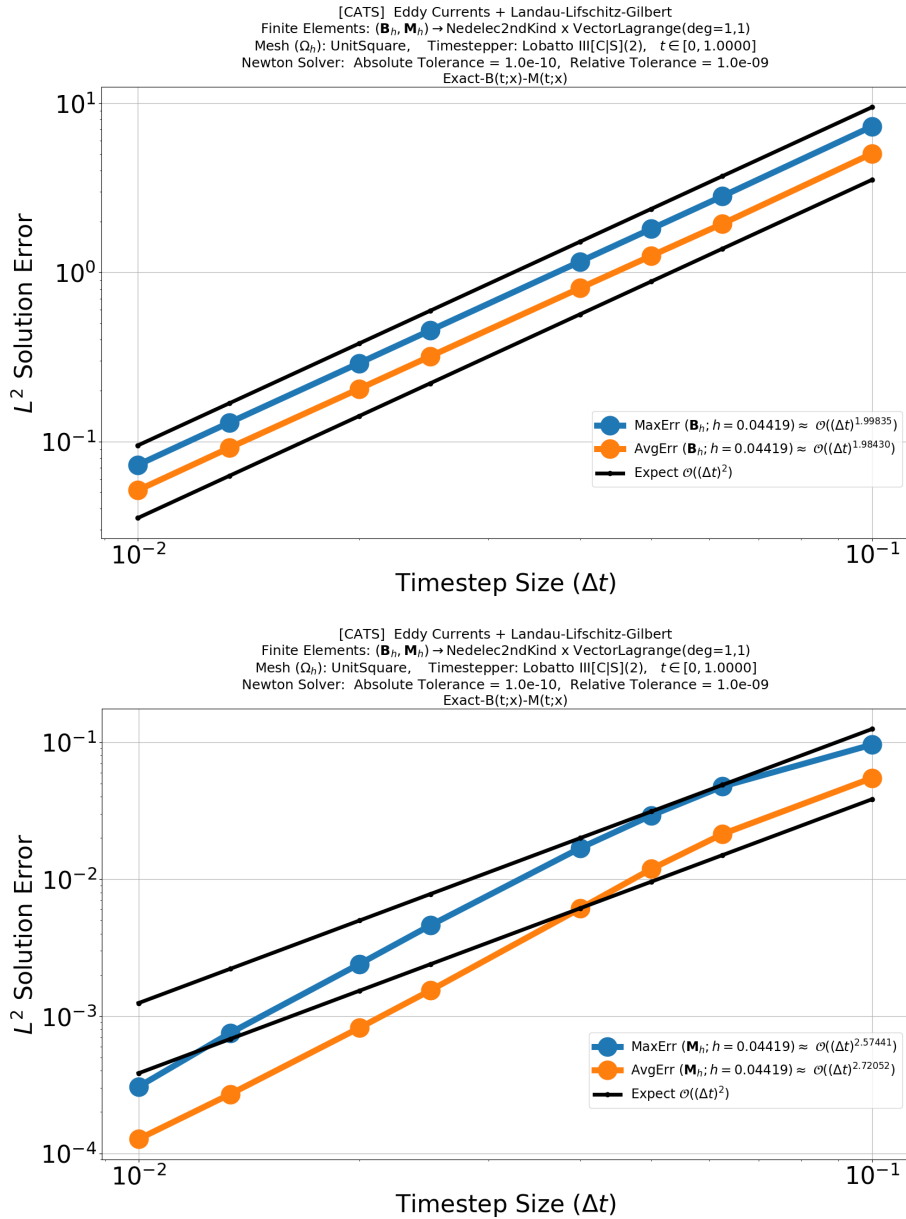


Figure 6.8: CATS Plots of LLG+EC with Space-Time Exact Solutions

Notice that finer fixed cell diameters and fixed timestep sizes are necessary in this case. In fact, the other six exact solution configurations should exhibit similar tracking behavior and refinement requirements for both CAMR and CATS validation plots. Because of this, further example plots are omitted to reduce clutter.

6.4 Summary of all Validation Tests for Unit Square $\Omega := [0, 1]^2$

Table 6.3: Validation Tests on the Unit Square. Here, $p := (2s - 2)$ is the order of timestepper. The physical constants were chosen as follows: $\gamma_0 = -1$, $\hat{\mathbf{a}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mu = \sigma = k_{anis} = k_{exch} = \alpha = 1$.

$\Omega = [0, 1]^2$ $t \in [0, T]; T = 1$ $\mathbf{B}_h \in \text{Ned}_{k+1}^{1st}$ $\mathbf{B}_h \in \text{Ned}_k^{2nd}$ $\mathbf{M}_h \in \text{V Lag}_k$ LobIII[C S](s)	$B_{ex}(t)$	$B_{ex}(\mathbf{x})$	$B_{ex}(t; \mathbf{x})$
$M_{ex}(t)$	$B_h : \mathcal{O}(h^0)$ $M_h : \mathcal{O}(h^0)$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$
$M_{ex}(\mathbf{x})$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^0)$ $M_h : \mathcal{O}((\Delta t)^0)$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$
$M_{ex}(t; \mathbf{x})$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$	$B_h : \mathcal{O}(h^{k+1})$ $M_h : \mathcal{O}(h^{k+1})$ <hr/> $B_h : \mathcal{O}((\Delta t)^p)$ $M_h : \mathcal{O}((\Delta t)^p)$

CHAPTER VII
SIMULATION ERROR ESTIMATES

We are interested in constructing error estimates for the computed solution when performing simulations: (the symbols c_j 's & k_j 's were defined in subsection 5.1.2)

$$\left\{ \begin{array}{l} \mathbf{B}_t - c_2 \bar{\nabla}^2 \mathbf{B} - c_1 \bar{\nabla} \times \bar{\nabla} \times \mathbf{M} = \vec{\mathbf{0}} \quad \text{in } \Omega \times [0, T] \\ \nabla \cdot \mathbf{B} = 0 \quad \text{in } \Omega \times [0, T] \\ \hline k_2 \mathbf{B} - \mathbf{M} + k_3 (\mathbf{M} \cdot \hat{\mathbf{a}}) \hat{\mathbf{a}} + k_4 \bar{\nabla}^2 \mathbf{M} = \mathbf{H}_{eff} \quad \text{in } \Omega \times [0, T] \\ \mathbf{M}_t - c_6 (\mathbf{M} \times \mathbf{H}_{eff}) - c_7 \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{eff}) = \vec{\mathbf{0}} \quad \text{in } \Omega \times [0, T] \\ \hline \mathbf{B} = \mathbf{B}_D \quad \text{on } \partial\Omega \times [0, T] \\ \partial\mathbf{M}/\partial n = \vec{\mathbf{0}} \quad \text{on } \partial\Omega \times [0, T] \\ \hline \mathbf{B} = \mathbf{B}_0 \quad \text{at } \bar{\Omega} \times \{t = 0\} \\ \mathbf{M} = \mathbf{M}_0 \quad \text{at } \bar{\Omega} \times \{t = 0\} \end{array} \right. \quad (7.1)$$

Error estimates are useful as they provide upper bounds on solution errors. This informs how refined h and Δt should be to achieve a sufficiently small error.

7.1 A Priori QIP Error Estimate

First, a lemma to aid in the proof of the a priori QIP error estimate:

Lemma 7.1.1. (Nested Norm Lemma)

Let $\mathbf{A}(\mathbf{x}), \mathbf{B}(\mathbf{x}) \in [C(\bar{\Omega})]^d$ with piecewise-smooth bounded domain Ω . Then:

$$(NNL) \quad \int_{\Omega} |\mathbf{A} \cdot \mathbf{B}| \, d\mathbf{x} \leq \|\mathbf{A}\|_{[L^2(\Omega)]^d} \cdot \|\mathbf{B}\|_{[L^2(\Omega)]^d}$$

Proof. Apply Cauchy-Schwarz Inequality (CSI) to the integrand via the 2-norm. Then apply CSI to the integral via the L^2 norm, resulting in nested norms:

$$\begin{aligned} \int_{\Omega} |\mathbf{A} \cdot \mathbf{B}| \, d\mathbf{x} &\stackrel{CSI}{\leq} \int_{\Omega} \|\mathbf{A}\|_2 \cdot \|\mathbf{B}\|_2 \, d\mathbf{x} = \left| \int_{\Omega} \|\mathbf{A}\|_2 \cdot \|\mathbf{B}\|_2 \, d\mathbf{x} \right| \\ &\stackrel{CSI}{\leq} \|\|\mathbf{A}\|_2\|_{L^2(\Omega)} \cdot \|\|\mathbf{B}\|_2\|_{L^2(\Omega)} \stackrel{L^2}{=} \left(\int_{\Omega} \mathbf{A} \cdot \mathbf{A} \, d\mathbf{x} \right)^{1/2} \cdot \left(\int_{\Omega} \mathbf{B} \cdot \mathbf{B} \, d\mathbf{x} \right)^{1/2} \quad (7.2) \\ &\stackrel{[L^2]^d}{=} \|\mathbf{A}\|_{[L^2(\Omega)]^d} \cdot \|\mathbf{B}\|_{[L^2(\Omega)]^d} \end{aligned}$$

□

Proposition 7.1.1. (A Priori QIP Error Estimate)

Let problem (7.1) have mesh $\Omega_h \in \mathbb{R}^d$ and N_T total timesteps & timestep size Δt . Let \mathbf{M} be the exact solution to problem (7.1). Let Hilbert space $V := [L^2(\Omega_h)]^d$. Let $\mathbf{M}_h^{(k)}$ be the computed solution using FEM and an s -stage PIRK timestepper. Let $\tau > 0$ be the absolute $\|\cdot\|_V$ Newton tolerance in applying PIRK on $\mathbf{M}_h^{(k)}$ once.

Then an a priori error estimate for the error in the 2-norm of $\mathbf{M}_h^{(N_T)}$ is:

$$\left| \int_{\Omega_h} \|\mathbf{M}_h^{(N_T)}\|_2^2 - \|\mathbf{M}_0\|_2^2 \, d\mathbf{x} \right| \leq |\Omega_h|^{1/2} \cdot [(2M_s + N_T(\Delta t)\tau)] \cdot \|\mathbf{M}_h^{(N_T)} - \mathbf{M}\|_V$$

Proof. Errors in $\mathbf{M}_h^{(1)}, \mathbf{M}_h^{(N_T)}$ due to timestepper are $(\Delta t)\tau, N_T(\Delta t)\tau$ respectively.

We apply appropriate Triangle Inequalities (TI), the Nested Norm Lemma (NNL) and the QIP-ness of the exact solution \mathbf{M} ($\|\mathbf{M}\|_2 := M_s$):

$$\begin{aligned} & \left| \int_{\Omega_h} \|\mathbf{M}_h^{(N_T)}\|_2^2 - \|\mathbf{M}_0\|_2^2 \, d\mathbf{x} \right| \\ &= \left| \int_{\Omega_h} \mathbf{M}_h^{(N_T)} \cdot \mathbf{M}_h^{(N_T)} - \mathbf{M}_0 \cdot \mathbf{M}_0 \, d\mathbf{x} \right| \\ &\stackrel{QIP}{=} \left| \int_{\Omega_h} \mathbf{M}_h^{(N_T)} \cdot \mathbf{M}_h^{(N_T)} - \mathbf{M} \cdot \mathbf{M} \, d\mathbf{x} \right| \\ &\stackrel{TI}{\leq} \int_{\Omega_h} \left| \mathbf{M}_h^{(N_T)} \cdot \mathbf{M}_h^{(N_T)} - \mathbf{M} \cdot \mathbf{M} \right| \, d\mathbf{x} \\ &\stackrel{CIZ}{=} \int_{\Omega_h} \left| \mathbf{M}_h^{(N_T)} \cdot \mathbf{M}_h^{(N_T)} + \left(\mathbf{M}_h^{(N_T)} \cdot \mathbf{M} - \mathbf{M}_h^{(N_T)} \cdot \mathbf{M} \right) - \mathbf{M} \cdot \mathbf{M} \right| \, d\mathbf{x} \\ &= \int_{\Omega_h} \left| \mathbf{M}_h^{(N_T)} \cdot \left[\mathbf{M}_h^{(N_T)} - \mathbf{M} \right] + \left[\mathbf{M}_h^{(N_T)} - \mathbf{M} \right] \cdot \mathbf{M} \right| \, d\mathbf{x} \tag{7.3} \\ &\stackrel{TI}{\leq} \int_{\Omega_h} \left| \mathbf{M}_h^{(N_T)} \cdot \left[\mathbf{M}_h^{(N_T)} - \mathbf{M} \right] \right| \, d\mathbf{x} + \int_{\Omega_h} \left| \left[\mathbf{M}_h^{(N_T)} - \mathbf{M} \right] \cdot \mathbf{M} \right| \, d\mathbf{x} \\ &\stackrel{NNL}{\leq} \|\mathbf{M}_h^{(N_T)}\|_V \cdot \|\mathbf{M}_h^{(N_T)} - \mathbf{M}\|_V + \|\mathbf{M}_h^{(N_T)} - \mathbf{M}\|_V \cdot \|\mathbf{M}\|_V \\ &\stackrel{QIP}{=} \left[\|\mathbf{M}_h^{(N_T)}\|_V + |\Omega_h|^{1/2} \cdot M_s \right] \cdot \|\mathbf{M}_h^{(N_T)} - \mathbf{M}\|_V \\ &\stackrel{PIRK}{\leq} \left[|\Omega_h|^{1/2} \cdot [M_s + N_T(\Delta t)\tau] + |\Omega_h|^{1/2} \cdot M_s \right] \cdot \|\mathbf{M}_h^{(N_T)} - \mathbf{M}\|_V \\ &= |\Omega_h|^{1/2} \cdot [(2M_s + N_T(\Delta t)\tau)] \cdot \|\mathbf{M}_h^{(N_T)} - \mathbf{M}\|_V \end{aligned}$$



7.2 A Posteriori QIP Error Estimate

We need a special norm and inner product for the a posteriori QIP error estimate:

Lemma 7.2.1. (Discretized Vector Lagrange Inner Product & Norm)

Let $\Omega \subset \mathbb{R}^d$ be a piecewise-smooth bounded domain with $d \in \{2, 3\}$.

Let $\Omega_h \subseteq \Omega$ be a mesh of domain Ω with a total of N_h nodes.

Let $\phi_1, \dots, \phi_{N_h}$ be the k^{th} -degree Vector Lagrange shape functions on Ω_h .

Let vector fields $\mathbf{A}_h, \mathbf{B}_h, \mathbf{C}_h \in [H^1(\Omega_h)]^d$.

Then, operations $\langle \cdot, \cdot \rangle_{DVL} : [H^1(\Omega_h)]^d \times [H^1(\Omega_h)]^d \rightarrow \mathbb{R}$ & $\|\cdot\|_{DVL} : [H^1(\Omega_h)]^d \rightarrow \mathbb{R}$

$$\begin{aligned} \langle \mathbf{A}_h, \mathbf{B}_h \rangle_{DVL} &:= \sum_{i=1}^{N_h} \sum_{j=1}^{N_h} \mathbf{A}_{h,i} \cdot \mathbf{B}_{h,j} \int_{\Omega_h} \phi_i \phi_j \, d\mathbf{x} \\ \|\mathbf{A}_h\|_{DVL} &:= \left(\sum_{i=1}^{N_h} \sum_{j=1}^{N_h} \mathbf{A}_{h,i} \cdot \mathbf{A}_{h,j} \int_{\Omega_h} \phi_i \phi_j \, d\mathbf{x} \right)^{1/2} \end{aligned}$$

are an inner product and corresponding norm on Ω_h , respectively.

Proof. Since the mass matrix M defined by $[m_{ij}]_{N_h \times N_h} := \int_{\Omega_h} \phi_i \phi_j \, d\mathbf{x}$ is symmetric positive definite, the inner product & norm are non-negative, being zero only if one of the arguments is the zero vector, and $\langle \mathbf{B}_h, \mathbf{A}_h \rangle_{DVL} = \langle \mathbf{A}_h, \mathbf{B}_h \rangle_{DVL}$.

Linearity implies that scalar multiples can be factored out of the inner product and norm; in addition, $\langle \mathbf{A}_h, \mathbf{B}_h + \mathbf{C}_h \rangle_{DVL} = \langle \mathbf{A}_h, \mathbf{B}_h \rangle_{DVL} + \langle \mathbf{A}_h, \mathbf{C}_h \rangle_{DVL}$.

Apply Cauchy-Schwarz Inequality (CSI) to verify norm's triangle inequality:

$$\begin{aligned} \|\mathbf{A}_h + \mathbf{B}_h\|_{DVL}^2 &:= \sum_{i=1}^{N_h} \sum_{j=1}^{N_h} (\mathbf{A}_{h,i} + \mathbf{B}_{h,i}) \cdot (\mathbf{A}_{h,j} + \mathbf{B}_{h,j}) \int_{\Omega_h} \phi_i \phi_j \, d\mathbf{x} \\ &= \|\mathbf{A}_h\|_{DVL}^2 + 2\langle \mathbf{A}_h, \mathbf{B}_h \rangle_{DVL} + \|\mathbf{B}_h\|_{DVL}^2 \\ &\leq \|\mathbf{A}_h\|_{DVL}^2 + 2|\langle \mathbf{A}_h, \mathbf{B}_h \rangle_{DVL}| + \|\mathbf{B}_h\|_{DVL}^2 \\ &\stackrel{CSI}{\leq} \|\mathbf{A}_h\|_{DVL}^2 + 2\|\mathbf{A}_h\|_{DVL} \cdot \|\mathbf{B}_h\|_{DVL} + \|\mathbf{B}_h\|_{DVL}^2 \\ &= (\|\mathbf{A}_h\|_{DVL} + \|\mathbf{B}_h\|_{DVL})^2 \end{aligned} \tag{7.4}$$

$$\therefore \|\mathbf{A}_h + \mathbf{B}_h\|_{DVL} \leq \|\mathbf{A}_h\|_{DVL} + \|\mathbf{B}_h\|_{DVL}$$

\therefore Inner product $\langle \cdot, \cdot \rangle_{DVL}$ and norm $\|\cdot\|_{DVL}$ are both well-defined.

□

Proposition 7.2.1. (A Posteriori QIP Error Estimate)

Let problem (7.1) have mesh $\Omega_h \in \mathbb{R}^d$ and timestep size $\Delta t > 0$.

Let $\mathbf{M}_h^{(k)}$ be the computed solution using FEM and an s -stage PIRK timestepper.

Let $\tau > 0$ be the absolute $\|\cdot\|_{V_h}$ Newton tolerance in applying PIRK on $\mathbf{M}_h^{(k)}$ once.

Then an a posteriori error estimate for the QIP error at timestep $t = t_{k+1}$ is:

$$\left| \|\mathbf{M}_h^{(k+1)}\|_{V_h}^2 - \|\mathbf{M}_h^{(k)}\|_{V_h}^2 \right| \leq 2(\Delta t)\tau \|\mathbf{M}_h^{(k)}\|_{V_h} + (\Delta t)^2\tau^2$$

where $\|\cdot\|_{V_h} := \|\cdot\|_{DVL}$ is the discretized Vector Lagrange norm on Ω_h .

Proof. First, define the Newton solver error in one step of the PIRK timestepper:

$$\mathbf{M}_h^{(k+1)} = \mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} + (\Delta t)\mathbf{E}^{(k)}, \text{ where } \begin{cases} \mathbf{S}^{(k)} & := \sum_{j=1}^s \gamma_j^{[M]} \mathbf{K}_{\mathbf{M},h}^{[j]} \\ \|\mathbf{E}^{(k)}\|_{V_h} & \leq \tau \quad \forall k \end{cases} \quad (7.5)$$

Next, take squared V_h -norms on both sides:

$$\begin{aligned} & \|\mathbf{M}_h^{(k+1)}\|_{V_h}^2 = \|\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} + (\Delta t)\mathbf{E}^{(k)}\|_{V_h}^2 \\ & = \left\langle \left(\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right) + (\Delta t)\mathbf{E}^{(k)}, \left(\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right) + (\Delta t)\mathbf{E}^{(k)} \right\rangle_{V_h} \\ & = \|\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)}\|_{V_h}^2 + 2 \left\langle (\Delta t)\mathbf{E}^{(k)}, \mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right\rangle_{V_h} + (\Delta t)^2 \|\mathbf{E}^{(k)}\|_{V_h}^2 \\ & \stackrel{QIP}{=} \|\mathbf{M}_h^{(k)}\|_{V_h}^2 + 2(\Delta t) \left\langle \mathbf{E}^{(k)}, \left(\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right) \right\rangle_{V_h} + (\Delta t)^2 \|\mathbf{E}^{(k)}\|_{V_h}^2 \end{aligned} \quad (7.6)$$

Subtracting both sides by $\|\mathbf{M}_h^{(k)}\|_{V_h}^2$ and taking absolute values results in:

$$\begin{aligned} & \left| \|\mathbf{M}_h^{(k+1)}\|_{V_h}^2 - \|\mathbf{M}_h^{(k)}\|_{V_h}^2 \right| \\ & = \left| 2(\Delta t) \left\langle \mathbf{E}^{(k)}, \left(\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right) \right\rangle_{V_h} + (\Delta t)^2 \|\mathbf{E}^{(k)}\|_{V_h}^2 \right| \\ & \stackrel{TI}{\leq} \left| 2(\Delta t) \left\langle \mathbf{E}^{(k)}, \left(\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right) \right\rangle_{V_h} \right| + |(\Delta t)^2 \|\mathbf{E}^{(k)}\|_{V_h}^2| \\ & \stackrel{CSI}{\leq} 2(\Delta t) \|\mathbf{E}^{(k)}\|_{V_h} \left\| \left(\mathbf{M}_h^{(k)} + (\Delta t)\mathbf{S}^{(k)} \right) \right\|_{V_h} + (\Delta t)^2 \|\mathbf{E}^{(k)}\|_{V_h}^2 \\ & \stackrel{QIP}{=} 2(\Delta t) \|\mathbf{E}^{(k)}\|_{V_h} \|\mathbf{M}_h^{(k)}\|_{V_h} + (\Delta t)^2 \|\mathbf{E}^{(k)}\|_{V_h}^2 \\ & \leq 2(\Delta t)\tau \|\mathbf{M}_h^{(k)}\|_{V_h} + (\Delta t)^2\tau^2 \end{aligned} \quad (7.7)$$



CHAPTER VIII

CONCLUSION

In this dissertation, we have successfully constructed and tested a numerical algorithm to solve the coupled LLG equation with eddy currents in 2D equipped with inhomogeneous Dirichlet BC's for the magnetic inductance field \mathbf{B} and homogeneous Neumann BC's for the magnetization field \mathbf{M} .

The first chapter began with overviews of micromagnetics and eddy currents from physics with connections to everyday life. Next, a survey of previous work on the Landau-Lifschitz-Gilbert (LLG) Equation coupled with the Eddy Currents Equation follows; we observed that the employed numerical methods were either high-order but specialized or general-purpose but low-order. The chapter concluded with a derivation of the coupled problem.

Chapter 2 provides essential background on linear multistep and Runge-Kutta timesteppers, both explicit and implicit. Several common Runge-Kutta timesteppers are constructed via quadrature or collocation. Also, their order of accuracy is stated and sometimes proven.

In Chapter 3, the notions of consistency and stability of timesteppers are discussed. There are several types of stability that are illustrated with examples. Also, quadratic invariant-preserving (QIP) timesteppers are introduced and visually demonstrated. Finally, a table summarizing well-known timesteppers' stability and QIP properties (or lack thereof.) We end by making the case that we should only use implicit Runge-Kutta (IRK) timesteppers as some of them bear sufficient stability properties or are QIP.

Finite elements are discussed in Chapter 4 starting with an overview of the Finite Element Method (FEM) as applied to Poisson's Equation in 2D. We talk about weak forms, function spaces, meshes, and assembled algebraic systems of equations. Afterwards, a catalog of relevant finite elements are defined. We wrap up the chapter with tips on implementing FEM in software and why we picked FEniCS as our finite

element framework.

Chapter 5 is where we decided on appropriate finite elements and our constructed partitioned implicit Runge-Kutta (PIRK) timestepper to apply to the coupled problem. We justified our choices based on the problem's weak form, the fact that the Eddy Currents Equation is stiff, and the fact that the LLG Equation has an inherent quadratic invariant. We produced the problem's final weak form with the timestepper and finite elements in effect and then discussed how we handled the inhomogeneous Dirichlet BC's with respect to our PIRK timestepper.

Chapter 6 begins with discussion of the features of FEniCS. We then identified SymPy our library for performing symbolic computations. SymPy is used in the simulation code to automatically impose the inhomogeneous Dirichlet BC's. Lastly, we illustrated the Method of Manufactured Solutions and validation testing with an example on the unit square domain.

Finally, two types of error estimates for the QIP error in the computed solution are stated and proven in Chapter 7.

8.1 Future Work

In order for this work to be useful in simulation of real-world experiments of micromagnetics with eddy currents, the code must become more efficient as well as thoroughly tested on a 3D mesh.

8.1.1 Block Preconditioner

To solve the linearized algebraic system at each Newton iteration [52, 64, 65, 121, 122], we employed GMRES [64, 104, 122] with Incomplete LU preconditioning [104] as this linear solver is very general and robust. Bañas [9] built a multigrid [17] preconditioner for the 3D problem with homogeneous BC's for the Eddy Currents equation, linear finite elements and low-order timestepping scheme. However, the performance of linear solves can be improved by constructing a block preconditioner [80, 104] tailored specifically for our numerical schemes applied to the coupled LLG+EC PDE

system. Doing so requires careful inspection of the structured block system arising from the PIRK timestepper including the distribution of the eigenvalues.

Also helpful when constructing preconditioners are sparsity patterns of the Jacobian matrix which visually indicate the distribution of non-zero entries:

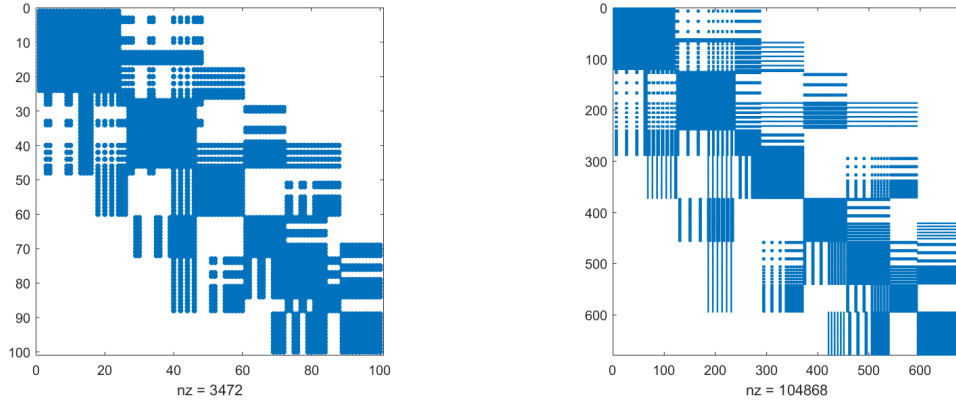


Figure 8.1: Jacobian Matrix Sparsity Pattern on a Coarse Unit Square Mesh ($h \approx 0.7071$) using the 2-stage PIRK, 1st-degree elements (left) and 3-stage PIRK, 3rd-degree elements (right). The number of zero entries is denoted by 'nz'.

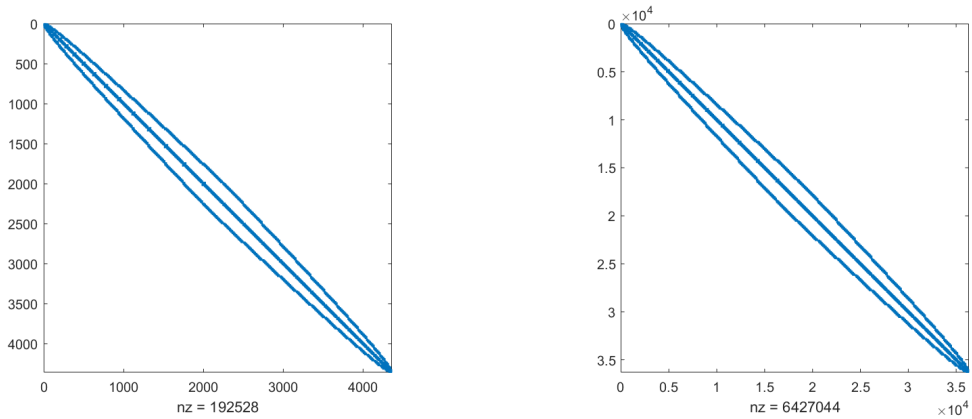


Figure 8.2: Jacobian Matrix Sparsity Pattern on a Fine Unit Square Mesh ($h \approx 0.0844$) using the 2-stage PIRK, 1st-degree elements (left) and 3-stage PIRK, 3rd-degree elements (right). Second kind Nédélec elements are used for \mathbf{B} . The number of zero entries is denoted by 'nz'.

Finally, the matrix shape and percentage of non-zeros in the Jacobian matrix for various mesh refinements is shown in the following table:

Table 8.1: Matrix Size (Percentage of Non-Zeros in Parentheses) in Jacobian Matrix for various Refinements of Unit Square Mesh. The computer used to produce these matrices has 32GB of RAM. For example, the top-left values of 100 (34.7%) indicate that the Jacobian matrix is 100x100 and about 34.7% of its entries are non-zero. Red entries estimate the required RAM necessary to produce the Jacobian matrix.

Cell Diameter h	Lobatto[$C S$](2)	Lobatto[$C S$](3)
	N2xVLag(1, 1)	N2xVLag(3, 3)
0.7071	100 (34.7%)	678 (22.8%)
0.0884	4,356 (1.0%)	36,294 (0.49%)
0.0110	264,196 (0.02%)	2,268,678 (0.0079%)
0.0055	1,052,676 (0.004%)	(> 32GB)
0.0028	4,202,500 (0.001%)	(> 64GB?)
0.0014	16,793,604 (0.0003%)	(> 128GB?)

8.1.2 Validation Tests in 3D on the Unit Cube

Our validation tests were performed exclusively in 2D on the unit square. Ideally, we desire for the validation tests to successively run in 3D on the unit cube. However, there are a few issues that currently impede this endeavor.

Early on in this research work, back in 2014, assembling a matrix using Nédélec elements in FEniCS with the PETSc backend [7] would cause a memory leak that would eventually exhaust all available computer memory. Fortunately, it was discovered that the copy constructor for PETSc matrices was the root cause of the memory leak and was fixed prior to the release of version 1.4 of FEniCS [29].

Unfortunately, another issue regarding matrix assembly with Nédélec elements still persists. This occurs specifically with 3^{rd} -degree Nédélec elements or higher

with a 3D mesh. Attempts to assemble a stiffness or mass matrix causes FEniCS to loop indefinitely [54]. Catastrophically, a workaround was proposed to use an internal representation for weak forms called 'quadrature', but it is deprecated in FEniCS version 2017.2.

To demonstrate the deprecation, running the following FEniCS Python code:

```
from dolfin import *
parameters['form_compiler']['representation'] = 'quadrature'
mesh = UnitCubeMesh(2,2,2)
V = FunctionSpace(mesh,'N1curl', 3) # 3rd-degree 1st-kind Nedelec
u = TrialFunction(V)
v = TestFunction(V)
a = dot(curl(u),curl(v))*dx
l = dot(u,v)*dx
A = PETScMatrix()
L = PETScMatrix()
assemble(a, tensor=A)
assemble(l, tensor=L)
```

results in the following output:

```
*** =====
*** FFC: quadrature representation is deprecated!
*** It will likely be removed in 2018.1.0 release.
*** Use uflacs representation instead.
*** =====
```

Alas, replacing the second line with the suggestion to use 'uflacs' instead:

```
parameters['form_compiler']['representation'] = 'uflacs'
```

causes the aforementioned infinite looping.

The only hope is that FEniCS is being completely overhauled [125] for a future release in a few years. Perhaps then these issues will be completely resolved. In addition, significant performance improvements will be realized. If the overhauled version can achieve both goals, only then will validation testing in 3D be feasible.

BIBLIOGRAPHY

- [1] C. Adams, R. Franzosa, Introduction to Topology: Pure & Applied, Pearson Prentice Hall, 2008.
- [2] A. Aharoni, Introduction to Theory of Ferromagnetism, Oxford Press, 2000.
- [3] R. Alexander, “Diagonally Implicit Runge-Kutta Methods for Stiff ODE’s”, SIAM Journal on Numerical Analysis, 14 (1977), 1006-1021.
- [4] M.S. Alnaes et al., “Unified Form Language: A Domain-specific Language for Weak Formulations of Partial Differential Equations”, ACM Transactions on Mathematical Software, 40 (2014), 1-38.
- [5] A.H. Al-Rabeh, “Optimal Order Diagonally Implicit Runge-Kutta Methods”, BIT, 33 (1993), 620-633.
- [6] ANSYS[®], Ansys, Inc., Canonsburg, PA, USA, 2018. www.ansys.com
- [7] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, “Efficient Management of Parallelism in Object Oriented Numerical Software Libraries”, Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997. www.mcs.anl.gov/petsc
- [8] W. Bangerth, R. Hartmann, G. Kanschat, “deal.II – A general-purpose Object-oriented Finite Element Library”, ACM Transactions on Mathematical Software, 33 (2007), 24/1-24/27. www.dealii.org
- [9] L. Bañas, “An Efficient Multigrid Preconditioner for Maxwell’s Equations in Micromagnetism”, Mathematics & Computers in Simulation, 80 (2010), 1657-1663.
- [10] L. Bañas, S. Bartels, A. Prohl, “A Convergent Implicit Finite Element Discretization of the Maxwell-Landau-Lifshitz-Gilbert Equation”, SIAM Journal on Numerical Analysis, 46 (2008), 1399-1422.
- [11] G. Bertotti, A. Magni, I.D. Mayergoyz, C. Serpico, “Landau-Lifshitz Magnetization Dynamics and Eddy Currents in Metallic Thin Films”, Journal of Applied Physics, 91 (2002), 7559-7561.
- [12] T.D. Blacker et al., “CUBIT Mesh Generation Environment, Volume 1: User’s Manual”, Technical Report SAND94-1100, Sandia National Laboratories, 1994. cubit.sandia.gov
- [13] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Springer, 2013.
- [14] P.D. Boom, D.W. Zingg, “Optimization of High-order Diagonally-implicit Runge-Kutta Methods”, Journal of Computational Physics, 371 (2018), 168-191.

- [15] J.-W. B. Bragg, J.C. Dickens, A.A. Neuber, K.R. Long, “High Field Magnetic Diffusion into Nonlinear Ferrimagnetic Materials for Pulse Sharpening Applications”, COMSOL Conference 2011, Boston, MA, 1-5.
- [16] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, 3rd Ed, Springer, 2008.
- [17] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, 2nd Edition, SIAM, 2000.
- [18] W.F. Brown Jr, Micromagnetics, Wiley, 1962.
- [19] J.C. Butcher, “Implicit Runge-Kutta Processes”, Mathematics of Computation, 18 (1964), 50-64.
- [20] J.C. Butcher, “Integration Processes based on Radau Quadrature Formulas”, Mathematics of Computation, 18 (1964), 233-244.
- [21] R.P.K. Chan, “On Symmetric Runge-Kutta Methods of High Order”, Computing, 45 (1990), 301-309.
- [22] R. Chang, V. Lomakin, E. Michielssen, “Coupled Electromagnetic – Micromagnetic Simulations of Complex Magnetic Structures”, 2013 IEEE Antennas and Propagation Society International Symposium, Orlando, FL, 1658-1659.
- [23] L. Chen, “Finite Element Methods for Maxwell Equations”, Lecture Notes, 2016. <https://www.math.uci.edu/~chenlong/226/FEMMaxwell.pdf> (Accessed June 28, 2018).
- [24] F.H. Chipman, “A-stable Runge-Kutta Processes”, BIT, 11 (1971), 384-388.
- [25] S.J. Colley, Vector Calculus, 4th Edition, Pearson, 2011.
- [26] COMSOL Multiphysics[®], COMSOL AB, Stockholm, Sweden, 2018. www.comsol.com
- [27] J.B. Conway, A Course in Point Set Topology, Springer, 2014.
- [28] G.J. Cooper, “Stability of Runge-Kutta Methods for Trajectory Problems”, IMA Journal of Numerical Analysis, 7 (1987), 1-13.
- [29] J. Córcoles, “Repeatedly creating a PETSc Matrix fills up Memory and kills Process”, <https://bitbucket.org/fenics-project/dolfin/issues/272> (Accessed June 24, 2018).
- [30] S. Couture et al., “Coupled Finite-Element Micromagnetic-Integral Equation Electromagnetic Simulator for Modeling Magnetization-Eddy Currents Dynamics”, IEEE Transactions on Magnetics, 53 (2017), 1-9.

- [31] G. Dahlquist, “Convergence and Stability in the Numerical Integration of Ordinary Differential Equations”, *Mathematica Scandinavica*, 4 (1956), 33-53.
- [32] G. Dahlquist, “A Special Stability Problem for Linear Multistep Methods”, *BIT*, 3 (1963), 27-43.
- [33] P.J. Davis, P. Rabinowitz, *Methods of Numerical Integration*, 2nd Edition, Academic Press, 1984.
- [34] B. Delaunay, “Sur la sphère vide”, *Izvestiya Akademii Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7 (1934), 793-800.
- [35] E. Della Torre, J.G. Eicke, “Eddy Currents in Micromagnetic Calculations”, *IEEE Transactions on Magnetics*, 33 (1997), 1251-1254.
- [36] H. Edelsbrunner, *Geometry and Topology for Mesh Generation*, Cambridge University Press, 2001.
- [37] B.L. Ehle, “On Padé Approximations to the Exponential Function and A-stable Methods for the Numerical Solution of Initial Value Problems”, Thesis, University of Waterloo, 1969.
- [38] A. Einstein, “Die Grundlage der allgemeinen Relativitätstheorie”, *Annalen der Physik*, 354 (1916), 769-822.
- [39] C.F. Gauss, “Methodus nova integralium valores per approximationem inveniendi”, *Commentationes Societatis Regiae Scientiarum Gottingensis Recentiores*, 3 (1814), 39-76.
- [40] S. Geng, “Construction of High Order Symplectic Runge-Kutta Methods”, *Journal of Computational Mathematics*, 11 (1993), 250-260.
- [41] C. Geuzaine, J.-F. Remacle, “Gmsh: A 3D Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities”, *International Journal for Numerical Methods in Engineering*, 79 (2009), 1309-1331. gmsh.info
- [42] T.L. Gilbert, “A Lagrangian Formulation of the Gyromagnetic Equation of the Magnetic Field”, *Physical Review*, 100 (1955), 1243.
- [43] T.L. Gilbert, “A Phenomenological Theory of Damping in Ferromagnetic Materials”, *IEEE Transactions on Magnetics*, 40 (2004), 3443-3449.
- [44] M.S. Gockenbach, *Understanding and Implementing the Finite Element Method*, SIAM, 2006.
- [45] D.J. Griffiths, *Introduction to Electrodynamics*, 3rd Edition, Prentice Hall, 1999.

- [46] P. Grinfeld, Introduction to Tensor Analysis and the Calculus of Moving Surfaces, Springer, 2013.
- [47] H. Haddar, P. Joly, “Effective Boundary Conditions for Thin Ferromagnetic Layers: The One-dimensional Model”, SIAM Journal on Applied Mathematics, 61 (2001), 1386-1417.
- [48] E. Hairer, “Highest Possible Order of Algebraically Stable Diagonally Implicit Runge-Kutta Methods”, BIT, 20 (1980), 254-256.
- [49] E. Hairer, G. Wanner, Solving ODE’s II: Stiff & Differential-Algebraic Problems, 2nd Edition, Springer, 2010.
- [50] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for ODE’s, 2nd Edition, Springer, 2006.
- [51] P.C. Hammer, J.W. Hollingsworth, “Trapezoidal Methods of Approximating Solutions of Differential Equations”, MTAC, 9 (1955), 92-96.
- [52] M.T. Heath, Scientific Computing: An Introductory Survey, 2nd Edition, McGraw-Hill, 2002.
- [53] F. Hecht, “New Development in FreeFem++”, Journal of Numerical Mathematics, 20 (2012), 251-265. www.freefem.org
- [54] J. Heller, “Assembling Matrices fails in 3D with High-order Nédélec Elements”, <https://bitbucket.org/fenics-project/dolfin/issues/469> (Accessed June 24, 2018).
- [55] E. Hofer, “A Partially Implicit Method for Large Stiff Systems of ODEs with Only Few Equations Introducing Small Time-Constants”, SIAM Journal on Numerical Analysis, 13 (1976), 645-663.
- [56] G. Hrkac et al., “Three-dimensional Micromagnetic Finite Element Simulations including Eddy Currents”, Journal of Applied Physics, 97 (2005), 10E311-1 – 10E311-3.
- [57] G. Hrkac et al., “Influence of Eddy Current on Magnetization Processes in Submicrometer Permalloy Structures”, IEEE Transactions on Magnetics, 41 (2005), 3097-3099.
- [58] J.D. Hunter, “Matplotlib: A 2D Graphics Environment”, Computing in Science & Engineering, 9 (2007), 90-95. www.matplotlib.org
- [59] A. Iserles, A First Course in the Numerical Analysis of Diff. Eqns, 2nd Edition, Cambridge University Press, 2009.

- [60] R. Iyer, J. Millhollon, K. Long, “Micromagnetics with Eddy Currents”, *Journal of Physics: Conference Series*, 268 (2011), 1-7.
- [61] D. Jiles, *Introduction to Magnetism and Magnetic Materials*, 3rd Edition, CRC Press, 2016.
- [62] J. Jin, *The Finite Element Method in Electromagnetics*, 3rd Edition, Wiley, 2014.
- [63] P. Joly, O. Vacus, “Mathematical and Numerical Studies of Nonlinear Ferromagnetic Materials”, *Modélisation Mathématique et Analyse Numérique*, 33 (1999), 593-626.
- [64] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.
- [65] C.T. Kelley, *Solving Nonlinear Equations with Newton’s Method*, SIAM, 2003.
- [66] C.A. Kennedy, M.H. Carpenter, “Diagonally Implicit Runge-Kutta Methods for ODE’s. A Review”, *Technical Memo*, NASA Langley Research Center, 2016.
- [67] T. Kluyver et al., “Jupyter Notebooks – A Publishing Format for Reproducible Computational Workflows”, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. (2016), 87-90. www.jupyter.org
- [68] J.F.B.M. Kraaijevanger, M.N. Spijker, “Algebraic Stability and Error Propagation in Runge-Kutta Methods”, *Applied Numerical Mathematics*, 5 (1989), 71-87.
- [69] F. Kukichi (菊地 文雄), “Numerical Analysis of Nedelec’s Edge Elements”, *Mathematical Analysis Lab. Record (数理解析研究所講究録)*, 1145 (2000), 1-9.
- [70] H. Kronmüller, M. Fähnle, *Micromagnetism and the Microstructure of Ferromagnetic Solids*, Cambridge University Press, 2003.
- [71] L.D. Landau, E.M. Lifschitz, “Zur Theorie der Dispersion der magnetische Permeabilität der ferromagnetische Körpern”, *Physikalische Zeitschrift der Sowjetunion*, 8 (1935), 153.
- [72] L.D. Landau, E.M. Lifschitz, *Collected Papers of L.D. Landau*, Gordon and Breach, 1965.
- [73] H.P. Langtangen, A. Logg, *Solving PDE’s in Python: The FEniCS Tutorial I*, Springer, 2017. Web. 25 March 2018. www.fenicsproject.org/tutorial
- [74] M.G. Larson, F. Bengzon, *The Finite Element Method*, Springer, 2013.
- [75] R. Larson, *Elementary Linear Algebra*, 8th Edition, Cengage, 2017. www.larsonlinearalgebra.com

- [76] K. Le, T. Tran, “A Convergent Finite Element Approximation for the Quasi-static Maxwell-Landau-Lifshitz-Gilbert Equations”, *Computers and Mathematics with Applications*, 66 (2013), 1389-1402.
- [77] R.J. Leveque, *Finite Difference Methods for ODE’s and PDE’s*, SIAM, 2007.
- [78] T.L. Lindstrøm, *Spaces: An Introduction to Real Analysis*, AMS, 2017.
- [79] R. Lobatto, *Lessen over de differentiaal- en integraal-rekening, Tweede Deel*, De Gebroeders Van Cleef, 1852.
- [80] A. Logg, K.-A. Mardal, G.N. Wells et al., *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, Springer, 2012. www.fenicsproject.org
- [81] K.R. Long, R.C. Kirby, B. van Bloemen Waanders, “Unified Embedded Parallel Finite Element Computations via Software-based Fréchet Differentiation”, *SIAM Journal on Scientific Computing*, 32 (2010), 3323-3351. trilinos.org
- [82] M. Lyly, J. Ruokolainen, E. Järvinen, “ELMER – A Finite Element Solver for Multiphysics”, Technical Report, CSC Ltd., Espoo, Finland, 2000. www.csc.fi/web/elmer
- [83] Maple[®], Waterloo Maple Inc., Waterloo, Ontario, Canada, 2018. www.maplesoft.com/products/maple
- [84] Mathematica[®], Wolfram Research Inc., Champaign, IL, USA, 2018. www.wolfram.com/mathematica
- [85] MATLAB[®], The MathWorks, Inc., Natick, MA, USA, 2018. www.mathworks.com/products/matlab.html
- [86] I.D. Mayergoyz, C. Serpico, Y. Shimizu, “Coupling between Eddy Currents and Landau-Lifshitz Dynamics”, *Journal of Applied Physics*, 87 (2000), 5529-5531.
- [87] A. Meurer et al., “SymPy: Symbolic Computing in Python”, *PeerJ Computer Science*. 3 (2017), e103. www.sympy.org
- [88] K. Miller, “Diagonally-implicit Runge-Kutta Methods for PDE’s and Stiff ODE’s”, Unpublished Lecture Notes, UC Berkeley, Berkeley, CA, USA, 1973.
- [89] J. Millhollon, “A Study of Numerical Methods for Ferromagnetic Materials with Dynamic Boundary Conditions”, Thesis, Texas Tech University, 2009.
- [90] P.B. Monk, *Finite Element Methods for Maxwell’s Eqns*, Oxford Press, 2003.
- [91] P.B. Monk, O. Vacus, “Accurate Discretisation of a Nonlinear Micromagnetic Problem”, *Computer Methods in Applied Mechanics and Engineering*, 190 (2001), 5243-5269.

- [92] MSC Nastran[®], MSC Software Corp., Newport Beach, CA, USA, 2018.
www.mscsoftware.com/product/msc-nastran
- [93] G. Mur, A.T. de Hoop, “A Finite-element Method for Computing Three-dimensional Electromagnetic Fields in Inhomogeneous Media”, IEEE Transactions on Magnetics, 21 (1985), 2188-2191.
- [94] J.C. Nédélec, “Mixed Finite Elements in \mathbb{R}^3 ”, Numerische Mathematik, 35 (1980), 315-341.
- [95] J.C. Nédélec, “A New Family of Mixed Finite Elements in \mathbb{R}^3 ”, Numerische Mathematik, 50 (1986), 57-81.
- [96] N.M. Newmark, “A Method of Computation for Structural Dynamics”, Journal of Engineering Mechanics, 85 (1959), 67-94.
- [97] W.L. Oberkampf, C.J. Roy, Verification and Validation in Scientific Computing, Cambridge University Press, 2010.
- [98] T.E. Oliphant, “Python for Scientific Computing”, Computing in Science & Engineering, 9 (2007), 10-20. www.numpy.org
- [99] M. Ostrogradsky, “Première note sur la théorie de la chaleur”, Mémoires de l’Académie impériale des sciences de St. Pétersbourg, 1 (1831), 129-133.
- [100] J. Pommier, Y. Renard, GetFEM++. An open source generic C++ library for finite element methods. getfem.org
- [101] R. Radau, “Etude sur les formules d’approximation qui servent à calculer la valeur numérique d’une intégrale définie”, Journal de Mathématiques Pures et Appliquées, 6 (1880), 283-336.
- [102] P.J. Roache, “Building PDE Codes to be Verifiable and Validatable”, Computing in Science & Engineering, 6 (2004), 30-38.
- [103] T. Rylander, A. Bondeson, P. Ingelström, Computational Electromagnetics, 2nd Edition, Springer, 2013.
- [104] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd Edition, SIAM, 2003.
- [105] P. Sacks, Techniques of Functional Analysis for Differential and Integral Equations, Academic Press, 2017.
- [106] E. Saff, A.D. Snider, Fundamentals of Complex Analysis, 3rd Ed, Pearson, 2003.
- [107] K. Salari, P. Knupp, “Code Verification by the Method of Manufactured Solutions”, Technical Report SAND2000-1444, Sandia National Laboratories, 2000.

- [108] G.M. Sandler, H.N. Bertram, “Micromagnetic Simulations with Eddy Currents of Rise Time in Thin Film Write Heads”, *Journal of Applied Physics*, 81 (1997), 4513-4515.
- [109] R. Scherer, “A Necessary Condition for B-stability”, *BIT*, 19 (1979), 111-115.
- [110] J. Schöberl, “NETGEN - An Advancing Front 2D/3D Mesh Generator based on Abstract Rules”, *Computing and Visualization in Science*, 1 (1997), 41-52. www.ngsolve.org
- [111] C. Serpico, I.D. Mayergoyz, G. Bertotti, “Analysis of Eddy Currents with Landau-Lifshitz Equation as a Constitutive Relation”, *IEEE Transactions on Magnetics*, 37 (2001), 3546-3549.
- [112] R.A. Serway, J.W. Jewett, *Physics for Scientists and Engineers*, 8th Edition, Brooks Cole, 2010.
- [113] J.R. Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator”, *Lecture Notes in Computer Science*, 1148 (1996), 203-222. www.cs.cmu.edu/~quake/triangle.html
- [114] J.R. Shewchuk, “Delaunay Refinement Algorithms for Triangular Mesh Generation”, *Computational Geometry: Theory and Applications*, 22 (2002), 21-74.
- [115] H. Si, “TetGen, a Delaunay-based Quality Tetrahedral Mesh Generator”, *ACM Transactions on Mathematical Software*, 41 (2015), 1-36. www.wias-berlin.de/software/index.jsp
- [116] M. Slodička, L. Bañas, “A Numerical Scheme for a Maxwell-Landau-Lifshitz-Gilbert System”, *Applied Mathematics and Computation*, 158 (2004), 79-99.
- [117] G. Strang, *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press, 2016.
- [118] A.H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice Hall, 1971.
- [119] L. Torres, L. Lopez-Diaz, E. Martínez, O. Alejos, “Micromagnetic Dynamic Computations including Eddy Currents”, *IEEE Transactions on Magnetics*, 39 (2003), 2498-2500.
- [120] L. Torres, E. Martínez, L. Lopez-Diaz, O. Alejos, “About the Inclusion of Eddy Currents in Micromagnetic Computations”, *Physica B*, 343 (2004), 257-261.
- [121] J.A. Trangenstein, *Scientific Computing: Vol. I – Linear & Nonlinear Equations*, Springer, 2017.

- [122] J.A. Trangenstein, *Scientific Computing: Vol. II – Eigenvalues & Optimization*, Springer, 2017.
- [123] J.A. Trangenstein, *Scientific Computing: Vol. III – Approximation & Integration*, Springer, 2017.
- [124] A. Visintin, “On Landau-Lifshitz’ Equations for Ferromagnetism”, *Japan Journal of Applied Mathematics*, 2 (1985), 69-84.
- [125] G.N. Wells, “FEniCS Core Developments”,
<https://groups.google.com/forum/#!topic/fenics-dev/ZHyW6ZjDToo>
(Accessed June 24, 2018).
- [126] H. Whitney, *Geometric Integration Theory*, Princeton University Press, 1957.
- [127] L. Yanik, E. Della Torre, M.J. Donahue, “A Testbed for a Finite-Difference Time Domain Micromagnetic Program with Eddy Currents”, *Physica B*, 343 (2004), 216-221.
- [128] L. Yanik, E. Della Torre, M.J. Donahue, “Micromagnetic Calculations of Eddy Currents with Time-varying Fields”, *Physica B*, 372 (2006), 290-293.
- [129] L. Zhang, T. Cui, H. Liu, “A Set of Symmetric Quadrature Rules on Triangles and Tetrahedra”, *Journal of Computational Mathematics*, 27 (2009), 89-96.

APPENDIX

A Taylor Series Expansions

Recall the Single-Variable Taylor Series Expansion (1TE) from calculus [25]:

Theorem A.1. (Single-Variable Taylor Series Expansion – Point Form)

Let $\Omega \subseteq \mathbb{R}$ be a domain with $x_0 \in \Omega$.

Let function $u(x) \in C^{k+1}(\Omega)$. Then:

$$u(x) = u(x_0) + \frac{x - x_0}{1!} \cdot \frac{du}{dx} \Big|_{x=x_0} + \cdots + \frac{(x - x_0)^k}{k!} \cdot \frac{d^k u}{dx^k} \Big|_{x=x_0} + r_k(x; x_0)$$

where $r_k(x; x_0) = \frac{(x - x_0)^{k+1}}{(k + 1)!} \cdot \frac{d^{k+1} u}{dx^{k+1}} \Big|_{x=\xi}$ with $\xi \in (\min\{x, x_0\}, \max\{x, x_0\})$.

Theorem A.2. (Single-Variable Taylor Series Expansion – Perturbed Form)

Let $\Omega \subseteq \mathbb{R}$ be a domain with $(x_0, x_0 + \Delta x) \subset \Omega$.

Let function $u(x) \in C^{k+1}(\Omega)$. Then:

$$u(x_0 + \Delta x) = u(x_0) + \frac{\Delta x}{1!} \cdot \frac{du}{dx} \Big|_{x=x_0} + \cdots + \frac{(\Delta x)^k}{k!} \cdot \frac{d^k u}{dx^k} \Big|_{x=x_0} + r_k(x; \Delta x)$$

where remainder term $r_k(x; \Delta x) = \frac{(\Delta x)^{k+1}}{(k + 1)!} \cdot \frac{d^{k+1} u}{dx^{k+1}} \Big|_{x=\xi}$ with $\xi \in (x_0, x_0 + \Delta x)$.

Generalizing to a Two-Variable Taylor Expansion (2TE) [25] is necessary for certain timestep construction and analysis:

Theorem A.3. (Two-Variable Taylor Series Expansion – Point Form)

Let $\Omega \subseteq \mathbb{R}^2$ be a star-shaped domain with $\mathbf{x}_0 = (x_{1,0}, x_{2,0}) = (x_0, y_0) \in \Omega$.

Let function $u(\mathbf{x}) = u(x_1, x_2) = u(x, y) \in C^{k+1}(\Omega)$. Then:

$$\begin{aligned} u(x, y) = & u(x_0, y_0) + \frac{(x - x_0)}{1!} \cdot \frac{\partial u}{\partial x} \Big|_{\mathbf{x}=\mathbf{x}_0} + \frac{(y - y_0)}{1!} \cdot \frac{\partial u}{\partial y} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ & + \frac{(x - x_0)^2}{2!} \cdot \frac{\partial^2 u}{\partial x^2} \Big|_{\mathbf{x}=\mathbf{x}_0} + \frac{(y - y_0)^2}{2!} \cdot \frac{\partial^2 u}{\partial y^2} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ & + \frac{(x - x_0)(y - y_0)}{2!} \cdot \frac{\partial^2 u}{\partial x \partial y} \Big|_{\mathbf{x}=\mathbf{x}_0} + \frac{(x - x_0)(y - y_0)}{2!} \cdot \frac{\partial^2 u}{\partial y \partial x} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ & + \cdots + \frac{1}{k!} \left[\sum_{j=1}^2 (x_j - x_{j,0}) \frac{\partial}{\partial x_j} \right]^k u \Big|_{\mathbf{x}=\mathbf{x}_0} + r_k(x, y; x_0, y_0) \end{aligned}$$

where remainder term $r_k(x, y; x_0, y_0) = \frac{1}{(k + 1)!} \left[\sum_{j=1}^2 (x_j - x_{j,0}) \frac{\partial}{\partial x_j} \right]^{k+1} u \Big|_{\mathbf{x}=\xi}$

with $\xi \in \ell[\mathbf{x}_0, \mathbf{x}]$ being a point on the line segment from point \mathbf{x}_0 to point \mathbf{x} .

Theorem A.4. (Two-Variable Taylor Series Expansion – Perturbed Form)

Let $\Omega \subseteq \mathbb{R}^2$ be star-shaped domain with open ball $B((x_0, y_0); \Delta x, \Delta y) \subset \Omega$.
 Let function $u(\mathbf{x}) = u(x_1, x_2) = u(x, y) \in C^{k+1}(\Omega)$. Then:

$$\begin{aligned} u(x + \Delta x, y + \Delta y) &= u(x_0, y_0) + \frac{\Delta x}{1!} \cdot \frac{\partial u}{\partial x} \Big|_{\mathbf{x}=\mathbf{x}_0} + \frac{\Delta y}{1!} \cdot \frac{\partial u}{\partial y} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ &+ \frac{(\Delta x)^2}{2!} \cdot \frac{\partial^2 u}{\partial^2 x} \Big|_{\mathbf{x}=\mathbf{x}_0} + \frac{(\Delta y)^2}{2!} \cdot \frac{\partial^2 u}{\partial^2 y} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ &+ \frac{\Delta x \Delta y}{2!} \cdot \frac{\partial^2 u}{\partial x \partial y} \Big|_{\mathbf{x}=\mathbf{x}_0} + \frac{\Delta x \Delta y}{2!} \cdot \frac{\partial^2 u}{\partial y \partial x} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ &+ \cdots + \frac{1}{k!} \left[\sum_{j=1}^2 (\Delta x_j) \frac{\partial}{\partial x_j} \right]^k u \Big|_{\mathbf{x}=\mathbf{x}_0} + r_k(x, y; \Delta x, \Delta y) \end{aligned}$$

where remainder $r_k(x, y; \Delta x, \Delta y) = \frac{1}{(k+1)!} \left[\sum_{j=1}^2 (\Delta x_j) \frac{\partial}{\partial x_j} \right]^{k+1} u \Big|_{\mathbf{x}=\boldsymbol{\xi}}$ with
 $\boldsymbol{\xi} \in \ell[\mathbf{x}_0, \mathbf{x}_0 + \Delta \mathbf{x}]$ being a point on line segment from \mathbf{x}_0 to $\mathbf{x}_0 + \Delta \mathbf{x}$.

B Chain Rules

Recall the Single-Variable Chain Rule (1CR) from calculus [25]:

Proposition B.1. (Single-Variable Chain Rule)

Let $\Omega_1, \Omega_2 \subset \mathbb{R}$ be bounded domains with $t_0 \in \Omega_1$ and $x_0 \in \Omega_2$.
 Let functions $x(t) \in C^1(\Omega_1)$ and $u(x) \in C^1(\Omega_2)$ such that $\text{range}[x] \subseteq \text{domain}[u]$.
 Then, composite function $u \circ x : \Omega_1 \rightarrow \mathbb{R}$ is well-defined and in $C^1(\Omega_1)$ such that

$$\frac{du}{dt} \Big|_{t=t_0} = \frac{du}{dx} \Big|_{x=x_0} \cdot \frac{dx}{dt} \Big|_{t=t_0}$$

The following Two-Variable Chain Rule (2CR) [25] is key in Taylor Series and certain Runge-Kutta timestepper constructions:

Proposition B.2. (Two-Variable Chain Rule)

Let $\Omega_1 \subset \mathbb{R}$ and $\Omega_2 \subset \mathbb{R}^2$ be bounded domains with $t_0 \in \Omega_1$ and $\mathbf{x}_0 = (x_0, y_0) \in \Omega_2$.
 Let functions $\mathbf{x}(t) \in [C^1(\Omega_1)]^2$ and $u(\mathbf{x}) \in C^1(\Omega_2)$ such that $\text{range}[\mathbf{x}] \subseteq \text{domain}[u]$.
 Then, composite function $u \circ \mathbf{x} : \Omega_1 \rightarrow \mathbb{R}$ is well-defined and in $C^1(\Omega_1)$ such that

$$\frac{du}{dt} \Big|_{t=t_0} = \frac{\partial u}{\partial x} \Big|_{\mathbf{x}=\mathbf{x}_0} \cdot \frac{dx}{dt} \Big|_{t=t_0} + \frac{\partial u}{\partial y} \Big|_{\mathbf{x}=\mathbf{x}_0} \cdot \frac{dy}{dt} \Big|_{t=t_0}$$

or written more compactly in terms of a dot product of two vectors

$$\frac{du}{dt} \Big|_{t=t_0} = \nabla u \Big|_{\mathbf{x}=\mathbf{x}_0} \cdot \frac{d\mathbf{x}}{dt} \Big|_{t=t_0}$$

C Vector Algebra Identities

The following vector algebra identities [25] are essential:

Proposition C.1. (Useful Vector Algebra Identities)

Let $\Omega \subset \mathbb{R}^d$ be an open connected set with $d \in \{2, 3\}$.

Let vector fields $\mathbf{A}(\mathbf{x}), \mathbf{B}(\mathbf{x}), \mathbf{C}(\mathbf{x}), \mathbf{D}(\mathbf{x}) \in [C(\overline{\Omega})]^d$.

Let scalar $\alpha \in \mathbb{R}$. Then:

$$\begin{aligned}
 (VA1) \quad \mathbf{A} \cdot (\mathbf{B} + \mathbf{C}) &= \mathbf{A} \cdot \mathbf{B} + \mathbf{A} \cdot \mathbf{C} \\
 (VA2) \quad \mathbf{A} \times (\mathbf{B} + \mathbf{C}) &= \mathbf{A} \times \mathbf{B} + \mathbf{A} \times \mathbf{C} \\
 (VA3) \quad \alpha(\mathbf{A} \cdot \mathbf{B}) &= (\alpha\mathbf{A}) \cdot \mathbf{B} = \mathbf{A} \cdot (\alpha\mathbf{B}) \\
 (VA4) \quad \alpha(\mathbf{A} \times \mathbf{B}) &= (\alpha\mathbf{A}) \times \mathbf{B} = \mathbf{A} \times (\alpha\mathbf{B}) \\
 (VA5) \quad \mathbf{A} \cdot \mathbf{B} &= \mathbf{B} \cdot \mathbf{A} \\
 (VA6) \quad \mathbf{A} \times \mathbf{B} &= -(\mathbf{B} \times \mathbf{A}) \\
 (VA7) \quad \mathbf{A} \times (\mathbf{B} \times \mathbf{C}) &= (\mathbf{A} \cdot \mathbf{C})\mathbf{B} - (\mathbf{A} \cdot \mathbf{B})\mathbf{C} \\
 (VA8) \quad (\mathbf{A} \times \mathbf{B}) \cdot (\mathbf{C} \times \mathbf{D}) &= (\mathbf{A} \cdot \mathbf{C})(\mathbf{B} \cdot \mathbf{D}) - (\mathbf{A} \cdot \mathbf{D})(\mathbf{B} \cdot \mathbf{C}) \\
 (VA9) \quad \mathbf{A} \times (\mathbf{B} \times \mathbf{C}) &\neq (\mathbf{A} \times \mathbf{B}) \times \mathbf{C} \quad (\text{in general})
 \end{aligned}$$

D Vector Calculus Identities

The following vector calculus identities [25] will come in handy:

Proposition D.1. (Useful Vector Calculus Identities)

Let $\Omega \subset \mathbb{R}^d$ be an open connected set with $d \in \{2, 3\}$.

Let scalar field $\varphi(\mathbf{x}) \in C(\overline{\Omega}) \cap C^1(\Omega)$.

Let vector field $\mathbf{A}(\mathbf{x}) \in [C(\overline{\Omega})]^d \cap [C^2(\Omega)]^d$.

Let vector field $\mathbf{B}(\mathbf{x}) \in [C(\overline{\Omega})]^d \cap [C^1(\Omega)]^d$. Then:

$$\begin{aligned}
 (VC1) \quad \nabla \cdot (\mathbf{A} \times \mathbf{B}) &= \mathbf{B} \cdot (\overline{\nabla} \times \mathbf{A}) - \mathbf{A} \cdot (\overline{\nabla} \times \mathbf{B}) \\
 (VC2) \quad \overline{\nabla} \times (\varphi\mathbf{A}) &= \varphi(\overline{\nabla} \times \mathbf{A}) + (\overline{\nabla}\varphi) \times \mathbf{A} \\
 (VC3) \quad \overline{\nabla} \times \overline{\nabla} \times \mathbf{A} &= \overline{\nabla}(\nabla \cdot \mathbf{A}) - \overline{\nabla}^2 \mathbf{A}
 \end{aligned}$$

E Divergence Theorem

Recall the Divergence Theorem [25, 44, 74], also known as Ostrogradsky's Theorem [99], from elementary multivariable calculus:

Theorem E.1. (Divergence Theorem)

Let $\Omega \subset \mathbb{R}^d$ be a piecewise-smooth bounded domain with $d \in \{2, 3\}$.

Let \mathbf{n} be the outward unit normal vector to the boundary $\partial\Omega$.

Let vector field $\mathbf{F}(\mathbf{x}) \in [C(\overline{\Omega})]^d \cap [C^1(\Omega)]^d$.

Then:

$$\int_{\Omega} \nabla \cdot \mathbf{F} \, d\mathbf{x} = \oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \, dS$$

F Green's Identities

Recall the integration-by-parts technique from elementary single-variable calculus. Both the standard statement (IBP) is provided as well as the version used in producing finite element weak forms of 2^{nd} -order ODE's & 1D PDE's (IBP')

Theorem F.1. (Integration-by-Parts)

Let functions $u(x) \in C[a, b] \cap C^2(a, b)$ and $v(x) \in C[a, b] \cap C^1(a, b)$ where $a < b$. Then:

$$(IBP) \quad \int_{[a,b]} v u_x dx = [vu]_{x=a}^{x=b} - \int_{[a,b]} v_x u dx$$

$$(IBP') \quad \int_{[a,b]} v u_{xx} dx = [v u_x]_{x=a}^{x=b} - \int_{[a,b]} v_x u_x dx$$

Green's identities [25, 44, 74] generalize integration-by-parts (IBP') to integrals involving certain 2^{nd} -order derivatives in \mathbb{R}^2 and \mathbb{R}^3 :

Theorem F.2. (Green's Identities)

Let $\Omega \subset \mathbb{R}^d$ be a piecewise-smooth bounded domain with $d \in \{2, 3\}$.

Let \mathbf{n} be the outward unit normal vector to the boundary $\partial\Omega$.

Let scalar fields $u(\mathbf{x}) \in C(\bar{\Omega}) \cap C^2(\Omega)$ and $v(\mathbf{x}) \in C(\bar{\Omega}) \cap C^1(\Omega)$.

Let vector fields $\mathbf{A}(\mathbf{x}) \in [C(\bar{\Omega})]^d \cap [C^2(\Omega)]^d$ and $\mathbf{B}(\mathbf{x}) \in [C(\bar{\Omega})]^d \cap [C^1(\Omega)]^d$.

Then:

$$(GI1) \quad \int_{\Omega} v \nabla^2 u d\mathbf{x} = \oint_{\partial\Omega} [v \bar{\nabla} u] \cdot \mathbf{n} dS - \int_{\Omega} \bar{\nabla} v \cdot \bar{\nabla} u d\mathbf{x}$$

$$(GI2) \quad - \int_{\Omega} \mathbf{B} \cdot (\bar{\nabla} \times \bar{\nabla} \times \mathbf{A}) d\mathbf{x} = \oint_{\partial\Omega} [\mathbf{B} \times (\bar{\nabla} \times \mathbf{A})] \cdot \mathbf{n} dS \\ - \int_{\Omega} (\bar{\nabla} \times \mathbf{B}) \cdot (\bar{\nabla} \times \mathbf{A}) d\mathbf{x}$$

G Linear Algebra Lemmas

The following lemma is needed for the proof in Chapter 3 that the GaussLegendre(s) family of timesteppers are QIP:

Lemma G.1. (Quadratic Form Symmetric Matrix Lemma – QFSML)

Given arbitrary real $m \times m$ matrix A , then quadratic form $\mathbf{x}^T A \mathbf{x}$ can be represented as $\mathbf{x}^T S \mathbf{x}$ where matrix S is symmetric.

Proof. Let $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$.

Coefficient of term x_i^2 in $\mathbf{x}^T A \mathbf{x}$ is a_{ii} .

Coefficient of term $x_i x_j$ ($i \neq j$) in $\mathbf{x}^T A \mathbf{x}$ is $a_{ij} + a_{ji}$.

Set entries of S $s_{ij} = s_{ji} = (a_{ij} + a_{ji})/2$. Then S is symmetric.

\therefore Coefficient of term x_i^2 in $\mathbf{x}^T S \mathbf{x}$ is $s_{ii} = a_{ii}$.

\therefore Coefficient of term $x_i x_j$ ($i \neq j$) in $\mathbf{x}^T S \mathbf{x}$ is $s_{ij} + s_{ji} = a_{ij} + a_{ji}$. □

The next lemma is also used in the proof that the GaussLegendre(s) timestepper is QIP in Chapter 3. It is stated without proof in §IV.2 of [50]:

Lemma G.2. (Quadratic Invariant Lemma – QIL)

Note that $\dot{\mathbf{u}}$ is shorthand for the time-derivative $\frac{d}{dt}[\mathbf{u}(t)]$.

Given model vector IVP (MVIVP):
$$\begin{cases} \dot{\mathbf{u}}(t) &= \mathbf{\Psi}(t; \mathbf{u}) \\ \mathbf{u}(t_0) &= \mathbf{u}_0 \end{cases}$$

Then, quadratic $\mathbf{u}^T S \mathbf{u}$ is invariant $\iff \mathbf{u}^T S \mathbf{\Psi}(t; \mathbf{u}) = 0 \quad \forall (\mathbf{u}, t) \in \mathbb{R}^m \times [t_0, \infty)$

Proof. WLOG, assume matrix S is symmetric, courtesy of QFSML.

(\implies): Suppose $\mathbf{u}^T S \mathbf{u}$ is invariant. Then $\frac{d}{dt} [\mathbf{u}^T S \mathbf{u}] = 0 \quad \forall (\mathbf{u}, t) \in \mathbb{R}^m \times [t_0, \infty)$

$$\frac{d}{dt} [\mathbf{u}^T S \mathbf{u}] = \dot{\mathbf{u}}^T S \mathbf{u} + \mathbf{u}^T S \dot{\mathbf{u}} = \mathbf{u}^T S^T \dot{\mathbf{u}} + \mathbf{u}^T S \dot{\mathbf{u}} \stackrel{SYM}{=} \mathbf{u}^T S \dot{\mathbf{u}} + \mathbf{u}^T S \dot{\mathbf{u}} = 2\mathbf{u}^T S \dot{\mathbf{u}}$$

$$\therefore 0 = \frac{d}{dt} [\mathbf{u}^T S \mathbf{u}] = 2\mathbf{u}^T S \dot{\mathbf{u}} \implies \mathbf{u}^T S \dot{\mathbf{u}} = 0 \stackrel{MVIVP}{\implies} \mathbf{u}^T S \mathbf{\Psi}(t; \mathbf{u}) = 0 \quad \forall (\mathbf{u}, t)$$

(\impliedby): Suppose $\mathbf{u}^T S \mathbf{\Psi}(t; \mathbf{u}) = 0 \quad \forall (\mathbf{u}, t) \in \mathbb{R}^m \times [t_0, \infty)$. Then:

$$0 = \mathbf{u}^T S \mathbf{\Psi}(t; \mathbf{u}) \stackrel{MVIVP}{=} \mathbf{u}^T S \dot{\mathbf{u}} \stackrel{SYM}{=} \frac{1}{2} \dot{\mathbf{u}}^T S \mathbf{u} + \frac{1}{2} \mathbf{u}^T S \dot{\mathbf{u}} = \frac{1}{2} \frac{d}{dt} [\mathbf{u}^T S \mathbf{u}]$$

$$\therefore \frac{1}{2} \frac{d}{dt} [\mathbf{u}^T S \mathbf{u}] = 0 \implies \frac{d}{dt} [\mathbf{u}^T S \mathbf{u}] = 0 \implies \mathbf{u}^T S \mathbf{u} \text{ is invariant.} \quad \square$$

H Tensor Notation

In order to determine the pre-timestepper weak form of the micromagnetics problem, we will employ tensor notation [46] which is sometimes called Einstein index notation as it was first introduced in [38]. Since there will be no explicit appeals to manifolds and differential geometry, all tensor notation indices used will be subscripts, never superscripts. What follows are the essentials necessary for our purposes.

Definition H.1. (Kronecker Symbols)

The Kronecker symbol $\delta_{ij} : \mathbb{N}^2 \rightarrow \{0, 1\}$ is defined as:

$$(TN\delta) \quad \delta_{ij} := \begin{cases} 1 & , \text{ if } i = j \\ 0 & , \text{ if } i \neq j \end{cases}$$

Definition H.2. (Permutation Symbols)

The permutation symbol $e_{ijk} : \mathbb{N}^3 \rightarrow \{-1, 0, +1\}$ is defined as:

$$(TNe) \quad e_{ijk} := \begin{cases} +1 & , \text{ if even permutation} \\ -1 & , \text{ if odd permutation} \\ 0 & , \text{ if an index is repeated} \end{cases}$$

Since we're working exclusively in Cartesian coordinates, the volume element is always unity which leads to a trivial relationship between permutation symbols and the desired Levi-Civita symbols [46]:

Definition H.3. (Levi-Civita Symbols)

In Cartesian coordinates, the Levi-Civita symbol ϵ_{ijk} is identical to the corresponding permutation symbol e_{ijk} .

Occasionally, the Levi-Civita symbol needs to be reindexed to better align with an involved vector calculus operation. Reindexing is achieved by swapping two of the three indices [46]:

Proposition H.1. (Levi-Civita Symbol Index Swap)

Swapping two of the indices of a Levi-Civita symbol changes the sign of the result:

$$(TN\epsilon) \quad \epsilon_{ijk} = -\epsilon_{kji}, \quad \epsilon_{ijk} = -\epsilon_{jik}, \quad \epsilon_{ijk} = -\epsilon_{ikj}$$

Armed with the Kronecker and Levi-Civita symbols, we can now state their fundamental relation [46]:

Theorem H.1. (Kronecker-Levi-Civita Identity)

$$(TN\delta\epsilon) \quad \epsilon_{ijk}\epsilon_{ilm} = \delta_{il}\delta_{km} - \delta_{jm}\delta_{kl}$$

Note that tensor indices cannot be used without restraint. The following fundamental rules of tensor index utility must always be adhered to [46]:

Theorem H.2. (Tensor Notation Rule for Repeated Indices)

Given a term in an expression written in tensor notation, any repeated index must occur exactly twice. Otherwise, the term is ill-formed and may be remedied by introducing new indices. For example:

$$\begin{array}{lll}
 \delta_{ij}A_jB_j & \text{can be rewritten as} & \delta_{ij}A_jB_k \\
 \epsilon_{ijk}A_iB_iC_i & \text{can be rewritten as} & \epsilon_{ijk}A_iB_jC_k \\
 \delta_{ij}A_{ij}\delta_{ij}B_{ij} & \text{can be rewritten as} & \delta_{ij}A_{ij}\delta_{km}B_{km} \\
 \epsilon_{ijk}A_iB_jC_k\epsilon_{ijk}D_iE_jF_k & \text{can be rewritten as} & \epsilon_{ijk}A_iB_jC_k\epsilon_{mnp}D_mE_nF_p
 \end{array}$$

Theorem H.3. (Tensor Notation Rule for Live Indices)

Given an equation written in tensor notation, any live index must occur on both sides of the equation. Otherwise, the equation is ill-formed. For example:

$$D_m = \delta_{ij}A_iB_jC_k \quad \text{can be rewritten as} \quad \begin{array}{l} D_k = \delta_{ij}A_iB_jC_k \text{ or} \\ D_m = \delta_{ij}A_iB_jC_m \end{array}$$

Table H.2: Tensor Form of Vector Algebra & Vector Calculus Operations

OPERATION	NON-TENSOR FORM	TENSOR FORM
Standard Basis	$\hat{\mathbf{e}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \hat{\mathbf{e}}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\hat{\mathbf{e}}_i$ (TN0)
Dot Product	$\mathbf{u} \cdot \mathbf{v} := u_1v_1 + u_2v_2$	$\mathbf{u} \cdot \mathbf{v} = u_i v_i$ (TN1)
Cross Product	$\mathbf{u} \times \mathbf{v} := \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix}$	$\mathbf{u} \times \mathbf{v} = \epsilon_{ijk}u_i v_j \hat{\mathbf{e}}_k$ (TN2)
Scalar Triple Product	$\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w})$	$\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w}) = \epsilon_{ijk}u_i v_j w_k$ (TN3)
Frobenius Product	$A : B := a_{11}b_{11} + a_{12}b_{12} + a_{21}b_{21} + a_{22}b_{22}$	$A : B = a_{ij}b_{ij}$ (TN4)
Divergence	$\nabla \cdot \mathbf{F}(\mathbf{x}) := \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2}$	$\nabla \cdot \mathbf{F}(\mathbf{x}) = \partial_k F_k$ (TN5)
Curl	$\bar{\nabla} \times \mathbf{F}(\mathbf{x}) := \begin{bmatrix} \frac{\partial F_3}{\partial x_2} - \frac{\partial F_2}{\partial x_3} \\ \frac{\partial F_1}{\partial x_3} - \frac{\partial F_3}{\partial x_1} \\ \frac{\partial F_2}{\partial x_1} - \frac{\partial F_1}{\partial x_2} \end{bmatrix}$	$\bar{\nabla} \times \mathbf{F}(\mathbf{x}) = \epsilon_{ijk}\partial_i F_j \hat{\mathbf{e}}_k$ (TN6)
Vector Gradient	$\bar{\nabla} f(\mathbf{x}) := \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)$	$\bar{\nabla} f(\mathbf{x}) = \partial_i f \hat{\mathbf{e}}_i^T$ (TN7)
Tensor Gradient	$\bar{\bar{\nabla}} \mathbf{F}(\mathbf{x}) := (\bar{\nabla} F_1, \bar{\nabla} F_2)^T$	$\bar{\bar{\nabla}} \mathbf{F}(\mathbf{x}) = \partial_k F_j \hat{\mathbf{e}}_j \hat{\mathbf{e}}_k^T$ (TN7)
Scalar Laplacian	$\nabla^2 f(\mathbf{x}) := \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2}$	$\nabla^2 f(\mathbf{x}) = \partial_i \partial_i f$ (TN8)
Vector Laplacian	$\bar{\nabla}^2 \mathbf{F}(\mathbf{x}) := (\nabla^2 F_1, \nabla^2 F_2)^T$	$\bar{\nabla}^2 \mathbf{F}(\mathbf{x}) = \partial_j \partial_j F_k \hat{\mathbf{e}}_k$ (TN8)

I Normed Linear Spaces

Normed linear spaces [78, 105, 117] provide a fundamental means of measuring the “size” of various mathematical objects:

Definition I.1. (Linear Space)

A set V is a linear space (or vector space) if vector addition and scalar multiplication are well-defined for all elements in V .

Definition I.2. (Norm)

Let V be a linear space. Let vectors $\mathbf{u}, \mathbf{v} \in V$ and scalar $\alpha \in \mathbb{R}$.
A norm on V is a mapping $\|\cdot\| : V \rightarrow \mathbb{R}$ satisfying the axioms:

$$\begin{aligned} \|\mathbf{u}\| &\geq 0 && \text{(Non-negativity of norm)} \\ \|\mathbf{u}\| = 0 &\iff \mathbf{u} = \vec{\mathbf{0}} && \text{(Only zero vector has zero norm)} \\ \|\alpha\mathbf{u}\| &= \alpha\|\mathbf{u}\| && \text{(Scalar multiple is factorable)} \\ \|\mathbf{u} + \mathbf{v}\| &\leq \|\mathbf{u}\| + \|\mathbf{v}\| && \text{(Triangle Inequality for norm)} \end{aligned}$$

Definition I.3. (Normed Linear Space)

A linear space V with a norm $\|\cdot\|$ is called an normed linear space.
A compact notation for a normed linear space is $(V, \|\cdot\|)$.

J Inner Product Spaces

Inner product spaces [75, 78, 105, 117] are the essential makeup of the Hilbert spaces encountered in Chapters 4,5,6 and 7:

Definition J.1. (Inner Product)

Let V be a linear space. Let vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and scalar $\alpha \in \mathbb{R}$.
An inner product on V is a mapping $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ satisfying the axioms:

$$\begin{aligned} \langle \mathbf{u}, \mathbf{u} \rangle &\geq 0 && \text{(Non-negativity of self-inner product)} \\ \langle \mathbf{u}, \mathbf{u} \rangle = 0 &\iff \mathbf{u} = \vec{\mathbf{0}} && \text{(Only zero vector has zero self-inner product)} \\ \langle \mathbf{u}, \mathbf{v} \rangle &= \langle \mathbf{v}, \mathbf{u} \rangle && \text{(Commutativity of inner product)} \\ \langle \alpha\mathbf{u}, \mathbf{v} \rangle &= \alpha\langle \mathbf{u}, \mathbf{v} \rangle && \text{(Scalar multiple is factorable)} \\ \langle \mathbf{u}, \mathbf{v} + \mathbf{w} \rangle &= \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle && \text{(Distributivity of vector addition)} \end{aligned}$$

Definition J.2. (Inner Product Space)

A linear space V with an inner product $\langle \cdot, \cdot \rangle$ is called an inner product space.
A compact notation for an inner product space is $(V, \langle \cdot, \cdot \rangle)$.

K Hilbert Spaces

Hilbert spaces [78, 105] are core to finite element methods. Loosely speaking, Hilbert spaces are inner product spaces with no “gaps”, which is formally called completeness of the space:

Definition K.1. (Hilbert Space)

A Hilbert space is a complete inner product space.

The Hilbert spaces used here from Chapter 4 to Chapter 7 [16, 13, 80, 44, 62, 74, 90] are function spaces defined in terms of the special Hilbert space $L^2(\Omega)$.

Proposition K.1. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with $d \in \{1, 2, 3\}$.

Then the function space

$$L^2(\Omega) := \left\{ u : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} u^2 \, d\mathbf{x} < \infty \right\}$$

is a Hilbert space with associated inner product

$$\langle f(\mathbf{x}), g(\mathbf{x}) \rangle := \int_{\Omega} fg \, d\mathbf{x}$$

where the integral is understood to be with respect to Lebesgue measure [78, 105].

For $L^2(\Omega)$, the lack of “gaps” is formally expressed as follows [78, 105]:

Corollary K.1. (Corollary to Proposition K.1)

Let (u_k) be a sequence of functions in $L^2(\Omega)$ such that the infinite series $\sum_{k=1}^{\infty} u_k$ absolutely converges, meaning $\sum_{k=1}^{\infty} \left(\int_{\Omega} u_k^2 \, d\mathbf{x} \right)^{1/2} < \infty$. Then, the infinite series $\sum_{k=1}^{\infty} u_k$ converges to a function u in $L^2(\Omega)$.

L Inequalities

The following inequalities [44, 74, 90] stem from functional analysis [105], a branch of mathematics with its building blocks from linear algebra [75, 117], real analysis [78] and point-set topology [1, 27]. These inequalities are key to many proofs involving error estimates:

Theorem L.1. (Triangle Inequality)

Let $(V, \|\cdot\|)$ be a normed linear space. Then:

$$\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\| \quad \forall \mathbf{u}, \mathbf{v} \in V$$

Theorem L.2. (Cauchy-Schwarz Inequality)

Let $(V, \langle \cdot, \cdot \rangle)$ be an inner product space. Then:

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \langle \mathbf{u}, \mathbf{u} \rangle^{1/2} \langle \mathbf{v}, \mathbf{v} \rangle^{1/2} \quad \forall \mathbf{u}, \mathbf{v} \in V$$

M B-Stability

Two new stability notions are needed for one of the barriers in section 3.12 [49]:

Definition M.1. (B-Stability)

A Runge-Kutta timestepper is B-stable if when applied to the model IVP (MIVP), the contractivity condition

$$\langle \Psi(t, u) - \Psi(t, v), u - v \rangle \leq 0 \quad (\text{M.1})$$

implies for all timestep sizes Δt

$$\|u^{(1)} - \hat{u}^{(1)}\| \leq \|u_0 - \hat{u}_0\| \quad (\text{M.2})$$

where $u^{(1)}, \hat{u}^{(1)}$ are the computed solution values after a single timestep starting with initial values u_0, \hat{u}_0 , respectively.

Definition M.2. (Algebraic Stability)

Given an arbitrary s -stage Runge-Kutta timestepper $\text{RK}(s)$, define matrix M as

$$M := [m_{ij}]_{s \times s} \text{ such that } m_{ij} := \gamma_i \beta_{ij} - \gamma_j \beta_{ji} - \gamma_i \gamma_j \quad (\text{M.3})$$

$\text{RK}(s)$ is algebraically stable if $\gamma_1, \dots, \gamma_s \geq 0$ and M is non-negative definite.

Definition M.3. (Non-confluent Runge-Kutta Timesteppers)

A Runge-Kutta timestepper is non-confluent if its α_j 's are all distinct.

For many timesteppers, B-stability and algebraic stability are synonymous [49]:

Proposition M.1. (Algebraic Stability & B-Stability Equivalence)

Algebraic stability and B-stability are equivalent for non-confluent or S -irreducible Runge-Kutta timesteppers. See [49] for the definition of S -irreducibility.

Some IRK timesteppers built from collocation are B-stable but others are not [49]:

Theorem M.1. (Collocation & B-stability)

GaussLegendre(s), RadauIA(s), RadauIIA(s) and LobattoIIC(s) are all B-stable. Lobatto IIIA(s) and Lobatto IIIB(s) are not B-stable.

B-stability is a measure of non-linear stability such that B-stability implies A-stability [49]. However, B-stability is different from L-stability – as an example, Scherer's DIRK(2) timestepper [109] in Table 2.9 is L-stable but not B-stable.

For this dissertation, algebraic stability and B-stability are only used in the statement of one of Hairer's barriers for DIRK timesteppers in section 3.12. Elsewhere, we will never mention algebraic stability or B-stability again as L-stability is a sufficiently strong notion of stability for our needs.