

SYMBOLIC COMPUTATION OF CONSERVATION LAWS OF
NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS
USING HOMOTOPY OPERATORS

by
Loren Douglas Poole

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree Doctor of Philosophy (Mathematical and Computer Sciences)

Golden, Colorado

Date _____

Signed: _____
Loren Douglas Poole

Signed: _____
Dr. Willy Hereman
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Dinesh P. Mehta
Professor and Head
Department of Mathematical and Computer Sciences

ABSTRACT

A mathematical method is developed for the purpose of constructing conservation laws for nonlinear partial differential equations (PDEs) in multiple space dimensions. The method is developed in the language of calculus, the calculus of variations and linear algebra, so that it is accessible to researchers in different fields. The method is algorithmic and has been implemented in the syntax of *Mathematica*, a major and commonly used computer algebra system. The software package, `ConservationLawsMD.m`, symbolically computes conservation laws for polynomial PDEs that can be written as nonlinear evolution equations.

With `ConservationLawsMD.m`, conservation laws are computed for many PDEs from mathematical physics, fluid dynamics, and soliton theory. Test cases are formed using conservation laws for the Zakharov-Kuznetsov equation, the Kadomtsev-Petviashvili equation and other well-known PDEs. Previously unknown conservation laws are given for several PDEs, including the Manakov-Santini system and the two-dimensional Gardner equation.

A second software package, `HomotopyIntegrator.m`, consists of code for the homotopy operator in `ConservationLawsMD.m`. The homotopy operator integrates one-dimensional exact expressions involving unspecified functions, or inverts a divergence on two- or three-dimensional exact expressions. The one-dimensional homotopy code is designed to supplement *Mathematica's* `Integrate` function. Since *Mathematica* does not have a function to invert divergences, the two- and three-dimensional homotopy codes provide a new and versatile tool for vector calculus.

When computing conservation laws, verification of their independence is of key importance. A third software package, `IndependenceTest.m` implements a comprehensive method for testing if densities are independent. Although this code is also part of `ConservationLawsMD.m`, the package `IndependenceTest.m` can be used as a stand-alone tool for researchers working on conservation laws.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF FIGURES	ix
LIST OF TABLES	xi
ACKNOWLEDGEMENTS	xii
CHAPTER 1 INTRODUCTION	1
1.1 Previous Work	3
1.2 New Contributions	5
CHAPTER 2 NOTATION AND DEFINITIONS	11
CHAPTER 3 EXAMPLES OF CONSERVATION LAWS OF NONLINEAR PDES IN (2+1)- AND (3+1)-DIMENSIONS	19
3.1 The Zakharov-Kuznetsov Equation	21
3.2 The (3+1)-Dimensional Equation for Non-stationary Transonic Gas Flow	24
3.3 The (2+1)-Dimensional Gardner Equation	28
CHAPTER 4 TOOLS FOR THE COMPUTATION OF CONSERVATION LAWS: THE ZEROTH-EULER OPERATOR	31
4.1 General Definitions	31
4.2 The Euler-Lagrange Equations	33
4.3 A Proof for the Exactness Theorem	38
CHAPTER 5 INDEPENDENCE OF CONSERVED DENSITIES	43
5.1 Terms that are Divergences or Divergence-Equivalent	44
5.2 Trivial Conservation Laws	47
5.3 Equivalent Densities	48
5.4 Linear Combinations	49

CHAPTER 6	TOOLS FOR THE COMPUTATION OF CONSERVATION LAWS:	
	THE HOMOTOPY OPERATOR	55
6.1	The Original Homotopy Operator	57
6.2	Where is the Homotopy in the Homotopy Operator?	59
6.3	Reformulation of the Homotopy Operator Integrand	60
6.3.1	The Homotopy Operator for One Independent Variable	61
6.3.2	The Homotopy Operator for Two Independent Variables	70
6.3.3	The Homotopy Operator for Three Independent Variables	76
6.4	Removing Divergence-Free Terms	77
6.5	Limitations of the Homotopy Operator	80
CHAPTER 7	CONSTRUCTION OF CONSERVATION LAWS FOR NONLIN-	
	EAR PDES	83
7.1	Conservation Laws for the Zakharov-Kuznetsov Equation	83
7.1.1	Establishing a Scaling Symmetry	84
7.1.2	Construction of a Candidate Density	87
7.1.3	Determination of the Actual Density	90
7.1.4	Calculation of the Flux	93
7.2	Conservation Laws for the Non-stationary Transonic Gas Flow Equation	94
7.2.1	Scaling Symmetry for the NTGF Equation	95
7.2.2	Construction of Candidate Densities	96
7.2.3	Determination of the Actual Densities	98
7.2.4	Calculation of the Flux and Inversion of the Transformation	100
7.3	Conservation Laws for the Gardner Equation	101
7.3.1	Establishing a Scaling Symmetry	101
7.3.2	Determination of the Density	102
7.3.3	Calculation of the Flux and Inversion of the Transformation	104
CHAPTER 8	CONSERVATION LAWS FOR PDES IN MULTI-DIMENSIONS	107
8.1	Kadomtsev-Petviashvili Equations	108
8.1.1	The Kadomtsev-Petviashvili Equation	108
8.1.2	The Potential Kadomtsev-Petviashvili Equation	111

8.2	The Fluid Dynamics Equations	114
8.2.1	The Shallow Water Equations	114
8.2.2	The Shallow Water Magnetohydrodynamics Equations	116
8.2.3	The Thermal Shallow Water Magnetohydrodynamics Equations	118
8.3	The Khokhlov-Zabolotskaya Equation	120
8.3.1	The (2+1)-Dimensional Khokhlov-Zabolotskaya Equation	120
8.3.2	The (3+1)-Dimensional Khokhlov-Zabolotskaya Equation	122
8.4	The Coupled Korteweg-de Vries Equations	125
8.5	The Manakov-Santini System	127
8.6	The Camassa-Holm Equation	128
8.7	Navier's Equation	130
CHAPTER 9 SOFTWARE PACKAGES		135
9.1	The Software Package ConservationLawsMD.m	135
9.1.1	Running a Test Example	136
9.1.2	The Data File	140
9.1.3	Testing a Conservation Law	143
9.1.3.1	Testing Densities of Evolution PDEs	144
9.1.3.2	Testing Densities of Non-evolution PDEs	145
9.2	The Software Package HomotopyIntegrator.m	149
9.3	The Software Package IndependenceTest.m	154
CHAPTER 10 CONCLUSIONS		159
REFERENCES CITED		163
APPENDIX ADDITIONAL PROOFS FOR THE HOMOTOPY OPERATOR		169
A.1	Lemmas for the Proof of Theorem 6.4	169
A.2	Extended Proof of Theorem 6.5	171
A.3	Proof of Theorem 6.6	181

LIST OF FIGURES

9.1	The menu that appears when ConservationLawsMD[] is initiated. . .	136
9.2	Page 1 of the menu listing the first ten multi-dimensional PDEs.	137
9.3	A list of weights for the (2+1)-dimensional Zakharov-Kuznetsov equation.	138
9.4	Information provided by ConservationLawsMD.m when the program terminates. A conservation law for the (2+1)-dimensional Zakharov-Kuznetsov equation is shown.	139
9.5	The contents of the data file, d_zk2d.m , for the (2+1)-dimensional Zakharov-Kuznetsov equation.	141
9.6	The result for testing the density of (9.3) for the Riemann System (9.1). .	146
9.7	The result from ConservationLawsMD.m verifying the density for the Kadomtsev-Petvishvili evolution equations (8.2).	148
9.8	A sample data file of densities for the potential Kadomtsev-Petviashvili equation. Note that three densities in this file are independent and one is not.	155
9.9	The evaluation of a density for the potential Kadomtsev-Petviashvili equation based on previously established independent densities.	157

LIST OF TABLES

2.1	The order M_i^j is the order of u^j with respect to x^i	14
3.1	List of conservation laws for the generalized Zakharov-Kuznetsov equation (3.14) for specific values of n	24
6.1	Run time comparisons of one-dimensional homotopy integrands.	69
6.2	Run time comparisons of two-dimensional homotopy integrands.	74
7.1	Candidate densities for the non-stationary transonic gas flow equation.	98
8.1	A list of densities for the potential Kadomtsev-Petviashvili equation (8.15) with their generalizations.	112
8.2	Table of densities calculated by <code>ConservationLawsMD.m</code> for the (3+1)- dimensional Khokhlov-Zabolotskaya (8.74) equation.	123
8.3	Table of densities at rank 5 calculated by <code>ConservationLawsMD.m</code> for the (3+1)-dimensional Navier equation (8.122).	132

ACKNOWLEDGEMENTS

Many thanks to my advisor, Dr. Willy Hereman, for the numerous hours he has spent guiding me through this project. With his insight and considerable knowledge, he was constantly able to suggest improvements and get me through tough spots. His “what if” questions often led to new discoveries and better methods, and he kept me on my toes by being constantly able to pick the one example that would create difficulties for my programs.

I also wish to thank the Department of Mathematical and Computer Science at CSM for their support and for allowing me to teach classes. The opportunity to work in the classroom has provided valuable experience for my academic career.

An undergraduate student, John-Bosco Tran, was most helpful by running and testing the conservation laws program and checking to see that the conservation laws presented in this dissertation were typeset correctly.

Thanks also goes to the National Science Foundation for their support. My research was supported in part under grant no. CCF-083783.

As well, I would like to thank Dr. Thomas Wolf for allowing me access to his CONLAW programs, as well as Dr. Jan Sanders and Dr. Bhimsen Shivamoggi for responding with confirmation of results from my programs.

For Charles and Nina

CHAPTER 1

INTRODUCTION

To a person familiar with physics, conservation of mass, momentum, energy, electric charge, and other quantities are well known. A conserved quantity moves or flows from one place to another; portions of it do not magically appear or disappear. A conservation law is a mathematical equation that describes how the quantity is affected by flow. In a dynamics problem, the amount of material flowing out of a closed region, or the flux of the material, is equal to the change of the density times the change in volume of the material contained in the closed region. Irrespective of their physical meaning, the dynamics terms “conserved density” and its associated “flux” will be used to represent the two dependent pieces of a conservation law.

Many partial differential equations (PDEs) or systems thereof that model physical situations exhibit conservation laws such as the conservation of mass, momentum, or energy. However, special nonlinear PDEs may also have a multitude of conservation laws without a physical meaning. Conservation laws are related to the symmetries of a PDE, a connection shown in a remarkable theorem proved by Emmy Noether in 1918 [54]. Noether proved that every conservation law for an invariant Euler-Lagrange system corresponds to a Lie symmetry for that system. That is, if a variational principle can be determined for a symmetry of a PDE, a corresponding conservation law can be found. Her theorem has been used in a wide variety of applications.

A serious study of conservation laws of nonlinear PDEs began in the 1960s with the investigation of solitary wave and soliton solutions for the Korteweg-de Vries (KdV) equation. In the 1870s, both Joseph Boussinesq and Lord Rayleigh independently proposed a nonlinear PDE to model the motion of a solitary wave moving in a channel [28]. In 1895, Diederik Korteweg and Gustav de Vries derived the same equation [46], which now carries their names. The KdV equation drew little interest until the 1960s. In 1965, Martin Kruskal and Norman Zabusky were investigating wave motion in a one-dimensional anharmonic lattice of equal masses [81]. They were trying to solve the FPU problem, posed by Enrico Fermi, John Pasta, and Stanislaw Ulam, a problem investi-

gating how energy is distributed among the many possible oscillations in a nonlinear lattice [24]. Zabusky and Kruskal found that interacting solitary waves would retain their shapes after collisions, that is, obey the superposition principle. This occurred despite the fact that the waves were highly nonlinear. Zabusky and Kruskal named these solitary waves “solitons”, and discovered that in the continuum limit, the KdV equation modeled the wave behavior and energy distribution they were observing.

At this point, attention turned toward the KdV equation and methods for finding soliton solutions. Since conservation laws provide information about properties of a PDE, part of the investigation focused on finding conserved densities for the KdV equation. The first two conservation laws, representing the conservation of mass and conservation of energy, had been known for some time. While investigating nonlinear dispersive waves in 1965, Gerald Whitham discovered the third conservation law [74]. In the same year, Zabusky and Kruskal computed the fourth and fifth conservation laws, but failed to find a sixth due to an error in their calculations. The following year, Robert Miura was able to find a seventh conservation law, then corrected the error in the computations and found the sixth. Miura went on to discover three more conservation laws, then proved [52] that an infinite sequence of conservation laws exists for the KdV equation.

The work by Kruskal, Zabusky and Miura, along with John Greene and Clifford Gardner led to the discovery of a one-parameter family of Bäcklund transformations between the solutions of the KdV and modified KdV equations, the discovery of the Lax pair for the KdV equation, and the Inverse Scattering Transform (IST) for linearizing the KdV equation [26]. The IST [1] provides a method for finding solitons for a PDE of all orders. When this can be done, the PDE is said to be completely integrable. While applying the IST to a PDE can be an onerous task, the existence of an infinite set of conservation laws indicates that a PDE is completely integrable, and that the application of the IST is likely to completely solve the PDE. However, the existence of infinitely many conservation laws is not a prerequisite. Indeed, PDEs that have few conservation laws have been found to be integrable. The best known example is the Burgers equation which can be transformed into the heat equation using the Cole-Hopf transformation [18, 37]. Integrable equations or systems are essential to theoretical and mathematical physicists, whose goal is to describe and understand the behavior of

classical or quantum systems [82].

Why compute conservation laws? First of all, one may wish to know which physical quantities are conserved in the system described by the PDE. Investigation of conservation laws can lead to new discoveries in particular, new methods for finding solutions for PDEs. For example, in the KdV case the investigation of conservation laws led to the discovery of the IST, a key method to solve nonlinear PDEs. Conservation laws aid the study of qualitative properties of PDEs, such as bi-Hamiltonian structures and recursion operators [9]. If constitutive properties have been added to close a system, one may wish to verify if conserved quantities have remained intact [34]. Conserved densities also aid in the design of numerical solvers for PDEs [63].

The approach used in this dissertation to compute conservation laws requires tools from calculus, linear algebra and the calculus of variations. It implements and extends the method described by Hereman et al. in [35] and by Hereman in [33]. A candidate density is constructed by forming a linear combination with undetermined coefficients of differential terms that are invariant under the scaling symmetry of the PDE. The undetermined coefficients can be calculated by applying the variational derivative to the time derivative of the candidate density after the density has been rewritten in terms of the space variables using the PDE. Once the density is known, the flux can be computed by applying a homotopy operator to invert a divergence. Each phase of this process will be explained in detail, using examples to clarify the explanations.

1.1 Previous Work

There are many methods for computing conservation laws [35, 53]; all require a significant amount of computation, which is best done with a computer algebra system (CAS) such as *Mathematica*, *Maple*, or *REDUCE*. A summary of nine methods for computing conservation laws is given by Naz [53], of which some have been coded in CAS. The method used in this dissertation is not mentioned by Naz.

Two previously developed packages written in *Mathematica* use some of the techniques that will be presented in this dissertation. The package `CONDENS.M`, written by Göktaş and Hereman [28, 29] calculates conservation laws for (1+1)-dimensional PDEs, that is, PDEs that have one independent space variable and one independent time vari-

able. `TransPDEDensityFlux.m`, a package written by Adams and Hereman [2, 3], extends the code in `CONDENS.M` to work on (1+1)-dimensional PDEs with transcendental nonlinearities. Neither program will work on a PDE with multiple space dimensions, and both accept only evolution equations. The methods used in `CONDENS.M` have also been extended to compute conservation laws for (1+1)-dimensional differential difference equations [34].

Two other fully developed programs exist to compute conservation laws, one by Wolf [76] in *REDUCE* and one by Cheviakov [15] in *Maple*. The program `GeM` by Cheviakov [15, 16], computes a set of multipliers on the PDE. The multipliers lead to a determining system of differential equations, based on the symmetries of the PDE, that yield a conservation law [4, 5]. The method does not require the computation of a variational principle as required by Noether's Theorem, but does require that a PDE be written in Cauchy-Kovalevskaya form, that is, that the PDE be solved in terms of its highest order derivative(s). Since a conservation law is computed as a divergence, a homotopy operator is used to invert the divergence [16] to get the density and the flux.

The packages `CONLAW1` through `CONLAW4` written by Wolf in *REDUCE* [75, 78] find conservation laws by solving an over-determined system of differential equations. All four versions use the program `CRACK`, a sophisticated solver for systems of differential equations. `CONLAW1` tries to solve the continuity equation directly, while `CONLAW3` tries to find the conservation law and its characteristic [56] concurrently. A characteristic is an identifying expression that has the same properties as the conservation law. Both `CONLAW2` and `CONLAW4` try to find the characteristic function before computing the conserved density. `CONLAW2` uses substitutions based on the PDE, whereas `CONLAW4` does not. While each of these programs can compute conserved densities with explicit dependence on the independent variables, arbitrary functions, and transcendental nonlinearities, it may be necessary to use a combination of the programs to find all conservation laws for a PDE. The complexity of the problem increases very fast as the order for the characteristic function increases, and the systems of differential equations generated become very large and difficult to solve.

Another method applies Noether's Theorem to get conservation laws from variational symmetries [56]. This method is used in the *Maple* package `DE_APPLS`, a part

of the `Vessiot` suite written by Anderson for computations in the jet space [6]. This program requires continual input from the user while working with Noether’s theorem and requires that the symmetries of the PDE have variational principles. The `Vessiot` suite is now part of a differential geometry package in *Maple*, introduced in version 11. The package was designed by Anderson and Cheb-Terrab [7] and contains the functions necessary to compute conservation laws. However, all computations are done in the language of differential forms.

The homotopy operator is used on expressions containing unspecified functions. The homotopy operator can integrate (by parts) a one-dimensional expression or invert the divergence of a multi-dimensional expression. The homotopy operator originates in differential geometry in a proof showing the exactness of the variational complex [56]. It has since been “translated” from the language of differential geometry into the language of standard calculus by Hereman et al. [35]. The one-dimensional version given in [35] was implemented in `TransPDEDensityFlux.m` [2] to resolve issues with *Mathematica*’s `Integrate` function. The one-dimensional version has also been implemented in *Maple* by Deconinck and Nivala [21] and contains a feature for separating an expression into an integrable part and a nonintegrable part.

1.2 New Contributions

A significant part of the research for this dissertation is the development of three software packages. All three software packages are written in *Mathematica* syntax and do computations symbolically. The leading package, `ConservationLawsMD.m`, contains all of the algorithms needed to compute conservation laws. Two additional packages, `HomotopyIntegrator.m` and `IndependenceTest.m` are spin offs of `ConservationLawsMD.m`. These are stand-alone packages that can be used in different applications involving conservation laws, and beyond. The algorithms in all three packages follow the procedures and algorithms for computing conservation laws that will be described in this dissertation.

In contrast to the programs discussed previously, `ConservationLawsMD.m` has several advantages. The program can run on a desktop computer and works on any system that supports *Mathematica*. The program does not generate systems of differ-

ential equations, nor does it need to compute an Euler-Lagrange system as required by Noether’s Theorem. The program can be set to run automatically, or the user can interact with the program by providing information he or she may have already computed. The program runs fast, and reports the conservation laws in notation that is easy to read. The program also automatically verifies the independence of the densities it computes. In all test cases, the program was able to compute a complete set of conservation laws, identical to conservation laws given in literature. Several examples of PDEs are provided in the package for a user to test and explore.

The package `ConservationLawsMD.m` uses the same general principle based on the scaling symmetry of the PDE as `CONDENS.M`, and includes some of the same features. However, many new algorithms had to be developed to extend the code to (2+1)- and (3+1)-dimensional PDEs. In general, computing conservation laws in (2+1)- or (3+1)-dimensions is considerably more complex than computing conservation laws for (1+1)-dimensional PDEs. As `ConservationLawsMD.m` was developed, a number of problems not encountered in `CONDENS.M` or `TransPDEDensityFlux.m` had to be overcome. This led to the design of several features not found in the older programs. These include

- an algorithm that converts a class of non-evolution equations into evolution form.
- a new algorithm for constructing the candidate density, adapted from the idea put forth in [35].
- a method for constructing candidate densities with explicit independent variables multiplied to dependent variables.
- a method for testing densities with rational and transcendental terms, as well as for testing densities with arbitrary functions.
- a method which involves a homotopy operator for finding the flux once the density is known. In higher dimensions, calculating the flux requires inverting a divergence, whereas in one dimension the flux can be found using integration by parts. CAS cannot invert divergences, and in some cases, fail to integrate one-dimensional expressions involving arbitrary functions that require integration by parts.
- a method for removing “curl” terms from a vector that is an inverted divergence.

- an algorithm for checking for trivial conservation laws or conservations laws that are either equivalent to one another, or a combination of lower order conservation laws.

The set of (1+1)-dimensional examples of PDEs in `CONDENS.M` consists almost entirely of evolution equations. Very few (2+1)- and (3+1)-dimensional PDEs are evolution equations, so an algorithm was developed to “transform” equations into evolution equations. The algorithm used for generating a candidate density in `CONDENS.M` does not produce the most desirable candidate when more than one independent space variable is involved. A new method, which uses the variational derivative and linear algebra, works efficiently and produces the most desirable candidate density by removing unnecessary terms.

The package `ConservationLawsMD.m` can only compute polynomial conservation laws. A polynomial conservation law is polynomial in the dependent variables, however, terms may be multiplied by polynomial functions of independent variables. Most (1+1)-dimensional PDEs have polynomial conservation laws where terms are not multiplied by independent variables. The (1+1)-dimensional version, `CONDENS.M`, does not attempt to compute terms multiplied by independent variables. The procedure used in `CONDENS.M` produced very few conservation laws for multi-dimensional PDEs. Yet some of these PDEs were known to be completely integrable, suggesting that an infinite variety of conservation laws may exist. Conservation laws for higher dimensional PDEs found in literature exhibited explicit time and space variables in both the density and flux, so the algorithm for generating a candidate density was modified to generate terms with explicit time and space variables. Once these modifications were made, `ConservationLawsMD.m` was able to consistently reproduce conservation laws in test cases.

Once a candidate density is known, the program `CONDENS.M` computes the flux using *Mathematica’s* `Integrate` function to do the necessary integration by parts. When transcendental nonlinearities were added to densities in `TransPDEDensityFlux.m`, *Mathematica’s* `Integrate` often failed to do the required integration. In multi-dimensions, it is necessary to invert a divergence to find the flux. CAS cannot do this. The homotopy operator solves both of these problems, providing a convenient method for computing fluxes.

The homotopy operator presented in this dissertation has been revised to work efficiently in code. Two issues arose with the original version that needed to be addressed before the homotopy operator would become a useful tool. First, the homotopy operator would produce a “swell” of terms internally, most of which would cancel before producing a solution. This was especially bad in the three-dimensional case. Using properties of combinatorics, the homotopy operator has been revised to eliminate the expression swell, thus generating considerably fewer terms. Secondly, in the multi-dimensional forms, the vector returned by the homotopy operator after it has inverted a divergence often contains divergence free or “curl” terms. A method to remove curl terms, based on linear algebra, has been developed. With these modifications, the homotopy operator has become a reliable tool for computing fluxes, even on differential expressions with transcendental or rational terms. However, like any integrator, there are limitations on what expressions the homotopy operator can integrate.

Since the homotopy operator has been successful on a wide variety of expressions, a second package, `HomotopyIntegrator.m`, composed of the homotopy operator code in `ConservationLawsMD.m` has been created. The package `HomotopyIntegrator.m` operates independently of `ConservationLawsMD.m` and can be used strictly as an integrator. The (1+1)-dimensional code is designed to work in tandem with *Mathematica*'s `Integrate` function. Supposedly, `Integrate` works on all types of functions, however, it often has difficulty integrating expressions containing unspecified functions involving transcendental or rational terms. The (2+1)- and (3+1)-dimensional code is an added feature for *Mathematica* since there currently is no function or package available for inverting divergences.

A third package accompanying this dissertation, `IndependenceTest.m`, also works independently of `ConservationLawsMD.m`. The package `IndependenceTest.m` takes a list of densities and compares them, checking that each density is independent of the other densities in the list, then prints a report. It is quite easy to compute two equivalent densities, but often difficult to recognize that they are equivalent. This package will be especially useful to researchers using other methods for computing conservation laws, giving a quick method for verification.

Several multi-dimensional PDEs with applications in fluid dynamics, gas dynam-

ics and mechanics are presented with a list of conservation laws for each PDE. Conservation laws have been published by various researchers for the Zakharov-Kuznetsov equation, the non-stationary transonic gas flow equation, the Kadomtsev-Petviashvili equation, the Khokhlov-Zabolotskaya equation, the shallow water fluid dynamics equations and the shallow water magnetohydrodynamics equations. These equations are presented as test cases and their conservation laws are compared to conservation laws presented in literature. Conservation laws for the Gardner equation, the Manakov-Santini system, the Camassa-Holm equation and the thermal shallow water magnetohydrodynamics equations have not been found in literature to date. The conservation laws presented for these equations are new. Partial results have been found for Navier's equation and the coupled Korteweg-de Vries equation. Some of the conservation laws for these equations are new.

In Chapter 2, notations are introduced, along with some basic definitions. Chapter 3 contains examples of PDEs along with several of their conservation laws. These examples will be used in later chapters to demonstrate and emphasize procedures for calculating conservation laws. Chapters 4 and 6 show the key tools used in the computation of conservation laws. The zeroth-Euler operator, otherwise known as the variational derivative from the calculus of variations, is introduced and its role in the Exactness Theorem is explained in Chapter 4. Chapter 6 shows the development of the homotopy operator used in the `ConservationLawsMD.m` and `HomotopyIntegrator.m`. A discussion of trivial conservation laws and a method for checking conservation laws for independence are given in Chapter 5. Chapter 7 contains a detailed description of how to compute conservation laws. The PDEs from Chapter 3 will be used to show the process and to highlight different techniques required for special situations that arise in the calculations. Chapter 8 shows additional examples of PDEs with some of their conservation laws, demonstrating the versatility of the program. Chapter 9 gives a description of the three software packages, `ConservationLawsMD.m`, `HomotopyIntegrator.m`, and `IndependenceTest.m` and explains how they run. Several examples showing results from a *Mathematica* notebook will demonstrate how each algorithm works and highlight the special features. Finally, Chapter 10 discusses future areas of research, including suggestions to make the algorithms more universal.

CHAPTER 2

NOTATION AND DEFINITIONS

Theorems and procedures shown in this dissertation are written in a form that allows for easy adaptation into the language of *Mathematica* or other CAS. Indeed, the procedures that will be described have all been programmed in *Mathematica*. It will be left to others to develop similar code in other CAS.

Notation for independent and dependent variables.

Conservation laws are computed for PDEs in (1+1)-, (2+1)-, or (3+1)-dimensions. An $(n+1)$ -dimensional PDE has n independent space variables, $\mathbf{x} = (x^1, \dots, x^n)$ and a time variable, t . The wording one-, two-, or three-dimensions will be used to denote the number of independent space variables under consideration, when t acts as a parameter. The function $\mathbf{u} = \mathbf{g}(\mathbf{x}, t)$ will represent a solution to a nonlinear PDE, where dependent variable \mathbf{u} has N components, that is, $\mathbf{u} = (u^1, u^2, \dots, u^j, \dots, u^N)$. The superscript notation is used in definitions and proofs involving the zeroth-Euler operator in Chapter 4 and the homotopy operator in Chapter 6. This notation closely follows the notation used by Olver [56] and has been widely adopted [14, 38, 39, 75] in this field of research. All concrete examples involving PDEs that occur throughout this dissertation will have no more than five dependent variables or three independent space variables. In all examples, \mathbf{x} will be taken as x , (x, y) , or (x, y, z) for one-, two- or three-dimensional problems, respectively. Also, for simplicity, the components of \mathbf{u} will be (u, v) instead of (u^1, u^2) . In PDEs where the dependent variables represent given quantities, appropriate dependent variables will replace components of \mathbf{u} . For example, in the shallow water hydrodynamics equations, (u, v, θ, ψ, h) will be used instead of $(u^1, u^2, u^3, u^4, u^5)$.

Notation for partial derivatives.

Partial derivatives on components of \mathbf{u} are denoted with a number-variable subscript notation. For example, $\frac{\partial^2 u}{\partial x^2}$ will be u_{2x} , $\frac{\partial^5 u}{\partial x^3 \partial y^2}$ will be u_{3x2y} , and $\frac{\partial^k u}{\partial x^k}$ will be u_{kx} . This notation serves two purposes; it makes multiple partial derivatives easy to repre-

sent in operators which involve sums over partial derivatives and it shortens lengthy differential expressions, making them easier to read.

Notation for jet spaces and differential functions.

The construction of a conservation law requires building a differential expression based on symmetries of the PDE. The differential expression has terms consisting of dependent variables and partial derivatives on dependent variables. The coefficients of these terms are constant or may be functions of the independent variables. Operations applied to differential expressions will act on the jet space, which is defined as follows.

Definition 2.1. *Let X be the space whose coordinate consists of the components of \mathbf{x} . Additionally, let U_0 be the space whose coordinate consists of all components of \mathbf{u} and U_M be the space whose coordinate consists of all partial derivatives of order M on all components of \mathbf{u} with respect to \mathbf{x} . The Cartesian product $U_0 \times U_1 \times \dots \times U_M$ is the space U^M with coordinate $\mathbf{u}^{(M)}$ [56]. The M th order jet space, J^M [38] is the Cartesian product $X \times U^M$.*

A differential function $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ is defined on J^M where f is determined by $f : X \times U^M \rightarrow \mathbb{R}$. Differentiations and integrations are carried out with respect to variables in J^M . In all cases where $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$, while f may contain terms where components of \mathbf{x} are multiplied by components of \mathbf{u} and derivatives of components of \mathbf{u} , f will not have terms containing only components of \mathbf{x} .

Example 2.1. *Let $\mathbf{u} = (u, v)$ and $\mathbf{x} = (x, y)$, then U_0 is the space with coordinate (u, v) , U_1 is the space with coordinate (u_x, u_y, v_x, v_y) , and U_2 is the space with coordinate $(u_{2x}, u_{xy}, u_{2y}, v_{2x}, v_{xy}, v_{2y})$. The jet space $J^2 = X \times U^2 = X \times U_0 \times U_1 \times U_2$ has the coordinate $(\mathbf{x}, \mathbf{u}^{(2)}) = (x, y, u, v, u_x, u_y, v_x, v_y, u_{2x}, u_{xy}, u_{2y}, v_{2x}, v_{xy}, v_{2y})$. A differential function $f(\mathbf{x}, \mathbf{u}^{(2)}(\mathbf{x}))$ is then defined on J^2 .*

Notation for operators acting on differential functions.

Several operators will appear that involve sums over derivatives on the jet space. The upper limit on the sums will be denoted by M_i^j , the order of differential function f for partial derivatives on u^j with respect to x^i .

Example 2.2. Let $\mathbf{x} = (x^1, x^2)$, $\mathbf{u} = (u^1, u^2)$, and $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$. An example of an operator acting in the jet space J^M is the partial derivative operator,

$$\mathcal{O} = \sum_{j=1}^N \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} \frac{\partial^{k_1+k_2}}{\partial u_{k_1 x^1 k_2 x^2}^j}, \quad (2.1)$$

where $u_{k_1 x^1 k_2 x^2}^j$ means $\frac{\partial^{k_1+k_2} u^j}{\partial (x^1)^{k_1} \partial (x^2)^{k_2}}$. For $\mathcal{O}f$, note that the second sum has index k_1 for partial derivatives on u^j with respect to x_1 , and the third sum has index k_2 for the x^2 -partial derivatives on u^j . Thus, M_1^j is the order for x^1 -partial derivatives on u^j in f and M_2^j is the order for x^2 -partial derivatives on u^j in f . Now, \mathcal{O} will be applied to a concrete example of a differential function where $(x^1, x^2) = (x, y)$ and $(u^1, u^2) = (u, v)$. Take $f(x, y, \mathbf{u}^{(4)}(x, y)) = v + xu_{xy}v_{2x} + y^2u_xu_{4xy}v_{3y}$. The order of u ($= u^1$) with respect to x ($= x^1$) is 4, so $M_1^1 = 4$. The value of M_2^1 is 1 since the order of u with respect to y ($= x^2$) is 1. The order for v ($= u^2$) with respect to x is 2, so $M_1^2 = 2$. Finally, $M_2^2 = 3$ since the order of v with respect to y is 3. Hence,

$$\begin{aligned} \mathcal{O}f &= \sum_{k_1=0}^4 \sum_{k_2=0}^1 \frac{\partial f}{\partial u_{k_1 x k_2 y}} + \sum_{k_1=0}^2 \sum_{k_2=0}^3 \frac{\partial f}{\partial v_{k_1 x k_2 y}} \\ &= \frac{\partial f}{\partial u_x} + \frac{\partial f}{\partial u_{xy}} + \frac{\partial f}{\partial u_{4xy}} + \frac{\partial f}{\partial v} + \frac{\partial f}{\partial v_{2x}} + \frac{\partial f}{\partial v_{3y}} \\ &= y^2u_{4xy}v_{3y} + xv_{2x} + y^2u_xv_{3y} + 1 + xu_{xy} + y^2u_xu_{4xy} \\ &= 1 + x(u_{xy} + v_{2x}) + y^2(u_xv_{3y} + u_xu_{4xy} + u_{4xy}v_{3y}). \end{aligned}$$

Identification of the order of dependent variables with respect to the independent variables is important for creating efficient algorithms. Table 2.1 shows how M_i^j is determined according to x^i and u^j for $1 \leq i \leq 3$ and $1 \leq j \leq 3$.

Total derivative operators.

The total derivative operator is the first operator to be defined using the notation discussed in Example 2.2. The total derivative is an algorithmic tool to compute derivatives with respect to a single independent variable on differential expressions, defined on the jet space.

The problems in this dissertation never require more than three independent space variables, $\mathbf{x} = (x, y, z)$ and one independent time variable, t . Of the four independent variables, one will act as a parameter in the problem, and in many cases that parameter is

Table 2.1: The order M_i^j is the order of u^j with respect to x^i .

			Dependent Variable Components		
			u^1	u^2	u^3
			u	v	w
Independent Variable Components	x^1	x	M_1^1	M_1^2	M_1^3
	x^2	y	M_2^1	M_2^2	M_2^3
	x^3	z	M_3^1	M_3^2	M_3^3

t . When t acts as a parameter, the jet space is $J^M = T \times X \times U^M$, where the coordinate for T is t , the coordinate for X is (x, y, z) and U^M contains partial derivatives with respect to variables in X only. A function defined on J^M may contain t explicitly, but none of the terms will have partial derivatives with respect to t . There are cases where a transformation requires that t be interchanged with one of the space variables, say t is exchanged with y . Now, y is the parameter so the coordinate for T is y and the coordinate for X is (t, x, z) . Again, the jet space is $J^M = T \times X \times U^M$ and U^M contains partial derivatives with respect to variables in X only.

The total derivative operator is defined for the components of \mathbf{x} only. The definition recognizes that there may be interchanges of independent variables, thus \mathbf{x} can be any combination of variables taken from x, y, z , and t .

Definition 2.2. Let $\mathbf{x} = (x^1, x^2, x^3)$ and $\mathbf{u} = (u^1, u^2, \dots, u^N)$. The total derivative operator D_{x^1} acting on the differential function $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ is defined as

$$D_{x^1} f = \frac{\partial f}{\partial x^1} + \sum_{j=1}^N \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} \sum_{k_3=0}^{M_3^j} u_{(k_1+1)x^1 k_2 x^2 k_3 x^3}^j \frac{\partial f}{\partial u_{k_1 x^1 k_2 x^2 k_3 x^3}^j}, \quad (2.2)$$

on the jet space J^M . The partial derivative $\frac{\partial f}{\partial x^1}$ acts on x^1 that appear explicitly in f , but not on u^j or any partial derivatives of u^j . The expression $u_{k_1 x^1 k_2 x^2 k_3 x^3}^j$ means $\frac{\partial^{k_1+k_2+k_3} u^j}{\partial^{k_1} x^1 \partial^{k_2} x^2 \partial^{k_3} x^3}$. D_{x^2} and D_{x^3} are defined analogously.

In a one-dimensional problem with independent variable x , (2.2) reduces to

$$D_x f = \frac{\partial f}{\partial x} + \sum_{j=1}^N \sum_{k=0}^{M_1^j} u_{(k+1)x}^j \frac{\partial f}{\partial u_{kx}^j}, \quad (2.3)$$

which is the familiar form given in [33, 35, 36]. In this case, one may have dependent variable $u^j(t, x)$, where t is a parameter.

Example 2.3. Let $\mathbf{x} = x$, $\mathbf{u} = u^1 = u$, and let t be a parameter. Taking $f(t, x, \mathbf{u}^{(2)}(t, x)) = x^2u^3 + u_x^2 + uu_{2x}$, by (2.3),

$$\begin{aligned} D_x f &= \frac{\partial f}{\partial x} + \sum_{k_1=0}^2 u_{(k_1+1)x} \frac{\partial f}{\partial u_{k_1x}} \\ &= \frac{\partial f}{\partial x} + u_x \frac{\partial f}{\partial u} + u_{2x} \frac{\partial f}{\partial u_x} + u_{3x} \frac{\partial f}{\partial u_{2x}} \\ &= 2xu^3 + u_x(3x^2u^2 + u_{2x}) + u_{2x}(2u_x) + u_{3x}(u) \\ &= 2xu^3 + 3x^2u^2u_x + 3u_xu_{2x} + uu_{3x}. \end{aligned}$$

Application of the total derivative operator is identical to applying the product rule and chain rule to f . However, the total derivative operator (2.2) is algorithmic and easy to code.

The next example shows the difference between D_x and D_y when applied to differential functions with two independent variables and with t as an independent parameter. In each case, (2.2) is reduced to a simpler form.

Example 2.4. Let $\mathbf{x} = (x, y)$ and $\mathbf{u} = (u^1, u^2) = (u, v)$, and let t be a parameter. Take $f(t, x, y, \mathbf{u}^{(2)}(t, x, y)) = yu_x - u_x^2v_{xy} + x^3u_{2x}v$. By (2.2),

$$\begin{aligned} D_x f &= \frac{\partial f}{\partial x} + \sum_{j=1}^N \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} u_{(k_1+1)xk_2y}^j \frac{\partial f}{\partial u_{k_1xk_2y}^j} \\ &= \frac{\partial f}{\partial x} + \sum_{k_1=0}^2 u_{(k_1+1)x} \frac{\partial f}{\partial u_{k_1x}} + \sum_{k_1=0}^1 \sum_{k_2=0}^1 v_{(k_1+1)xk_2y} \frac{\partial f}{\partial v_{k_1xk_2y}} \\ &= \frac{\partial f}{\partial x} + u_{2x} \frac{\partial f}{\partial u_x} + u_{3x} \frac{\partial f}{\partial u_{2x}} + v_x \frac{\partial f}{\partial v} + v_{2xy} \frac{\partial f}{\partial v_{xy}} \\ &= 3x^2u_{2x}v + u_{2x}(y - 2u_xv_{xy}) + u_{3x}(x^3v) + v_x(x^3u_{2x}) - v_{2xy}(u_x^2) \\ &= yu_{2x} - 2u_xu_{2x}v_{xy} - u_x^2v_{2xy} + 3x^2u_{2x}v + x^3u_{3x}v + x^3u_{2x}v_x, \\ D_y f &= \frac{\partial f}{\partial y} + \sum_{j=1}^N \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} u_{k_1x(k_2+1)y}^j \frac{\partial f}{\partial u_{k_1xk_2y}^j} \\ &= \frac{\partial f}{\partial y} + \sum_{k_1=0}^2 \sum_{k_2=0}^0 u_{k_1x(k_2+1)y} \frac{\partial f}{\partial u_{k_1xk_2y}} + \sum_{k_1=0}^1 \sum_{k_2=0}^1 v_{k_1x(k_2+1)y} \frac{\partial f}{\partial v_{k_1xk_2y}} \\ &= \frac{\partial f}{\partial y} + u_{xy} \frac{\partial f}{\partial u_x} + u_{2xy} \frac{\partial f}{\partial u_{2x}} + v_y \frac{\partial f}{\partial v} + v_{x2y} \frac{\partial f}{\partial v_{xy}} \end{aligned}$$

$$\begin{aligned}
&= u_x + u_{xy}(y - 2u_x v_{xy}) + u_{2xy}(x^3 v) + v_y(x^3 u_{2x}) - v_{x2y}(u_x^2) \\
&= u_x + yu_{xy} - 2u_x u_{xy} v_{xy} - u_x^2 v_{x2y} + x^3 u_{2xy} v + x^3 u_{2x} v_y.
\end{aligned}$$

The jet space for problems using a total derivative operator as given in Example 2.4 is $J^M = T \times X \times U^M$ where the coordinate for X is (x, y) and the coordinate for T is t . In the next example, the jet space is also $J^M = T \times X \times U^M$, but now (t, x) is the coordinate for X and y is the coordinate for T .

Example 2.5. Let $\mathbf{x} = (t, x)$ and $\mathbf{u} = u^1 = u$, and let y be a parameter. Take $f(y, \mathbf{x}, \mathbf{u}^{(3)}(y, \mathbf{x})) = u^3 + tu_x^2 + xu_t - u_{t3x}$. By (2.2),

$$\begin{aligned}
D_t f &= \frac{\partial f}{\partial t} + \sum_{k_1=0}^1 \sum_{k_2=0}^3 u^{(k_1+1)tk_2x} \frac{\partial f}{\partial u_{k_1tk_2x}} \\
&= \frac{\partial f}{\partial t} + u_t \frac{\partial f}{\partial u} + u_{2t} \frac{\partial f}{\partial u_t} + u_{tx} \frac{\partial f}{\partial u_x} + u_{2t3x} \frac{\partial f}{\partial u_{t3x}} \\
&= u_x^2 + u_t(3u^2) + u_{2t}(x) + u_{tx}(2tu_x) - u_{2t3x}(1) \\
&= 3u^2 u_t + u_x^2 + 2tu_x u_{tx} + xu_{2t} - u_{2t3x}.
\end{aligned}$$

In the case where t is a parameter and x, y , and z are the independent variables, the total t -derivative has the short form

$$D_t f = \frac{\partial f}{\partial t} + \sum_{j=1}^2 \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} \sum_{k_3=0}^{M_3^j} \frac{\partial f}{\partial u_{k_1 x k_2 y k_3 z}^j} D_x^{k_1} D_y^{k_2} D_z^{k_3} u_t^j. \quad (2.4)$$

The partial derivative $\frac{\partial f}{\partial t}$ acts on explicit t in f only. Since t is a parameter, there are no partial derivatives on \mathbf{u} with respect to t in f . The total derivative operators $D_x^{k_1}$, $D_y^{k_2}$, and $D_z^{k_3}$ are acting on u_t only, and not on f . This case occurs often when testing the density of a conservation law.

Example 2.6. Let $\mathbf{x} = (x, y, z)$ and $\mathbf{u} = (u^1, u^2) = (u, v)$. Taking $f(t, \mathbf{x}, \mathbf{u}^{(2)}(t, \mathbf{x})) = uu_x v_{2x} - xuv_{xy} + t^2 u_z^2$,

$$\begin{aligned}
D_t f &= \frac{\partial f}{\partial t} + \sum_{k_1=0}^1 \sum_{k_3=0}^1 \frac{\partial f}{\partial u_{k_1 x k_3 z}} D_x^{k_1} D_z^{k_3} u_t + \sum_{k_1=2}^1 \sum_{k_2=1}^1 \frac{\partial f}{\partial u_{k_1 x k_2 y}} D_x^{k_1} D_y^{k_2} v_t \\
&= \frac{\partial f}{\partial t} + \frac{\partial f}{\partial u} u_t + \frac{\partial f}{\partial u_x} D_x u_t + \frac{\partial f}{\partial u_z} D_z u_t + \frac{\partial f}{\partial v_{xy}} D_x D_y v_t + \frac{\partial f}{\partial v_{2x}} D_x^2 v_t \\
&= 2tu_z^2 + (u_x v_{2x} - x v_{xy}) u_t + (u v_{2x}) u_{tx} + (2t^2 u_z) u_{tz} - (xu) v_{txy} + (uu_x) v_{t2x} \\
&= u_t u_x v_{2x} + uu_{tx} v_{2x} + uu_x v_{t2x} - xu_t v_{xy} - xuv_{txy} + 2tu_z^2 + 2t^2 u_z u_{tz}.
\end{aligned}$$

Again, (2.4) gives an algorithmic definition of D_t . The total t -derivative can also be computed in a straightforward manner by applying the chain rule and product rule.

Notation for gradient and divergence operators.

The gradient operator will be denoted by ∇ , where $\nabla u^j = (u_x^j, u_y^j)$ when $\mathbf{x} = (x, y)$ and $\nabla u^j = (u_x^j, u_y^j, u_z^j)$ when $\mathbf{x} = (x, y, z)$, $j = 1 \dots N$. The symbol Δ denotes the Laplacian operator in Cartesian coordinates where $\Delta u^j = u_{2x}^j + u_{2y}^j$ when $\mathbf{x} = (x, y)$ and $\Delta u^j = u_{2x}^j + u_{2y}^j + u_{2z}^j$ when $\mathbf{x} = (x, y, z)$, $j = 1 \dots N$. The divergence of a differential vector function on the jet space is called the total divergence and requires the use of the total derivative operators. To stress the use of total derivatives, the notation $\text{Div } \mathbf{F}$ is used for total divergence instead of $\nabla \cdot \mathbf{F}$.

Definition 2.3. *The total divergence of a differential vector function, $\mathbf{F} = (F^1, F^2)$, with two independent variables, $\mathbf{x} = (x, y)$, is given by*

$$\text{Div } \mathbf{F} = D_x F^1 + D_y F^2. \tag{2.5}$$

Likewise, the divergence on a differential vector function, $\mathbf{F} = (F^1, F^2, F^3)$ with three independent variables, $\mathbf{x} = (x, y, z)$ is given by

$$\text{Div } \mathbf{F} = D_x F^1 + D_y F^2 + D_z F^3. \tag{2.6}$$

CHAPTER 3

EXAMPLES OF CONSERVATION LAWS OF NONLINEAR PDES IN (2+1)- AND (3+1)-DIMENSIONS

The purpose of this chapter is to introduce three examples of PDEs, together with a list of their conservation laws. The conservation laws shown in this chapter have all been obtained using the conservation laws algorithm, `ConservationLawsMD.m`, and provide the reader with a sampling of the program's capabilities. Techniques for computing conservation laws for these PDEs will be shown in detail in Chapter 7.

In this dissertation, the methods used to calculate conservation laws require PDEs to be written in evolution form. Taking $\mathbf{x} = (x, y, z)$, the evolution form for a (3+1)-dimensional PDE is

$$\mathbf{u}_t = \mathbf{G}(\mathbf{u}^{(M)}(\mathbf{x})) = \mathbf{G}(u^1, u_x^1, u_y^1, u_z^1, u_{2x}^1, u_{2y}^1, u_{2z}^1, u_{xy}^1, \dots, u_{M_1^N M_2^N M_3^N}^N), \quad (3.1)$$

where \mathbf{G} is assumed to be smooth and does not explicitly depend on \mathbf{x} and t . Although requiring evolution equations is a serious restriction, many PDEs can be written as a single evolution equation or system of evolution equations after some simple transformations. One such transformation may use an interchange of variables. Another may introduce one or more new dependent variables to transform the PDE into an evolution system.

Example 3.1. *The (3+1)-dimensional non-stationary transonic gas flow (NTGF) equation [27, 44],*

$$2u_{tx} + u_x u_{2x} - u_{2y} - u_{2z} = 0, \quad (3.2)$$

is not an evolution equation, but can be written as an evolution system by using two simple transformations. First interchange the independent variables t and z to get $2u_{xz} + u_x u_{2x} - u_{2y} - u_{2t} = 0$, then introduce $v = u_t$ to obtain the system

$$\begin{aligned} u_t &= v, \\ v_t &= 2u_{xz} + u_x u_{2x} - u_{2y}. \end{aligned} \quad (3.3)$$

Both equations in the system are evolution equations. Note that it is also possible to get an evolution system by interchanging independent variables t and y , so that (3.2) becomes $2u_{xy} + u_x u_{2x} - u_{2t} - u_{2z} = 0$. Next introduce $v = u_t$ to form the system

$$\begin{aligned} u_t &= v, \\ v_t &= 2u_{xy} + u_x u_{2x} - u_{2z}. \end{aligned} \tag{3.4}$$

The choice of transformations has no effect on the final form of the conservation laws. The inverse transformation is applied after the conservation laws are obtained, giving results corresponding to the PDE in its original form.

A conservation law for a PDE is defined as follows.

Definition 3.1. *A conservation law for the PDE (3.1) is a PDE itself in the form*

$$\mathbf{D}_t \rho + \text{Div } \mathbf{J} = 0, \tag{3.5}$$

where \mathbf{D}_t is the total derivative with respect to t , Div is the operator described in (2.5) and (2.6), $\rho = \rho(t, \mathbf{x}, \mathbf{u}^{(M)}(t, \mathbf{x}))$ is the conserved density, and $\mathbf{J} = \mathbf{J}(t, \mathbf{x}, \mathbf{u}^{(P)}(t, \mathbf{x}))$ is the associated flux. Equation (3.5) is satisfied for all solutions of (3.1) [1].

`ConservationLawsMD.m` can produce conserved densities that are polynomial in $\mathbf{u}^{(M)}$, including conserved densities with polynomial functions of independent variables multiplied to terms. Fluxes have fewer restrictions and are determined by (3.5) once $\mathbf{D}_t \rho$ is computed. If the PDE has rational terms, the flux may also contain rational terms. Transcendental terms are not allowed at this point, but may be included at a later date. See [2, 3] for methods for computing conservation laws for (1+1)-dimensional PDEs with transcendental terms.

If both ρ and \mathbf{J} are local functionals, then the conservation law is local. A local functional requires that the value of a functional f at a particular \mathbf{x} depend only on the values of \mathbf{u} in an arbitrary small neighborhood of \mathbf{x} [30, 52]. In the multi-dimensional examples, conservation laws that are both local and polynomial only in the dependent variables do not occur frequently, so adaptations to both the PDEs and the methods to compute conservation laws have been made to pursue conservation laws that have terms multiplied by polynomial functions of independent variables.

The conservation law given by (3.5) is the differential form, which holds over any volume. A conservation law in three dimensions can be stated in the integral form

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\mathbf{D}_t \rho + \text{Div } \mathbf{J}) dV = 0. \quad (3.6)$$

If \mathbf{u} and all partial derivatives of \mathbf{u} go to zero as $V \rightarrow \infty$, then by the divergence theorem,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{Div } \mathbf{J} dV = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{J} \cdot \mathbf{n} dS = 0,$$

where S is the boundary of V and \mathbf{n} is normal to S . Then

$$\mathbf{D}_t \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho dV = 0,$$

so

$$P = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho dV$$

is constant in time. P called a constant of motion [1, 34]. The first few constants of motion express physical conservation laws, such as conservation of mass or conservation of energy.

The PDEs presented in this chapter are chosen to demonstrate and highlight different features of the process that `ConservationLawsMD.m` uses to compute a conservation law. Full details will be shown in subsequent chapters. In Chapter 7 a conservation law for each equation will be calculated in a step-by-step process to highlight the algorithms in `ConservationLawsMD.m`.

3.1 The Zakharov-Kuznetsov Equation

The Zakharov-Kuznetsov (ZK) equation characterizes three-dimensional ion-sound solitons in a low pressure uniform magnetized plasma [83]. The ZK equation is an evolution equation. After re-scaling [72], it has the form

$$u_t + \alpha u u_x + \beta \Delta u_x = 0,$$

where α and β are parameters. The (2+1)-dimensional equation,

$$u_t + \alpha u u_x + \beta (u_{2x} + u_{2y})_x = 0, \quad (3.7)$$

is one of several two-dimensional generalizations of the Korteweg-de Vries (KdV) [52] equation,

$$u_t + \alpha u u_x + u_{3x} = 0. \quad (3.8)$$

`ConservationLawsMD.m` computes four conservation laws for the ZK equation (3.7). Three do not have explicit independent variables, and one has terms multiplied by independent variables. They are

$$D_t(u) + D_x\left(\frac{\alpha}{2}u^2 + \beta u_{2x}\right) + D_y(\beta u_{xy}) = 0, \quad (3.9)$$

$$D_t(u^2) + D_x\left(\frac{2\alpha}{3}u^3 - \beta(u_x^2 - u_y^2) + 2\beta u(u_{2x} + u_{2y})\right) + D_y(-2\beta u_x u_y) = 0, \quad (3.10)$$

$$\begin{aligned} D_t\left(u^3 - \frac{3\beta}{\alpha}(u_x^2 + u_y^2)\right) + D_x\left(\frac{3\alpha}{4}u^4 + 3\beta u^2 u_{2x} - 6\beta u(u_x^2 + u_y^2) + \frac{3\beta^2}{\alpha}(u_{2x}^2 - u_{2y}^2)\right. \\ \left. - \frac{6\beta^2}{\alpha}(u_x(u_{3x} + u_{x2y}) + u_y(u_{2xy} + u_{3y}))\right) + D_y\left(3\beta u^2 u_{xy}\right. \\ \left. + \frac{6\beta^2}{\alpha}u_{xy}(u_{2x} + u_{2y})\right) = 0, \end{aligned} \quad (3.11)$$

$$\begin{aligned} D_t\left(tu^2 - \frac{2}{\alpha}xu\right) + D_x\left(t\left(\frac{2\alpha}{3}u^3 - \beta(u_x^2 - u_y^2) + 2\beta u(u_{2x} + u_{2y})\right) - \frac{2}{\alpha}x\left(\frac{\alpha}{2}u^2 + \beta u_{2x}\right)\right. \\ \left. + \frac{2\beta}{\alpha}u_x\right) - D_y\left(2\beta(tu_x u_y + \frac{1}{\alpha}xu_{xy})\right) = 0. \end{aligned} \quad (3.12)$$

Note that conservation law (3.9) is the ZK equation itself. No other conservation laws were found. Zakharov and Kuznetsov [83] were able to identify three integrals of motion which are identical to (3.9), the conservation of mass, (3.10), the conservation of momentum, and (3.12), the conservation of center of mass. Shivamoggi et al. [68] and Infeld [40] both claim that there are only four conservation laws for the ZK equation, but give different results. Infeld gives the same results as Zakharov and Kuznetsov, but with one additional conservation law. Shivamoggi et al. computed conservation laws for the potential ZK equation,

$$u_t + \frac{1}{2}u_x^2 + u_{3x} + u_{x2y} = 0. \quad (3.13)$$

Doing so, they produced four nonlocal conservation laws for the ZK equation. The densities given by Shivamoggi et al. [68] are correct for (3.13), but `ConservationLawsMD.m` computes different fluxes for three of the four conservation laws. Shivamoggi has confirmed that there are typographical errors in [66] and has provided corrected versions [67]. A fifth conservation law for (3.13) where the terms have explicit independent variables was computed by `ConservationLawsMD.m`. This fifth law was not reported by Shivamoggi.

The existence of just four conservation laws for (3.7) supports the proposition that the ZK equation is not integrable. The ZK equation has also been found not integrable

by the inverse scattering transform method [72]. Furthermore, the ZK equation does not possess the Painlevé property¹ There is some confusion about the Painlevé test performed on the ZK equation. Shivamoggi verified that the ZK equation (3.7) has the Painlevé property [66], then later refuted his results [68] with detailed computations for (3.13). An algorithm by Baldwin and Hereman [8] that performs the Painlevé test confirms the results in [68]. Indeed, neither the ZK equation nor the potential ZK equation possess the Painlevé property.

If the αuu_x term in (3.7) is replaced by $\alpha u^n u_x$, where n is a rational number, the new equation is called the generalized (2+1)-dimensional ZK equation [48],

$$u_t + \alpha u^n u_x + \beta(u_{2x} + u_{2y})_x = 0. \quad (3.14)$$

The generalized ZK equation has three conservation laws where terms are not multiplied by independent variables. These are

$$D_t(u) + D_x\left(\frac{\alpha}{n+1}u^{n+1} + \beta u_{2x}\right) + D_y(\beta u_{xy}) = 0, \quad (3.15)$$

$$D_t(u^2) + D_x\left(\frac{2\alpha}{n+2}u^{n+2} - \beta(u_x^2 - u_y^2) + 2\beta u(u_{2x} + u_{2y})\right) - D_y(2\beta u_x u_y) = 0, \quad (3.16)$$

$$D_t\left(u^{n+2} - \frac{(n+1)(n+2)\beta}{2\alpha}(u_x^2 + u_y^2)\right) + D_x\left(\frac{(n+2)\alpha}{2(n+1)}u^{2(n+1)} + (n+2)\beta u^{n+1}u_{2x} - (n+1)(n+2)\beta u^n(u_x^2 + u_y^2) + \frac{(n+1)(n+2)\beta^2}{2\alpha}(u_{2x}^2 - u_{2y}^2) - \frac{(n+1)(n+2)\beta^2}{\alpha}(u_x(u_{3x} + u_{x2y}) + u_y(u_{2xy} + u_{3y}))\right) + D_y\left((n+2)\beta u^{n+1}u_{xy} + \frac{(n+1)(n+2)\beta^2}{\alpha}u_{xy}(u_{2x} + u_{2y})\right) = 0. \quad (3.17)$$

Equation (3.14) with $n = \frac{1}{2}$ is similar to the Schamel equation [70] generalized to two-dimensions, which also has three conservation laws.

`ConservationLawsMD.m` cannot directly compute the density for (3.14) since it has an unspecified exponent. To find (3.15), the program was run using (3.14) with $n = 2, 3, 4, 6,$ and 12 . Table 3.1 lists the conservation laws reported by `ConservationLawsMD.m`. These results could also be computed by hand in a straightforward manner.

From the conservation laws in Table 3.1, it is easy to see the pattern for unspecified n , resulting in (3.15), which is easy to compute by hand. Once the pattern is

¹If a PDE has no movable singularities other than poles, it possesses the Painlevé property [1]. PDEs that possess the Painlevé property are said to be integrable.

Table 3.1: List of conservation laws for the generalized Zakharov-Kuznetsov equation (3.14) for specific values of n .

n	Conservation Law
2	$D_t(u) + D_x(\frac{1}{3}\alpha u^3 + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$
3	$D_t(u) + D_x(\frac{1}{4}\alpha u^4 + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$
4	$D_t(u) + D_x(\frac{1}{5}\alpha u^5 + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$
6	$D_t(u) + D_x(\frac{1}{7}\alpha u^7 + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$
12	$D_t(u) + D_x(\frac{1}{13}\alpha u^{13} + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$
$\frac{1}{2}$	$D_t(u) + D_x(\frac{2}{3}\alpha u^{3/2} + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$
$-\frac{1}{3}$	$D_t(u) + D_x(\frac{3}{2}\alpha u^{2/3} + \beta u_{2x}) + D_y(\beta u_{xy}) = 0$

recognized, both equation (3.14) with the exponent n left unspecified and the density, u , are input into `ConservationLawsMD.m`. The program attaches arbitrary coefficients to every term, recalculates these coefficients, and returns the revised density if changes are necessary. After the density is determined, the program calculates the flux. If the expression given as a candidate density is not a density, the program will state so. In this case, the program confirms that u is indeed the density for (3.14) and calculates the flux as given in (3.15) without any restrictions on the exponent n . Conservation laws are given for $n = \frac{1}{2}$ and $n = -\frac{1}{3}$ in Table 3.1 to show that n does not have to be a positive integer. However, `ConservationLawsMD.m` will not test irrational values for n . The program automatically verifies every density-flux pair in (3.5) and reports an error if the resulting expression does not simplify to zero. Using a similar procedure, but with considerably more effort, (3.16) and (3.17) were obtained.

3.2 The (3+1)-Dimensional Equation for Non-stationary Transonic Gas Flow

The PDE for non-stationary transonic gas flow (NTGF),

$$2u_{tx} + u_x u_{2x} - u_{2y} - u_{2z} = 0, \quad (3.18)$$

is introduced as a (3+1)-dimensional example. The NTGF equation arises from the study of gas dynamics [27, 44], where u represents the velocity of a fluid. Like the NTGF equation, many multidimensional wave and gas flow equations have a mixed-derivative term, u_{tx} , where the t -derivative is combined with a space variable derivative.

Since these are not evolution equations, `ConservationLawsMD.m` will apply the transformations described in Example 3.1 to form an evolution equation or system of evolution equations. The evolution form is then used by the program to compute conservation laws.

The NTGF equation is a conservation law itself,

$$D_t\left(2u_x\right) + D_x\left(\frac{1}{2}u_x^2\right) - D_y\left(u_y\right) - D_z\left(u_z\right) = 0, \quad (3.19)$$

representing conservation of momentum. Four conservation laws without independent variables multiplied to terms can be generated for (3.18). They are

$$D_t\left(u_x^2\right) + D_x\left(\frac{1}{2}\left(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2\right)\right) + D_y\left(\frac{1}{2}(uu_{xy} - u_xu_y)\right) - D_z\left(u_xu_z\right) = 0, \quad (3.20)$$

$$D_t\left(u_xu_y\right) - D_x\left(\frac{1}{3}(3uu_{ty} - u_x^2u_y + uu_xu_{xy})\right) + D_y\left(\frac{1}{2}(2uu_{tx} + \frac{2}{3}uu_xu_{2x} - u_y^2 + u_z^2)\right) - D_z\left(u_yu_z\right) = 0, \quad (3.21)$$

$$D_t\left(u_xu_z\right) + D_x\left(u_z(u_t + \frac{1}{2}u_x^2)\right) - D_y\left(u_yu_z\right) - D_z\left(\frac{1}{2}(2u_tu_x + \frac{1}{3}u_x^3 - u_y^2 + u_z^2)\right) = 0, \quad (3.22)$$

$$D_t\left(uu_xu_{2x} - \frac{3}{2}uu_{2y} + \frac{3}{2}u_z^2\right) + D_x\left(3u_t^2 + u_tu_x^2 - uu_xu_{tx}\right) + D_y\left(\frac{3}{2}(uu_{ty} - u_tu_y)\right) - D_z\left(3u_tu_z\right) = 0. \quad (3.23)$$

The NTGF equation also has conservation laws where the density and flux depend explicitly on the independent variables. `ConservationLawsMD.m` reports a large number of such conservation laws which have been grouped into sets. The first set consists of conservation laws that do not have a generalization,

$$D_t\left(u_x(yu_z - zu_y)\right) + D_x\left(yu_z(u_t + \frac{1}{2}u_x^2) + \frac{1}{3}z(3uu_{ty} - u_x^2u_y + uu_xu_{xy})\right) - D_y\left(yu_yu_z + \frac{1}{2}z(2uu_{tx} + \frac{2}{3}uu_xu_{2x} - u_y^2 + u_z^2)\right) - D_z\left(\frac{1}{2}y(2u_tu_x + \frac{1}{3}u_x^3 - u_y^2 + u_z^2) - zu_yu_z\right) = 0, \quad (3.24)$$

$$\begin{aligned}
& \mathbf{D}_t \left(t(uu_x u_{2x} - \frac{3}{2}uu_{2y} + \frac{3}{2}u_z^2) + \frac{3}{5}xu_x^2 + \frac{9}{5}u_x(yu_y + zu_z) \right) + \mathbf{D}_x \left(t(3u_t^2 + u_t u_x^2 \right. \\
& \quad \left. - uu_x u_{tx}) + \frac{3}{10}x(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2) - \frac{3}{5}y(3uu_{ty} - u_x^2 u_y + uu_x u_{xy}) \right. \\
& \quad \left. + \frac{9}{5}zu_z(u_t + \frac{1}{2}u_x^2) + u(\frac{9}{5}u_t + \frac{1}{10}u_x^2) \right) + \mathbf{D}_y \left(\frac{3}{2}t(uu_{ty} - u_t u_y) \right. \\
& \quad \left. + \frac{3}{10}x(uu_{xy} - u_x u_y) + \frac{9}{10}y(2uu_{tx} + \frac{2}{3}uu_x u_{2x} - u_y^2 + u_z^2) - \frac{9}{5}zu_y u_z \right) \\
& \quad \left. - \mathbf{D}_z \left(3tu_t u_z + \frac{3}{5}xu_x u_z + \frac{9}{5}yu_y u_z + \frac{9}{10}z(2u_t u_x + \frac{1}{3}u_x^3 - u_y^2 + u_z^2) + \frac{9}{5}uu_z \right) = 0, \quad (3.25)
\end{aligned}$$

$$\begin{aligned}
& \mathbf{D}_t \left(t^2(uu_x u_{2x} - \frac{3}{2}uu_{2y} + \frac{3}{2}u_z^2) + \frac{9}{5}(\frac{2}{3}tx + y^2 + z^2)u_x^2 + \frac{18}{5}tu_x(yu_y + zu_z) - \frac{12}{5}x^2 u_x \right) \\
& \quad + \mathbf{D}_x \left(t^2(3u_t^2 + u_t u_x^2 - uu_x u_{tx}) + \frac{9}{10}(\frac{2}{3}tx + y^2 + z^2)(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2) \right. \\
& \quad \left. - \frac{6}{5}ty(3uu_{ty} - u_x^2 u_y + uu_x u_{xy}) + \frac{18}{5}tzu_z(u_t + \frac{1}{2}u_x^2) \right. \\
& \quad \left. + \frac{1}{5}tu(18u_t + u_x^2) - \frac{9}{5}yuu_y - \frac{3}{5}x^2 u_x^2 \right) + \mathbf{D}_y \left(\frac{3}{2}t^2(uu_{ty} - u_t u_y) \right. \\
& \quad \left. + \frac{9}{10}(\frac{2}{3}tx + y^2 + z^2)(uu_{xy} - u_x u_y) + \frac{9}{5}ty(2uu_{tx} + \frac{2}{3}uu_x u_{2x} - u_y^2 + u_z^2) \right. \\
& \quad \left. - \frac{18}{5}tzu_y u_z + \frac{6}{5}x^2 u_y \right) - \mathbf{D}_z \left(3t^2 u_t u_z + \frac{9}{5}(\frac{2}{3}tx + y^2 + z^2)u_x u_z + \frac{18}{5}tyu_y u_z \right. \\
& \quad \left. + \frac{9}{5}tz(2u_t u_x + \frac{1}{3}u_x^3 - u_y^2 + u_z^2) + \frac{18}{5}tuu_z - \frac{6}{5}x^2 u_z \right) = 0. \quad (3.26)
\end{aligned}$$

Both (3.23) and (3.24) are listed as conservation laws by Khamitova [44], but none of the others are given. In Chapter 5, where the equivalence of conservation laws is discussed, it will be shown that (3.23) is equivalent to the conservation law given in [44].

All other conservation laws calculated by `ConservationLawsMD.m` for the NTGF equation can be grouped into three sets where each set can be generalized to a single equation involving an arbitrary function, $f(t)$. One set of conservation laws is

$$\begin{aligned}
& \mathbf{D}_t \left(tu_x^2 - 4xu_x \right) + \mathbf{D}_x \left(\frac{1}{2}t(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2) - xu_x^2 \right) + \mathbf{D}_y \left(\frac{1}{2}t(uu_{xy} - u_x u_y) \right. \\
& \quad \left. + 2xu_y \right) + \mathbf{D}_z \left(-tu_x u_z + 2xu_z \right) = 0, \quad (3.27)
\end{aligned}$$

$$\begin{aligned}
& \mathbf{D}_t \left(t^2 u_x^2 + 8tu \right) + \mathbf{D}_x \left(\frac{1}{2}t^2(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2) - 2(tx + z^2)(4u_t + u_x^2) \right) \\
& \quad + \mathbf{D}_y \left(\frac{1}{2}t^2(uu_{xy} - u_x u_y) + 4(tx + z^2)u_y \right) + \mathbf{D}_z \left(-t^2 u_x u_z - 8zu \right. \\
& \quad \left. + 4(tx + z^2)u_z \right) = 0, \quad (3.28)
\end{aligned}$$

$$\begin{aligned}
& \mathbf{D}_t \left(t^3 u_x^2 + 12t^2 u \right) + \mathbf{D}_x \left(\frac{1}{2}t^3(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2) - 3(t^2 x + 2tz^2)(4u_t + u_x^2) \right) \\
& \quad + \mathbf{D}_y \left(\frac{1}{2}t^3(uu_{xy} - u_x u_y) + 6(t^2 x + 2tz^2)u_y \right) + \mathbf{D}_z \left(-t^3 u_x u_z - 24tzu \right. \\
& \quad \left. + 6(t^2 x + 2tz^2)u_z \right) = 0. \quad (3.29)
\end{aligned}$$

Conservation law (3.27) requires some modification before it will fit the pattern in this

set. The term $-4xu_x$ can be moved into the flux while concurrently moving $4u$ into the density,

$$\begin{aligned} D_t(-4xu_x) &= -4xu_{tx} - 4u_t + 4u_t \\ &= D_t(4u) + D_x(-4xu_t). \end{aligned} \tag{3.30}$$

All other conservation laws in this set calculated by `ConservationLawsMD.m` have the form of (3.29) with higher powers of t . Indeed, all factors with the form t^n in the conservation law can be replaced by an arbitrary function of t , say $f = f(t)$, and the derivatives of f . This leads to a general conservation law

$$\begin{aligned} D_t\left(fu_x^2 + 4f'u\right) + D_x\left(\frac{1}{2}f\left(\frac{2}{3}u_x^3 - uu_{2y} + u_z^2\right) - (xf' + z^2f'')(4u_t + u_x^2)\right) \\ + D_y\left(\frac{1}{2}f(uu_{xy} - u_xu_y) + 2(xf' + z^2f'')u_y\right) \\ + D_z\left(-fu_xu_z + 2(xf' + z^2f'')u_z - 4f''zu\right) = 0. \end{aligned} \tag{3.31}$$

When $f = 1$, (3.31) is the same as conservation law (3.20), and when $f = t$, (3.31) reduces to (3.27), using (3.30). For $f = t^2$ and $f = t^3$, (3.31) reduces to (3.28) and (3.29), respectively.

`ConservationLawsMD.m` cannot directly calculate a density like (3.31) that contains arbitrary functions not found in the PDE. However, once a general conservation law, for example (3.31), is deduced, the density and the PDE can be submitted together to `ConservationLawsMD.m` for further analysis. The program will assign arbitrary coefficients to each term, recalculate the coefficients, then return a modified density or state that the expression given is not a density. This verification procedure works for densities that have arbitrary functions that do not appear in the PDE. Once the density is computed and verified, the program will calculate the flux, and finally check the density-flux pair in (3.5). A drawback of this procedure is that `ConservationLawsMD.m` requires that the PDE be submitted as an evolution equation or system. Any transformations used to change the PDE into evolution form must also be applied to the density before submission. More details about this are given in Section 9.1.3.

`ConservationLawsMD.m` calculates two additional sets of conservation laws. They lead to the generalized forms,

$$\begin{aligned}
& D_t \left(f u_x u_y + y f' u_x^2 + 4y f'' u \right) - D_x \left(\frac{1}{3} f (3u u_{ty} - u_x^2 u_y + u u_x u_{xy}) \right. \\
& \quad \left. - \frac{1}{2} y f' \left(\frac{2}{3} u_x^3 - u u_{2y} + u_z^2 \right) + \frac{1}{2} f' u u_y + y (x f'' + z^2 f''') (4u_t + u_x^2) \right) \\
& \quad + D_y \left(\frac{1}{2} f (2u u_{tx} + \frac{2}{3} u u_x u_{2x} - u_y^2 + u_z^2) + \frac{1}{2} y f' (u u_{xy} - u_x u_y) \right. \\
& \quad \left. - 2(x f'' + z^2 f''') (u - y u_y) \right) - D_z \left(f u_y u_z + y f' u_x u_z \right. \\
& \quad \left. - 2y (x f'' + z^2 f''') u_z + 4y z f''' u \right) = 0, \tag{3.32}
\end{aligned}$$

and

$$\begin{aligned}
& D_t \left(f u_x u_z + z f' u_x^2 + 4z f'' u \right) + D_x \left(f u_z \left(\frac{1}{2} u_x^2 + u_t \right) + \frac{1}{2} z f' \left(\frac{2}{3} u_x^3 - u u_{2y} + u_z^2 \right) \right. \\
& \quad \left. - z (x f'' + \frac{1}{3} z^2 f''') (4u_t + u_x^2) \right) - D_y \left(f u_y u_z - \frac{1}{2} z f' (u u_{xy} - u_x u_y) \right. \\
& \quad \left. - 2z (x f'' + \frac{1}{3} z^2 f''') u_y \right) - D_z \left(\frac{1}{2} f (2u_t u_x + \frac{1}{3} u_x^3 - u_y^2 + u_z^2) + z f' u_x u_z \right. \\
& \quad \left. + 2x f'' (u - z u_z) + 2z^2 f''' (u - \frac{1}{3} z u_z) \right) = 0, \tag{3.33}
\end{aligned}$$

where $f = f(t)$. Again, when $f(t) = 1$, conservation laws (3.32) and (3.33) reduce to (3.21) and (3.22), respectively.

3.3 The (2+1)-Dimensional Gardner Equation

The (2+1)-dimensional Gardner equation was derived by Konopelchenko and Dubrovsky [45] in an effort to find higher-dimensional generalizations of the original Gardner equation. The original (1+1)-dimensional Gardner equation,

$$u_t = -\frac{3}{2} \alpha u^2 u_x + 6\beta u u_x + u_{3x}, \tag{3.34}$$

is an integrable combination of the KdV and modified KdV equations. Using (3.34) as a starting point, Konopelchenko and Dubrovsky constructed Lax operators for a two-dimensional version, then applied the Lax commutativity condition which allows for integrability. The resulting equation is

$$u_t = -\frac{3}{2} \alpha^2 u^2 u_x + 6\beta u u_x + u_{3x} - 3\alpha u_x \partial_x^{-1} u_y + 3\partial_x^{-1} u_{2y}, \tag{3.35}$$

where, for example, $\partial_x^{-1} u_y = \int u_y dx$. Note that if $\alpha = 0$ in (3.35), the equation reduces to the Kadomtsev-Petviashvili (KP) equation (given in Chapter 8). Setting $\beta = 0$, produces a modified KP equation, and if $u_y = 0$, the (2+1)-dimensional Gardner equation (3.35) reduces to (3.34).

For simplicity, in all following conservation laws for the Gardner equation, let $v = \partial_x^{-1}u_y$. `ConservationLawsMD.m` found two conservation laws for (3.35) where terms are not multiplied by independent variables. These are

$$\mathbf{D}_t(u) + \mathbf{D}_x\left(\frac{1}{2}\alpha^2u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) - \mathbf{D}_y\left(\frac{3}{2}(\alpha u^2 + 2v)\right) = 0, \quad (3.36)$$

$$\begin{aligned} \mathbf{D}_t(u^2) + \mathbf{D}_x\left(\frac{3}{4}\alpha^2u^4 - 4\beta u^3 + 3\alpha u^2v + 3v^2 + u_x^2 - 2uu_{2x}\right) \\ - \mathbf{D}_y\left(u(\alpha u^2 + 6v)\right) = 0. \end{aligned} \quad (3.37)$$

Similar to the NTGF equation, all conservation laws containing explicit independent variables can be grouped into three sets, then each set can be generalized to a single equation. One set of conservation laws calculated by `ConservationLawsMD.m` contains

$$\begin{aligned} \mathbf{D}_t(tu) + \mathbf{D}_x\left(t\left(\frac{1}{2}\alpha^2u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) + yv\right) \\ - \mathbf{D}_y\left(\frac{3}{2}t(\alpha u^2 + 2v) + yu\right) = 0, \end{aligned} \quad (3.38)$$

$$\begin{aligned} \mathbf{D}_t(t^2u) + \mathbf{D}_x\left(t^2\left(\frac{1}{2}\alpha^2u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) + 2tyv\right) \\ - \mathbf{D}_y\left(\frac{3}{2}t^2(\alpha u^2 + 2v) + 2tyu\right) = 0, \end{aligned} \quad (3.39)$$

$$\begin{aligned} \mathbf{D}_t(t^3u) + \mathbf{D}_x\left(t^3\left(\frac{1}{2}\alpha^2u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) + 3t^2yv\right) \\ - \mathbf{D}_y\left(\frac{3}{2}t^3(\alpha u^2 + 2v) + 3t^2yu\right) = 0. \end{aligned} \quad (3.40)$$

All other conservation laws in this set have the same form as (3.40), but with higher powers of t . Like the NTGF equation, all factors that have the form t^n in the conservation laws can be replaced with an arbitrary function $f(t)$ and derivatives of f where needed. The general conservation law is

$$\begin{aligned} \mathbf{D}_t(fu) + \mathbf{D}_x\left(f\left(\frac{1}{2}\alpha^2u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) + f'yu\right) \\ - \mathbf{D}_y\left(\frac{3}{2}f(\alpha u^2 + 2v) + f'yu\right) = 0. \end{aligned} \quad (3.41)$$

Although `ConservationLawsMD.m` cannot compute densities with arbitrary functions, when the generalized density together with the PDE are submitted, the program verifies whether or not the given density is correct.

Two other conservation laws with an arbitrary function, $f(t)$ are

$$\begin{aligned} \mathbf{D}_t\left(fu^2 + \frac{2}{3\alpha}yf'u\right) + \mathbf{D}_x\left(f\left(\frac{3}{4}\alpha^2u^4 - 4\beta u^3 + 3\alpha u^2v + 3v^2 + u_x^2 - 2uu_{2x}\right) \right. \\ \left. + \frac{2}{3\alpha}yf'\left(\frac{1}{2}\alpha^2u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) + \frac{1}{\alpha}v(2xf' + \frac{1}{3}y^2f'')\right) \\ - \mathbf{D}_y\left(fu(\alpha u^2 + 6v) + \frac{1}{\alpha}yf'(\alpha u^2 + 2v) + \frac{1}{\alpha}u(2xf' + \frac{1}{3}y^2f'')\right) = 0, \end{aligned} \quad (3.42)$$

and

$$\begin{aligned}
& \mathbf{D}_t \left(u^2 (\beta f + \frac{\alpha}{6} y f') + \frac{1}{3} u (x f' + \frac{1}{6} y^2 f'') \right) + \mathbf{D}_x \left((\beta f + \frac{\alpha}{6} y f') (\frac{3}{4} \alpha^2 u^4 - 4 \beta u^3 \right. \\
& \quad \left. + 3 \alpha u^2 v + 3 v^2 + u_x^2 - 2 u u_{2x}) + \frac{1}{3} (x f' + \frac{1}{6} y^2 f'') (\frac{1}{2} \alpha^2 u^3 - 3 \beta u^2 + 3 \alpha u v - u_{2x}) \right. \\
& \quad \left. + \frac{1}{3} f' u_x + \frac{1}{3} y v (x f'' + \frac{1}{18} y^2 f''') \right) - \mathbf{D}_y \left(u (\beta f + \frac{\alpha}{6} y f') (\alpha u^2 + 6 v) \right. \\
& \quad \left. + \frac{1}{2} (x f' + \frac{1}{6} y^2 f'') (\alpha u^2 + 2 v) + \frac{1}{3} y u (x f'' + \frac{1}{18} y^2 f''') \right) = 0. \tag{3.43}
\end{aligned}$$

Setting $f = 1$ in (3.41) and (3.42), yields (3.36) and (3.37), respectively.

CHAPTER 4

TOOLS FOR THE COMPUTATION OF CONSERVATION LAWS: THE ZEROTH-EULER OPERATOR

Two tools from the calculus of variations play an important role in the computation of conservation laws. The first tool is the zeroth-Euler operator which provides a method to test a differential expression for exactness. The second tool is the homotopy operator, which will be introduced in Chapter 6.

The zeroth-Euler operator originates from the calculus of variations and is also known as the Euler-Lagrange operator or as the variational derivative. The zeroth-Euler operator has several uses in the construction and verification of conservation laws and is a key tool for the Exactness Theorem, which will be introduced later in this chapter. The Exactness Theorem provides a method for verifying whether a one-dimensional differential function is a total derivative or a multi-dimensional differential function is a total divergence. This chapter shows a derivation for the zeroth-Euler operator for scalar differential functions, $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$, defined on the jet space J^M , then gives a proof for the Exactness Theorem.

4.1 General Definitions

The zeroth-Euler operator acting on a scalar function f in multiple dimensions is defined as follows.

Definition 4.1. *The zeroth-Euler operator acting on a scalar differential function, $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$, is given by*

$$\mathcal{L}_{\mathbf{u}(\mathbf{x})}f = (\mathcal{L}_{u^1(\mathbf{x})}f, \mathcal{L}_{u^2(\mathbf{x})}f, \dots, \mathcal{L}_{u^j(\mathbf{x})}f, \dots, \mathcal{L}_{u^N(\mathbf{x})}f).^2 \quad (4.1)$$

²In [33, 35, 36] $\mathcal{L}_{\mathbf{u}(\mathbf{x})}$ is denoted as $\mathcal{L}_{\mathbf{u}(\mathbf{x})}^{(0)}$, where the superscript (0) is used to distinguish the zeroth-Euler operator from higher-Euler operators. Since higher-Euler operators are not used in this dissertation, the superscript has been dropped.

For $\mathbf{x} = (x^1, \dots, x^n)$,

$$\mathcal{L}_{u^j(\mathbf{x})}f = \sum_{k_1=0}^{M_1^j} \cdots \sum_{k_n=0}^{M_n^j} (-D_{x^1})^{k_1} \cdots (-D_{x^n})^{k_n} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j}, \quad (4.2)$$

where $j = 1, \dots, N$.

In practice, the computations to find conservation laws are done in one-, two-, and three-dimensions only. To make computations easier to follow, in subsequent chapters the zeroth-Euler operator will be shown with (x^1, x^2, x^3) replaced by (x, y, z) .

Definition 4.2. *The zeroth-Euler operator for the one-dimensional case where $f = f(x, \mathbf{u}^{(M)}(x))$ is*

$$\begin{aligned} \mathcal{L}_{u^j(x)}f &= \sum_{k=0}^{M_1^j} (-D_x)^k \frac{\partial f}{\partial u_{kx}^j} \\ &= \frac{\partial f}{\partial u^j} - D_x \frac{\partial f}{\partial u_x^j} + D_x^2 \frac{\partial f}{\partial u_{2x}^j} - D_x^3 \frac{\partial f}{\partial u_{3x}^j} + \cdots + (-D_x)^{M_1^j} \frac{\partial f}{\partial u_{M_1^j x}^j}, \end{aligned} \quad (4.3)$$

$j = 1, \dots, N$. For the two-dimensional case where $f = f(x, y, \mathbf{u}^{(M)}(x, y))$, the zeroth-Euler operator is

$$\mathcal{L}_{u^j(x,y)}f = \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}, \quad (4.4)$$

$j = 1, \dots, N$. Analogously, for the three-dimensional case where $f = f(x, y, z, \mathbf{u}^{(M)}(x, y, z))$, the operator is

$$\mathcal{L}_{u^j(x,y,z)}f = \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} \sum_{k_3=0}^{M_3^j} (-D_x)^{k_1} (-D_y)^{k_2} (-D_z)^{k_3} \frac{\partial f}{\partial u_{k_1 x k_2 y k_3 z}^j}, \quad (4.5)$$

$j = 1, \dots, N$.

When computing conservation laws, it is necessary to verify if a differential expression is exact, that is, whether or not a differential expression can be fully integrated.

Definition 4.3. *Let $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ be a differential function of order M . When $\mathbf{x} = x$, f is called exact if there exists a differential function $F(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$ such that $f = D_x F$. When $\mathbf{x} = (x^1, \dots, x^n)$, $n \geq 2$, f is exact if there exists a differential vector function $\mathbf{F}(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$ such that $f = \text{Div } \mathbf{F}$.*

The following theorem employs the zeroth-Euler operator to give a test to determine when a differential function is exact. This theorem is key to the computation of conservation laws.

Theorem 4.1 (Exactness Theorem). *A differential function $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ is exact if and only if $\mathcal{L}_{\mathbf{u}(\mathbf{x})}f \equiv \mathbf{0}$. Here, $\mathbf{0}$ is the vector $(0, 0, \dots, 0)$ which has N components matching the number of components of \mathbf{u} .*

The proof of the Exactness Theorem is given in Section 4.3. The proof requires the development of the Euler-Lagrange equation for multiple independent variables and multi-component differential functions, shown in the next section.

4.2 The Euler-Lagrange Equations

In the calculus of variations, for specific f , the integral

$$\int_{x_1}^{x_2} f(x, y(x), y'(x)) dx \quad (4.6)$$

may represent the time it takes to move along an arc, $y(x)$, from point (x_1, y_1) to point (x_2, y_2) . The Euler-Lagrange equation,

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \frac{\partial f}{\partial y'} = 0, \quad (4.7)$$

is a differential equation whose solution for $y(x)$ finds the time of quickest descent from (x_1, y_1) to (x_2, y_2) [73]. In other words, the solution $y(x)$ of (4.7) minimizes (4.6). A more general version of the Euler-Lagrange equation is a key element in the proof of the Exactness Theorem.

The Euler-Lagrange equation covering cases with multiple dependent variables and multiple derivatives, but one independent variable is given in [28]. In this dissertation, the Euler-Lagrange equation will be derived for n independent variables, $\mathbf{x} = (x^1, \dots, x^n)$, and N dependent variables, $\mathbf{u} = (u^1, \dots, u^N)$ and mixed derivatives to finite order, so that multi-dimensional, multi-component cases can be considered in the proof of the Exactness Theorem.

Let $\Omega \in \mathbb{R}^n$ be a simple connected region with boundary $\partial\Omega$. The multi-dimensional version of (4.6) is

$$\int_{\Omega} f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\mathbf{x}, \quad (4.8)$$

where f is defined on the jet space J^M , and $d\mathbf{x} = dx^1 \dots dx^n$. Now, the goal is to find the Euler-Lagrange equations whose solution $\mathbf{u}(\mathbf{x})$ extremizes (4.8).

The following lemma is needed to derive the appropriate Euler-Lagrange equation. The proof for the lemma is constructed in a similar manner to the proof given for the one variable case in [73].

Lemma 4.1. *Let $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ be a continuous function defined on J^M , and $\eta(\mathbf{x})$ be a continuous function on Ω such that $\eta(\mathbf{x}_1) = \eta(\mathbf{x}_2) = 0$ for \mathbf{x}_1 and \mathbf{x}_2 on $\partial\Omega$. If*

$$\int_{\Omega} \eta(\mathbf{x}) f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\mathbf{x} = 0 \quad (4.9)$$

for every possible choice of $\eta(\mathbf{x})$, then $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) \equiv 0$ on Ω .

PROOF: Choose $\mathbf{x}^* \in \Omega$ and assume that $f(\mathbf{x}^*, \mathbf{u}^{(M)}(\mathbf{x})) > 0$. Since f is continuous, there is a neighborhood around \mathbf{x}^* , $\Omega^* \in \Omega$, where $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) > 0$ when $\mathbf{x} \in \Omega^*$. With \mathbf{x}_1^* and \mathbf{x}_2^* on $\partial\Omega^*$, choose

$$\eta(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Omega \setminus \Omega^*, \\ (\mathbf{x}^* - \mathbf{x}_1^*)^T (\mathbf{x}^* - \mathbf{x}_1^*) (\mathbf{x}^* - \mathbf{x}_2^*)^T (\mathbf{x}^* - \mathbf{x}_2^*), & \mathbf{x} \in \Omega^*. \end{cases} \quad (4.10)$$

This choice for $\eta(\mathbf{x})$ is never negative and satisfies the conditions of Lemma 4.1. Next, by substituting (4.10) into the left-hand side of (4.9),

$$\begin{aligned} & \int_{\Omega} \eta(\mathbf{x}) f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\mathbf{x} \\ &= \int_{\Omega^*} (\mathbf{x}^* - \mathbf{x}_1^*)^T (\mathbf{x}^* - \mathbf{x}_1^*) (\mathbf{x}^* - \mathbf{x}_2^*)^T (\mathbf{x}^* - \mathbf{x}_2^*) f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\mathbf{x}. \end{aligned} \quad (4.11)$$

Clearly, (4.11) is greater than 0. This contradicts (4.9), so $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) \not\equiv 0$. A similar argument yields $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) \not\leq 0$. Therefore, $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ must be identically 0. ■

The next theorem gives the Euler-Lagrange equation and shows that when a $\mathbf{u}(\mathbf{x})$ extremizes (4.8), $\mathbf{u}(\mathbf{x})$ must be a solution to the Euler-Lagrange equation (4.8).

Theorem 4.2. *Let $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ be a differential function where $\mathbf{u}(\mathbf{x}) \in C^M(\Omega)$ satisfies given values $\mathbf{u}(\mathbf{x}_1)$ and $\mathbf{u}(\mathbf{x}_2)$ at \mathbf{x}_1 and \mathbf{x}_2 on $\partial\Omega$, no matter the choice for \mathbf{u} . If $\mathbf{u}(\mathbf{x})$ extremizes*

$$I = \int_{\Omega} f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\mathbf{x}, \quad (4.12)$$

then $\mathbf{u}(\mathbf{x})$ is a solution to the system of differential equations given by

$$\sum_{k_1=0}^{M_1^j} \cdots \sum_{k_n=0}^{M_n^j} (-D_{x_1})^{k_1} \cdots (-D_{x_n})^{k_n} \frac{\partial f}{\partial u_{k_1 x^1 \cdots k_n x^n}^j} = 0, \quad j = 1, \dots, N. \quad (4.13)$$

Equations (4.13) are the Euler-Lagrange equations for n independent variables and N dependent variables.

PROOF: Let $\mathbf{u}(\mathbf{x})$ be the function that extremizes (4.12). Define $U^j(\mathbf{x})$ so that

$$U^j(\mathbf{x}) = u^j(\mathbf{x}) + \epsilon \eta^j(\mathbf{x}), \quad j = 1, \dots, N, \quad (4.14)$$

where $\eta^j(\mathbf{x})$ is an arbitrary M th-order differentiable function, ϵ is a real parameter, and

$$\eta_{k_1 x^1 \cdots k_n x^n}^j(\mathbf{x}_1) = \eta_{k_1 x^1 \cdots k_n x^n}^j(\mathbf{x}_2) = 0, \quad (4.15)$$

where $0 \leq k_i \leq M_i^j$, $i = 1, \dots, n$. The functions $U^j(\mathbf{x})$ form a family of differential functions that connect the endpoints $(\mathbf{x}_1, u^j(\mathbf{x}_1))$ and $(\mathbf{x}_2, u^j(\mathbf{x}_2))$. Since $u^j(\mathbf{x})$ and $\eta^j(\mathbf{x})$ are differentiable up to order M , all mixed derivatives up to order M are defined for $U^j(\mathbf{x})$.

Next, form the integral

$$\begin{aligned} I(\epsilon) = \int_{\Omega} & f(x^1 \cdots, x^n, U^1, \cdots, U^N, U_{x^1}^1, \cdots, U_{x^1}^N, \\ & U_{x^2}^1, \cdots, U_{x^2}^N, \cdots, U_{x^n}^1, \cdots, U_{x^n}^N, U_{2x^1}^1, \cdots, U_{2x^1}^N \cdots, \\ & U_{k_1 x^1 \cdots k_n x^n}^1, \cdots, U_{k_1 x^1 \cdots k_n x^n}^N, \cdots, \\ & U_{M_1^1 x^1 \cdots M_n^1 x^n}^1, \cdots, U_{M_1^N x^1 \cdots M_n^N x^n}^N) d\mathbf{x}, \end{aligned} \quad (4.16)$$

where $\sum_{i=1}^n M_i^j \leq M$, the order of the coordinate for J^M . For fixed functions $\eta^j(\mathbf{x})$, (4.16) defines a function of the parameter ϵ . When $\epsilon = 0$, $U^j(\mathbf{x}) = u^j(\mathbf{x})$ are the components of the extremizing function for (4.12). By setting $\epsilon = 0$, (4.16) reduces to (4.12), so $I(\epsilon)$ is extremized with respect to ϵ when $\epsilon = 0$. Using standard calculus, this requires $\frac{dI}{d\epsilon}(0) = 0$.

Now, use the chain rule to differentiate $I(\epsilon)$,

$$\begin{aligned}
\frac{dI}{d\epsilon}(\epsilon) &= \int_{\Omega} \sum_{j=1}^N \left(\frac{\partial f}{\partial U^j} \frac{\partial U^j}{\partial \epsilon} + \frac{\partial f}{\partial U_{x^1}^j} \frac{\partial U_{x^1}^j}{\partial \epsilon} + \cdots + \frac{\partial f}{\partial U_{k_1 x^1 \dots k_n x^n}^j} \frac{\partial U_{k_1 x^1 \dots k_n x^n}^j}{\partial \epsilon} + \cdots \right. \\
&\quad \left. + \frac{\partial f}{\partial U_{M_1^j x^1 \dots M_n^j x^n}^j} \frac{\partial U_{M_1^j x^1 \dots M_n^j x^n}^j}{\partial \epsilon} \right) d\mathbf{x} \\
&= \int_{\Omega} \sum_{j=1}^N \left(\eta^j \frac{\partial f}{\partial U^j} + \eta_{x^1}^j \frac{\partial f}{\partial U_{x^1}^j} + \cdots + \eta_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial U_{k_1 x^1 \dots k_n x^n}^j} + \cdots \right. \\
&\quad \left. + \eta_{M_1^j x^1 \dots M_n^j x^n}^j \frac{\partial f}{\partial U_{M_1^j x^1 \dots M_n^j x^n}^j} \right) d\mathbf{x}. \tag{4.17}
\end{aligned}$$

Hence, $\frac{dI}{d\epsilon}(0) = 0$ is equivalent to

$$\begin{aligned}
\frac{dI}{d\epsilon}(0) &= \int_{\Omega} \sum_{j=1}^N \left(\eta^j \frac{\partial f}{\partial u^j} + \eta_{x^1}^j \frac{\partial f}{\partial u_{x^1}^j} + \cdots + \eta_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + \cdots \right. \\
&\quad \left. + \eta_{M_1^j x^1 \dots M_n^j x^n}^j \frac{\partial f}{\partial u_{M_1^j x^1 \dots M_n^j x^n}^j} \right) d\mathbf{x} \\
&= 0. \tag{4.18}
\end{aligned}$$

To split the sum over j in (4.18), for each j , $1 \leq j \leq N$, choose η such that η^j is arbitrary, but satisfies (4.15) and $\eta^i = 0$ when $i \neq j$. Thus, $\eta = (0, \dots, 0, \eta^j, 0, \dots, 0)$.

Then (4.18) can be written as the system

$$\frac{dI}{d\epsilon}(0) = \int_{\Omega} \sum_{k_1=0}^{M_1^j} \cdots \sum_{k_n=0}^{M_n^j} \eta_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} d\mathbf{x} = 0, \quad j = 1, \dots, N. \tag{4.19}$$

Next, it is necessary to repeatedly apply the divergence theorem. The manner in which the divergence theorem is applied depends on the order of f with respect to each independent variable, M_1^j, \dots, M_n^j . Details for the case where $f = f(x, y, u^{(2)}(x, y))$ are shown in Example 4.1 immediately following this proof. With repeated applications of the divergence theorem, (4.19) becomes

$$\int_{\Omega} \eta^j \left(\sum_{k_1=0}^{M_1^j} \cdots \sum_{k_n=0}^{M_n^j} (-D_{x^1})^{k_1} \cdots (-D_{x^n})^{k_n} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \right) d\mathbf{x} = 0 \quad j = 1, \dots, N. \tag{4.20}$$

Since the integral in (4.20) must vanish for all choices of η^j , application of Lemma 4.1 produces the system of Euler-Lagrange equations (4.13). \blacksquare

The following example demonstrates the repeated application of the divergence theorem needed to transform (4.19) to (4.20) for a simple case where $u^1 = u$ and $f = f(x, y, u^{(2)}(x, y))$.

Example 4.1. *To set the stage, let $\mathbf{F} = (F^1, F^2)$. Then*

$$\iint_{\Omega} (\mathbf{D}_x(\eta F^1) + \mathbf{D}_y(\eta F^2)) dydx = \iint_{\Omega} (\eta(\mathbf{D}_x F^1 + \mathbf{D}_y F^2) + \eta_x F^1 + \eta_y F^2) dydx. \quad (4.21)$$

Also, by Green's theorem³ [73],

$$\iint_{\Omega} (\mathbf{D}_x(\eta F^1) + \mathbf{D}_y(\eta F^2)) dydx = \int_{\partial\Omega} \eta(F^1 n^1 + F^2 n^2) dS, \quad (4.22)$$

where $\mathbf{n} = (n^1, n^2)$ is a vector normal to the boundary, $\partial\Omega$. By combining (4.21) and (4.22), the identity

$$\iint_{\Omega} (\eta_x F^1 + \eta_y F^2) dydx = \int_{\partial\Omega} \eta(F^1 n^1 + F^2 n^2) dS - \iint_{\Omega} \eta(\mathbf{D}_x F^1 + \mathbf{D}_y F^2) dydx \quad (4.23)$$

is established.

For the case with $f = f(x, y, u^{(2)}(x, y))$,

$$I(\epsilon) = \int_{\Omega} f(x, y, U, U_x, U_y, U_{2x}, U_{xy}, U_{2y}) dydx, \quad (4.24)$$

where $U(x, y) = u(x, y) + \epsilon\eta(x, y)$ as in (4.14), and $\eta(x, y)$ is subject to the conditions in (4.15). Following the steps (4.17) - (4.19),

$$\begin{aligned} \frac{dI}{d\epsilon}(0) &= \iint_{\Omega} \left(\eta \frac{\partial f}{\partial u} + \eta_x \frac{\partial f}{\partial u_x} + \eta_y \frac{\partial f}{\partial u_y} + \eta_{2x} \frac{\partial f}{\partial u_{2x}} + \eta_{xy} \frac{\partial f}{\partial u_{xy}} + \eta_{2y} \frac{\partial f}{\partial u_{2y}} \right) dydx \\ &= 0. \end{aligned} \quad (4.25)$$

First, split the integral in (4.25), then use (4.23) so that

$$\begin{aligned} &\iint_{\Omega} \eta \frac{\partial f}{\partial u} dydx + \iint_{\Omega} \left(\eta_x \frac{\partial f}{\partial u_x} + \eta_y \frac{\partial f}{\partial u_y} \right) dydx \\ &\quad + \iint_{\Omega} \left(\eta_{2x} \frac{\partial f}{\partial u_{2x}} + \eta_{xy} \frac{\partial f}{\partial u_{xy}} + \eta_{2y} \frac{\partial f}{\partial u_{2y}} \right) dydx \end{aligned}$$

³The multi-dimensional version of Green's theorem is called the divergence theorem. See, for example, [19].

$$\begin{aligned}
&= \iint_{\Omega} \eta \frac{\partial f}{\partial u} dydx + \int_{\partial\Omega} \eta \left(\frac{\partial f}{\partial u_x} n^1 + \frac{\partial f}{\partial u_y} n^2 \right) dS - \iint_{\Omega} \eta \left(D_x \frac{\partial f}{\partial u_x} + D_y \frac{\partial f}{\partial u_y} \right) dydx \\
&\quad + \int_{\partial\Omega} \left(\eta_x \frac{\partial f}{\partial u_{2x}} n^1 + \eta_y \left(\frac{\partial f}{\partial u_{xy}} + \frac{\partial f}{\partial u_{2y}} \right) n^2 \right) dS \\
&\quad - \iint_{\Omega} \left(\eta_x D_x \frac{\partial f}{\partial u_{2x}} + \eta_y \left(D_x \frac{\partial f}{\partial u_{xy}} + D_y \frac{\partial f}{\partial u_{2y}} \right) \right) dydx \\
&= \iint_{\Omega} \eta \frac{\partial f}{\partial u} dydx - \iint_{\Omega} \eta \left(D_x \frac{\partial f}{\partial u_x} + D_y \frac{\partial f}{\partial u_y} \right) dydx \\
&\quad - \iint_{\Omega} \left(\eta_x D_x \frac{\partial f}{\partial u_{2x}} + \eta_y \left(D_x \frac{\partial f}{\partial u_{xy}} + D_y \frac{\partial f}{\partial u_{2y}} \right) \right) dydx, \tag{4.26}
\end{aligned}$$

since arbitrary η and derivatives of η vanish on $\partial\Omega$. Again, use (4.23) on the third integral in (4.26) to get

$$\begin{aligned}
&\iint_{\Omega} \eta \left(\frac{\partial f}{\partial u} - D_x \frac{\partial f}{\partial u_x} - D_y \frac{\partial f}{\partial u_y} \right) dydx - \left[\int_{\partial\Omega} \eta \left(D_x \left(\frac{\partial f}{\partial u_{2x}} + \frac{\partial f}{\partial u_{xy}} \right) n^1 + D_y \frac{\partial f}{\partial u_{2y}} n^2 \right) dS \right. \\
&\quad \left. - \iint_{\Omega} \eta \left(D_x^2 \frac{\partial f}{\partial u_{2x}} + D_x D_y \frac{\partial f}{\partial u_{xy}} + D_y^2 \frac{\partial f}{\partial u_{2y}} \right) dydx \right] \\
&= \iint_{\Omega} \eta \left(\frac{\partial f}{\partial u} - D_x \frac{\partial f}{\partial u_x} - D_y \frac{\partial f}{\partial u_y} + D_x^2 \frac{\partial f}{\partial u_{2x}} + D_x D_y \frac{\partial f}{\partial u_{xy}} + D_y^2 \frac{\partial f}{\partial u_{2y}} \right) dydx \\
&= \iint_{\Omega} \eta \sum_{k_1=0}^2 \sum_{k_2=0}^2 (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}} dydx. \tag{4.27}
\end{aligned}$$

Note that (4.27) is the same as (4.20) with a double sum and with $M_1^1 = 2$ and $M_2^1 = 2$.

4.3 A Proof for the Exactness Theorem

With the derivation of the Euler-Lagrange system finished, the proof for the Exactness Theorem can be completed.

PROOF OF THEOREM 4.1: Following the conditions for Theorem 4.2, let $f = f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ be a differential function where $\mathbf{u}(\mathbf{x}) \in C^M(\Omega)$ satisfies given values $\mathbf{u}(\mathbf{x}_1)$ and $\mathbf{u}(\mathbf{x}_2)$ at \mathbf{x}_1 and \mathbf{x}_2 on $\partial\Omega$, no matter the choice for \mathbf{u} .

First, assume that f is exact. In other words, there is $\mathbf{F} = \mathbf{F}(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$ such that $f = \text{Div } \mathbf{F}$. By the divergence theorem for multi-dimensions [19],

$$\int_{\Omega} f d\mathbf{x} = \int_{\Omega} \text{Div } \mathbf{F} d\mathbf{x} = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS, \tag{4.28}$$

where \mathbf{n} is a vector normal to the boundary, $\partial\Omega$. The integral $\int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS$ describes how \mathbf{F} behaves at $\partial\Omega$, thus is dependent only of the values of \mathbf{u} at $\partial\Omega$. Since the values of \mathbf{u} are prescribed at $\partial\Omega$, the integral $\int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS = C$, where C is constant.

By (4.28), $\int_{\Omega} f \, d\mathbf{x} = C$. This means that $\int_{\Omega} f \, d\mathbf{x}$ is independent of the choice for \mathbf{u} . Therefore, every \mathbf{u} extremizes $\int_{\Omega} f \, d\mathbf{x}$ and the Euler-Lagrange equations (4.13) are satisfied identically.

Conversely, assume that (4.13) holds no matter the choice for \mathbf{u} , that is, $\mathcal{L}_{\mathbf{u}(\mathbf{x})} f \equiv 0$. Let $f = f(\mathbf{x}, \epsilon \mathbf{u}^{(M)}(\mathbf{x}))$ [56], where u^j is replaced by ϵu^j , u_x^j is replaced by ϵu_x^j , and so on. Then

$$\begin{aligned} \frac{df}{d\epsilon} &= \sum_{j=1}^N \left(u^j \frac{\partial f}{\partial u^j} + u_{x^1}^j \frac{\partial f}{\partial u_{x^1}^j} + \cdots + u_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + u_{M_1^j x^1 \dots M_n^j x^n}^j \frac{\partial f}{\partial u_{M_1^j x^1 \dots M_n^j x^n}^j} \right) \\ &= \sum_{j=1}^N \sum_{k_1=0}^{M_1^j} \cdots \sum_{k_n=0}^{M_n^j} u_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j}. \end{aligned} \quad (4.29)$$

Repeatedly integrate each term in (4.29) by parts. For example, integrating the term

$$u_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \text{ gives}$$

$$\begin{aligned} & u_{k_1 x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \\ &= -u_{(k_1-1)x^1 \dots k_n x^n}^j D_{x^1} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + D_{x^1} \left(u_{(k_1-1)x^1 \dots k_n x^n}^j \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \right) \\ &= u_{(k_1-2)x^1 \dots k_n x^n}^j (D_{x^1})^2 \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + D_{x^1} \left(\left(u_{(k_1-1)x^1 \dots k_n x^n}^j \right. \right. \\ &\quad \left. \left. - u_{(k_1-2)x^1 \dots k_n x^n}^j D_{x^1} \right) \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \right) \\ &= \cdots \\ &= u_{k_2 x^2 \dots k_n x^n} (-D_{x^1})^{k_1} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + D_{x^1} \underbrace{\left(\sum_{i=1}^{k_1} u_{(k_1-i)x^1 \dots k_n x^n} (-D_{x^1})^{i-1} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \right)}_{P^1} \\ &= -u_{(k_2-1)x^2 \dots k_n x^n} (-D_{x^1})^{k_1} (D_{x^2}) \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + D_{x^1} P^1 \\ &\quad + D_{x^2} \left(u_{(k_2-1)x^2 \dots k_n x^n} (-D_{x^1})^{k_1} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \right) \\ &= \cdots \\ &= u_{k_3 x^3 \dots k_n x^n} (-D_{x^1})^{k_1} (-D_{x^2})^{k_2} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} + D_{x^1} P^1 \\ &\quad + D_{x^2} \underbrace{\left(\sum_{i=1}^{k_2} u_{(k_2-i)x^1 \dots k_n x^n} (-D_{x^1})^{k_1-1} (-D_{x^2})^{i-1} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}^j} \right)}_{P^2} \end{aligned}$$

$$\begin{aligned}
&= u_{k_3 x^3 \dots k_n x^n} (-D_{x^1})^{k_1} (-D_{x^2})^{k_2} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}} + D_{x^1} (P^1) + D_{x^2} (P^2) \\
&= \dots \\
&= u^j (-D_{x^1})^{k_1} \dots (-D_{x^n})^{k_n} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}} + D_{x^1} (P^1) + \dots + D_{x^n} (P^n).
\end{aligned}$$

After integrating all terms by parts and rearranging, (4.29) becomes

$$\begin{aligned}
\frac{df}{d\epsilon} &= \sum_{j=1}^N u^j \sum_{k_1=0}^{M_1^j} \dots \sum_{k_n=0}^{M_n^j} (-D_{x^1})^{k_1} \dots (-D_{x^n})^{k_n} \frac{\partial f}{\partial u_{k_1 x^1 \dots k_n x^n}} + \text{Div } \tilde{\mathbf{P}} \\
&= \sum_{j=1}^N u^j \mathcal{L}_{u^j(\mathbf{x})} f + \text{Div } \tilde{\mathbf{P}} \\
&= \mathbf{u} \cdot \mathcal{L}_{\mathbf{u}(\mathbf{x})} f + \text{Div } \tilde{\mathbf{P}}, \tag{4.30}
\end{aligned}$$

where $\tilde{\mathbf{P}}$ is the sum of all $\mathbf{P} = (P^1, \dots, P^n)$. Since $\mathcal{L}_{\mathbf{u}(\mathbf{x})} f \equiv 0$,

$$\frac{df}{d\epsilon}(\mathbf{x}, \epsilon \mathbf{u}^{(M)}(\mathbf{x})) = \text{Div } \tilde{\mathbf{P}}. \tag{4.31}$$

Integrating both sides with respect to ϵ from $\epsilon = \epsilon_0$ to $\epsilon = 1$ gives

$$\begin{aligned}
f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) - f(\mathbf{x}, \epsilon_0 \mathbf{u}^{(M)}(\mathbf{x})) &= \int_{\epsilon_0}^1 \text{Div } \tilde{\mathbf{P}} d\epsilon \\
&= \text{Div } \tilde{\mathbf{P}} - \text{Div } \epsilon_0 \tilde{\mathbf{P}}, \tag{4.32}
\end{aligned}$$

since the divergence is independent of ϵ . The lower limit, ϵ_0 , is chosen so that all $\epsilon_0 \mathbf{u}$ and derivatives of $\epsilon_0 \mathbf{u}$ vanish, leaving $f(\mathbf{x}, \epsilon_0 \mathbf{u}^{(M)}(\mathbf{x})) = f(\mathbf{x}, \mathbf{0})$. For polynomial functions and most other functions, $\epsilon_0 = 0$. For rational functions, it may be necessary to choose $\epsilon_0 = \infty$. Since explicit functions of \mathbf{x} are not considered, $f(\mathbf{x}, \mathbf{0}) = 0$. Therefore $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ is a divergence. \blacksquare

The zeroth-Euler operator for dependent variable u^j in (4.2) is the left hand side of (4.13). Setting $n = 1, 2, 3$ produces (4.3), (4.4), and (4.5), respectively.

The following example shows an application of the Exactness Theorem.

Example 4.2. Let $\mathbf{u} = (u^1, u^2) = (u, v)$, and let

$$f(x, y, \mathbf{u}^{(3)}(x, y)) = v_y + xv_{xy} + u_x v_{xy} + uv_{2xy} + 2yv_x + y^2 v_{xy} - v_y u_{2x} - vv_{2xy}.$$

Since there are two independent variables, apply the zeroth-Euler operator (4.4) for each dependent variable to get

$$\begin{aligned}
\mathcal{L}_{u(x,y)}f &= \frac{\partial f}{\partial u} - D_x \frac{\partial f}{\partial u_x} + D_y^2 \frac{\partial f}{\partial u_{2x}} - D_x^2 D_y \frac{\partial f}{\partial u_{2xy}} \\
&= v_{2xy} - D_x(v_{xy}) + D_x^2(-v_y) - D_x^2 D_y(-v) \\
&= v_{2xy} - v_{2xy} - v_{2xy} + v_{2xy} \\
&= 0,
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}_{v(x,y)}f &= \frac{\partial f}{\partial v} - D_x \frac{\partial f}{\partial v_x} - D_y \frac{\partial f}{\partial v_y} + D_x D_y \frac{\partial f}{\partial v_{xy}} - D_x^2 D_y \frac{\partial f}{\partial v_{2xy}} \\
&= -u_{2xy} - D_x(2y) - D_y(1 - u_{2x}) + D_x D_y(x + u_x + y^2) - D_x^2 D_y(u) \\
&= -u_{2xy} + u_{2xy} + u_{2xy} - u_{2xy} \\
&= 0.
\end{aligned}$$

Since $\mathcal{L}_{\mathbf{u}(x,y)} = (0, 0)$, f is the divergence of some function \mathbf{F} . Indeed, it is easy to verify that $f = \text{Div} \mathbf{F}$ for $\mathbf{F} = (F^1, F^2) = (xv_y + uv_{xy}, y^2v_x - vu_{2x})$.

Although $\mathbf{F} = (F^1, F^2) = (xv_y + uv_{xy}, y^2v_x - vu_{2x})$ is given in Example 4.2, infinitely many choices exist for \mathbf{F} . Indeed, any addition of “curl” terms to \mathbf{F} will produce an identical divergence. Thus, $f = \text{Div} \mathbf{G} = v_y + xv_{xy} + u_x v_{xy} + uv_{2xy} + 2yv_x + y^2v_{xy} - v_y u_{2x} - vu_{2xy}$ where $\mathbf{G} = (F^1 + D_y \theta, F^2 - D_x \theta)$ with θ any differential expression.

CHAPTER 5

INDEPENDENCE OF CONSERVED DENSITIES

The algorithms used in `ConservationLawsMD.m` require that the program first compute a density for a conservation law, then use the continuity equation (3.5) and that density to compute the flux. For each new density generated, several questions must be addressed before the density is reported. Does the density lead to a trivial conservation law? Is the density written in the simplest possible form? Does the density lead to a *new* conservation law, or to previously calculated conservation laws in disguise? Is the density a linear combination of smaller densities?

Each section in this chapter addresses these questions. A density will be independent and in simplest form when it passes the tests described in each section. Since two densities that appear to be different may lead to the same conservation law, it is impossible to determine their independence by straightforward inspection. The verification of densities calls for a systematic approach and sophisticated tools. One method requires constructing a characteristic (differential function) for the conservation law. A characteristic is uniquely determined for each independent conservation law [56, p. 266]. If two characteristics can be shown to be independent, the corresponding conservation laws are independent. A second more algorithmic approach that uses the zeroth-Euler operator is implemented in `ConservationLawsMD.m` and has its own stand-alone code in `IndependenceTest.m`.

The tests described in this chapter are part of the `ConservationLawsMD.m` code, so that the code will produce independent densities in the simplest form. However, when an interchange of independent variables is used to transform a PDE into an evolution equation (3.1), or when a user wishes to verify a density by providing it to the program, some additional analysis is required. An accompanying program, `IndependenceTest.m`, also written in *Mathematica* language, performs the tests described in this chapter on a set of densities obtained from any source. The code `IndependenceTest.m` will take a list of densities for a PDE, order them by increasing complexity, then compare the densities with one-another. The program can both simplify a density and establish whether or

not it is independent of other densities in the list.

5.1 Terms that are Divergences or Divergence-Equivalent

When computing a density, `ConservationLawsMD.m` uses the scaling symmetry of the PDE to construct a candidate density which may contain a large number of terms. Many terms can be removed from the candidate immediately with use of a simple algorithm. If these terms are part of the conservation law, they will be found in the flux at the time the flux is calculated. Terms that are divergences (total derivatives in the one-dimensional case) are the first to be removed.

Definition 5.1. *A term or expression f is a divergence when there exists a vector \mathbf{F} such that $f = \text{Div } \mathbf{F}$. In the one-dimensional case, f is a total derivative when there exists a function F such that $f = D_x F$. Note that $D_x f$ is the same as a one-dimensional divergence, so from this point onward, the term “divergence” will include one-dimensional cases that are total derivatives.*

To find terms that are divergences, the zeroth-Euler operator is applied to the density, term by term. By Theorem 4.1, a divergence is any term whose image under the zeroth-Euler operator is $\mathbf{0}$. Terms that are divergences should not be included in the density.

Example 5.1. *The scaling symmetry of the ZK equation (3.7) determines that a candidate density will have the terms $\{u^2, u_{2x}, u_{xy}, u_{2y}\}$. Obviously, the term u_{2x} is a divergence since $u_{2x} = \text{Div}(u_x, 0)$. The zeroth-Euler operator identifies the divergence easily,*

$$\mathcal{L}_{u(x,y)}(u_{2x}) = (-D_x)^2 \frac{\partial u_{2x}}{\partial u_{2x}} = 0.$$

The terms u_{xy} and u_{2y} are also divergences. All three divergences are removed leaving a single term in the candidate density list, namely $\{u^2\}$.

As shown in Example 5.1, only single terms are considered. If a single term is a divergence, it is removed.

Secondly, if two terms are divergence-equivalent, one of the terms must be removed from the density.

Definition 5.2. *Two or more terms are divergence-equivalent when a linear combination of the terms is a divergence.*

For example, $u_x u_y$ and $u u_{xy}$ are divergence-equivalent since $u_x u_y + u u_{xy} = \text{Div}(u u_y, 0)$. The zeroth-Euler operator is also used to identify divergence-equivalent terms by applying the following theorem.

Theorem 5.1. *When the zeroth-Euler operator is applied to a set of divergence-equivalent terms, their images under the zeroth-Euler operator are linearly dependent.*

PROOF: Let f_1, \dots, f_n be divergence-equivalent terms in a differential expression. By Definition 5.2,

$$p_1 f_1 + \dots + p_n f_n = \text{Div } \mathbf{G} \quad (5.1)$$

for some vector \mathbf{G} . Note that p_1, \dots, p_n cannot be zero. Applying the zeroth-Euler operator to both sides of (5.1) gives

$$\begin{aligned} \mathcal{L}_{\mathbf{u}(\mathbf{x})}(p_1 f_1 + \dots + p_n f_n) &= \mathcal{L}_{\mathbf{u}(\mathbf{x})} \text{Div } G, \\ p_1 \mathcal{L}_{\mathbf{u}(\mathbf{x})} f_1 + \dots + p_n \mathcal{L}_{\mathbf{u}(\mathbf{x})} f_n &= 0. \end{aligned}$$

Since p_1, \dots, p_n cannot be zero, the images of f_1, \dots, f_n under the zeroth-Euler operator are linearly dependent. ■

Once the divergence-equivalent terms have been identified by the zeroth-Euler operator, all but one of these terms are removed from the density. As the next example shows, the identification of divergence-equivalent terms reduces to a linear algebra problem.

Example 5.2. *A partial list of terms for a candidate density is*

$$\{u^6, u^2 v^2, u_{3x} v, u_{2x} v_x, u_x v_{2x}, u v_{3x}, u_{xy} v, u_x v_y, u_y v_x, u v_{xy}\}. \quad (5.2)$$

Apply the zeroth-Euler operator to each term to get

$$\begin{aligned} (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u^6) &= (6u^5, 0), & (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u^2 v^2) &= (2u v^2, 2u^2 v), \\ (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u_{3x} v) &= (v_{3x}, -u_{3x}), & (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u_{2x} v_x) &= (-v_{3x}, u_{3x}), \\ (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u_x v_{2x}) &= (-v_{3x}, u_{3x}), & (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u v_{3x}) &= (v_{3x}, -u_{3x}), \\ (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u_{xy} v) &= (v_{xy}, u_{xy}), & (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u_x v_y) &= (-v_{xy}, -u_{xy}), \\ (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u_y v_x) &= (-v_{xy}, -u_{xy}), & (\mathcal{L}_{u(x,y)}, \mathcal{L}_{v(x,y)})(u v_{xy}) &= (v_{xy}, u_{xy}). \end{aligned}$$

A linear combination of the components of the images under the zeroth-Euler operator

with undetermined coefficients p_1, \dots, p_{10} is

$$\begin{aligned}
& p_1(6u^5, 0) + p_2(2uv^2, 2u^2v) + p_3(v_{3x}, -u_{3x}) + p_4(-v_{3x}, u_{3x}) + p_5(-v_{3x}, u_{3x}) \\
& + p_6(v_{3x}, -u_{3x}) + p_7(v_{xy}, u_{xy}) + p_8(-v_{xy}, -u_{xy}) + p_9(-v_{xy}, -u_{xy}) \\
& + p_{10}(v_{xy}, u_{xy}) = (0, 0).
\end{aligned} \tag{5.3}$$

After collecting like terms,

$$\begin{pmatrix} 6p_1u^5 + 2p_2u^2v + (p_3 - p_4 - p_5 + p_6)v_{3x} + (p_7 - p_8 + p_9 - p_{10})v_{xy} \\ 2p_2uv^2 + (-p_3 + p_4 + p_5 - p_6)u_{3x} + (p_7 - p_8 + p_9 - p_{10})u_{xy} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{5.4}$$

Clearly, $p_1 = p_2 = 0$, $p_3 = p_4 + p_5 - p_6$ and $p_7 = p_8 + p_9 - p_{10}$ with p_4, p_5, p_6, p_8, p_9 , and p_{10} arbitrary satisfy (5.4). Since the components with coefficients p_3, p_4, p_5 , and p_6 form a linearly dependent set, the terms $u_{3x}v, u_{2x}v_x, u_xv_{2x}$, and uv_{3x} are divergence-equivalent. The components with coefficients p_7, p_8, p_9 , and p_{10} form another linearly dependent set, so the terms $u_{xy}v, u_xv_y, u_yv_x$, and uv_{xy} are a second set of divergence-equivalent terms. Six terms can be removed from the list (5.2), leaving $\{u^6, u^2v^2, u_xv_{2x}, u_xv_y\}$. Note that in this case, there are several choices for the final list. The lowest order terms are chosen to remain in the list.

The techniques used in Examples 5.1 and 5.2 to identify terms that are either divergences or divergence-equivalent are summarized in the following algorithm. The algorithm is used by `ConservationLawsMD.m` to analyze a list of terms for a candidate density and by `IndependenceTest.m` to analyze densities given by the user or previously computed.

Algorithm 5.1 (An algorithm for removing divergences and divergence terms.). *The algorithm takes a differential expression that has been established to be a density for a PDE, or a list of terms for a candidate density.*

- i. *Expand the density, then place the terms into a list D . Attach an undetermined coefficient to each term in the list.*
- ii. *Form list E by applying the zeroth-Euler operator to each term in list D .*
- iii. *All zeros in list E correspond to divergences in D . Report the divergences, then remove them from D .*

- iv. Add the terms in list E and collect like terms.
- v. Find the values for the undetermined coefficients that make the linear combination of list terms in E equal zero.
- vi. Any undetermined coefficients whose solutions are in terms of another identify terms that are divergence-equivalent. Report all divergence-equivalent terms.

5.2 Trivial Conservation Laws

A trivial conservation law will fit one of two possible cases. The first case occurs when the density ρ and the flux \mathbf{J} vanish independently for solutions of the given PDE [56]. Trivial conservation laws for evolution equations can be avoided by requiring that the density and flux be written in terms of the independent space variables, \mathbf{u} , and the spatial derivatives of \mathbf{u} . However, conservation laws generated for PDEs that are not evolution equations require some evaluation.

Example 5.3. A possible conservation law for the NTGF equation (3.18) is

$$D_t(zu_x) + D_x(\frac{1}{4}zuiu_x^2) - D_y(\frac{1}{2}zu_y) + D_z(\frac{1}{2}(u - zu_z)) = 0.$$

This conservation law is trivial since the density zu_x is a divergence and can be moved to the flux, that is $D_t(zu_x) = D_x(zu_t)$. When this is done, the conservation law becomes

$$D_t(0) + D_x(zu_t + \frac{1}{4}zuiu_x^2) - D_y(\frac{1}{2}zu_y) + D_z(\frac{1}{2}(u - zu_z)) = 0.$$

The density and flux vanish independently for solutions of (8.1).

`ConservationLawsMD.m` will report this type of trivial conservation law and state that it is trivial. `IndependenceTest.m` will state that the density is a divergence.

The second case occurs when the conservation law (3.5) holds identically for \mathbf{u} , without \mathbf{u} being a solution of the PDE [56]. A simple example is

$$D_t(u_{2x} - u_{2y}) + D_x(-u_{tx}) + D_y(u_{ty}) = 0,$$

which clearly holds for any arbitrary smooth function $u(t, x, y)$. A linear combination of trivial conservation laws of both kinds is also trivial. `ConservationLawsMD.m` automatically rejects this type of trivial conservation law. Since `IndependenceTest.m` works with densities only, it cannot identify this type of trivial conservation law.

5.3 Equivalent Densities

Two conservation laws are equivalent when their difference is a trivial conservation law. Thus, two densities, ρ_1 and ρ_2 , are equivalent when their difference is a divergence, that is when $\rho_1 - \rho_2 = \text{Div } \mathbf{Q}$ for some vector $\mathbf{Q}(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$. Two equivalent densities represent the same conservation law.

Example 5.4. *The conservation law for the KdV equation (3.8),*

$$\mathbf{D}_t(u^3 - 3u_x^2) + \mathbf{D}_x\left(\frac{3}{4}u^4 - buu_x^2 + 3u^2u_{2x} + 3u_{2x}^2 - 6u_xu_{3x}\right) = 0, \quad (5.5)$$

corresponds to Boussinesq's moment of instability [30]. The density can be given in two ways,

$$\rho_1 = u^3 - 3u_x^2,$$

as given in (5.5), or

$$\rho_2 = u^3 + 3uu_{2x}.$$

Both densities are equivalent since $\rho_1 - \rho_2 = -3(u_x^2 + uu_{2x}) = \mathbf{D}_x(-3uu_x)$.

Once again, the zeroth-Euler operator is a useful identifying tool. Indeed, if two densities, ρ_1 and ρ_2 , are equivalent, then by Theorem 4.1,

$$\mathcal{L}_{\mathbf{u}(\mathbf{x})}(\rho_2 - \rho_1) = \mathbf{0}.$$

The test can also be used to compare densities obtained from different sources. The following example compares a density reported by `ConservationLawsMD.m` to one found in literature.

Example 5.5. *Conservation law (3.23) for the NTGF equation, (3.18) has the density*

$$\rho_1 = -\frac{1}{3}uu_xu_{2x} - \frac{1}{2}u_y^2 + \frac{1}{2}uu_{2z}$$

which was obtained by `ConservationLawsMD.m`. Khamitova [44] reports a density,

$$\rho_2 = \frac{1}{6}u_x^3 - \frac{1}{2}u_y^2 - \frac{1}{2}u_z^2.$$

ρ_1 and ρ_2 have one identical term, which suggests that the densities might be equivalent. To confirm this, the zeroth-Euler operator is applied to their difference,

$$\begin{aligned}
& \mathcal{L}_{u(\mathbf{x})}(\rho_2 - \rho_1) \\
&= \mathcal{L}_{u(\mathbf{x})}\left(\frac{1}{6}u_x^3 + \frac{1}{3}uu_xu_{2x} - \frac{1}{2}u_z^2 - \frac{1}{2}uu_{2z}\right) \\
&= -\frac{1}{6}\mathbf{D}_x(3u_x^2) + \frac{1}{3}u_xu_{2x} - \frac{1}{3}\mathbf{D}_x(uu_{2x}) + \frac{1}{3}\mathbf{D}_x^2(uu_x) + \frac{1}{2}\mathbf{D}_z(2u_z) - \frac{1}{2}\mathbf{D}_z^2(u) - \frac{1}{2}u_{2z} \\
&= 0.
\end{aligned}$$

Hence, $\rho_1 - \rho_2$ is a divergence, so the two densities are equivalent. Note that there are two sets of divergence-equivalent terms in the difference $\rho_2 - \rho_1$.

A second form of equivalence occurs when an independent variable can be factored from both $\mathbf{D}_t(\rho)$ and $\text{Div } \mathbf{J}$ in (3.5) and moved to the right hand side, producing an equivalent conservation law. This case often occurs when explicit independent variables show up in densities and fluxes.

Example 5.6. A conservation law for the ZK Equation (3.7) is

$$\mathbf{D}_t(yu) + \mathbf{D}_x\left(\frac{1}{2}\alpha yu^2 + \beta yu_{2x} + \beta yu_{2y}\right) + \mathbf{D}_y(0) = 0. \quad (5.6)$$

Factoring gives

$$y\mathbf{D}_t(u) + y\mathbf{D}_x\left(\frac{1}{2}\alpha u^2 + \beta u_{2x} + \beta u_{2y}\right) + \mathbf{D}_y(0) = 0,$$

which is identical to

$$\mathbf{D}_t(u) + \mathbf{D}_x\left(\frac{1}{2}\alpha u^2 + \beta u_{2x} + \beta u_{2y}\right) + \mathbf{D}_y(0) = 0, \quad (5.7)$$

Therefore, (5.6) is equivalent to (5.7). If the term βu_{2y} is shifted to \mathbf{D}_y , (5.7) is the ZK equation written as conservation law (3.9).

5.4 Linear Combinations

A linear combination of densities with constant coefficients is also a density. This is easily verified using the additive and scalar multiplicative properties of the total derivative and the divergence operators. However, the linear combination can be cumbersome to work with and does not represent a new conservation law.

A density that is a linear combination of other densities can be difficult to split apart, unless information is known about other densities. The algorithm described below can take a linear combination and split out both known densities and densities equivalent to known densities. The process requires establishing a list of independent densities by checking for equivalences. Any part of the linear combination that does not match a density in the established list is added to the list as a new independent density.

Example 5.7. Let $I = \{u_x^2\}$ be a list of established densities for the NTGF equation (3.18). A new density is given as $\rho_2 = u_x^2 + u_x u_y$. After comparing ρ_2 to the densities in I , u_x^2 is removed from ρ_2 , since it is already a density. This leaves $\rho_2 = u_x u_y$ as a new independent density, which is now added to list I . Now, $I = \{u_x^2, u_x u_y\}$. The next density is given as $\rho_3 = uu_{2x} + u_x u_y$. When compared to the densities in I , $u_x u_y$ is already there and uu_{2x} is equivalent to u_x^2 . Thus ρ_3 is not a new density since it is equivalent to a linear combination of other independent densities.

The algorithm described here uses linear algebra to determine if all or part of a density is equivalent to a linear combination of densities. The zeroth-Euler operator is applied to all densities before a linear combination is constructed. The use of the zeroth-Euler operator gives the algorithm the capability of finding a part of a density equivalent to a known independent density. This algorithm is part of the `IndependenceTest.m` code and will work on any list of densities for a particular PDE.

Algorithm 5.2 (Algorithm to separate linear combinations of densities). *The algorithm takes a differential expression that has been established to be a density for a PDE and a list I , containing P densities that have been established as independent densities for the same PDE.*

- i. Attach an undetermined coefficient, $p_i, i = 1, \dots, P$ to each density in list I .*
- ii. Expand the density being tested so that it is the sum of Q terms. Attach an undetermined constant, $q_i, i = 1, \dots, Q$, to each term in the density.*
- iii. Add the density being tested to I , then form list E by applying the zeroth-Euler operator to every density in list I .*
- iv. Add the elements in E and rearrange the sum according to like terms.*

- v. Form a system of linear equations for the q_i by setting the coefficients of like terms equal to zero. In this system, the p_i are parameters. Solve the system for all q_i .
- vi. Any solution for q_i that is in terms of the p_i identifies a density equivalent to an established density. These terms are removed from the density being tested. Any solution for q_i equal to zero identifies an independent term.

Algorithm 5.2 is demonstrated using densities for the ZK equation (3.7).

Example 5.8. For the ZK equation (3.7), a list of independent densities has been established as

$$I = \{u, u^2, u^3 - \frac{3\beta}{\alpha}(u_x^2 + u_y^2)\}. \quad (5.8)$$

A new density is given as

$$\rho = u - 3u^2 + tu^2 - \alpha u^3 + 3\beta(uu_{2x} + uu_{2y}) - \frac{2}{\alpha}xu. \quad (5.9)$$

An inspection shows that densities from list I form part of ρ . Algorithm 5.2 will identify and remove these independent densities from ρ .

- i. Undetermined coefficients, p_i , are attached to each density:

$$I = \{p_1u, p_2u^2, p_3(u^3 - \frac{3\beta}{\alpha}(u_x^2 + u_y^2))\}.$$

- ii. Undetermined constants, q_i are attached to each term in the density being evaluated:

$$\rho_e = q_1u - 3q_2u^2 + q_3tu^2 - q_4\alpha u^3 - 3q_5\beta uu_{2x} - 3q_6\beta uu_{2y} - \frac{2}{\alpha}q_7xu.$$

- iii. ρ_e is added to I , then the zeroth-Euler operator is applied to every density in the list:

$$E = \{p_1, 2p_2u, \frac{3}{\alpha}p_3(\alpha u^2 + 2\beta(u_{2x} + u_{2y})), q_1 - 6q_2u - 2q_3tu - 3q_4\alpha u^2 - 6q_5\beta u_{2x} - 6q_6\beta u_{2y} - \frac{2}{\alpha}q_7x\}.$$

- iv. The elements in E are summed and rearranged according to like terms:

$$(p_1 + q_1) + 2(p_2 - 3q_2)u - (q_3)tu + 3(p_3 - q_4\alpha)u^2 + 6\beta(\frac{1}{\alpha}p_3 - q_5)u_{2x} + 6\beta(\frac{1}{\alpha}p_3 - q_6)u_{2y} - \frac{2}{\alpha}(q_7)x.$$

- v. A linear system of equations is formed by setting the coefficients in iv equal to zero.

The solution to the system is:

$$q_1 = -p_1, \quad q_2 = -\frac{1}{3}p_2, \quad q_3 = 0, \quad q_4 = \frac{1}{\alpha}p_3, \quad q_5 = \frac{1}{\alpha}p_3, \quad q_6 = \frac{1}{\alpha}p_3, \quad q_7 = 0.$$

vi. The term with coefficient q_1 is equivalent to the density with coefficient p_1 , the term with coefficient q_2 is equivalent to the density with coefficient p_2 , and the terms with coefficients q_4 , q_5 , and q_6 are equivalent to the density with coefficient p_3 . The terms with coefficients q_3 and q_7 form an independent density, $tu^2 - \frac{2}{\alpha}xu$.

From ρ in (5.9), the terms u , $-3u^2$, and $-\alpha u^3 + 3\beta uu_{2x} + 3\beta uu_{2y}$ are removed since they are equivalent to established independent densities. The density $tu^2 - \frac{2}{\alpha}xu$ is therefore a new independent density originating from (5.9).

The method used in Algorithm 5.2 identifies $tu^2 - \frac{2}{\alpha}xu$ as independent even though u and u^2 have been established as independent densities. The independent variables t and x are considered to be part of the differential expression, not coefficients on the terms, so tu^2 is independent of u^2 and xu is independent of u .

A second method for splitting a density given as a linear combination of potentially independent densities works when the original PDE is an evolution equation, and eliminates the need for a list of independent densities. This method is implemented in `ConservationLawsMD.m` since a non-evolution PDE is transformed into evolution form internally. Since the ZK equation (3.7) is an evolution equation, this method will work for (5.9).

First attach undetermined coefficients to each term in the density,

$$\rho = c_1u - 3c_2u^2 + c_3tu^2 - c_4\alpha u^3 - 3\beta c_5uu_{2x} - 3\beta c_6uu_{2y} - \frac{2}{\alpha}c_7xu, \quad (5.10)$$

then take the total t -derivative of the density to get

$$\begin{aligned} D_t\rho = & c_1u_t - 6c_2uu_t + c_3u^2 + 2c_3tuu_t - 3c_4\alpha u^2u_t - 3c_5\beta u_tu_{2x} - 3c_5\beta uu_{t2x} \\ & - 3c_6\beta u_tu_{2y} - 3c_6\beta uu_{t2y} - \frac{2}{\alpha}c_7xu_t. \end{aligned} \quad (5.11)$$

Next, solve (3.7) for u_t ,

$$u_t = -(\alpha uu_x + \beta(u_{3x} + u_{x2y})). \quad (5.12)$$

Using (5.12), (5.11) becomes

$$\begin{aligned}
D_t \rho = & -c_1(\alpha uu_x + \beta(u_{3x} + u_{x2y})) + 6c_2u(\alpha uu_x + \beta(u_{3x} + u_{x2y})) + c_3u^2 \\
& - 2c_3tu(\alpha uu_x + \beta(u_{3x} + u_{x2y})) + 3c_4\alpha u^2(\alpha uu_x + \beta(u_{3x} + u_{x2y})) \\
& + 3c_5\beta(\alpha uu_x + \beta(u_{3x} + u_{x2y}))u_{2x} + 3c_5\beta u(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_{2x} \\
& + 3c_6\beta(\alpha uu_x + \beta(u_{3x} + u_{x2y}))u_{2y} + 3c_6\beta u(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_{2y} \\
& + \frac{2}{\alpha}c_7x(\alpha uu_x + \beta(u_{3x} + u_{x2y})). \tag{5.13}
\end{aligned}$$

Now, apply the zeroth-Euler operator to (5.13), to get

$$\begin{aligned}
\mathcal{L}_{u(x,y)}(D_t \rho) = & 2(c_3 - c_7)u + 18\alpha\beta(c_5 - c_4)u_x u_{2x} + 6\alpha\beta(c_6 - c_4)u_x u_{2y} \\
& + 12\alpha\beta(c_6 - c_4)u_y u_{xy}. \tag{5.14}
\end{aligned}$$

By (3.5), $D_t \rho$ is a divergence, thus Theorem 4.1 requires that $D_t \rho$ be zero. Clearly $c_5 = c_4$, $c_6 = c_4$ and $c_7 = c_3$ with c_1 , c_2 , c_3 and c_4 arbitrary satisfy $D_t \rho = 0$. Finally, substituting this solution into (5.10), the density becomes

$$\rho = c_1u - 3c_2u^2 + c_3(tu^2 - \frac{2}{\alpha}xu) - c_4(\alpha u^3 - 3\beta(uu_{2x} + 3uu_{2y})). \tag{5.15}$$

Since all but one of the coefficients can be set to zero and the remaining coefficient can be set to one, (5.15) can be split into four independent densities, namely

$$\begin{aligned}
\rho_1 &= u, \\
\rho_2 &= u^2, \\
\rho_3 &= tu^2 - \frac{2}{\alpha}xu, \\
\rho_4 &= u^3 - 3\beta(uu_{2x} + uu_{2y}).
\end{aligned}$$

Since the coefficients in (5.15) are arbitrary, there is some freedom of choice for how values are set. In this dissertation, densities are normalized on the highest degree terms, and other coefficients are adjusted accordingly.

CHAPTER 6

TOOLS FOR THE COMPUTATION OF CONSERVATION LAWS: THE HOMOTOPY OPERATOR

To compute the flux of a conservation law, it is necessary to integrate (by parts) a differential function with one independent variable that is known to be a total derivative. *Mathematica* and other CAS are often unable to do this integration. For example, *Mathematica* cannot integrate

$$f = 2u_x u_{2x} \cos u + u_x v \sin u - u_x^3 \sin u - v_x \cos u + 2v_x v_{2x}, \quad (6.1)$$

which is a total derivative since $f = D_x F$ with $F = u_x^2 \cos u - v \cos u + v_x^2$. Although F can be easily computed by hand, *Mathematica* fails to integrate f , most likely because f is of second order in both u and v . On functions with more than one independent variable, it is necessary to invert a divergence on an exact differential function. Currently, CAS are unable to do this. The homotopy operator provides a reliable method for integrating exact functions in one independent variable and for inverting divergences on multi-variable exact functions.

Homotopies play a large role in topological theory. A general description of a homotopy is as follows.

Definition 6.1. *A homotopy is a continuous transformation acting on two functions, say $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}_0(\mathbf{x})$. A homotopy between \mathbf{u} and \mathbf{u}_0 from space X to space Y is the continuous map $T: X \times [0, 1] \rightarrow Y$ such that $T(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ and $T(\mathbf{x}, 1) = \mathbf{u}(\mathbf{x})$ [17].*

To establish a path from $\mathbf{u}_0(\mathbf{x})$ to $\mathbf{u}(\mathbf{x})$, one uses an auxiliary variable $\lambda \in [0, 1]$. A simple example is $T(\mathbf{x}, \lambda) = (1 - \lambda)\mathbf{u}_0(\mathbf{x}) + \lambda\mathbf{u}(\mathbf{x})$.

The homotopy operator first appeared in works by Volterra, where he uses the homotopy operator in the inverse problem of the calculus of variations [71]. The Poincaré lemma⁴ states that on a smooth manifold, exact differential k -forms are closed [11, 25].

⁴There is some confusion historically about whether the Poincaré lemma is as stated in this dissertation or is the converse of this statement. Olver [56] refers to the converse as the Poincaré lemma.

The converse of the Poincaré lemma states that closed forms are exact. However, the converse holds only on simple topological domains, such as a star-shaped domain as described by [56]. Once the domain is appropriately restricted, the proof of the converse of the Poincaré lemma requires the construction of a homotopy operator. Indeed, the construction of suitable homotopy operators is the key to exactness proofs for many complexes such as the de Rham complex and the variational complex [56].

The idea for using the homotopy operator for the computations of conservation laws was suggested by Anco and Bluman [5, p. 582]. They developed a homotopy formula for directly computing densities and fluxes. Cheviakov [16] applies the method of Anco and Bluman and the homotopy operators from [35] to compute several conservation laws. Prior to Anco and Bluman’s work, Olver [56] showed a detailed construction of the homotopy operator for use in the proof of the exactness of the variational complex. He stated that the homotopy operator is a tool for inverting a divergence, and also points out the shortcomings of the homotopy operator in practical computations. The homotopy operator given in [56] is written in terms of differential forms and requires the use of higher-Euler operators, that is, higher order versions of the zeroth-Euler operator, (4.1). To make the homotopy operator algorithmic, the form given by Olver was rewritten in terms of standard calculus by Hereman et al. [35]. The formulas in this dissertation build on the formulation given in [35] using an idea put forth by Kruskal et al. [47].

In the form given in [35], the homotopy operator can be easily applied and is algorithmic. However, two issues must be addressed before the homotopy operator becomes a useful tool. First, application of the homotopy operator can quickly become unmanageable [56], especially in higher dimensions. Indeed, the use of higher-Euler operators in the integrand of the homotopy operator creates a swell of terms of which a large number cancel before the final result is produced. The second issue occurs in the inversion of a divergence, Div^{-1} , which does not produce a unique answer. In analogy with standard integration with D_x^{-1} , where the result is up to an arbitrary constant, Div^{-1} is defined up to a divergence-free (curl) term (see Example 4.2). The homotopy operator will produce a particular choice for the curl term [35], often creating very large vectors. Resolving these issues is an important piece of this dissertation, and the findings

give new insight in the theory and applicability of the homotopy operator.

With universal application in mind, the latest version of the homotopy operator has its own *Mathematica* code, `HomotopyIntegrator.m`, which runs independently and as a part of `ConservationLawsMD.m`. To our knowledge, `HomotopyIntegrator.m` is currently the only full-fledged implementation of the homotopy operator in *Mathematica*. In collaboration with Hereman [35], Deconinck and Nivala have recently developed similar code in *Maple* [21, 22] for the one-dimensional homotopy operator as an independent integrator. Both Anderson [6] and Cheviakov [15] have implemented the homotopy operator in *Maple* as a component of broader software packages.

The program `HomotopyIntegrator.m` has proven to be reliable by either integrating a variety of one-dimensional exact differential expressions or inverting divergences on a variety of multi-dimensional exact differential expressions. The code verifies if an expression is exact by applying the zeroth-Euler operator (see Theorem 4.1). If the function is not exact but has parameters, the code will identify values for the parameters that may make the expression exact. If successful, it then performs the required inversion of that exact function. In cases where there are multiple independent variables, the program will automatically remove curl terms from the solution vector.

6.1 The Original Homotopy Operator

The homotopy operator taken from [56] and translated into standard calculus first appears in [35]. In this dissertation, it is given in separate cases for one, two and three independent variables since all three cases are used in the conservation laws algorithm.

Definition 6.2. *Let $\mathbf{x} = x$ be the independent variable, and let $f(x, \mathbf{u}^{(M)}(x))$ be an exact differential function. The one-dimensional homotopy operator is defined by*

$$\mathcal{H}_{\mathbf{u}(x)}f = \int_{\lambda_0}^1 \sum_{j=1}^N (\mathcal{I}_{u^j(x)}f) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda}, \quad (6.2)$$

where $\mathbf{u} = (u^1, \dots, u^N)$. $(\mathcal{I}_{u^j(x)}f) [\lambda \mathbf{u}]$ means that in $\mathcal{I}_{u^j(x)}f$, all u^j are replaced by λu^j , all u_x^j are replaced by λu_x^j , and so on for all derivatives of u^j . The integrand, $\mathcal{I}_{u^j(x)}f$, is defined as

$$\mathcal{I}_{u^j(x)}f = \sum_{i=0}^{M_1^j-1} D_x^i \left(u^j \mathcal{L}_{u^j(x)}^{(i+1)}f \right), \quad (6.3)$$

where M_1^j is the order of the differential function f in dependent variable u^j with respect to x , and $\mathcal{L}_{u^j(x)}^{(i)}$ is the i^{th} higher-Euler operator, a higher-order variational derivative from the calculus of variations, given as

$$\mathcal{L}_{u^j(x)}^{(i)} f = \sum_{k=i}^{M_1^j} \binom{k}{i} (-D_x)^{k-i} \frac{\partial f}{\partial u_{kx}^j}. \quad (6.4)$$

In [35], The lower limit, λ_0 , on the integral in (6.2) is given as 0. However, the kernel of the homotopy operator is not trivial [22], so keeping $\lambda_0 = 0$ restricts the homotopy operator to exact polynomial differential functions with perhaps a few nonlinear transcendental terms. When differential functions with fractional form occur in the integrand, singularities may occur at $\lambda = 0$, thus the integral diverges as $\lambda_0 \rightarrow 0$. In cases where the integral diverges as $\lambda \rightarrow 0$, the integral will converge as $\lambda \rightarrow \infty$, so the lower limit is taken as $\lambda_0 \rightarrow \infty$. These cases will be discussed in Section 6.3.1.

Versions of the homotopy operator for two and three independent variables have a similar construction.

Definition 6.3. Let $\mathbf{x} = (x, y)$, and let $f(x, y, \mathbf{u}^{(M)}(x, y))$ be an exact differential function for two independent variables. The two-dimensional homotopy operator is a two-component vector,

$$\left(\mathcal{H}_{\mathbf{u}(x,y)}^{(x)} f, \mathcal{H}_{\mathbf{u}(x,y)}^{(y)} f \right), \quad (6.5)$$

where

$$\mathcal{H}_{\mathbf{u}(x,y)}^{(x)} f = \int_{\lambda_0}^1 \sum_{j=1}^N \left(\mathcal{I}_{u^j(x,y)}^{(x)} f \right) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda}, \quad (6.6)$$

and

$$\mathcal{H}_{\mathbf{u}(x,y)}^{(y)} f = \int_{\lambda_0}^1 \sum_{j=1}^N \left(\mathcal{I}_{u^j(x,y)}^{(y)} f \right) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda}. \quad (6.7)$$

The integrands, $\mathcal{I}_{u^j(x,y)}^{(x)} f$ and $\mathcal{I}_{u^j(x,y)}^{(y)} f$, are defined as

$$\mathcal{I}_{u^j(x,y)}^{(x)} f = \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \left(\frac{1+i_1}{1+i_1+i_2} \right) D_x^{i_1} D_y^{i_2} \left(u^j \mathcal{L}_{u^j(x,y)}^{(i_1+1, i_2)} f \right) \quad (6.8)$$

and

$$\mathcal{I}_{u^j(x,y)}^{(y)} f = \sum_{i_1=0}^{M_1^j} \sum_{i_2=0}^{M_2^j-1} \left(\frac{1+i_2}{1+i_1+i_2} \right) D_x^{i_1} D_y^{i_2} \left(u^j \mathcal{L}_{u^j(x,y)}^{(i_1, i_2+1)} f \right). \quad (6.9)$$

For both cases,

$$\mathcal{L}_{u^j(x,y)}^{(i_1,i_2)} f = \sum_{k_1=i_1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \binom{k_1}{i_1} \binom{k_2}{i_2} (-D_x)^{k_1-i_1} (-D_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (6.10)$$

is the (i_1, i_2) higher-Euler operator acting on functions f with two independent variables.

Definition 6.4. Let $\mathbf{x} = (x, y, z)$, and let $f(x, y, z, \mathbf{u}^{(M)}(x, y, z))$ be a differential function of three independent variables. The three-dimensional homotopy operator is a three-component vector,

$$\left(\mathcal{H}_{\mathbf{u}(x,y,z)}^{(x)} f, \mathcal{H}_{\mathbf{u}(x,y,z)}^{(y)} f, \mathcal{H}_{\mathbf{u}(x,y,z)}^{(z)} f \right), \quad (6.11)$$

where the x -component is given by

$$\mathcal{H}_{\mathbf{u}(x,y,z)}^{(x)} f = \int_{\lambda_0}^1 \sum_{j=1}^N \left(\mathcal{I}_{u^j(x,y,z)}^{(x)} f \right) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda}. \quad (6.12)$$

The integrand for the x -component is given as

$$\mathcal{I}_{u^j(x,y,z)}^{(x)} f = \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \sum_{i_3=0}^{M_3^j} \left(\frac{1+i_1}{1+i_1+i_2+i_3} \right) D_x^{i_1} D_y^{i_2} D_z^{i_3} \left(u^j \mathcal{L}_{u^j(x,y,z)}^{(i_1+1,i_2,i_3)} f \right). \quad (6.13)$$

$\mathcal{H}_{\mathbf{u}(x,y,z)}^{(y)} f$, and $\mathcal{H}_{\mathbf{u}(x,y,z)}^{(z)} f$ are defined analogously. The higher-Euler operator acting on differential functions f with three independent variables is given as

$$\begin{aligned} & \mathcal{L}_{u^j(x,y,z)}^{(i_1,i_2,i_3)} f \\ &= \sum_{k_1=i_1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \sum_{k_3=i_3}^{M_3^j} \binom{k_1}{i_1} \binom{k_2}{i_2} \binom{k_3}{i_3} (-D_x)^{k_1-i_1} (-D_y)^{k_2-i_2} (-D_z)^{k_3-i_3} \frac{\partial f}{\partial u_{k_1 x k_2 y k_3 z}^j}. \end{aligned}$$

6.2 Where is the Homotopy in the Homotopy Operator?

The homotopy in the homotopy operator is not easily apparent. The method used to show the homotopy below is patterned after the proof for the converse of the Poincaré lemma given in [25].

From Definition 2.1, $(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ is the coordinate for jet space J^M . Let $I = [0, 1]$ be the unit interval on the λ -axis. Then the space $I \times J^M$ has the coordinate $(\lambda, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$. Define

$$y_1 : J^M \longrightarrow I \times J^M \text{ such that } y_1(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) = (1, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})),$$

$$\text{and } y_0 : J^M \longrightarrow I \times J^M \text{ such that } y_0(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) = (0, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})).$$

Let $g : I \times J^M \longrightarrow \mathbb{R}$ be a real valued function on $I \times J^M$. Define y_i^* , $i = 0, 1$ such that

$$y_i^* g = g \circ y_i, \quad i = 0, 1. \quad (6.14)$$

Thus, $y_i^* : C(I \times J^M) \longrightarrow C(J^M)$, where C is the space of all continuous differential functions.

Let C^1 be the space of a continuous exact differential functions, and let $\omega = g(\lambda, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$. The homotopy $H : C^1(I \times J^M) \longrightarrow C(J^M)$ is defined as

$$H\omega = \int_0^1 g(\lambda, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\lambda. \quad (6.15)$$

To see the homotopy as defined in Definition 6.1, take the partial derivative with respect to λ on both sides of (6.15),

$$\begin{aligned} \frac{\partial H\omega}{\partial \lambda} &= \frac{\partial}{\partial \lambda} \int_0^1 g(\lambda, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) d\lambda \\ &= g(1, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) - g(0, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) \\ &= y_1^* \omega - y_0^* \omega. \end{aligned}$$

Therefore, H assigns g to y_1^* when $\lambda = 1$ and assigns g to y_0^* when $\lambda = 0$.

The homotopy used in (6.2), (6.5), and (6.11) takes an exact function and inverts the divergence. The integrand has $g(\lambda, \mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x})) = (\mathcal{I}_{u^j(\mathbf{x})} f) [\lambda \mathbf{u}]_{\lambda}^{\frac{1}{\lambda}}$, which will be derived in the next section. The same homotopy (6.15) is used in the proof of the converse of the Poincaré theorem [11, 25] and in the exactness proofs for the de Rham complex and the variational complex [56].

6.3 Reformulation of the Homotopy Operator Integrand

While attempting to explain why the homotopy operator inverts the derivative of a differential function with one independent variable or the divergence of a differential function with multiple independent variables, it was discovered that the integrand (6.3) could be written in a simpler manner by eliminating the binomial coefficient and reducing the number of applications of the total derivative operator. Following this discovery for the one-dimensional case, new simpler integrands for two- and three-dimensional homotopy operators were developed. These new integrands considerably reduce computation time since they prevent the calculation of large sets of total derivatives.

The next three sections will introduce the new and improved homotopy operators for one, two and three independent variables. At the same time, proofs will be given to show that the one-dimensional homotopy operator does indeed invert a total derivative and that the two-dimensional homotopy operator will invert a total divergence.

6.3.1 The Homotopy Operator for One Independent Variable

Based on work by Kruskal et al. [47], the integrand of the one-dimensional homotopy operator with independent variable, $\mathbf{x} = x$, can be revised as follows.

Definition 6.5. *The homotopy operator for one independent variable, x , applied to $f = f(x, \mathbf{u}^{(M)}(x))$ is (6.2), with (6.3) replaced by*

$$\mathcal{I}_{u^j(x)} f = \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j}. \quad (6.16)$$

The higher-Euler operators (6.4) are no longer needed.

Before showing that the one-dimensional homotopy operator given in Definition 6.5 inverts a total derivative, some preliminary definitions and theorems are required. The theorems shown here follow the layout for the case with one independent variable in [36], but are shown in greater detail and for multiple dependent variables.

The degree operator \mathbf{M} [47, p. 953] is introduced to show how a homotopy is applied to the inversion problem.

Definition 6.6. *The degree operator \mathbf{M} acting on a differential function $f = f(x, \mathbf{u}^{(M)}(x))$ with one independent variable is defined [47] as*

$$\mathbf{M}f = \sum_{j=1}^N \sum_{i=0}^{M_1^j} u_{ix}^j \frac{\partial f}{\partial u_{ix}^j}, \quad (6.17)$$

where f has order M_1^j in u^j with respect to x .

When \mathbf{M} is applied to a polynomial differential function $f = f(x, \mathbf{u}^{(M)}(x))$, the result contains each term of f multiplied by its total degree.

Example 6.1. *To illustrate the degree operator, let $\mathbf{u} = (u^1, u^2) = (u, v)$, and take $f(x, \mathbf{u}^{(5)}(x)) = (u)^p (v_{2x})^q (u_{5x})^r$, where p , q , and r are nonnegative integers. Applying*

\mathbf{M} to f gives

$$\begin{aligned}\mathbf{M}f &= u \frac{\partial}{\partial u} (u)^p (v_{2x})^q (u_{5x})^r + u_{5x} \frac{\partial}{\partial u_{5x}} (u)^p (v_{2x})^q (u_{5x})^r + v_{2x} \frac{\partial}{\partial v_{2x}} (u)^p (v_{2x})^q (u_{5x})^r \\ &= pu (u)^{p-1} (v_{2x})^q (u_{5x})^r + ru_{5x} (u)^p (v_{2x})^q (u_{5x})^{r-1} + qv_{2x} (u)^p (v_{2x})^{q-1} (u_{5x})^r \\ &= (p + q + r) (u)^p (v_{2x})^q (u_{5x})^r.\end{aligned}$$

Note that the total degree of f has become a factor. Compare this with $x(x^n)' = x(nx^{n-1}) = nx^n$ in standard calculus.

Now, the inverse of the degree operator, \mathbf{M}^{-1} , will be derived. The inverse of the degree operator uses a homotopy (6.15) to return the given f . To avoid singularities when evaluating λ at $\lambda = 0$, the lower limit on the integral is taken as λ_0 [22]. Thus, the homotopy used here is on $[\lambda_0, 1] \times J^M$.

Theorem 6.1. Let $g(x, \mathbf{u}^{(M)}(x))$ be a differential monomial such that $g(x, \mathbf{u}^{(M)}(x)) = \mathbf{M}f(x, \mathbf{u}^{(M)}(x))$, where $\mathbf{M}f(x, \mathbf{u}^{(M)}(x)) \neq 0$. Also, let $g[\lambda \mathbf{u}]$ denote $g(x, \mathbf{u}^{(M)}(x))$ in which \mathbf{u} is replaced by $\lambda \mathbf{u}$, \mathbf{u}_x is replaced by $\lambda \mathbf{u}_x$, and so on, where λ is an auxiliary parameter.

Then

$$f(x, \mathbf{u}^{(M)}(x)) = \mathbf{M}^{-1}g(x, \mathbf{u}^{(M)}(x)) = \int_{\lambda_0}^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda}. \quad (6.18)$$

PROOF: If $g(x, \mathbf{u}^{(M)}(x))$ has order M_1^j in u^j with respect to x , then $g[\lambda \mathbf{u}]$ also has order M_1^j in u^j . Furthermore,

$$\begin{aligned}\frac{d}{d\lambda}g[\lambda \mathbf{u}] &= \sum_{j=1}^N \sum_{i=0}^{M_1^j} \frac{\partial g[\lambda \mathbf{u}]}{\partial \lambda u_{ix}^j} \frac{d\lambda u_{ix}^j}{d\lambda} \\ &= \frac{1}{\lambda} \sum_{j=1}^N \sum_{i=0}^{M_1^j} u_{ix}^j \frac{\partial g[\lambda \mathbf{u}]}{\partial u_{ix}^j} \\ &= \frac{1}{\lambda} \mathbf{M}g[\lambda \mathbf{u}],\end{aligned}$$

by the definition of \mathbf{M} . Integrating both sides with respect to λ gives

$$\begin{aligned}\int_{\lambda_0}^1 \frac{d}{d\lambda}g[\lambda \mathbf{u}] d\lambda &= \int_{\lambda_0}^1 \mathbf{M} \frac{g[\lambda \mathbf{u}]}{\lambda} d\lambda, \\ g[\lambda \mathbf{u}] \Big|_{\lambda=\lambda_0}^{\lambda=1} &= \mathbf{M} \int_{\lambda_0}^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda}, \\ g(x, \mathbf{u}^{(M)}(x)) - g[\lambda_0 \mathbf{u}] &= \mathbf{M} \int_{\lambda_0}^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda}.\end{aligned}$$

Next, one must evaluate $g[\lambda_0 \mathbf{u}]$. Using (6.17),

$$\begin{aligned} g[\lambda_0 \mathbf{u}] &= \mathbf{M}f[\lambda_0 \mathbf{u}] = \sum_{j=1}^N \sum_{i=0}^{M_1^j} \lambda_0 u_{ix}^j \frac{\partial f[\lambda_0 \mathbf{u}]}{\partial (\lambda_0 u_{ix}^j)} \\ &= \lambda_0 \sum_{j=1}^N \sum_{i=0}^{M_1^j} u_{ix}^j \frac{\partial f[\lambda_0 \mathbf{u}]}{\partial (\lambda_0 u_{ix}^j)} \end{aligned} \quad (6.19)$$

Depending on the form of f , there are three types of monomials that can occur, leading to two choices for λ_0 that make $g[\lambda_0 \mathbf{u}] = 0$.

Case 1: If $g[\lambda_0 \mathbf{u}]$ is a monomial in fractional form and λ_0 is a factor in the denominator, then let $\lambda_0 \rightarrow \infty$ to get $g[\lambda_0 \mathbf{u}] = 0$. Then

$$g(x, \mathbf{u}^{(M)}(x)) = \mathbf{M} \int_{\infty}^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda}, \quad (6.20)$$

and the integral converges. Apply \mathbf{M}^{-1} to both sides of (6.20) to get (6.18).

Case 2: For all other forms of (6.19), provided that λ_0 does not drop out of (6.19), set $\lambda_0 = 0$. Thus,

$$g(x, \mathbf{u}^{(M)}(x)) = \mathbf{M} \int_0^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda}, \quad (6.21)$$

and the integral converges. Apply \mathbf{M}^{-1} to both sides of (6.21) to get (6.18).

Case 3: For fractional monomials where all λ_0 drop out of (6.19), $g(x, \mathbf{u}^{(M)}(x))$ is in the kernel of the degree operator \mathbf{M} . Then $\mathbf{M} \int_{\lambda_0}^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda} = C$, where C is a constant, and (6.18) does not hold. A shift of coordinates is needed to move g out of the kernel of \mathbf{M} so that (6.18) will apply. Details will be given in Section 6.5. ■

To show that \mathbf{M}^{-1} does indeed invert the effect of applying \mathbf{M} to a monomial, \mathbf{M}^{-1} will be applied to the result from Example 6.1, a simple one-dimensional monomial. This is an example where case 2 applies.

Example 6.2. To illustrate $\mathbf{M}^{-1}g$, let $g(x, \mathbf{u}^{(5)}(x)) = \mathbf{M}f(x, \mathbf{u}^{(5)}(x))$ from Example 6.1. That is,

$$g = (p + q + r) (u)^p (v_{2x})^q (u_{5x})^r.$$

Inverting the \mathbf{M} operator yields

$$\begin{aligned}
\mathbf{M}^{-1}g(x, \mathbf{u}^{(M)}) &= \int_{\lambda_0}^1 (p+q+r) (\lambda u)^p (\lambda v_{2x})^q (\lambda u_{5x})^r \frac{d\lambda}{\lambda} \\
&= (p+q+r) (u)^p (v_{2x})^q (u_{5x})^r \int_0^1 \lambda^{p+q+r-1} d\lambda \\
&= (p+q+r) (u)^p (v_{2x})^q (u_{5x})^r \left[\frac{\lambda^{p+q+r}}{p+q+r} \right]_{\lambda_0}^1. \tag{6.22}
\end{aligned}$$

Since (6.22) converges as $\lambda \rightarrow 0$, take $\lambda_0 = 0$ so that $\mathbf{M}^{-1}g(x, \mathbf{u}^{(M)}) = (u)^p (v_{2x})^q (u_{5x})^r$.

Note that (6.22) is identical to f given in Example 6.1.

At this point, only monomial functions have been considered. Let $f(x, \mathbf{u}^{(M)}(x)) = f_1 + f_2 + \dots + f_i + \dots + f_q$, where each $f_i = f_i(x, \mathbf{u}^{(M)}(x))$ is a single-term differential function (not necessarily a monomial). Theorem 6.18 holds for f provided that $\mathbf{M}f_i \neq 0$ for $i = 1, \dots, q$. The following examples illustrate cases 1 and 2 for more general differential functions.

Example 6.3. Let $g(x, \mathbf{u}^{(1)}(x)) = 3u^2u_x + u \cos u$. The inverse of the degree operator gives

$$\begin{aligned}
\mathbf{M}^{-1}g(x, \mathbf{u}^{(M)}) &= \int_{\lambda_0}^1 (3\lambda^2u^2u_x + u \cos(\lambda u)) d\lambda \\
&= [\lambda^3u^2u_x + \sin(\lambda u)]_{\lambda_0}^1.
\end{aligned}$$

Since there is no denominator, take $\lambda_0 = 0$ as in case 2. Thus $g(x, \mathbf{u}^{(1)}(x)) = u^2u_x + \sin u$.

Case 1 occurs when g is rational.

Example 6.4. To illustrate case 1, let $g(x, \mathbf{u}^{(1)}(x)) = -\frac{2}{u_x^2 + v^2}$. Application of the inverse of the degree operator yields

$$\begin{aligned}
\mathbf{M}^{-1}g(x, \mathbf{u}^{(M)}) &= \int_{\lambda_0}^1 \frac{-2}{\lambda^3(u_x^2 + v^2)} d\lambda \\
&= \left[\frac{1}{\lambda^2(u_x^2 + v^2)} \right]_{\lambda_0}^1.
\end{aligned}$$

In this case, a singularity occurs at $\lambda_0 = 0$, thus the integral diverges. Take $\lambda_0 \rightarrow \infty$ so that $\mathbf{M}^{-1}g(x, \mathbf{u}^{(M)}) = \frac{1}{u_x^2 + v^2}$.

If g is a polynomial differential expression, then $g[\lambda \mathbf{u}]$ is polynomial in λ . Therefore, M^{-1} does polynomial integration on a single variable, resulting in an adjustment of the coefficients of g . However, if g is transcendental or rational, then M^{-1} does single variable integration by parts with respect to λ . The integration by parts over λ in the homotopy is simpler than doing integration by parts on a differential expression.

The next theorem shows the commutative relation between the degree operator M and the total derivative D_x .

Theorem 6.2. *The degree operator as well as its inverse commute with the total derivative operator; that is, $MD_x = D_xM$ and $M^{-1}D_x = D_xM^{-1}$.*

PROOF: Let $f = f(x, \mathbf{u}^{(M)}(x))$. Then,

$$\begin{aligned}
MD_x f &= \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}+1} u_{ix}^{j_1} \frac{\partial}{\partial u_{ix}^{j_1}} \left(\frac{\partial f}{\partial x} + \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{(k+1)x}^{j_2} \frac{\partial f}{\partial u_{kx}^{j_2}} \right) \\
&= \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}+1} u_{ix}^{j_1} \left(\frac{\partial^2 f}{\partial u_{ix}^{j_1} \partial x} + \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} \left(\frac{\partial u_{(k+1)x}^{j_2}}{\partial u_{ix}^{j_1}} \frac{\partial f}{\partial u_{kx}^{j_2}} + u_{(k+1)x}^{j_2} \frac{\partial^2 f}{\partial u_{ix}^{j_1} \partial u_{kx}^{j_2}} \right) \right) \\
&= \frac{\partial}{\partial x} \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}+1} u_{ix}^{j_1} \frac{\partial f}{\partial u_{ix}^{j_1}} + \sum_{j_1=1}^N \sum_{i=1}^{M_1^{j_1}+1} u_{ix}^{j_1} \frac{\partial f}{\partial u_{(i-1)x}^{j_1}} \\
&\quad + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}+1} \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{ix}^{j_1} u_{(k+1)x}^{j_2} \frac{\partial^2 f}{\partial u_{ix}^{j_1} \partial u_{kx}^{j_2}} \\
&= \frac{\partial}{\partial x} \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{ix}^{j_1} \frac{\partial f}{\partial u_{ix}^{j_1}} + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{(i+1)x}^{j_1} \frac{\partial f}{\partial u_{ix}^{j_1}} + \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{(k+1)x}^{j_2} \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{ix}^{j_1} \frac{\partial^2 f}{\partial u_{ix}^{j_1} \partial u_{kx}^{j_2}} \\
&= \frac{\partial}{\partial x} \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{kx}^{j_2} \frac{\partial f}{\partial u_{kx}^{j_2}} + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{(i+1)x}^{j_1} \frac{\partial f}{\partial u_{ix}^{j_1}} + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{(i+1)x}^{j_1} \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{kx}^{j_2} \frac{\partial^2 f}{\partial u_{kx}^{j_2} \partial u_{ix}^{j_1}} \\
&= \frac{\partial}{\partial x} \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{kx}^{j_2} \frac{\partial f}{\partial u_{kx}^{j_2}} + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{(i+1)x}^{j_1} \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} \frac{\partial u_{kx}^{j_2}}{\partial u_{ix}^{j_1}} \frac{\partial f}{\partial u_{kx}^{j_2}} \\
&\quad + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{(i+1)x}^{j_1} \sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{kx}^{j_2} \frac{\partial^2 f}{\partial u_{ix}^{j_1} \partial u_{kx}^{j_2}} \\
&= \left(\frac{\partial}{\partial x} + \sum_{j_1=1}^N \sum_{i=0}^{M_1^{j_1}} u_{(i+1)x}^{j_1} \frac{\partial}{\partial u_{ix}^{j_1}} \right) \left(\sum_{j_2=1}^N \sum_{k=0}^{M_1^{j_2}} u_{kx}^{j_2} \frac{\partial f}{\partial u_{kx}^{j_2}} \right) \\
&= D_x M f. \tag{6.23}
\end{aligned}$$

To show that $M^{-1}D_x f = D_x M^{-1}f$ apply M to both sides of $M^{-1}D_x f = D_x M^{-1}f$. The left hand side simplifies to $MM^{-1}D_x f = D_x f$ since $MM^{-1} = I$, where I is the identity operator. Likewise, the right hand side simplifies to $MD_x M^{-1}f = D_x MM^{-1}f = D_x f$ since M and D_x commute. \blacksquare

The next key theorem shows that the homotopy operator does indeed integrate an exact differential expression.

Theorem 6.3. *Let $f = f(x, \mathbf{u}^{(M)}(x))$ be exact, that is, $\mathcal{L}_{\mathbf{u}(x)}f = \mathbf{0}$. Then $D_x^{-1}f = \mathcal{H}_{\mathbf{u}(x)}f$.*

PROOF: To start, consider only component u^j of \mathbf{u} and multiply $\mathcal{L}_{\mathbf{u}(x)}f$ by u^j to restore the degree. First, split off $u^j \frac{\partial f}{\partial u^j}$. Next, integrate by parts and split off $u_x^j \frac{\partial f}{\partial u_x^j}$. Continue this process until $u_{M_1^j x}^j \frac{\partial f}{\partial u_{M_1^j x}^j}$ is split off. The details are as follows.

$$\begin{aligned}
u^j \mathcal{L}_{\mathbf{u}(x)}f &= u^j \sum_{k=0}^{M_1^j} (-D_x)^k \frac{\partial f}{\partial u_{kx}^j} \\
&= u^j \frac{\partial f}{\partial u^j} + u^j \sum_{k=1}^{M_1^j} (-D_x)^k \frac{\partial f}{\partial u_{kx}^j} \\
&= u^j \frac{\partial f}{\partial u^j} - D_x \left(u^j \sum_{k=1}^{M_1^j} (-D_x)^{k-1} \frac{\partial f}{\partial u_{kx}^j} \right) + u_x^j \sum_{k=1}^{M_1^j} (-D_x)^{k-1} \frac{\partial f}{\partial u_{kx}^j} \\
&= u^j \frac{\partial f}{\partial u^j} + u_x^j \frac{\partial f}{\partial u_x^j} - D_x \left(u^j \sum_{k=1}^{M_1^j} (-D_x)^{k-1} \frac{\partial f}{\partial u_{kx}^j} \right) + u_x^j \sum_{k=2}^{M_1^j} (-D_x)^{k-1} \frac{\partial f}{\partial u_{kx}^j} \\
&= u^j \frac{\partial f}{\partial u^j} + u_x^j \frac{\partial f}{\partial u_x^j} - D_x \left(u^j \sum_{k=1}^{M_1^j} (-D_x)^{k-1} \frac{\partial f}{\partial u_{kx}^j} + u_x^j \sum_{k=2}^{M_1^j} (-D_x)^{k-2} \frac{\partial f}{\partial u_{kx}^j} \right) \\
&\quad + u_{2x}^j \sum_{k=2}^{M_1^j} (-D_x)^{k-2} \frac{\partial f}{\partial u_{kx}^j} \\
&= \dots \\
&= u^j \frac{\partial f}{\partial u^j} + u_x^j \frac{\partial f}{\partial u_x^j} + \dots + u_{M_1^j x}^j \frac{\partial f}{\partial u_{M_1^j x}^j} - D_x \left(u^j \sum_{k=1}^{M_1^j} (-D_x)^{k-1} \frac{\partial f}{\partial u_{kx}^j} \right. \\
&\quad \left. + u_x^j \sum_{k=2}^{M_1^j} (-D_x)^{k-2} \frac{\partial f}{\partial u_{kx}^j} + \dots + u_{(M_1^j-1)x}^j \sum_{k=M_1^j}^{M_1^j} (-D_x)^{k-M_1^j} \frac{\partial f}{\partial u_{kx}^j} \right)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{i=0}^{M_1^j} u_{ix}^j \frac{\partial f}{\partial u_{ix}^j} - D_x \left(\sum_{i=0}^{M_1^j-1} u_{ix}^j \sum_{k=i+1}^{M_1^j} (-D_x)^{k-(i+1)} \frac{\partial f}{\partial u_{kx}^j} \right) \\
&= \sum_{i=0}^{M_1^j} u_{ix}^j \frac{\partial f}{\partial u_{ix}^j} - D_x \left(\sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j} \right). \tag{6.24}
\end{aligned}$$

Now, sum over all of the components u^j to get

$$\begin{aligned}
\sum_{j=1}^N u^j \mathcal{L}_{u^j(x)} f &= \sum_{j=1}^N \sum_{i=0}^{M_1^j} u_{ix}^j \frac{\partial f}{\partial u_{ix}^j} - \sum_{j=1}^N D_x \left(\sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j} \right) \\
&= Mf - D_x \left(\sum_{j=1}^N \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j} \right). \tag{6.25}
\end{aligned}$$

Since f is exact, $\mathcal{L}_{u^j(x)} f = 0$ for $j = 1, \dots, N$, which implies that $\sum_{j=1}^N \mathcal{L}_{u^j(x)} f = 0$. Hence,

$$Mf = D_x \left(\sum_{j=1}^N \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j} \right). \tag{6.26}$$

Apply M^{-1} to both sides and replace $M^{-1}D_x$ by $D_x M^{-1}$ using Theorem 6.2. Then

$$f = D_x \left(M^{-1} \sum_{j=1}^N \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j} \right). \tag{6.27}$$

Apply D_x^{-1} to both sides and use (6.18) to obtain

$$D_x^{-1} f = \int_{\lambda_0}^1 \left(\sum_{j=1}^N \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j} \right) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda}. \tag{6.28}$$

The right hand side of (6.28) is identical to (6.2) with integrand (6.16). ■

The integrand (6.16) originated from Theorem 6.3. The original version (6.3), which used higher-Euler operators, had been established as the integrand by many sources [22, 35, 56]. The following theorem shows that both integrands are equivalent.

Theorem 6.4. *The integrands (6.3) and (6.16) are identical, that is,*

$$\sum_{i=0}^{M_1^j-1} D_x^i u^j \sum_{k=i+1}^{M_1^j} \binom{k}{i+1} (-D_x)^{k-(i+1)} \frac{\partial f}{\partial u_{kx}^j} = \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j}. \tag{6.29}$$

PROOF: Apply Leibnitz's rule to move all D_x inside the inner sum. Then,

$$\begin{aligned}
& \sum_{i=0}^{M_1^j-1} D_x^i \left(u^j \sum_{k=i+1}^{M_1^j} \binom{k}{i+1} (-D_x)^{k-(i+1)} \frac{\partial f}{\partial u_{kx}^j} \right) \\
&= \sum_{i=0}^{M_1^j-1} \sum_{m=0}^i \binom{i}{m} u_{mx}^j D_x^{i-m} \left(\sum_{k=i+1}^{M_1^j} \binom{k}{i+1} (-D_x)^{k-(i+1)} \frac{\partial f}{\partial u_{kx}^j} \right) \\
&= \sum_{i=0}^{M_1^j-1} \sum_{m=0}^i \binom{i}{m} u_{mx}^j (-1)^{m-i} \sum_{k=i+1}^{M_1^j} \binom{k}{i+1} (-D_x)^{k-(m+1)} \frac{\partial f}{\partial u_{kx}^j}.
\end{aligned}$$

Now, the sums over i and m are interchanged to bring u_{mx}^j up front. Then the sums over i and k are interchanged so that D_x and $\partial f / \partial u_{kx}^j$ come outside the sum over i . This leaves only a combinatoric expression inside the sum over i .

$$\begin{aligned}
& \sum_{i=0}^{M_1^j-1} \sum_{m=0}^i \binom{i}{m} u_{mx}^j (-1)^{m-i} \sum_{k=i+1}^{M_1^j} \binom{k}{i+1} (-D_x)^{k-(m+1)} \frac{\partial f}{\partial u_{kx}^j} \\
&= \sum_{m=0}^{M_1^j-1} u_{mx}^j \sum_{i=m}^{M_1^j-1} \binom{i}{m} (-1)^{m-i} \sum_{k=i+1}^{M_1^j} \binom{k}{i+1} (-D_x)^{k-(m+1)} \frac{\partial f}{\partial u_{kx}^j} \\
&= \sum_{m=0}^{M_1^j-1} u_{mx}^j \sum_{k=m+1}^{M_1^j} (-D_x)^{k-(m+1)} \frac{\partial f}{\partial u_{kx}^j} \sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \binom{k}{i+1}. \tag{6.30}
\end{aligned}$$

The identity

$$\sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \binom{k}{i+1} = 1 \tag{6.31}$$

holds for $k \geq m+1$, a condition that is met in (6.30). A proof for (6.31) is given in the appendix. In the final step, the order of the summations is changed,

$$\sum_{m=0}^{M_1^j-1} u_{mx}^j \sum_{k=m+1}^{M_1^j} (-D_x)^{k-(m+1)} \frac{\partial f}{\partial u_{kx}^j} = \sum_{k=1}^{M_1^j} \left(\sum_{m=0}^{k-1} u_{mx}^j (-D_x)^{k-(m+1)} \right) \frac{\partial f}{\partial u_{kx}^j},$$

which gives the right hand side of (6.29). ■

The integrand (6.16) has been implemented in `HomotopyIntegrator.m`. Due to the elimination of the binomial coefficient and the reduced number of times the D_x operator is called inside the sums, it should be obvious that (6.16) is more efficient than (6.3). In extensive testing, (6.16) consistently outperformed the original version (6.3),

dramatically reducing CPU time on complex expressions. Table 6.1 shows some test results comparing run times on the two integrands.

Furthermore, the code has functioned well on a variety of expressions. However, like any other symbolic integrator, if the expression being integrated is very complex, the homotopy operator may not be able to integrate it. Yet, it is able to integrate many expressions that *Mathematica's* Integrate function cannot integrate.

Table 6.1: Run time comparisons of one-dimensional homotopy integrands.

Trial	Form of Expression	Number of Terms	Highest Order	Original Integrand (6.3) CPU time (s)	New Integrand (6.16) CPU time (s)
1	Polynomial	3	7	0.0040	0.0020
2	Polynomial	6	10	0.017	0.0050
3	Polynomial	4	7	0.050	0.0090
4	Trigonometric	3	7	0.019	0.0030
5	Square Root	3	6	0.576	0.073
6	Rational	6	8	38.31	4.65
7	Rational	9	8	493.3	31.45

Example 6.5. *Mathematica was unable to integrate f in (6.1). Using the homotopy operator with integrand (6.16) and letting $(u^1, u^2) = (u, v)$, the integrands are*

$$\begin{aligned}
\mathcal{I}_u f &= (u\mathbf{l}) \frac{\partial f}{\partial u_x} + (u(-\mathbf{D}_x) + u_x\mathbf{l}) \frac{\partial f}{\partial u_{2x}} \\
&= u(2u_{2x} \cos u + v \sin u - 3u_x^2 \sin u) - u\mathbf{D}_x(2u_x \cos u) + u_x(2u_x \cos u) \\
&= uv \sin u - uu_x^2 \sin u + 2u_x^2 \cos u,
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{I}_v f &= (v\mathbf{l}) \frac{\partial f}{\partial v_x} + (v(-\mathbf{D}_x) + v_x\mathbf{l}) \frac{\partial f}{\partial v_{2x}} \\
&= v(-\cos u + 2v_{2x}) - v\mathbf{D}_x(2v_x) + v_x(2v_x) \\
&= -v \cos u + 2v_x^2.
\end{aligned}$$

Replace u and v with λu and λv , then integrate with respect to λ to get

$$\begin{aligned}
\mathcal{H}_{\mathbf{u}(x)} f &= \int_0^1 (\mathcal{I}_u f[\lambda \mathbf{u}] + \mathcal{I}_v f[\lambda \mathbf{u}]) \frac{d\lambda}{\lambda} \\
&= \int_0^1 (\lambda uv \sin(\lambda u) - \lambda^2 uu_x^2 \sin(\lambda u) + 2\lambda u_x^2 \cos(\lambda u) - v \cos(\lambda u) + 2\lambda v_x^2) d\lambda
\end{aligned}$$

$$\begin{aligned}
&= [\lambda^2 u_x^2 \cos u - \lambda v \cos u + \lambda^2 v_x^2]_{\lambda=0}^{\lambda=1} \\
&= u_x^2 \cos u - v \cos u + v_x^2
\end{aligned}$$

The code `HomotopyIntegrator.m` produces the same result.

6.3.2 The Homotopy Operator for Two Independent Variables

After the new integrand (6.16) was tested, an investigation began to see if similar versions could be derived for the two and three independent variables cases. Unlike the one-dimensional case, it was not possible to completely eliminate the binomial coefficients from the multi-dimensional integrands. However, a new version was obtained which greatly reduced the swell of terms that occurred with the integrands containing the higher-Euler operators, (6.8) and (6.9).

For the two independent variables case, let $\mathbf{x} = (x, y)$. The two-dimensional homotopy operator is revised as follows.

Definition 6.7. *The homotopy operator for two independent variables where $\mathbf{x} = (x, y)$, is the two-component vector (6.5), where the x - and y -components are given by (6.6) and (6.7), respectively. For $f(x, y, \mathbf{u}^{(M)}(x, y))$, the new integrand, $\mathcal{I}_{u^j(x,y)}^{(x)} f$, replacing (6.8), is given as*

$$\begin{aligned}
\mathcal{I}_{u^j(x,y)}^{(x)} f = & \\
& \sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \left(\sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} \frac{\binom{i_1+i_2}{i_1} \binom{k_1+k_2-i_1-i_2-1}{k_1-i_1-1}}{\binom{k_1+k_2}{k_1}} u_{i_1 x i_2 y}^j (-D_x)^{k_1-i_1-1} (-D_y)^{k_2-i_2} \right) \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}.
\end{aligned} \tag{6.32}$$

Similarly, instead of (6.9), the new integrand, $\mathcal{I}_{u^j(x,y)}^{(y)} f$, is

$$\begin{aligned}
\mathcal{I}_{u^j(x,y)}^{(y)} f = & \\
& \sum_{k_1=0}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2-1} \frac{\binom{i_1+i_2}{i_2} \binom{k_1+k_2-i_1-i_2-1}{k_2-i_2-1}}{\binom{k_1+k_2}{k_2}} u_{i_1 x i_2 y}^j (-D_x)^{k_1-i_1} (-D_y)^{k_2-i_2-1} \right) \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}.
\end{aligned} \tag{6.33}$$

Like the one-dimensional case, the higher-Euler operators (6.10) are no longer needed.

The key proof in this section will show that the two-dimensional homotopy operator will invert a divergence, producing a two-component vector. The preliminary definitions and theorems from the one-dimensional case are adapted to the two-dimensional case. First, the \mathbf{M} operator is changed as follows.

Definition 6.8. *For the two independent variables case, the degree operator \mathbf{M} applied to $f(x, y, \mathbf{u}^{(M)}(x, y))$ is given by*

$$g = \mathbf{M}f = \sum_{j=1}^N \sum_{i_1=0}^{M_1^j} \sum_{i_2=0}^{M_2^j} u_{i_1 x i_2 y}^j \frac{\partial f}{\partial u_{i_1 x i_2 y}^j}. \quad (6.34)$$

Analogous to Theorem 6.1,

$$\mathbf{M}^{-1}g(x, y, \mathbf{u}^{(M)}(x, y)) = \int_{\lambda_0}^1 g[\lambda \mathbf{u}] \frac{d\lambda}{\lambda}. \quad (6.35)$$

The proof for the two-dimensional case is the same as for the one-dimensional case.

The development of the homotopy operator for functions with two independent variables is shown in the following theorem. Since the proof is lengthy, only an outline of the proof is given here. A fuller development of the proof is given in the appendix.

Theorem 6.5. *Let $f(x, y, \mathbf{u}^{(M)}(x, y))$ be an exact differential function and define the vector $(\mathcal{H}_{\mathbf{u}(x,y)}^{(x)}f, \mathcal{H}_{\mathbf{u}(x,y)}^{(y)}f)$ to be (6.6) and (6.7) with integrands (6.32) and (6.33). Then*

$$\text{Div}^{-1}f = (\mathcal{H}_{\mathbf{u}(x,y)}^{(x)}f, \mathcal{H}_{\mathbf{u}(x,y)}^{(y)}f). \quad (6.36)$$

SKETCH OF THE PROOF: The proof for this theorem is constructed in the same manner as the proof of Theorem 6.3. To start, for a chosen u^j multiply $\mathcal{L}_{u^j(x,y)}f$ by u^j to restore the degree. The first step involves splitting off $u^j \frac{\partial f}{\partial u^j}$, then splitting the remaining sum into a part with k_1 set equal to 0, a part with k_2 set equal to 0, and a part with the remaining terms. The $k_1 = 0$ and $k_2 = 0$ parts are treated as in Theorem 6.3, that is by splitting off all $u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j}$ and all $u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j}$ terms. These are all terms found in the \mathbf{M} operator given in Definition 6.8. In detail,

$$\begin{aligned} u^j \mathcal{L}_{u^j(x,y)}f &= u^j \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\ &= u^j \frac{\partial f}{\partial u^j} + u^j \sum_{k_1=1}^{M_1^j} (-D_x)^{k_1} \frac{\partial f}{\partial u_{k_1 x}^j} + u^j \sum_{k_2=1}^{M_2^j} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_2 y}^j} \\ &\quad + u^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \end{aligned}$$

$$\begin{aligned}
&= u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^{M_1^j} u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} - D_x \left[\sum_{i_1=0}^{M_1^j-1} u_{i_1 x}^j \sum_{k_1=i_1+1}^{M_1^j} (-D_x)^{k_1-(i_1+1)} \frac{\partial f}{\partial u_{k_1 x}^j} \right] \\
&\quad + \sum_{k_2=1}^{M_2^j} u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} - D_y \left[\sum_{i_2=0}^{M_2^j-1} u_{i_2 y}^j \sum_{k_2=i_2+1}^{M_2^j} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_2 y}^j} \right] \\
&\quad + u^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \tag{6.37}
\end{aligned}$$

The next step splits the term $u^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}$ in (6.37). Keeping in mind that the goal is to produce an expression in the form $Mf - ((D_x(\dots) + D_y(\dots)))$, it is necessary to split this term into a part that comes under the D_x operator and a part that comes under the D_y operator, followed by integration by parts. The fraction that goes in each part is determined by comparing the number of times D_x is applied in the term to the number of times D_y is applied. In the next step, the total number of derivatives taken is $k_1 + k_2$. In the D_x part, the fraction of the number of derivatives taken is $\frac{k_1}{k_1 + k_2}$. In the D_y part, the fraction of the number of derivatives taken is $\frac{k_2}{k_1 + k_2}$. Following this approach, the last term (6.37) becomes

$$\begin{aligned}
&u^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&= -D_x \left[u^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\frac{k_1}{k_1 + k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
&\quad - D_y \left[u^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\frac{k_2}{k_1 + k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
&\quad + u_x^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\frac{k_1}{k_1 + k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&\quad + u_y^j \sum_{k_1=1}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\frac{k_2}{k_1 + k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \tag{6.38}
\end{aligned}$$

Continue by first integrating the last two terms of (6.38) by parts, then splitting each sum into a fraction that goes in the D_x part and a fraction that goes in the D_y part. This process continues until there are no derivatives left to split. Consolidating the D_x

and D_y parts eventually yields

$$\begin{aligned}
u^j \mathcal{L}_{u^j(x,y)} f &= \sum_{i_1=0}^{M_1^j} \sum_{i_2=0}^{M_2^j} u_{i_1 x i_2 y}^j \frac{\partial f}{\partial u_{i_1 x i_2 y}^j} \\
&- \left[D_x \left(\sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \left(\sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} C_1 u_{i_1 x i_2 y} (-D_x)^{k_1-i_1-1} (-D_y)^{k_2-i_2} \right) \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \right. \\
&\left. + D_y \left(\sum_{k_1=0}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2-1} C_2 u_{i_1 x i_2 y} (-D_x)^{k_1-i_1} (-D_y)^{k_2-i_2-1} \right) \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \right], \quad (6.39)
\end{aligned}$$

where

$$C_1 = \frac{\binom{i_1+i_2}{i_1} \binom{k_1+k_2-i_1-i_2-1}{k_1-i_1-1}}{\binom{k_1+k_2}{k_1}} \quad \text{and} \quad C_2 = \frac{\binom{i_1+i_2}{i_2} \binom{k_1+k_2-i_1-i_2-1}{k_2-i_2-1}}{\binom{k_1+k_2}{k_2}}.$$

Note that in (6.39), the expression that D_x operates on is $\mathcal{I}_{u^j(x,y)}^{(x)}$ in (6.32) and the expression that D_y operates on is $\mathcal{I}_{u^j(x,y)}^{(y)}$ in (6.33). Next, sum over all of the components u^j to get

$$\sum_{j=1}^N u^j \mathcal{L}_{u^j(x,y)} f = Mf - \sum_{j=1}^N \left(D_x \mathcal{I}_{u^j(x,y)}^{(x)} f + D_y \mathcal{I}_{u^j(x,y)}^{(y)} f \right), \quad (6.40)$$

Since f is exact, $\mathcal{L}_{u^j(x,y)} f = 0$ for $j = 1, \dots, N$, thus $\sum_{j=1}^N u^j \mathcal{L}_{u^j(x,y)} f = 0$. Equation (6.40) can be rewritten as

$$Mf = \sum_{j=1}^N \left(D_x \mathcal{I}_{u^j(x,y)}^{(x)} f + D_y \mathcal{I}_{u^j(x,y)}^{(y)} f \right). \quad (6.41)$$

Applying M^{-1} to both sides of (6.41) and using Theorem 6.2 gives

$$\begin{aligned}
f &= \sum_{j=1}^N M^{-1} \left(D_x \mathcal{I}_{u^j(x,y)}^{(x)} f + D_y \mathcal{I}_{u^j(x,y)}^{(y)} f \right) \\
&= M^{-1} D_x \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(x)} f + M^{-1} D_y \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(y)} f \\
&= D_x M^{-1} \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(x)} f + D_y M^{-1} \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(y)} f \quad \text{by Theorem 6.2} \\
&= \text{Div} \left(M^{-1} \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(x)} f, M^{-1} \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(y)} f \right), \quad (6.42)
\end{aligned}$$

where $M^{-1} \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(x)} f = \mathcal{H}_{\mathbf{u}(x,y)}^{(x)}$ and $M^{-1} \sum_{j=1}^N \mathcal{I}_{u^j(x,y)}^{(y)} f = \mathcal{H}_{\mathbf{u}(x,y)}^{(y)}$. Applying the inverse of the divergence operator to both sides of (6.42) gives (6.36). \blacksquare

Again, it can be shown that the integrands developed in Theorem 6.5 and the integrand given in [35, 56] are the same. Indeed, Theorem 6.6 establishes the equivalence of the two x -integrands, (6.8) and (6.32). The proof for the y -integrands is analogous.

Theorem 6.6.

$$\begin{aligned} & \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \left(\frac{1+i_1}{1+i_1+i_2} \right) D_x^{i_1} D_y^{i_2} \left(u^j \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \\ &= \sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \left(\sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} \frac{\binom{i_1+i_2}{i_1} \binom{k_1+k_2-i_1-i_2-1}{k_1-(i_1+1)}}{\binom{k_1+k_2}{k_1}} u_{i_1 x i_2 y} (-D_x)^{k_1-i_1-1} (-D_y)^{k_2-i_2} \right) \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \end{aligned}$$

The proof of Theorem 6.6 is similar to that of Theorem 6.4. However, some of the combinatoric manipulations require several lengthy steps and new combinatoric identities must be established, so the proof is given in the appendix.

As in the one-dimensional case, integrands (6.32) and (6.33) are implemented in the code of `HomotopyIntegrator.m`. Extensive testing has shown that the new integrands significantly reduce both expression swell and run time. Table 6.2 shows a sample of test results comparing both versions. Trial 5 in Table 6.2 especially emphasizes the reduction in expression swell.

Table 6.2: Run time comparisons of two-dimensional homotopy integrands.

Trial	Original Integrands (6.8) and (6.9)		New Integrands (6.32) and (6.33)		Both Integrands			
	Number of terms before applying outer sum	Run Time (s)	Number of terms before applying outer sum	Run Time (s)	Number of terms after applying outer sum			
	x	y	x	y	x	y		
1	31	38	0.0090	25	38	0.0080	8	10
2	521	750	0.130	165	195	0.050	67	76
3	3674	10015	2.73	718	1369	0.312	264	488
4	17802	16843	6.55	3680	2729	0.767	1312	1033
5	1332643	591251	997.3	83289	49879	138.7	83276	49867
6	10747	180250	101.0	2723	2189	7.80	2027	1635

Although the new integrands have improved the efficiency of the homotopy operator, they both produce exactly the same result. As can be seen in the last column

of Table 6.2, the output can be quite large. The inversion of a divergence on a function with multiple independent variables can only be determined up to a divergence-free vector, \mathbf{K} . For the two-dimensional case $\mathbf{K} = (D_x\phi, -D_y\phi)$, where ϕ is an arbitrary scalar differential function. A reduction algorithm has been created to remove \mathbf{K} leaving the output in much simpler form. The reduction algorithm will be presented in Section 6.4.

In the next example, the homotopy operator inverts a divergence, with the result being a vector consisting of differential functions.

Example 6.6. Let $\mathbf{u} = (u^1, u^2) = (u, v)$ and $f(x, y, \mathbf{u}^{(4)}(x, y)) = 3u_y^2u_{xy} + 3v_{3xy}$, where f is exact. First, the integrands are calculated using (6.32) and (6.33), so that

$$\begin{aligned}\mathcal{I}_{u(x,y)}^{(x)}f &= \left(-\frac{1}{2}uD_y + \frac{1}{2}u_y\mathbf{l}\right) \frac{\partial f}{\partial u_{xy}} \\ &= -3uu_yu_{2y} + \frac{3}{2}u_y^3, \\ \mathcal{I}_{u(x,y)}^{(y)}f &= (u\mathbf{l}) \frac{\partial f}{\partial u_y} - \left(\frac{1}{2}uD_x - \frac{1}{2}u_x\mathbf{l}\right) \frac{\partial f}{\partial u_{xy}} \\ &= 3uu_yu_{xy} + \frac{3}{2}u_xu_y^2, \\ \mathcal{I}_{v(x,y)}^{(x)}f &= \left(-\frac{3}{4}vD_x^2D_y + \frac{1}{4}v_yD_x^2 + \frac{1}{2}v_xD_xD_y - \frac{1}{2}v_{xy}D_x - \frac{1}{4}v_{2x}D_y + \frac{3}{4}v_{2xy}\mathbf{l}\right) \frac{\partial f}{\partial v_{3xy}} \\ &= \frac{9}{4}v_{2xy}, \\ \mathcal{I}_{v(x,y)}^{(y)}f &= \left(-\frac{1}{4}vD_x^3 + \frac{1}{4}v_xD_x^2 - \frac{1}{4}v_{2x}D_x + \frac{1}{4}v_{3x}\mathbf{l}\right) \frac{\partial f}{\partial v_{3xy}} \\ &= \frac{3}{4}v_{3x}.\end{aligned}$$

First, replace u with λu , v with λv , etc., then apply the homotopy, (6.6) and (6.7) to get

$$\begin{aligned}\mathcal{H}_{\mathbf{u}(x,y)}^{(x)}f &= \int_0^1 \left(\mathcal{I}_{u(x,y)}^{(x)}f[\lambda\mathbf{u}] + \mathcal{I}_{v(x,y)}^{(x)}f[\lambda\mathbf{u}]\right) \frac{d\lambda}{\lambda} \\ &= \int_0^1 \left(\frac{3}{2}\lambda^2u_y^3 - 3\lambda^2uu_yu_{2y} + \frac{9}{4}v_{2xy}\right) d\lambda \\ &= \left[\frac{1}{2}\lambda^3u_y^3 - \lambda^3uu_yu_{2y} + \frac{9}{4}\lambda v_{2xy}\right]_0^1 \\ &= \frac{1}{2}u_y^3 - uu_yu_{2y} + \frac{9}{4}v_{2xy}, \\ \mathcal{H}_{\mathbf{u}(x,y)}^{(y)}f &= \int_0^1 \left(\mathcal{I}_{u(x,y)}^{(y)}f[\lambda\mathbf{u}] + \mathcal{I}_{v(x,y)}^{(y)}f[\lambda\mathbf{u}]\right) \frac{d\lambda}{\lambda} \\ &= \int_0^1 \left(3\lambda^2uu_yu_{xy} + \frac{3}{2}\lambda^2u_xu_y^2 + \frac{3}{4}v_{3x}\right) d\lambda \\ &= \left[\lambda^3uu_yu_{xy} + \frac{1}{2}\lambda^3u_xu_y^2 + \frac{3}{4}\lambda v_{3x}\right]_0^1 \\ &= uu_yu_{xy} + \frac{1}{2}u_xu_y^2 + \frac{3}{4}v_{3x}.\end{aligned}$$

The homotopy operator gives the vector

$$\mathbf{F} = \text{Div}^{-1} f = \left(\frac{1}{2}u_y^3 - uu_yu_{2y} + \frac{9}{4}v_{2xy}, uu_yu_{xy} + \frac{1}{2}u_xu_y^2 + \frac{3}{4}v_{3x} \right). \quad (6.43)$$

This result can be reduced to

$$\tilde{\mathbf{F}} = (u_y^3, 3v_{3x}) \quad (6.44)$$

by removing the divergence-free vector $\mathbf{K} = (D_y\phi, -D_x\phi)$, where $\phi = -\frac{1}{2}uu_y^2 + \frac{9}{4}v_{2x}$.

The code in `HomotopyIntegrator.m` returns (6.44).

6.3.3 The Homotopy Operator for Three Independent Variables

The homotopy operator for three independent variables is presented here. The development of three-dimensional version uses the same techniques as those for two-dimensional version. The proofs are similar, but lengthier, and therefore omitted. For brevity, only the simplified form of the operator is shown.

Definition 6.9. *The homotopy operator for three independent variables, $\mathbf{x} = (x, y, z)$, produces a three-component vector (6.11), where the x -component is given in (6.12) and the y - and z -components are defined analogously. The new integrand replacing (6.13) is*

$$\begin{aligned} I_{u^j(x,y,z)}^{(x)} f &= \sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \sum_{k_3=0}^{M_3^j} \sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} \sum_{i_3=0}^{k_3} \\ &\quad \frac{\binom{i_1+i_2+i_3}{i_1} \binom{i_2+i_3}{i_2} \binom{k_1+k_2+k_3-i_1-i_2-i_3-1}{k_1-i_1-1} \binom{k_2+k_3-i_2-i_3}{k_2-i_2}}{\binom{k_1+k_2+k_3}{k_1} \binom{k_2+k_3}{k_2}} u_{i_1x i_2y i_3z}^j \\ &\quad (-D_x)^{k_1-i_1-1} (-D_y)^{k_2-i_2} (-D_z)^{k_3-i_3} \frac{\partial f}{\partial u_{k_1x k_2y k_3z}^j}. \end{aligned} \quad (6.45)$$

The integrands $I_{u^j(x,y,z)}^{(y)} f$ and $I_{u^j(x,y,z)}^{(z)} f$ are similar. Indeed,

$$\begin{aligned} I_{u^j(x,y,z)}^{(y)} f &= \sum_{k_1=0}^{M_1^j} \sum_{k_2=1}^{M_2^j} \sum_{k_3=0}^{M_3^j} \sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2-1} \sum_{i_3=0}^{k_3} \\ &\quad \frac{\binom{i_1+i_2+i_3}{i_2} \binom{i_1+i_3}{i_3} \binom{k_1+k_2+k_3-i_1-i_2-i_3-1}{k_2-i_2-1} \binom{k_1+k_3-i_1-i_3}{k_3-i_3}}{\binom{k_1+k_2+k_3}{k_2} \binom{k_1+k_3}{k_3}} u_{i_1x i_2y i_3z}^j \\ &\quad (-D_x)^{k_1-i_1} (-D_y)^{k_2-i_2-1} (-D_z)^{k_3-i_3} \frac{\partial f}{\partial u_{k_1x k_2y k_3z}^j}, \end{aligned} \quad (6.46)$$

and

$$\begin{aligned}
I_{u^j(x,y,z)}^{(z)} f &= \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} \sum_{k_3=1}^{M_3^j} \sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2} \sum_{i_3=0}^{k_3-1} \\
&\quad \frac{\binom{i_1+i_2+i_3}{i_3} \binom{i_1+i_2}{i_1} \binom{k_1+k_2+k_3-i_1-i_2-i_3-1}{k_3-i_3-1} \binom{k_1+k_2-i_1-i_2}{k_1-i_1}}{\binom{k_1+k_2+k_3}{k_3} \binom{k_1+k_2}{k_1}} u_{i_1 x i_2 y i_3 z}^j \\
&\quad (-D_x)^{k_1-i_1} (-D_y)^{k_2-i_2} (-D_z)^{k_3-i_3-1} \frac{\partial f}{\partial u_{k_1 x k_2 y k_3 z}^j}. \tag{6.47}
\end{aligned}$$

The three-dimensional version has also been extensively tested. The results were similar to those for the one- and two-dimensional versions. Expression swell is more pronounced as the number of independent variables increases. In contrast to the integrands that were expressed in terms of higher-Euler operators, the new integrands, (6.45), (6.46), and (6.47), largely prevent unwieldy expression swell. The three-dimensional version also produces a result containing a divergence-free vector, $\nabla \times \mathbf{K}$, where \mathbf{K} is an arbitrary differential vector function. Again, the vector $\nabla \times \mathbf{K}$ needs to be removed from the result.

6.4 Removing Divergence-Free Terms

As mentioned in the previous section, the homotopy operator acting on exact expressions with more than one independent variable will produce a vector result that often contains a divergence-free (curl) vector. For a particular expression, the choice for divergence-free vectors is infinite. The homotopy operator computes one choice for that expression and will consistently reproduce the divergence-free vector. A result containing a divergence-free vector can be of unmanageable size, whereas the result without the divergence-free part is generally concise.

Several methods exist to remove divergence-free vectors. To do this by hand requires educated guesses as to what the divergence-free piece contains, a process difficult to program. The method implemented in `HomotopyIntegrator.m` uses linear algebra. Indeed, all it requires is the solution of a linear system which is efficient and easy to program.

The method for removing curl terms will be explained using (6.43), the result from Example 6.6. To begin with, attach undetermined coefficients, k_1, \dots, k_p , to the

terms of (6.43) and call the new vector $\tilde{\mathbf{F}}$. This gives

$$\tilde{\mathbf{F}} = \begin{pmatrix} \frac{1}{2}k_1u_y^3 - k_2uu_yu_{2y} + \frac{9}{4}k_3v_{2xy} \\ k_4uu_yu_{xy} + \frac{1}{2}k_5u_xu_y^2 + \frac{3}{4}k_6v_{3x} \end{pmatrix}. \quad (6.48)$$

Apply the divergence operator (2.5) to both (6.43) and (6.48), producing two corresponding divergences,

$$\begin{aligned} f &= \text{Div } \mathbf{F} \\ &= 3u_y^2u_{xy} + 3v_{3xy} \end{aligned} \quad (6.49)$$

and

$$\begin{aligned} \tilde{f} &= \text{Div } \tilde{\mathbf{F}} \\ &= \frac{3}{2}k_1u_y^2u_{xy} - k_2u_xu_yu_{2y} - k_2uu_{xy}u_{2y} - k_2uu_yu_{x2y} + \frac{9}{4}k_3v_{3xy} + k_4u_y^2u_{xy} \\ &\quad + k_4uu_{xy}u_{2y} + k_4uu_yu_{x2y} + \frac{1}{2}k_5u_y^2u_{xy} + k_5u_xu_yu_{2y} + \frac{3}{4}k_6v_{3xy} \\ &= \left(\frac{3}{2}k_1 + k_4 + \frac{1}{2}k_5\right)u_y^2u_{xy} + (k_5 - k_2)u_xu_yu_{2y} + (k_4 - k_2)uu_{xy}u_{2y} \\ &\quad + (k_4 - k_2)uu_yu_{x2y} + \left(\frac{9}{4}k_3 + \frac{3}{4}k_6\right)v_{3xy}. \end{aligned} \quad (6.50)$$

Since \mathbf{F} and $\tilde{\mathbf{F}}$ differ only by a curl vector, their divergences are equal, that is, f must equal \tilde{f} . Gathering like terms in (6.49) and (6.50) produces the linear system

$$\begin{aligned} \frac{1}{2}(3k_1 + 2k_4 + k_5) &= 3, \\ k_5 - k_2 &= 0, \\ k_4 - k_2 &= 0, \\ \frac{3}{4}(3k_3 + k_6) &= 3. \end{aligned} \quad (6.51)$$

Removing the divergence-free vector from \mathbf{F} can be tricky. It is possible to remove only part of the divergence-free vector, leaving a divergence-free part behind. Furthermore, there may be more than one choice for the vector that completely removes the divergence-free part. Thus, the manner in which the system is solved becomes important.

Mathematica's `Solve` function solves a system of equations working through a list of variables in order. It takes the first variable in a list and will either find a specific solution, or it will solve for that variable in terms of subsequent variables. If the variable

is found to be arbitrary, `Solve` moves on to the next variable in the list and does not report the arbitrary variable. If `Solve` is given the list $\{k_1, \dots, k_6\}$, the solution to the system (6.51) is

$$k_1 = 2 - k_5, \quad k_2 = k_5, \quad k_3 = \frac{4}{3} - \frac{1}{3}k_6, \quad k_4 = k_5, \quad (6.52)$$

where k_5 and k_6 are arbitrary. Since it is desirable to eliminate as many terms as possible, choose $k_5 = 0$ and $k_6 = 0$. With this choice, $k_1 = 2$, $k_2 = 0$, $k_3 = \frac{4}{3}$, and $k_4 = 0$. Substituting this solution into (6.48) produces the vector

$$\mathbf{F} = (u_y^3 + 3v_{3x}, 0), \quad (6.53)$$

a simpler form, but perhaps not the most desirable result.

To find the solution to the system of undetermined coefficients that will produce a result without any divergence-free part, take the coefficients, k_i , from the equations where the right-hand side is not zero and place them in the list of variables for `Solve` first, then add all other coefficients. From the system (6.51), k_1 , k_3 , and k_6 are placed at the front of the list, and the rest are added to get $\{k_1, k_3, k_6, k_2, k_4, k_5\}$, which is the order that `Solve` will use. The coefficients k_1 , k_3 , and k_6 are taken first because they are coefficients of terms that must be in the divergence. For this simple case, the solution to the system produces the result in (6.53).

Reversing the order of the k_i may change the divergence-free result by moving a term that has mixed partial derivatives to a different component, possibly leading to a more desirable result. For the system (6.51), if the list for `Solve` is ordered as $\{k_6, k_3, k_1, k_2, k_4, k_5\}$, the solution is

$$k_6 = 4 - 3k_3, \quad k_1 = 2 - k_5, \quad k_2 = k_5, \quad k_4 = k_5, \quad (6.54)$$

with k_3 and k_5 arbitrary. Since k_3 and k_5 are free, set them equal to zero. This gives $k_1 = 2$, $k_2 = 0$, $k_4 = 0$, and $k_6 = 4$. Replacing all k_i in (6.48) with these solutions produces (6.44), the simplest possible vector that can be found with balanced components.

The reduction algorithm in `HomotopyIntegrator.m` has been tested extensively on both two and three independent variable vector functions. By placing the constants into the solve list in the correct order, the algorithm runs quickly and returns the simplest possible vector, even when thousands of terms are involved. The next example

shows a simple three-dimensional case for which the homotopy operator returns a large divergence-free term.

Example 6.7. Let $\mathbf{u} = (u, v)$, and let

$$\mathbf{F}(x, y, z, \mathbf{u}^{(7)}(x, y, z)) = (u_{y3z}^2 - vu_{xy}, u^3u_{x6y}, v_y^2 - uv_{5x2z}). \quad (6.55)$$

Applied to $f = \text{Div}\mathbf{F}$, the homotopy operator produces a vector with 42 terms in the x -component, 36 terms in the y -component, and 32 terms in the z -component. The reduction algorithm is able to return (6.55) using 0.25 seconds of CPU time. When f is given to `HomotopyIntegrator.m`, the program will return (6.55).

6.5 Limitations of the Homotopy Operator

The homotopy operator has proven to be quite versatile in tests using a wide variety of functions. However, there are two possible limitations to its use. When either one is encountered, it may be necessary to pursue another avenue to complete the integration.

One advantage of the homotopy operator is that it replaces an integration by parts problem on a function with one independent variable with an integration over an auxiliary variable that can often be done using direct methods. In multi-variable cases, the integration over the auxiliary variable replaces a possible integration over multiple variables. It is possible, however, that if the given expression is a complex composition of functions, even the integral for the auxiliary variable may be too complex to integrate. The code `HomotopyIntegrator.m` will print a message when this happens.

The second limitation occurs with exact expressions in rational form. If a function $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ is rational and f is in the kernel of the degree operator, \mathbf{M} , then the homotopy operators as defined in Definitions 6.5, 6.7, and 6.9 will not work, as discussed in Case 3 of the proof for Theorem 6.1. Consider the one-dimensional case. In Theorem 6.3, it was shown that if f is an exact differential function, then

$$\mathbf{M}f = D_x \sum_{j=1}^N (\mathcal{I}_{u^j(x)} f), \quad (6.56)$$

for dependent variable u^j . When $f \in \text{Ker } \mathbf{M}$, that is, when $\mathbf{M}f = 0$, then $\sum_{j=1}^n \mathcal{I}_{u^j(x)} f = C$, $C \in \mathbb{R}$, so the homotopy operator will return a constant.

Example 6.8. Let $\mathbf{u} = (u^1, u^2) = (u, v)$, and $F = \frac{u_x}{v}$. Then $f = \mathbf{D}_x F = \frac{u_{2x}v - u_x v_x}{v^2}$.

For this case,

$$\begin{aligned} \mathbf{M}f &= u_x \frac{\partial f}{\partial u_x} + u_{2x} \frac{\partial f}{\partial u_{2x}} + v \frac{\partial f}{\partial v} + v_x \frac{\partial f}{\partial v_x} \\ &= -u_x \frac{v_x}{v^2} + u_{2x} \frac{1}{v} + v \frac{2u_x v_x - u_{2x} v}{v^3} - v_x \frac{u_x}{v^2} \\ &= 0. \end{aligned}$$

If we try to integrate f using the homotopy operator, the integrands are

$$\begin{aligned} I_{u(x)}^{(x)} &= u \frac{\partial f}{\partial u_x} - (u \mathbf{D}_x - u_x \mathbf{l}) \frac{\partial f}{\partial u_{2x}} & I_{v(x)}^{(x)} &= v \frac{\partial f}{\partial v_x} \\ &= -u \frac{v_x}{v^2} + u \frac{v_x}{v^2} + \frac{u_x}{v} & \text{and} & &= -v \frac{u_x}{v^2} \\ &= \frac{u_x}{v}, & & &= -\frac{u_x}{v}. \end{aligned}$$

Clearly, $I_{u(x)}^{(x)} + I_{v(x)}^{(x)} = 0$. Thus, the homotopy operator returns a constant instead of F .

The same problem occurs for functions with two or three independent variables.

If f is composed of several terms and one of the terms is in $\text{Ker } \mathbf{M}$, then the homotopy operator will give a false result, since it will integrate all terms not in $\text{Ker } \mathbf{M}$, but not the terms in $\text{Ker } \mathbf{M}$. The following example highlights the problem.

Example 6.9. Let $\mathbf{u} = (u^1, u^2) = (u, v)$. Take $F = \frac{u^2 + v}{u - v}$. Then

$$f = \mathbf{D}_x F = \frac{u^2 u_x + u^2 v_x - 2u u_x v + u v_x - u_x v}{(u - v)^2}$$

is exact. Applying the degree operator to f gives

$$\mathbf{M}f = \frac{u^2 u_x}{(u - v)^2} + \frac{u^2 v_x}{(u - v)^2} - \frac{2u v u_x}{(u - v)^2} + 0 + 0.$$

Because 0 occurs for two of the terms, the homotopy operator cannot integrate this expression. If it is allowed to continue, it will produce $F = \frac{u^2}{u - v}$, which is an incorrect solution.

The origin for the jet space J^M is set at $(0, \dots, 0)$ which is inappropriate for cases where $f \in \text{Ker } \mathbf{M}$. If the center is shifted away from $(0, \dots, 0)$, then f is shifted out of $\text{Ker } \mathbf{M}$. It suffices to shift the coordinates that appear in the denominator of the term in $\text{Ker } \mathbf{M}$. Once the integration is complete, the shift can be inverted, putting the result back at the original center. The following example illustrates how the shift works.

Example 6.10. Let $f(x, \mathbf{u}^{(2)}(x)) = \frac{u_{2x}v - u_x v_x}{v^2}$, where $f \in \text{Ker } \mathbf{M}$. Since v is in the denominator, shift v by v_0 , that is, replace v with $v - v_0$, to get $f_0 = \frac{u_{2x}(v - v_0) - u_x v_x}{(v - v_0)^2}$. Using the homotopy operator (6.2) to integrate f_0 , first compute the integrands (6.16) to get

$$\mathcal{I}_{\mathbf{u}(x)}^{(x)} f_0 = \frac{u_x}{v - v_0} \quad \text{and} \quad \mathcal{I}_{v(x)}^{(x)} f_0 = \frac{-u_x v}{(v - v_0)^2}.$$

Then,

$$\begin{aligned} \mathcal{H}_{\mathbf{u}(x)} f_0 &= \int_0^1 \left(\frac{u_x}{v - v_0} - \frac{u_x v}{(v - v_0)^2} \right) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda} \\ &= - \int_0^1 \frac{u_x v_0}{(\lambda v - v_0)^2} d\lambda \\ &= \left. \frac{u_x v_0}{v(\lambda v - v_0)} \right|_0^1 \\ &= \frac{u_x}{v - v_0}. \end{aligned}$$

Lastly, remove the shift by replacing v with $v + v_0$ (or simply set $v_0 = 0$) to get

$$F = \mathcal{H}_{\mathbf{u}(x)} f = \frac{u_x}{v}.$$

Similar problems occur with rational functions involving multiple independent variables. Again, a shift of the coordinate(s) in the denominator will solve the problem. However, the complexity of the integration over the auxiliary variable λ increases rapidly when there are multiple independent variables. Thus, applying the shifts can cause `HomotopyIntegrator.m` to run for a long time.

CHAPTER 7

CONSTRUCTION OF CONSERVATION LAWS FOR NONLINEAR PDES

The techniques for constructing conservation laws will be shown by using the PDEs given in Chapter 3. Each PDE will illustrate various aspects of the computations used to determine conservation laws. The program `ConservationLawsMD.m` consists of algorithms based on these computations.

The procedure begins by computing the density, ρ , in equation (3.5). To do this, a candidate density is built by constructing lists of terms invariant under the scaling symmetry of the PDE, then forming an expression by summing the terms after multiplying each term by an undetermined coefficient. Values for the undetermined coefficients are found by applying the zeroth-Euler operator (4.2) to the t -derivative of the candidate density. Once the density is obtained, the flux, \mathbf{J} is found by inverting the divergence on $D_t(\rho)$ using the homotopy operator. In summary, the steps are:

- i. Determine a scaling symmetry for the PDE.
- ii. Construct a candidate density with undetermined coefficients using the scaling symmetry.
- iii. Calculate the unknown coefficients for the density.
- iv. Calculate the flux, once the density is known.

The ZK equation (3.7) is a (2+1)-dimensional example that will be used to illustrate the main steps listed above. The NTGF equation (3.18) is a good (3+1)-dimensional example to demonstrate how a single candidate density can be split into several densities. Gardner's equation (3.35) does not have a scaling symmetry unless the parameter β is included in the computations for the scaling symmetry.

7.1 Conservation Laws for the Zakharov-Kuznetsov Equation

The (2+1)-dimensional ZK equation,

$$u_t + \alpha uu_x + \beta(u_{2x} + u_{2y})_x = 0. \quad (7.1)$$

is taken as the first example because it is one of the few multi-dimensional examples that is an evolution equation, and it has polynomial conservation laws. Therefore, the method for computing conservation laws is straightforward. Also, conservation laws for the ZK equation have been previously studied [40, 68, 83], thus, the results presented in this section can be easily verified.

7.1.1 Establishing a Scaling Symmetry

Every PDE has a unique set of Lie-point symmetries which can include translations, rotations, scalings and other symmetries. The Lie-point symmetries are generated by constructing a set of infinitesimal generators. In turn, this requires solving an over-determined system of linear partial differential equations which lead to a set of spanning vector fields that can be translated into symmetry groups [12, 56]. The application of symmetries allows one to generate additional solutions from known solutions. In this dissertation, one type of Lie-point symmetry, the scaling or dilation symmetry, will be used to formulate conservation laws.

A candidate density is formed by assuming that the given PDE is invariant under a scaling symmetry, one of the Lie-point symmetries.

Definition 7.1. *A PDE is scaling invariant when a scaling transformation applied to the PDE returns the PDE multiplied by a constant factor.*

Although not all PDEs will have a scaling symmetry, it is possible to “induce” a scaling symmetry by introducing one or more parameters into the PDE. A scaling symmetry can be calculated using linear algebra, which avoids having to solve a system of differential equations. This is done by constructing a set of *weights* for every dependent and independent variable in the PDE. Once the weights are computed, a candidate density can be built as a linear combination of terms that are scaling invariant for a prescribed *rank* [52].

The next example illustrates how the ZK equation (7.1) is invariant under a particular scaling symmetry. The scaling factor, λ , used in examples in this chapter should not be confused with the auxiliary variable used in the homotopy operator.

Example 7.1. *The ZK equation (7.1) is scaling invariant under the scaling symmetry*

$$(t, x, y, u) \rightarrow (\lambda^{-3}t, \lambda^{-1}x, \lambda^{-1}y, \lambda^2u), \quad (7.2)$$

where λ is an arbitrary scaling parameter, different from the λ in the homotopy operator. It is not necessary to scale the constant parameters, α and β , in (7.1). To verify the scaling invariance for (7.1), let $\tilde{t} = \lambda^{-3}t$, $\tilde{x} = \lambda^{-1}x$, $\tilde{y} = \lambda^{-1}y$, and $\tilde{u} = \lambda^2u$. Applying the scaling symmetry to equation (7.1) transforms the left-hand side of the equation as follows,

$$\begin{aligned} & u_t + \alpha uu_x + \beta u_{3x} + \beta u_{x2y} \\ &= D_t u + \alpha u D_x u + \beta D_x^3 u + \beta D_x D_y^2 u \\ &= D_t \frac{\tilde{u}}{\lambda^2} + \alpha \frac{\tilde{u}}{\lambda^2} D_x \frac{\tilde{u}}{\lambda^2} + \beta D_x^3 \frac{\tilde{u}}{\lambda^2} + \beta D_x D_y^2 \frac{\tilde{u}}{\lambda^2} \\ &= \frac{1}{\lambda^2} D_{\tilde{t}} \tilde{u} (D_{\tilde{t}} \tilde{t}) + \frac{\alpha}{\lambda^4} \tilde{u} D_{\tilde{x}} \tilde{u} (D_{\tilde{x}} \tilde{x}) + \frac{\beta}{\lambda^2} D_{\tilde{x}}^3 \tilde{u} (D_{\tilde{x}} \tilde{x})^3 + \frac{\beta}{\lambda^2} D_{\tilde{x}} D_{\tilde{y}}^2 \tilde{u} (D_{\tilde{x}} \tilde{x}) (D_{\tilde{y}} \tilde{y})^2 \\ &= \frac{1}{\lambda^2} D_{\tilde{t}} \tilde{u} \left(D_{\tilde{t}} \frac{t}{\lambda^3} \right) + \frac{\alpha}{\lambda^4} \tilde{u} D_{\tilde{x}} \tilde{u} \left(D_{\tilde{x}} \frac{x}{\lambda} \right) + \frac{\beta}{\lambda^2} D_{\tilde{x}}^3 \tilde{u} \left(D_{\tilde{x}} \frac{x}{\lambda} \right)^3 + \frac{\beta}{\lambda^2} D_{\tilde{x}} D_{\tilde{y}}^2 \tilde{u} \left(D_{\tilde{x}} \frac{x}{\lambda} \right) \left(D_{\tilde{y}} \frac{y}{\lambda} \right)^2 \\ &= \frac{1}{\lambda^5} (D_{\tilde{t}} \tilde{u} + \alpha \tilde{u} D_{\tilde{x}} \tilde{u} + \beta D_{\tilde{x}}^3 \tilde{u} + \beta D_{\tilde{x}} D_{\tilde{y}}^2 \tilde{u}) \\ &= \frac{1}{\lambda^5} (\tilde{u}_{\tilde{t}} + \alpha \tilde{u} \tilde{u}_{\tilde{x}} + \beta \tilde{u}_{3\tilde{x}} + \beta \tilde{u}_{\tilde{x}2\tilde{y}}). \end{aligned} \quad (7.3)$$

The ZK equation is scaling invariant under the scaling symmetry (7.2) since the system in the scaled variables is identical to (7.1) to within the constant factor, $\frac{1}{\lambda^5}$.

Clearly, the transformed PDE (7.3) has the same solutions as the original. Furthermore, the scaling symmetry can be used to find densities, fluxes, generalized symmetries, recursion operators, and so on. Focusing on conservation laws, the defining equations (3.5) must hold on solutions of the PDE. Therefore, ρ and \mathbf{J} must obey the scaling symmetry of the PDE.

To construct a possible density, the scaling symmetry is used to find a *weight* for each variable, from which a *rank* can be found for the PDE.

Definition 7.2. *Let λ be an arbitrary scaling parameter. The weight of a variable (dependent or independent) is defined as the exponent p in the factor λ^p that multiplies the variable. For variable x , if the factor is λ^{-p} , the weight is denoted $W(x) = -p$. Total derivatives also carry a weight. If $W(x) = -p$, then $W(D_x) = p$.*

Definition 7.3. *The rank of a monomial is the sum of the weights of the variables in the monomial. A differential function is uniform in rank if all monomials in the differential function have the same rank. A differential function is multi-uniform in rank if it is uniform in rank for more than one scaling symmetry.*

Example 7.2. *Using the scaling symmetry (7.2) for the ZK equation the weights for the variables are*

$$W(u) = 2, \quad W(D_t) = 3, \quad W(D_x) = 1, \quad W(D_y) = 1. \quad (7.4)$$

The rank of the monomial $\alpha u u_x$ is $W(u) + W(u) + W(D_x) = 5$.

A PDE that has a scaling symmetry is uniform in rank. To calculate the weights, first assume that the PDE is uniform in rank, or that each equation is a system of PDEs is uniform in rank. This allows the formation of a linear system of weight balance equations. The equations in a system do not have to carry the same rank. Requiring uniformity of rank for (7.1) gives the system of weight balance equations for the ZK equation,

$$\begin{aligned} W(u) + W(D_t) &= 2W(u) + W(D_x) = W(u) + 3W(D_x) \\ &= W(u) + W(D_x) + 2W(D_y). \end{aligned} \quad (7.5)$$

The parameters α and β are assumed to have weight zero, hence they are not included in the system. If the system has no solution, that is, if the system is not scaling invariant, then weights can be added to the parameters to try to induce a scaling symmetry. In general, the solution to the system will leave at least one variable with an arbitrary weight, since λ is arbitrary. Furthermore, the order in which systems like (7.5) is solved is important. It is convenient to set $W(D_x) = 1$ later, so the system will be solved in terms of $W(D_x)$. In case there is more than one arbitrary weight in the system, weights for the dependent variables and weighted parameters are found first, followed by $W(D_t)$, followed by the weights for the independent space variables. The solution for the weight system (7.5) for the ZK equation is

$$W(u) = 2W(D_x), \quad W(D_t) = 3W(D_x), \quad W(D_y) = W(D_x). \quad (7.6)$$

Choosing $W(D_x) = 1$ gives (7.4). Using Definition 7.2, (7.4) are the weights for the scaling symmetry (7.2).

The choice $W(D_x) = 1$ is convenient, but it is possible that $W(D_x)$ may have to be assigned a different value. When formulating the candidate density the weights of the dependent variables, weighted parameters, and total derivatives of independent variables must be positive. Thus, the choice for $W(D_x)$ may depend on the values of the other weights. The weights do not have to be integer values.

7.1.2 Construction of a Candidate Density

Once the scaling symmetry is established, the terms of a candidate density can be constructed by fixing a value for the rank and combining dependent variables, partial derivatives on dependent variables, and independent variables into monomials so that each monomial has that preselected rank. Linearly combining these monomials with undetermined coefficients, the density in its final form will have a rank equal to the rank initially chosen. A candidate density for the ZK equation (7.1) with a rank of 6 will be constructed as the steps are explained.

First, construct a list \mathcal{P} containing all powers of dependent variables and products of dependent variables that have rank 6 or less, where 6 is the rank chosen for the candidate density.

Example 7.3. *For the ZK equation (7.1), using the weights (7.4), all possible powers and products of dependent variables with rank 6 or less are*

$$\mathcal{P} = \{u^3, u^2, u\}. \quad (7.7)$$

Second, construct a new list, \mathcal{Q} , by first taking all terms in \mathcal{P} with a rank of 6 and placing them into \mathcal{Q} . For all terms left in \mathcal{P} adjust their ranks up to rank 6 by applying total derivatives with respect to the independent space variables, then place the results in \mathcal{Q} . There may be several choices for adjusting the rank of the term and all of these choices should be added to \mathcal{Q} . When moving terms into \mathcal{Q} , all numerical coefficients can be ignored as they will be recalculated later.

Example 7.4. *In \mathcal{P} (7.3), the term u^3 has rank 6, so it is moved directly into \mathcal{Q} . The term u^2 has rank 4 and can be adjusted to rank 6 by applying D_x twice, by applying D_y twice, and by applying each of D_x and D_y once. Each time a total derivative is applied*

to u^2 , the rank of the term increases by 1 since the weight for each total derivative is 1, by (7.5). The term u has rank 2 and can be adjusted to rank six in five ways. List \mathcal{Q} is

$$\mathcal{Q} = \{u^3, u_x^2, uu_{2x}, u_y^2, uu_{2y}, u_x u_y, uu_{xy}, u_{4x}, u_{3xy}, u_{2x2y}, u_{x3y}, u_{4y}\}. \quad (7.8)$$

All twelve terms in \mathcal{Q} have rank 6.

Third, from \mathcal{Q} , remove all terms that are divergences or divergence-equivalent (see Definitions 5.1 and 5.2). If these terms belong in the conservation law, they will occur in the flux. If they are left in the candidate density, the final form for the density would be a linear combination of equivalent and/or trivial densities. Once all divergences and divergence-equivalent terms have been removed from \mathcal{Q} , put the remaining terms in list \mathcal{R} . Use Algorithm 5.1 to find divergences and divergence equivalent terms.

Example 7.5. Apply the zeroth-Euler operator (4.4) to \mathcal{Q} in (7.8) term by term to get

$$\mathcal{L}_{u(x,y)}\mathcal{Q} = \{3u^2, -2u_{2x}, 2u_{2x}, -2u_{2y}, 2u_{2y}, -2u_{xy}, 2u_{xy}, 0, 0, 0, 0, 0\}. \quad (7.9)$$

Divergences are terms corresponding to 0 in $\mathcal{L}_{u(x,y)}\mathcal{Q}$, so u_{4x} , u_{3xy} , u_{2x2y} , u_{x3y} , and u_{4y} can be removed from \mathcal{Q} . Next, attach undetermined coefficients to each term in (7.9), then set the sum of these terms equal to zero,

$$3p_1u^2 - 2p_2u_{2x} + 2p_3u_{2x} - 2p_4u_{2y} + 2p_5u_{2y} - 2p_6u_{xy} + 2p_7u_{xy} = 0.$$

After gathering like terms and forming a system of coefficient equations, the following relationships are found among the undetermined coefficients,

$$p_1 = 0, \quad p_2 = p_3, \quad p_4 = p_5, \quad p_6 = p_7.$$

Thus, the terms with coefficients p_3 , p_5 , and p_7 are divergence-equivalent to the terms with coefficients p_2 , p_4 , and p_6 , respectively. For each divergence-equivalent pair, the terms with the highest order are removed from \mathcal{Q} in (7.8).

This leaves

$$\mathcal{R} = \{u^3, u_x^2, u_y^2, u_x u_y\}, \quad (7.10)$$

which contains the terms left in \mathcal{Q} after all divergences and divergence-equivalent terms have been removed.

The candidate density is constructed by putting an undetermined coefficient, c_i , on each term in \mathcal{R} and adding all of the terms, that is, linearly combining the terms in \mathcal{R} , to create the expression

$$\rho_1 = c_1 u^3 + c_2 u_x^2 + c_3 u_y^2 + c_4 u_x u_y. \quad (7.11)$$

ρ_1 is a candidate density of rank 6 for the ZK equation. The next step is to determine values for the c_i , which is done in the next section.

In the next example, following the steps used to construct (7.11), a second candidate density of rank 1 containing terms which are products of independent and dependent variables will be constructed for the ZK equation (7.1).

Example 7.6. *The candidate density for rank 1 will have independent variables of degree 1 multiplied explicitly to the dependent variables. Since $W(D_x) = 1, W(D_y) = 1$, and $W(D_t) = 3$ by (7.4), $W(x) = -1, W(y) = -1$, and $W(t) = -3$, respectively, as described in Definition 7.2. First, find all powers and products of dependent variables with rank 1 or less. In this case, there are none, so $\mathcal{P} = \{\}$. To find all terms with explicit independent variable x , find all polynomial terms of rank 2 or less, then multiply each term by x . This creates the list $\mathcal{P}_{(x)}$ with terms of rank 1 or less,*

$$\mathcal{P}_{(x)} = \{xu\}.$$

Since $W(y) = -1$, find all polynomial terms of rank 2 or less, then multiply each term by y to create the rank 1 list,

$$\mathcal{P}_{(y)} = \{yu\}.$$

The last set of terms is found by multiplying t to all polynomial terms of rank 4 or less, giving

$$\mathcal{P}_{(t)} = \{tu^2, tu\}$$

In the construction of a the candidate density, t is a parameter. By keeping t a parameter, the possibility of constructing trivial conservation laws is eliminated. Bring all terms in $\mathcal{P}, \mathcal{P}_{(x)}, \mathcal{P}_{(y)}$, and $\mathcal{P}_{(t)}$ up to rank 1 by applying D_x and D_y , then combine the lists to get

$$\mathcal{Q} = \{xu, yu, tu^2, tu_{2x}, tu_{xy}, tu_{2y}\}.$$

Remove any divergences and divergent-equivalent terms from \mathcal{Q} to get

$$\mathcal{R} = \{xu, yu, tu^2\}.$$

A second candidate density for the ZK equation is

$$\rho_2 = c_1xu + c_2yu + c_3tu^2. \quad (7.12)$$

Note that it is necessary to set an upper bound for the degree of x , y , and t , otherwise ρ_2 would have an infinite number of terms, all of rank 1. In this case, a degree of 1 is sufficient to find all rank 1 densities.

7.1.3 Determination of the Actual Density

All, part, or none of the candidate density may be an actual density for a PDE. Indeed, the candidate density may be trivial or a linear combination of two or more independent densities. Analysis of the candidate density is done using the continuity equation (3.5). By (3.5), $D_t\rho = -\text{Div } \mathbf{J}$, so $D_t\rho$ must be a divergence with respect to the space variables. If the PDE is an evolution equation written in the form (3.1), all t -derivatives can be replaced with the right-hand side of the evolution equation. The next example shows $D_t\rho_1$ for candidate density (7.11) of rank 6 after all t -derivatives have been replaced.

Example 7.7. *Using (2.4), the total t -derivative of the rank 6 candidate density (7.11) is*

$$D_t\rho_1 = 3c_1u^2u_t + 2c_2u_xu_{tx} + 2c_3u_yu_{ty} + c_4(u_{tx}u_y + u_xu_{ty}).$$

Let E_1 be $-D_t\rho_1$ where all u_t in $D_t\rho_1$ have been replaced by the right-hand side of the ZK equation (5.12). Thus

$$\begin{aligned} E_1 = & 3c_1u^2(\alpha uu_x + \beta(u_{3x} + u_{x2y})) + 2c_2u_x(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_x \\ & + 2c_3u_y(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_y + c_4(u_y(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_x \\ & + u_x(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_y). \end{aligned} \quad (7.13)$$

Since $D_t\rho_1$ is required to be a divergence, the zeroth-Euler operator (4.4) applied to E_1 must be 0 by Theorem 4.1. This forms an equation where the coefficients on like terms must add up to zero. Thus, a linear system of equations is obtained, making it possible to compute values for the undetermined coefficients using linear algebra.

Example 7.8. *Applying the zeroth-Euler operator to (7.13) gives*

$$\begin{aligned} 0 &\equiv \mathcal{L}_{\mathbf{u}(x,y)} E_1 \\ &= -2((3c_1\beta + c_3\alpha)u_x u_{2y} + 2(3c_1\beta + c_3\alpha)u_y u_{xy} + 2c_4\alpha u_x u_{xy} + c_4\alpha u_y u_{2x} \\ &\quad + 3(3c_1\beta + c_2\alpha)u_x u_{2x}). \end{aligned}$$

A linear system of equations for the undetermined coefficients is formed by setting each factor equal to zero. After eliminating duplicate equations, the system is

$$3c_1\beta + c_3\alpha = 0, \quad c_4\alpha = 0, \quad 3c_1\beta + c_2\alpha = 0. \quad (7.14)$$

Note that this system is parameterized by the parameters α and β , appearing in the PDE. Before solving the system (7.14), it is necessary to compute compatibility conditions on the parameters α and β , if any exist, assuming that the parameters are nonzero. A detailed algorithm for computing the compatibility conditions is given in [30]. If a compatibility condition exists, but is ignored, then the solutions for all c_i in (7.14) will be zero which means there is no density. Before computing compatibility conditions, from the system of coefficient equations remove all equations where the solution for the c_i must be zero. Next, take all remaining c_i in the system of coefficient equations, and set them equal to 1, one at a time, then algebraically eliminate all other c_i .

Example 7.9. *To check for compatibility conditions on the parameters in (7.14), first remove $c_4\alpha = 0$ from the system since clearly $c_4 = 0$ which means that the term with coefficient c_4 in the candidate density (7.11) is not part of the actual density. For the remaining equations*

$$3c_1\beta + c_3\alpha = 0, \quad 3c_1\beta + c_2\alpha = 0, \quad (7.15)$$

set $c_1 = 1$ and algebraically eliminate c_2 and c_3 from (7.15). With $c_2 = c_3 = -\frac{3\beta}{\alpha}$, all equations in the system are satisfied. Next set $c_2 = 1$ and algebraically eliminate c_1 and c_3 from (7.15). With $c_1 = -\frac{\alpha}{3\beta}$ and $c_3 = 1$, the system is solved. Finally, set $c_3 = 1$ and algebraically eliminate c_1 and c_2 from (7.15). With $c_1 = -\frac{\alpha}{3\beta}$ and $c_2 = 1$, again, the system is solved.

Each time a c_i is set equal to one, the term with that coefficient is forced to remain in the density while the parameters are evaluated. If the candidate density is a linear

combination of densities, more than one compatibility condition could occur, so it is necessary to check each term in the density.

There are three possible outcomes when a particular $c_k = 1$ and all other c_i are eliminated.

Outcome 1: All remaining c_i are eliminated from the system leaving only identities. There are no compatibility conditions on the parameters.

Outcome 2: All remaining c_i are eliminated from the system leaving a new consistent system of equations containing parameters only. The solution to this new system provides the compatibility conditions on the parameters.

Outcome 3: All remaining c_i are eliminated from the system leaving a new system of equations containing parameters only. This new system either requires one of the parameters to be zero, or is inconsistent. In either case, the particular c_k set equal to 1 must actually be zero.

The compatibility check for candidate density (7.11) in Example 7.9 has Outcome 1; there are no compatibility conditions.

Once the compatibility conditions are known, it is possible to find a solution to the system of coefficient equations.

Example 7.10. *The solution to (7.14) is*

$$c_2 = -\frac{3\beta}{\alpha}c_1, \quad c_3 = -\frac{3\beta}{\alpha}c_1, \quad c_4 = 0, \quad (7.16)$$

where c_1 is arbitrary. This solution yields the density

$$\rho_1 = c_1 u^3 - c_1 \frac{3\beta}{\alpha} u_x^2 - c_1 \frac{3\beta}{\alpha} u_y^2.$$

Since c_1 is arbitrary, set $c_1 = 1$ to get

$$\rho_1 = u^3 - \frac{3\beta}{\alpha}(u_x^2 + u_y^2), \quad (7.17)$$

the density given in conservation law (3.11).

The next example shows a complete set of computations for the undetermined coefficients for candidate density (7.12) of rank 1.

Example 7.11. First, apply the total t -derivative to the candidate density (7.12) for rank 1 to get

$$D_t \rho_2 = c_1 x u_t + c_2 y u_t + c_3 u^2 + 2c_3 t u u_t.$$

Let E_2 be $-D_t \rho_2$ where all u_t in $D_t \rho_2$ have been replaced by the right-hand side of the ZK equation (5.12). Thus

$$\begin{aligned} E_2 &= c_1 x (\alpha u u_x + \beta (u_{3x} + u_{x2y})) + c_2 y (\alpha u u_x + \beta (u_{3x} + u_{x2y})) - c_3 u^2 \\ &\quad + 2c_3 t u (\alpha u u_x + \beta (u_{3x} + u_{x2y})). \end{aligned} \quad (7.18)$$

Require E_2 to be a divergence. Hence,

$$\begin{aligned} 0 &\equiv \mathcal{L}_{\mathbf{u}(x,y)} E_2 \\ &= -(\alpha c_1 + 2c_3)u. \end{aligned}$$

There is only one equation which does not generate any compatibility conditions on the parameters α and β . The solution for this equation is

$$c_1 = -\frac{2}{\alpha} c_3, \quad (7.19)$$

where c_3 is arbitrary. This leads to the density

$$\rho_2 = c_2 y u + c_3 t u^2 - \frac{2}{\alpha} c_3 x u. \quad (7.20)$$

where c_2 and c_3 are arbitrary.

The presence of two arbitrary coefficients indicates that ρ_2 consists of two independent densities. However, Example 5.6 shows that the conservation law with density yu is equivalent to the ZK equation (3.9), which is obtained at rank 2. Thus, by taking $c_3 = 1$, the only new density is

$$\rho_2 = t u^2 - \frac{2}{\alpha} x u. \quad (7.21)$$

This is the density for conservation law (3.12).

7.1.4 Calculation of the Flux

Once a density has been found, the corresponding flux can be calculated by using the homotopy operator (6.5) with integrands (6.32) and (6.33), since by (3.5),

$\mathbf{J} = \text{Div}^{-1}(-\mathbf{D}_t \rho)$. The undetermined coefficients for each density were calculated so that E_1 (7.13) and E_2 (7.18) are exact. Thus all that is needed is to replace the c_i in E_1 and E_2 using the solutions for the undetermined coefficients, then apply the homotopy operator.

Example 7.12. For ρ_1 , (7.17), $E_1 = -\mathbf{D}_t \rho_1$ is (7.13) with the solutions (7.16) and $c_1 = 1$ replacing all c_i so that

$$E_1 = 3u^2(\alpha uu_x + \beta(u_{3x} + u_{x2y})) - \frac{6\beta}{\alpha}u_x(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_x \\ - \frac{6\beta}{\alpha}u_y(\alpha uu_x + \beta(u_{3x} + u_{x2y}))_y.$$

Now, to find the flux, apply the homotopy operator to E_1 ,

$$\mathbf{J}_1 = \left(\mathcal{H}_{\mathbf{u}(x,y)}^{(x)} E_1, \mathcal{H}_{\mathbf{u}(x,y)}^{(y)} E_1 \right) \\ = \left(\frac{3}{4}\alpha u^4 + \beta u^2(3u_{2x} + 2u_{2y}) - \beta u(6u_x^2 + 2u_y^2) + \frac{3\beta^2}{4\alpha}u(u_{2x2y} + u_{4y}) \right. \\ \left. - \frac{\beta^2}{\alpha}u_x\left(\frac{7}{2}u_{x2y} + 6u_{3x}\right) - \frac{\beta^2}{\alpha}u_y(4u_{2xy} + \frac{3}{2}u_{3y}) + \frac{\beta^2}{\alpha}(3u_{2x}^2 + \frac{5}{2}u_{xy}^2 + \frac{3}{4}u_{2y}^2) \right. \\ \left. + \frac{5\beta^2}{4\alpha}u_{2x}u_{2y}, \beta u^2u_{xy} - 4\beta uu_xu_y - \frac{3\beta^2}{4\alpha}u(u_{x3y} + u_{3xy}) - \frac{\beta^2}{4\alpha}u_x(13u_{2xy} + 3u_{3y}) \right. \\ \left. - \frac{5\beta^2}{4\alpha}u_y(u_{3x} + 3u_{x2y}) + \frac{9\beta^2}{4\alpha}u_{xy}(u_{2x} + u_{2y}) \right). \quad (7.22)$$

After removing the divergence-free term $\mathbf{K} = (\mathbf{D}_y \theta, -\mathbf{D}_x \theta)$ with

$$\theta = 2\beta u^2 u_y + \frac{3\beta^2}{4\alpha}u(u_{2xy} + u_{3y}) + \frac{5\beta^2}{2\alpha}u_x u_{xy} + \frac{15\beta^2}{4\alpha}u_y u_{2y} + \frac{5\beta^2}{4\alpha}u_{2x} u_y,$$

$\mathbf{J}_1 - \mathbf{K}$ is the flux given in conservation law (3.11).

The flux corresponding to the rank 1 density (7.21) is found in the same manner.

Example 7.13. For ρ_2 (7.21), $E_2 = -\mathbf{D}_t \rho_2$ is (7.18) with $c_1 = -\frac{2}{\alpha}$, $c_2 = 0$, and $c_3 = 1$,

$$E_2 = -\frac{2}{\alpha}x(\alpha uu_x + \beta(u_{3x} + u_{x2y})) - u^2 + 2tu(\alpha uu_x + \beta(u_{3x} + u_{x2y})). \quad (7.23)$$

Again, to find the flux, apply the homotopy operator to E_2 . After removing the small divergence-free term, the flux is the same as in conservation law (3.12).

7.2 Conservation Laws for the Non-stationary Transonic Gas Flow Equation

The NTGF equation (3.18) serves as a good model for the computation of conservation laws in (3+1)-dimensions. The construction of two polynomial conservation laws, (3.21) and (3.22), will be shown in this section.

Unlike the ZK equation, the NTGF equation is not an evolution equation. The transformation of the NTGF equation into two evolution equations was shown in Example 3.1. The evolution equations used for calculations in this section are

$$\begin{aligned} u_t &= v, \\ v_t &= 2u_{xz} + u_x u_{2x} - u_{2y}. \end{aligned} \tag{7.24}$$

The reader should be aware that with the interchange of t and z in the transformation, the densities being calculated are actually the z -component of the flux in (3.21) and (3.22). All calculations that follow are based on (7.24).

7.2.1 Scaling Symmetry for the NTGF Equation

The system of weight equations for (7.24) is

$$\begin{aligned} W(v) &= W(u) + W(D_t), \\ W(v) + W(D_t) &= W(u) + W(D_x) + W(D_z) = 2W(u) + 3W(D_x) \\ &= W(u) + 2W(D_y). \end{aligned} \tag{7.25}$$

The weight system has to be solved in terms of $W(D_x)$ and $W(D_y)$, thus, there are two free weights

$$\begin{aligned} W(u) &= 2W(D_y) - 3W(D_x), & W(v) &= 3W(D_y) - 3W(D_x), \\ W(D_z) &= 2W(D_y) - W(D_x), & W(D_t) &= W(D_y). \end{aligned} \tag{7.26}$$

Taking $W(D_x) = 1$ and $W(D_y) = a$, the scaling symmetry for (7.24) is

$$(x, y, z, t, u, v) \longrightarrow (\lambda^{-1}x, \lambda^{-a}y, \lambda^{1-2a}z, \lambda^{-a}t, \lambda^{2a-3}u, \lambda^{3a-3}v), \tag{7.27}$$

where a is arbitrary. In this case, there is a one-parameter family of scaling symmetries. For example, setting $a = 2$ produces the symmetry

$$(x, y, z, t, u, v) \longrightarrow (\lambda^{-1}x, \lambda^{-2}y, \lambda^{-3}z, \lambda^{-2}t, \lambda u, \lambda^3v), \tag{7.28}$$

whereas setting $a = 3$ produces the symmetry

$$(x, y, z, t, u, v) \longrightarrow (\lambda^{-1}x, \lambda^{-3}y, \lambda^{-5}z, \lambda^{-3}t, \lambda^3u, \lambda^6v).$$

Weights do not have to be integers. For example, setting $a = \frac{5}{2}$ produces the symmetry

$$(x, y, z, t, u, v) \longrightarrow (\lambda^{-1}x, \lambda^{-5/2}y, \lambda^{-4}z, \lambda^{-5/2}t, \lambda^2u, \lambda^{9/2}v).$$

All three symmetries are valid choices under the conditions set in Section 7.1.1, hence by Definition 7.3, the NTGF equation is multi-uniform in rank. The symmetry (7.28) will be used to find a candidate density, with the weights,

$$W(u) = 1, \quad W(v) = 3, \quad W(D_t) = 2, \quad W(D_x) = 1, \quad W(D_y) = 2, \quad W(D_z) = 3. \quad (7.29)$$

7.2.2 Construction of Candidate Densities

Since the NTGF evolution equations are multi-uniform in rank, a candidate density for a particular rank can be split further by using additional symmetries in the family. The procedure will be shown by calculating all polynomial candidate densities at rank 6 for (7.24) using (7.29).

First, it is necessary to find lists \mathcal{P} , \mathcal{Q} , and \mathcal{R} following the steps described in Section 7.1.2.

Example 7.14. *Using the weights (7.29), the list of all combinations of dependent variables of rank 6 or less is*

$$\mathcal{P} = \{u^6, u^3v, v^2, u^5, u^2v, u^4, uv, u^3, v, u^2, u\}.$$

Bringing all terms up to rank 6 by applying D_x , D_y , and D_z gives the list

$$\begin{aligned} \mathcal{Q} = \{ & u^6, u^3v, v^2, u^4u_x, u^2v_x, uu_xv, u^2u_x^2, u^3u_{2x}, u^3u_y, u_xv_x, u_{2x}v, uv_{2x}, u_yv, uv_y, u_x^3, \\ & uu_xu_{2x}, u^2u_{3x}, uu_xu_y, u^2u_{xy}, u^2u_z, v_{3x}, v_{xy}, v_z, u_{2x}^2, u_xu_{3x}, uu_{4x}, u_xu_{xy}, u_{2x}u_y, \\ & uu_{2xy}, u_xu_z, uu_{xz}, u_y^2, uu_{2y}, u_{5x}, u_{3xy}, u_{x2y}, u_{2xz}\}. \end{aligned} \quad (7.30)$$

Use Algorithm 5.1 to remove all divergences and divergence-equivalent terms. First,

apply the zeroth-Euler operator (4.5) to every term in (7.30),

$$\begin{aligned}
\mathcal{L}_{\mathbf{u}(x,y,z)}\mathcal{Q} = & \\
& \{(6u^5, 0), (3u^2v, u^3), (0, 2v), (0, 0), (2uv_x, -2uu_x), (-uv_x, uu_x), \\
& (-2u(u_x^2 + uu_{2x}), 0), (6u(u_x^2 + uu_{2x}), 0), (0, 0), (-v_{2x}, -u_{2x}), (v_{2x}, u_{2x}), \\
& (v_{2x}, u_{2x}), (-v_y, u_y), (v_y, -u_y), (-6u_xu_{2x}, 0), (3u_xu_{2x}, 0), (-6u_xu_{2x}, 0), \\
& (-u_xu_y - 2uu_{xy}), (2(u_xu_y - 2uu_{xy}), 0), (0, 0), (0, 0), (0, 0), (0, 0), \\
& (2u_{4x}, 0), (-2u_{4x}, 0), (2u_{4x}, 0), (0, 0), (0, 0), (0, 0), (-2u_{xz}, 0), \\
& (2u_{xz}, 0), (-2u_{2y}, 0), (2u_{2y}, 0), (0, 0), (0, 0), (0, 0), (0, 0)\}, \tag{7.31}
\end{aligned}$$

where the variational derivatives correspond to each term in \mathcal{Q} . The terms u^4u_x , u^3u_y , u^2u_z , v_{3x} , v_{xy} , v_z , u_xu_{xy} , $u_{2x}u_y$, uu_{2xy} , u_{5x} , u_{3xy} , u_{x2y} , and u_{2xz} are divergences. A linear combination of the terms in (7.31) set equal to $(0, 0)$ yields

$$\begin{aligned}
& (6p_1u^5 + 3p_2u^2v + (2p_5 - p_6)uv_x - (2p_7 - 6p_8)u(u_x^2 + uu_{2x}) - (p_{10} - p_{11} - p_{12})v_{2x} \\
& - (p_{13} - p_{14})v_y - (6p_{15} - 3p_{16} + 6p_{17})u_xu_{2x} - (p_{18} - 2p_{19})(u_xu_y + 2uu_{xy}) \\
& + (2p_{24} - 2p_{25} + 2p_{26})u_{4x} - (2p_{30} - 2p_{31})u_{xz} - (2p_{32} - 2p_{33})u_{2y}, p_2u^3 + 2p_3v \\
& + (2p_5 - p_6)uu_x - (p_{10} - p_{11} - p_{12})u_{2x} + (p_{13} - p_{14})u_y) = (0, 0), \tag{7.32}
\end{aligned}$$

where the subscript on each undetermined coefficient matches the position of the variational derivative in \mathcal{Q} . In (7.32), clearly $p_1 = p_2 = p_3 = 0$, $p_5 = \frac{1}{2}p_6$, $p_7 = 3p_8$, $p_{10} = p_{11} + p_{12}$, $p_{13} = p_{14}$, $p_{15} = \frac{1}{2}p_{16} - p_{17}$, $p_{18} = 2p_{19}$, $p_{24} = p_{25} - p_{26}$, $p_{30} = p_{31}$ and $p_{32} = p_{33}$, where $p_6, p_9, p_{11}, p_{12}, p_{14}, p_{16}, p_{17}, p_{19}, p_{25}, p_{26}, p_{31}$, and p_{33} are arbitrary. Each solution in terms of some p_i represents a pair or triplet of divergence-equivalent terms. All terms whose position in \mathcal{Q} is the same as the subscript on the arbitrary p_i are divergence-equivalent terms that can be removed from \mathcal{Q} . The terms left in \mathcal{Q} form

$$\mathcal{R} = \{u^6, u^3v, v^2, u^2v_x, u^2u_x^2, u_xv_x, u_yv, u_x^3, uu_xu_y, u_{2x}^2, u_xu_z, u_y^2\}. \tag{7.33}$$

The terms in \mathcal{R} will be used to construct candidate densities.

Since the scaling symmetry (7.27) is multi-uniform in rank, \mathcal{R} can be subdivided. With $W(D_x) = 1$ and $W(D_y) = a$, the weight equations (7.26) become

$$W(u) = 2a - 3 \quad W(v) = 3a - 3, \quad W(D_z) = 2a - 1, \quad W(D_t) = a. \tag{7.34}$$

The weights in (7.34) can be used to calculate symbolic ranks for each term in \mathcal{R} . For example, the symbolic rank for u^3v is

$$\begin{aligned} W(u^3v) &= 3W(u) + W(v) \\ &= 3(2a - 3) + 3a - 3 \\ &= 9a - 12. \end{aligned}$$

Terms with the same symbolic rank will be grouped together to form a candidate density. Several candidate densities can be formed, one for each symbolic rank.

Example 7.15. *After the symbolic ranks have been calculated for all terms in \mathcal{R} , seven candidate densities can be formed. Table 7.1 lists the symbolic ranks paired with terms that carry that symbolic rank. Candidate densities are shown for each symbolic rank.*

Table 7.1: Candidate densities for the non-stationary transonic gas flow equation.

Symbolic Rank	Terms From \mathcal{R} With This Symbolic Rank	Candidate Density
$4a - 2$	u_{2x}^2	$\rho_1 = c_1 u_{2x}^2$
$5a - 4$	$u_x v_x$	$\rho_2 = c_1 u_x v_x$
$6a - 6$	$v^2, u_x^3, u_x u_z, u_y v, u_y^2$	$\rho_3 = c_1 v^2 + c_2 u_x^3 + c_3 u_x u_z + c_4 u_y v + c_5 u_y^2$
$7a - 8$	$u^2 v_x, u u_x u_y$	$\rho_4 = c_1 u^2 v_x + c_2 u u_x u_y$
$8a - 10$	$u^2 u_x^2$	$\rho_5 = c_1 u^2 u_x^2$
$9a - 12$	$u^3 v$	$\rho_6 = c_1 u^3 v$
$12a - 18$	u^6	$\rho_7 = c_1 u^6$

7.2.3 Determination of the Actual Densities

Each candidate density in Table 7.1 must be evaluated separately, however, each candidate density contains only a few terms and is easy to evaluate. Densities ρ_1 and ρ_3 will be evaluated in detail in the following examples. The procedure is the same as that used for the ZK equation.

Example 7.16. *Applying the D_t operator (2.4) to ρ_1 in Table 7.1 gives*

$$D_t \rho_1 = 2c_1 u_{2x} u_{t2x}.$$

Use (7.24) to replace u_t with v so that

$$D_t \rho_1 = 2c_1 u_{2x} v_{2x}.$$

Next, require $D_t \rho_1$ to be a divergence which implies that the zeroth-Euler operator acting on $D_t \rho_1$ must be $(0, 0)$. This gives the equation

$$\mathcal{L}_{\mathbf{u}(x,y)}(D_t \rho_1) = (2c_1 v_{4x}, 2c_1 u_{4x}) \equiv (0, 0).$$

The system of undetermined coefficients has only one equation, $2c_1 = 0$. Since $c_1 = 0$, candidate ρ_1 does not form a density.

Similar results occur for candidate densities ρ_2 , ρ_4 , ρ_5 , ρ_6 , and ρ_7 . None of these candidates form a density. Next, evaluate ρ_3 .

Example 7.17. The total t -derivative (2.4) of ρ_3 in Table 7.1 gives

$$D_t \rho_3 = 2c_1 v v_t + 3c_2 u_x^2 u_{tx} + c_3 u_{tx} u_z + c_3 u_x u_{tz} + c_4 u_y v_t + c_4 u_{ty} v + 2c_5 u_y u_{ty}.$$

Set $E_3 = -D_t \rho_3$ and use (7.24) to replace u_t with v and v_t with $2u_{xz} + u_x u_{2x} - u_{2y}$ to get

$$\begin{aligned} E_3 &= -2c_1 v(2u_{xz} + u_x u_{2x} - u_{2y}) - 3c_2 u_x^2 v_x - c_3(v_x u_z + u_x v_z) \\ &\quad - c_4 u_y(2u_{xz} + u_x u_{2x} - u_{2y}) - c_4 v_y v - 2c_5 u_y v_y. \end{aligned} \quad (7.35)$$

Next, the zeroth-Euler operator is applied to E_3 , and the result is set equal to zero, yielding

$$\begin{aligned} &\mathcal{L}_{\mathbf{u}(x,y)} E_3 \\ &= \begin{pmatrix} (3c_2 - c_1)(u_x v_{2x} + u_{2x} v_x) + (c_3 - 2c_1)v_{xz} + (c_1 + c_5)v_{2y} \\ (3c_2 - c_1)u_x u_{2x} + (c_3 - 2c_1)u_{xz} + (c_1 + c_5)u_{2y} \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned} \quad (7.36)$$

From (7.36), the system for the undetermined coefficients is

$$3c_2 - c_1 = 0, \quad c_3 - 2c_1 = 0, \quad c_1 + c_5 = 0. \quad (7.37)$$

The solution is

$$c_1 = -c_5, \quad c_2 = -\frac{1}{3}c_5, \quad c_3 = -2c_5, \quad (7.38)$$

where c_5 is arbitrary. c_4 does not occur in the system, so it remains in ρ_3 as an arbitrary constant. Hence, the candidate density is

$$\rho_3 = -c_5 v^2 - \frac{1}{3}c_5 u_x^3 - 2c_5 u_x u_z + c_4 v u_y + c_5 u_y^2. \quad (7.39)$$

The candidate density (7.39) is a linear combination of two densities. One density is found by setting $c_5 = 0$ and choosing $c_4 = 1$. The other density is found by setting $c_4 = 0$ and letting $c_5 = -\frac{1}{2}$. The two densities are

$$\rho_{31} = u_x u_z + \frac{1}{6} u_x^3 - \frac{1}{2} u_y^2 + \frac{1}{2} v^2, \quad (7.40)$$

$$\rho_{32} = u_y v. \quad (7.41)$$

These choices for c_4 and c_5 lead to a simple form for the final result.

7.2.4 Calculation of the Flux and Inversion of the Transformation

Since the NTGF equation is three-dimensional, the homotopy operator (6.11) with integrands (6.45), (6.46), and (6.47) is needed to find the flux.

Example 7.18. *To get $E_{31} = -D_t \rho_{31}$ for (7.40), take E_3 (7.35), and apply the solution (7.38) with $c_4 = 0$ and $c_5 = -\frac{1}{2}$. Thus,*

$$E_{31} = -v(2u_{xz} + u_x u_{2x} - u_{2y}) - \frac{1}{2} u_x^2 v_x - v_x u_z - u_x v_z + u_y v_y. \quad (7.42)$$

By construction, E_{31} (7.42) is exact. Using the homotopy operator (6.11) on E_{31} , the flux is

$$\mathbf{J}_{31} = \begin{pmatrix} \mathcal{H}_{\mathbf{u}(x,y,z)}^{(x)} E_{31} \\ \mathcal{H}_{\mathbf{u}(x,y,z)}^{(y)} E_{31} \\ \mathcal{H}_{\mathbf{u}(x,y,z)}^{(z)} E_{31} \end{pmatrix} = \begin{pmatrix} -v(u_z + \frac{1}{2} u_x^2) \\ u_y v \\ -u_x v \end{pmatrix}. \quad (7.43)$$

There are no divergence free terms in the result.

The conservation law for (7.24) corresponding to $(\rho_{31}, \mathbf{J}_{31})$ is

$$D_t \left(u_x u_z + \frac{1}{6} u_x^3 - \frac{1}{2} u_y^2 + \frac{1}{2} v^2 \right) - D_x \left(v(u_z + \frac{1}{2} u_x^2) \right) + D_y (u_y v) - D_z (u_x v) = 0. \quad (7.44)$$

The actual conservation law for (3.18) appears when the transformation (described in Example 3.1) used to put (3.18) in evolution form is inverted. Applied to (7.44), the inversion requires v to be replaced by u_t , followed by an interchange of t with z . Note that with the interchange of t and z , the z -component of the flux becomes the actual density, and the calculated density becomes the z -component of the flux. After the inversion, (7.44) becomes (3.22).

Example 7.19. The flux for (7.41) is calculated by taking E_3 (7.35) and letting $c_1 = c_2 = c_3 = c_5 = 0$, and $c_4 = -1$ to get

$$E_{32} = -D_t \rho_{32} = -u_y(2u_{xz} + u_x u_{2x} - u_{2y}) - v_y v. \quad (7.45)$$

Again, by design, $E_{32} = -D_t \rho_{32}$ is exact. After applying the homotopy operator (6.11), to E_{32} , the flux is found to be

$$\mathbf{J}_{32} = \begin{pmatrix} \mathcal{H}_{\mathbf{u}(x,y,z)}^{(x)}(E_{32}) \\ \mathcal{H}_{\mathbf{u}(x,y,z)}^{(y)}(E_{32}) \\ \mathcal{H}_{\mathbf{u}(x,y,z)}^{(z)}(E_{32}) \end{pmatrix} = \begin{pmatrix} \frac{1}{3}(3uu_{yz} - u_x^2 u_y + uu_x u_{xy}) \\ -\frac{1}{2}(2uu_{xz} + \frac{2}{3}uu_x u_{2x} - u_y^2 + v^2) \\ -u_x u_y \end{pmatrix}, \quad (7.46)$$

with the divergence-free term removed.

Once again the transformation used to create the evolution equations from (3.18) must be inverted on (7.41) and (7.46) to find the true conservation law. When in the inversion is complete, (7.41) and (7.46) yield the conservation law (3.21).

7.3 Conservation Laws for the Gardner Equation

Although the Gardner equation (3.35) is an evolution equation, the antiderivative terms in the equation pose a problem for `ConservationLawsMD.m`. The antiderivative terms can be replaced by letting $v_x = u_y$ [84], to get

$$\begin{aligned} v_x &= u_y, \\ u_t &= u_{3x} + 6\beta uu_x - \frac{3}{2}\alpha^2 u^2 u_x + 3v_y - 3\alpha u_x v. \end{aligned} \quad (7.47)$$

Only one of the equations in (7.47) is an evolution equation, yet the computations require that both be evolution equations. Again, a transformation involving the interchange t with y in both equations produces the desired result,

$$\begin{aligned} u_t &= v_x, \\ v_t &= -\frac{1}{3}u_{3x} - 2\beta uu_x + \frac{1}{2}\alpha^2 u^2 u_x + \alpha u_x v + \frac{1}{3}u_y. \end{aligned} \quad (7.48)$$

All calculations in this section will be based on (7.48).

7.3.1 Establishing a Scaling Symmetry

Calculating the weights for the variables in the Gardner equation requires a modification not previously shown. If the weight equations are constructed following the tech-

niques used in Section 7.1.2 which did not put weights on the parameters, the solution to the weight system is trivial,

$$W(u) = 0, \quad W(v) = 0, \quad W(\mathbf{D}_t) = 0, \quad W(\mathbf{D}_x) = 0, \quad W(\mathbf{D}_y) = 0.$$

Clearly, zero weights are useless for the construction of candidate densities.

Since (7.48) does not have a scaling symmetry, it is necessary to induce a scaling symmetry by assigning a weight to the parameter β . The system of weight equations when β carries weight is

$$\begin{aligned} W(u) + W(\mathbf{D}_t) &= W(v) + W(\mathbf{D}_x), \\ W(v) + W(\mathbf{D}_t) &= W(u) + 3W(\mathbf{D}_x) = 2W(u) + W(\mathbf{D}_x) + W(\beta) \\ &= 3W(u) + W(\mathbf{D}_x) = W(v) + W(\mathbf{D}_y) = W(u) + W(v) + W(\mathbf{D}_x). \end{aligned} \quad (7.49)$$

Solving the system in terms of $W(\mathbf{D}_x)$ gives

$$\begin{aligned} W(u) &= W(\mathbf{D}_x), & W(v) &= 2W(\mathbf{D}_x), & W(\beta) &= W(\mathbf{D}_x) \\ W(\mathbf{D}_t) &= 2W(\mathbf{D}_x), & W(\mathbf{D}_y) &= 3W(\mathbf{D}_x). \end{aligned} \quad (7.50)$$

Taking $W(\mathbf{D}_x) = 1$ gives the numerical weights to be used to construct a candidate density,

$$W(u) = 1, \quad W(v) = 2, \quad W(\beta) = 1, \quad W(\mathbf{D}_t) = 2, \quad W(\mathbf{D}_y) = 3. \quad (7.51)$$

The weights in (7.51) are equivalent to the scaling symmetry

$$(x, y, t, u, v, \beta) \longrightarrow (\lambda^{-1}x, \lambda^{-3}y, \lambda^{-2}t, \lambda u, \lambda^2v, \lambda\beta). \quad (7.52)$$

Although there is a second parameter α in the equation, α does not need to carry a weight. A scaling symmetry can be found when α carries a weight, however the addition of another weighted variable would increase the complexity of the candidate densities and cause unnecessary computations.

7.3.2 Determination of the Density

The candidate density for (7.48) will have a rank of 3. Since β carries a weight, it is included in the list \mathcal{P} of combinations of dependent variables up to rank 3. Apart from this modification, the procedure for constructing a set of terms to use to form the candidate density is the same as described in Section 7.1.2.

Example 7.20. Using the scaling symmetry (7.52) and the preselected rank of 3, all combinations of dependent variables, u and v , along with β that have rank 3 or less are

$$\mathcal{P} = \{u^3, uv, \beta^3, \beta^2u, \beta v, u^2, v, \beta u, \beta^2, u, \beta\}. \quad (7.53)$$

Bringing all terms up to rank 3 by applying D_x and D_y produces the set

$$\mathcal{Q} = \{u^3, uv, \beta^3, \beta^2u, \beta v, uu_x, v_x, \beta u_x, u_{2x}\}, \quad (7.54)$$

where β is a constant parameter. There are no divergence-equivalent terms in \mathcal{Q} , so once all divergences are removed, \mathcal{Q} is reduced to

$$\mathcal{R} = \{u^3, uv, \beta u^2, \beta v, \beta^2 u\}, \quad (7.55)$$

the list of terms which will form the candidate density.

Since the Gardner evolution system is not multi-uniform in rank, the candidate density cannot be subdivided. Hence, the candidate density for rank 3 is

$$\rho = c_1 u^3 + c_2 uv + c_3 \beta u^2 + c_4 \beta v + c_5 \beta^2 u. \quad (7.56)$$

The procedure for finding values for the undetermined coefficients, c_i , closely follows that outlined in Section 7.1.3.

Example 7.21. To find the actual density using (2.4) for ρ in (7.56), compute

$$D_t \rho = 3c_1 u^2 u_t + c_2 (u_t v + uv_t) + 2c_3 \beta u u_t + c_4 \beta v_t + c_5 \beta^2 u_t. \quad (7.57)$$

All t -derivatives in (7.57) are replaced with the right-hand sides of both evolution equations (7.48) so that,

$$\begin{aligned} E = -D_t \rho = & -(3c_1 u^2 v_x + c_2 v v_x + c_2 u (-\frac{1}{3} u_{3x} - 2\beta u u_x + \frac{1}{2} \alpha^2 u^2 u_x + \alpha u_x v + \frac{1}{3} u_y) \\ & + 2c_3 \beta u v_x + c_4 \beta (-\frac{1}{3} u_{3x} - 2\beta u u_x + \frac{1}{2} \alpha^2 u^2 u_x + \alpha u_x v + \frac{1}{3} u_y) + c_5 \beta^2 v_x). \end{aligned} \quad (7.58)$$

Since E must be a divergence, the zeroth-Euler operator is applied,

$$\mathcal{L}_{\mathbf{u}(x,y)} E = \begin{pmatrix} (c_4 \alpha \beta - 2c_3 \beta) v_x + (c_2 \alpha - 6c_1) u v_x \\ (2c_3 \beta - c_4 \alpha \beta) u_x + (6c_1 - c_2 \alpha) u u_x \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (7.59)$$

yielding the system of equations,

$$(2c_3 - c_4 \alpha) \beta = 0, \quad 6c_1 - c_2 \alpha = 0.$$

There are no compatibility conditions on the parameters α and β . The system of undetermined coefficients yields

$$c_1 = \frac{1}{6}\alpha c_2, \quad c_3 = \frac{1}{2}\alpha c_4, \quad (7.60)$$

where c_2 , c_4 and c_5 are arbitrary. Substituting (7.60) into (7.56) gives

$$\rho = \frac{1}{6}\alpha c_2 u^3 + c_2 uv + \frac{1}{2}\alpha c_4 \beta u^2 + c_4 \beta v + c_5 \beta^2 u. \quad (7.61)$$

The three arbitrary constants in the candidate density allow the candidate to be split into three densities. Separating the densities and choosing $c_2 = 6$, $c_4 = \frac{3}{\beta}$, and $c_5 = \frac{1}{\beta^2}$ gives

$$\rho_1 = u, \quad (7.62)$$

$$\rho_2 = \frac{3}{2}(\alpha u^2 + 2v), \quad (7.63)$$

$$\rho_3 = u(\alpha u^2 + 6v). \quad (7.64)$$

Note that β is a constant parameter and $\beta \neq 0$, so $\frac{2}{\beta}$ and $\frac{1}{\beta^2}$ are legitimate values for c_4 and c_5 respectively. However, since $W(\beta) = 1$, replacing c_4 with $\frac{2}{\beta}$ in ρ_2 (7.62) reduces the rank of ρ_2 to 2. Indeed, if the density were calculated for rank 2, ρ_2 would be obtained. At higher ranks, the lower rank densities will reappear multiplied by powers of β . In general, densities will not be unique to each rank when a weighted parameter occurs in the problem. The densities found at different ranks must be checked for independence as they may appear in equivalent forms.

7.3.3 Calculation of the Flux and Inversion of the Transformation

Of the three densities computed for the Gardner evolution equations, ρ_1 (7.62) leads to a trivial conservation law.

Example 7.22. Evaluate E in (7.58) with $c_1 = c_2 = c_3 = c_4 = 0$ and $c_5 = \frac{1}{\beta^2}$ to get

$$E_1 = -D_t \rho_1 = -v_x.$$

The flux, \mathbf{J}_1 is found by inverting the divergence on E_1 . Clearly,

$$\mathbf{J}_1 = \begin{pmatrix} -v \\ 0 \end{pmatrix}. \quad (7.65)$$

Like the NTGF equation, the density-flux pair (ρ_1, \mathbf{J}_1) yields

$$D_t(u) - D_x(v) + D_y(0) = 0, \quad (7.66)$$

a conservation law for (7.48). To get a conservation law for (3.35), it is necessary to invert the transformation used to create the evolution system. The inversion of (7.66) requires interchanging y and t , which interchanges the density with the y -component of the flux at the same time. This leaves the trivial conservation law,

$$D_t(0) - D_x(\partial_x^{-1}u_y) + D_y(u) = 0,$$

which clearly holds for any choice for u . These “trivial” conservation laws may occur when an interchange of the independent variables is used to form an evolution system.

The other two densities lead to non-trivial conservation laws. The homotopy operator will find the corresponding flux for both cases.

Example 7.23. *To find the flux for (7.63), take E in (7.58) and substitute $c_1 = c_2 = c_5 = 0$, $c_3 = \frac{1}{2}\alpha c_4$ and $c_4 = \frac{3}{\beta}$ to get*

$$E_2 = -D_t\rho_2 = -3\alpha uv_x + 3\left(\frac{1}{3}u_{3x} + 2\beta uu_x - \frac{1}{2}\alpha^2 u^2 u_x - \alpha u_x v - \frac{1}{3}u_y\right).$$

Apply the homotopy operator (6.5) to E_2 so that

$$\mathbf{J}_2 = \begin{pmatrix} \mathcal{H}_{\mathbf{u}(x,y)}^{(x)}(E_2) \\ \mathcal{H}_{\mathbf{u}(x,y)}^{(y)}(E_2) \end{pmatrix} = - \begin{pmatrix} \frac{1}{2}\alpha^2 u^3 - 3\beta u^2 + 3\alpha uv - u_{2x} \\ u \end{pmatrix}.$$

The conservation law corresponding to (ρ_2, \mathbf{J}_2) is

$$D_t\left(\frac{3}{2}(\alpha u^2 + 2v)\right) - D_x\left(\frac{1}{2}\alpha^2 u^3 - 3\beta u^2 + 3\alpha uv - u_{2x}\right) - D_y(u) = 0, \quad (7.67)$$

which is the same as (3.36) after t and y have been interchanged to invert the transformation used to get (7.48).

Example 7.24. *To find the flux for (7.64), substitute $c_3 = c_4 = c_5 = 0$, $c_1 = \frac{1}{6}\alpha c_2$ and $c_2 = 6$ into E in (7.58) to get*

$$E_3 = -D_t\rho_3 = -3\alpha u^2 v_x - 6v v_x + 6u\left(\frac{1}{3}u_{3x} + 2\beta uu_x - \frac{1}{2}\alpha^2 u^2 u_x - \alpha u_x v - \frac{1}{3}u_y\right).$$

Apply the homotopy operator (6.5) to E_3 so that

$$\mathbf{J}_3 = \begin{pmatrix} \mathcal{H}_{\mathbf{u}(x,y)}^{(x)}(E_3) \\ \mathcal{H}_{\mathbf{u}(x,y)}^{(y)}(E_3) \end{pmatrix} = - \begin{pmatrix} \frac{3}{4}\alpha^2 u^4 - 4\beta u^3 + 3\alpha u^2 v + 3v^2 + u_x^2 - 2uu_{2x} \\ u^2 \end{pmatrix}.$$

The conservation law corresponding to (ρ_3, \mathbf{J}_3) is

$$\mathbf{D}_t(u(\alpha u^2 + 6v)) - \mathbf{D}_x\left(\frac{3}{4}\alpha^2 u^4 - 4\beta u^3 + 3\alpha u^2 v + 3v^2 + u_x^2 - 2uu_{2x}\right) - \mathbf{D}_y(u^2) = 0. \quad (7.68)$$

After inverting the transformations used to get an evolution form for the Gardner equation, (7.68) conservation law (3.37).

CHAPTER 8

CONSERVATION LAWS FOR PDES IN MULTI-DIMENSIONS

Conservation laws are given for a variety of multi-dimensional PDEs in this chapter. The PDEs chosen for this chapter either have a physical application or have been the subject of case studies in a number of papers. All of the results in this chapter are based on conservation laws computed using `ConservationLawsMD.m`, and all conservation laws reported have been verified by the program.

The conservation laws for several PDEs given in this chapter are well established. For example, conservation laws for the potential Kadomtsev-Petviashvili equation have been computed by Infeld and Frycz [41] and Rosenhaus [59] among others. These conservation laws are being reproduced here to demonstrate the capabilities and versatility of `ConservationLawsMD.m` and to point out errors found in a number of publications. While typographical errors are easy to resolve, there are cases where the results reported in literature appear to be in error. Computational errors in literature made verification of the conservation laws for the Khokhlov-Zabolotskaya equation difficult.

In addition to verifying known conservation laws, new conservation laws are presented in this chapter. The recently developed Manakov-Santini system [50, 51] yields several conservation laws. To date, no conservation laws have been found in any literature for the Camassa-Holm equation or for the Gardner equation (3.35).

Many of the PDEs in this chapter are not evolution equations. An algorithm in `ConservationLawsMD.m` can transform many PDEs into an evolution form automatically before computing conservation laws. However, if a user wishes to test a conservation law using the program, the evolution equations are required, therefore the test case must be transformed by hand into evolution form. To make matters completely transparent, the evolution form will be provided for non-evolution PDEs. Instructions for how to use `ConservationLawsMD.m` to test a density will be given in Section 9.1.3.

8.1 Kadomtsev-Petviashvili Equations

Like the ZK equation (3.7), the well-known Kadomtsev-Petviashvili (KP) equation is a (2+1)-dimensional generalization of the KdV equation. The KP equation was selected to be a test case for `ConservationLawsMD.m`. Tests were extended to the potential KP equation since a number of publications listed conservation laws for the potential form rather than the KP equation itself.

Like the NTGF equation (3.18), the KP and potential KP equations have time-space-derivative term, u_{tx} , in place of the time-derivative term found in an evolution equation. Thus, it is necessary to transform both equations into evolution equations before beginning the computations to find conservation laws.

8.1.1 The Kadomtsev-Petviashvili Equation

The (2+1)-dimensional KP equation describes the stability of shallow water waves with wavelengths much greater than their amplitude moving in the x -direction and subject to weak variations in the y -direction [43]. The KP equation is given as

$$(u_t + \alpha uu_x + u_{3x})_x + \sigma^2 u_{2y} = 0, \quad (8.1)$$

where the parameter α occurs after a re-scaling of the physical coefficients and $\sigma^2 = \pm 1$. With $\sigma^2 = 1$, the KP equation describes motion in shallow water waves. For $\sigma^2 = -1$, the equation describes motion in fluids with a high surface tension [1]. The KP equation also has applications in plasma physics and gas dynamics [59].

The KP equation can be written as an evolution system using the transformations applied in Example 3.1. After interchanging t and y , then setting v equal to u_t , the evolution system equivalent to (8.1) is

$$\begin{aligned} u_t &= v, \\ v_t &= -\frac{1}{\sigma^2}(u_{xy} + \alpha u_x^2 + \alpha uu_{2x} + u_{4x}). \end{aligned} \quad (8.2)$$

All conservation laws given below are for the original form (8.1) of the KP equation. The KP equation itself is a conservation law, representing the conservation of momentum,

$$D_t(u_x) + D_x(\alpha uu_x + u_{3x}) + D_y(\sigma^2 u_y) = 0. \quad (8.3)$$

All other conservation laws for the KP equation are either nonlocal, or have terms explicitly multiplied by the independent variables. Although (8.3) is physically relevant, according to the definition in Section 5.2, this law can be reduced to a divergence on the space variables only, making it a trivial conservation law.

The program `ConservationLawsMD.m` computes several density-flux pairs with an explicit occurrence of x , y and t . A partial list of conservation laws ordered by rank and computed by the program is listed below. The ranks are based on the scaling symmetry for (8.1) (after the transformation into evolution form is inverted) where

$$W(u) = 2, \quad W(D_t) = 3, \quad W(D_x) = 1, \quad W(D_y) = 2. \quad (8.4)$$

Conservation laws for rank 2 are (8.3) and

$$D_t(u) + D_x\left(\frac{1}{2}\alpha u^2 + u_{2x} - x(u_t + \alpha uu_x + u_{3x})\right) - D_y(\sigma^2 x u_y) = 0. \quad (8.5)$$

The conservation law for rank 1 is

$$D_t(yu_x) + D_x(y(\alpha uu_x + u_{3x})) - D_y(\sigma^2(u - yu_y)) = 0. \quad (8.6)$$

The conservation law for rank 0 is

$$D_t(yu) + D_x\left(y\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) - xy(u_t + \alpha uu_x + u_{3x})\right) + D_y(\sigma^2 x(u - yu_y)) = 0. \quad (8.7)$$

The conservation law for rank -1 is

$$D_t(tu) + D_x\left(t\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\frac{1}{2}\sigma^2 y^2 - tx\right)(u_t + \alpha uu_x + u_{3x})\right) - D_y\left(yu - u_y\left(\frac{1}{2}y^2 - \sigma^2 tx\right)\right) = 0. \quad (8.8)$$

The conservation law for rank -3 is

$$D_t(tyu) + D_x\left(ty\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\frac{1}{6}\sigma^2 y^3 - txy\right)(u_t + \alpha uu_x + u_{3x})\right) - D_y\left(u\left(\frac{1}{2}y^2 - \sigma^2 tx\right) - u_y\left(\frac{1}{6}y^3 - \sigma^2 txy\right)\right) = 0. \quad (8.9)$$

The conservation law for rank -4 is

$$D_t(t^2u) + D_x\left(t^2\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\sigma^2 ty^2 - t^2x\right)(u_t + \alpha uu_x + u_{3x})\right) - D_y\left(2tyu - u_y\left(ty^2 - \sigma^2 t^2x\right)\right) = 0, \quad (8.10)$$

and the conservation law for rank -6 is

$$\begin{aligned} & \mathbf{D}_t\left(t^2yu\right) + \mathbf{D}_x\left(t^2y\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\frac{1}{3}\sigma^2ty^3 - t^2xy\right)(u_t + \alpha uu_x + u_{3x})\right) \\ & - \mathbf{D}_y\left(u\left(ty^2 - \sigma^2t^2x\right) - u_y\left(\frac{1}{3}ty^3 - \sigma^2t^2xy\right)\right) = 0. \end{aligned} \quad (8.11)$$

Like (8.3), the density for conservation law (8.6) can be shifted into the flux, making it a trivial conservation law. The densities at ranks -7 and -10 will be t^3u and t^4u , respectively, and the pattern of $t^n u$ continues for $n = 5, 6, 7, \dots$. The densities at ranks -8 and -11 will be t^3yu and t^4yu , respectively, and the pattern of $t^n yu$ also continues for $n = 5, 6, 7, \dots$. It is clear from the pattern established by running `ConservationLawsMD.m` over a range of ranks that the conservation laws (8.10) and (8.11) can be generalized by replacing t^2 with t^n and t with $\frac{1}{2}nt^{n-1}$. If the density $\rho = t^n u$ is provided to `ConservationLawsMD.m`, the program verifies that $t^n u$ is indeed a density for (8.1) and returns the conservation law

$$\begin{aligned} & \mathbf{D}_t\left(t^n u\right) + \mathbf{D}_x\left(t^n\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\frac{1}{2}\sigma^2nt^{n-1}y^2 - t^n x\right)(u_t + \alpha uu_x + u_{3x})\right) \\ & - \mathbf{D}_y\left(nt^{n-1}yu - u_y\left(\frac{1}{2}nt^{n-1}y^2 - \sigma^2t^n x\right)\right) = 0. \end{aligned} \quad (8.12)$$

Although the program can compute a scaling symmetry for a PDE with an unspecified exponent, n , it would be unable to generate a candidate density without a specific value for the exponent, since the scaling symmetry would be given in terms of n . The program can verify the proposed general forms, compute the correct coefficients for the form of the density and compute the flux. Wolf's `CONLAW` program [75, 76] suggested a further generalization where t^n in (8.12) is replaced by an arbitrary function $f = f(t)$. Doing so, (8.12) becomes

$$\begin{aligned} & \mathbf{D}_t\left(fu\right) + \mathbf{D}_x\left(f\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\frac{1}{2}\sigma^2f'y^2 - fx\right)(u_t + \alpha uu_x + u_{3x})\right) \\ & - \mathbf{D}_y\left(f'yu - u_y\left(\frac{1}{2}f'y^2 - \sigma^2fx\right)\right) = 0, \end{aligned} \quad (8.13)$$

which is also verified by `ConservationLawsMD.m` in the same manner as (8.12). A second conservation law, generalizing (8.9) and (8.11), again with arbitrary function $f = f(t)$, is

$$\begin{aligned} & \mathbf{D}_t\left(fyu\right) + \mathbf{D}_x\left(fy\left(\frac{1}{2}\alpha u^2 + u_{2x}\right) + \left(\frac{1}{6}\sigma^2f'y^3 - fxy\right)(u_t + \alpha uu_x - u_{3x})\right) \\ & - \mathbf{D}_y\left(u\left(\frac{1}{2}f'y^2 - \sigma^2fx\right) - u_y\left(\frac{1}{6}f'y^3 - \sigma^2fxy\right)\right) = 0. \end{aligned} \quad (8.14)$$

Note that setting $f = t^2$ in (8.13) gives (8.10) and setting $f = t^2$ in (8.14) gives (8.11). To get (8.13) and (8.14), the densities, $\rho_1 = fu$ and $\rho_2 = fyu$ are given to `ConservationLawsMD.m`. The program verifies both densities with the arbitrary function $f(t)$ by assigning undetermined coefficients and recomputing their values, following the procedure described in Section 7.1.3. It then computes the fluxes using the homotopy operator. Generalizations of a similar nature occur in a variety of cases.

The generalizations (8.13) and (8.14) produce an infinite variety of conservation laws. The KP equation is known to be integrable, a fact supported by the existence of large families of conservation laws.

8.1.2 The Potential Kadomtsev-Petviashvili Equation

Nonlocal conservation laws for the KP equation are obtained by calculating conservation laws for the potential KP equation. The potential KP equation is formulated by replacing u in equation (8.1) with u_x , then integrating with respect to x , to get

$$u_{tx} + \alpha u_x u_{2x} + u_{4x} + \sigma^2 u_{2y} = 0. \quad (8.15)$$

The potential KP equation can be written as an evolution system using the same transformations applied to the KP equation. The evolution system is

$$\begin{aligned} u_t &= v, \\ v_t &= -\frac{1}{\sigma^2}(u_{xy} + \alpha u_x u_{2x} + u_{4x}). \end{aligned} \quad (8.16)$$

Like the KP equation, the potential KP equation (8.15) is itself a conservation law,

$$\mathbf{D}_t(u_x) + \mathbf{D}_x\left(\frac{1}{2}\alpha u_x^2 + u_{3x}\right) + \mathbf{D}_y(\sigma^2 u_y) = 0, \quad (8.17)$$

also representing conservation of momentum. All other conservation laws are given by rank. The ranks are based on the weights for (8.15),

$$W(u) = 1, \quad W(\mathbf{D}_t) = 3, \quad W(\mathbf{D}_x) = 1, \quad W(\mathbf{D}_y) = 2. \quad (8.18)$$

There are three other polynomial conservation laws where terms are not multiplied by independent variables, all computed by `ConservationLawsMD.m`.

The conservation law for rank 4 is

$$\mathbf{D}_t(u_x^2) + \mathbf{D}_x\left(\frac{2}{3}\alpha u_x^3 - u_{2x}^2 + 2u_x u_{3x} - \sigma^2 u_y^2\right) + \mathbf{D}_y(2\sigma^2 u_x u_y) = 0. \quad (8.19)$$

The conservation law for rank 5 is

$$D_t(u_x u_y) + D_x(\alpha u_x^2 u_y + u_t u_y + 2u_{3x} u_y - 2u_{2x} u_{xy}) - D_y\left(\frac{1}{3}\alpha u_x^3 - u_{2x}^2 + u_t u_x - \sigma^2 u_y^2\right) = 0. \quad (8.20)$$

The conservation law for rank 6 is

$$D_t\left(2\alpha u u_x u_{2x} + 3u u_{4x} - 3\sigma^2 u_y^2\right) + D_x\left(3u_t^2 + 2\alpha u_t u_x^2 - 2\alpha u u_x u_{tx} - 3u_{tx} u_{2x} + 3u_t u_{3x} + 3u_x u_{t2x} - 3u u_{t3x}\right) + D_y\left(6\sigma^2 u_t u_y\right) = 0. \quad (8.21)$$

In literature [41, 59], a density equivalent to the density in (8.21), $\rho = u_x^3 - 3u_{2x}^2 + 3\sigma^2 u_y^2$ is given. Normally `ConservationLawsMD.m` would return densities in the form given in [41, 59], however, the exchange of y and t needed to put (8.15) in evolution form shifts the computed density to the flux. That is, the density given in (8.21) is computed by the homotopy operator as the y -component of the flux, where the program has less control over the terms chosen. When the transformation is inverted, the y -component of the flux becomes the density.

Three generalizations can be formed to represent all conservation laws of the potential KP equation. Like the KP equation (8.1), the generalized conservation laws can be deduced by looking for patterns in the densities computed by `ConservationLawsMD.m`. Table 8.1 shows two lists of densities, followed by their generalizations.

Table 8.1: A list of densities for the potential Kadomtsev-Petviashvili equation (8.15) with their generalizations.

Generalization for the density of (8.22)		Generalization for the density of (8.23)	
Rank	Densities	Rank	Densities
5	u_x^2	6	$u_x u_y$
2	$t u_x^2 + \frac{2}{\alpha} u$	3	$t u_x u_y - \frac{1}{2} \sigma^2 y u_x^2$
-1	$t^2 u_x^2 + \frac{4}{\alpha} t u$	0	$t^2 u_x u_y - \sigma^2 t y u_x^2 - \frac{2}{\alpha} \sigma^2 y u$
-4	$t^3 u_x^2 + \frac{6}{\alpha} t^2 u$	-3	$t^3 u_x u_y - \frac{3}{2} \sigma^2 t^2 y u_x^2 - \frac{6}{\alpha} \sigma^2 t y u$
-7	$t^4 u_x^2 + \frac{8}{\alpha} t^3 u$	-6	$t^4 u_x u_y - 2\sigma^2 t^3 y u_x^2 - \frac{12}{\alpha} \sigma^2 t^2 y u$
-10	$t^5 u_x^2 + \frac{10}{\alpha} t^4 u$	-9	$t^5 u_x u_y - \frac{5}{2} \sigma^2 t^4 y u_x^2 - \frac{20}{\alpha} \sigma^2 t^3 y u$
GD	$f u_x^2 + \frac{2}{\alpha} f' u$	GD	$f u_x u_y - \frac{1}{2} \sigma^2 f' y u_x^2 - \frac{1}{\alpha} \sigma^2 f'' y u$

Note: GD = general density and $f = f(t)$ is an arbitrary function.

Again, `ConservationLawsMD.m` verifies the generalized densities, then computes the fluxes and reports the conservation laws, given as follows. First, to shorten the fluxes, let

$$\begin{aligned}
A &= \frac{2}{\alpha}u_t + u_x^2 + \frac{2}{\alpha}u_{3x}, \\
B &= \frac{2}{3}\alpha u_x^3 - u_{2x}^2 + 2u_x u_{3x} - \sigma^2 u_y^2, \\
C &= \alpha u_x^2 u_y + u_t u_y + 2u_{3x} u_y - 2u_{2x} u_{xy}, \\
D &= 3u_t^2 + 2\alpha u_t u_x^2 - 2\alpha u u_x u_{tx} - 3u_{tx} u_{2x} + 3u_t u_{3x} + 3u_x u_{t2x} - 3u u_{t3x}, \\
E &= \frac{1}{3}\alpha u_x^3 - u_{2x}^2 + u_t u_x - \sigma^2 u_y^2.
\end{aligned}$$

The generalized conservation laws are

$$\begin{aligned}
&D_t\left(fu_x^2 + \frac{2}{\alpha}f'u\right) + D_x\left(fB - (f'x + \frac{1}{2}\sigma^2 f''y^2)A + \frac{2}{\alpha}f'u_{2x}\right) \\
&+ D_y\left(2\sigma^2(fu_x - \frac{1}{\alpha}f'x)u_y - \frac{1}{\alpha}f''y(2u - yu_y)\right) = 0, \tag{8.22}
\end{aligned}$$

$$\begin{aligned}
&D_t\left(fu_x u_y - \frac{1}{2}\sigma^2 f'yu_x^2 - \frac{1}{\alpha}\sigma^2 f''yu\right) + D_x\left(fC - \frac{1}{2}\sigma^2 f'yB + (\frac{1}{2}\sigma^2 f''xy - \frac{1}{12}f'''y^3)A\right. \\
&\left. - \frac{1}{\alpha}\sigma^2 f''yu_{2x}\right) - D_y\left(fE + f'yu_x u_y + \frac{1}{\alpha}f''x(u - yu_y) - \frac{1}{2\alpha}\sigma^2 f'''y^2(u - \frac{1}{3}yu_y)\right) = 0, \tag{8.23}
\end{aligned}$$

$$\begin{aligned}
&D_t\left(f(2\alpha u u_x u_{2x} + 3u u_{4x} - 3\sigma^2 u_y^2) + f'u_x(xu_x + 2yu_y) - f''(\frac{1}{2}\sigma^2 y^2 u_x^2 - \frac{2}{\alpha}xu)\right. \\
&\left. - \frac{1}{\alpha}\sigma^2 f'''y^2 u\right) + D_x\left(fD + f'(2yC - u_x u_{2x} - u u_{3x} + 2u u_y) + (f'x - \frac{1}{2}\sigma^2 f''y^2)B\right. \\
&\left. - \frac{1}{2}(f''x^2 - f''' \sigma^2 x y^2 + \frac{1}{12}f^{(4)}y^4)A + \frac{2}{\alpha}f''(xu_{2x} - u_x) - \frac{1}{\alpha}f''' \sigma^2 y^2 u_{2x}\right) \\
&+ D_y\left(6\sigma^2 f u_t u_y - f'(2yE + \sigma^2 u_y(u + x u_x)) - f''u_y(\frac{1}{\alpha}\sigma^2 x^2 + y^2 u_x)\right. \\
&\left. - \frac{1}{\alpha}f'''xy(2u - yu_y) + \frac{1}{3\alpha}\sigma^2 f^{(4)}y^3(4u - yu_y)\right) = 0, \tag{8.24}
\end{aligned}$$

where $f = f(t)$ is an arbitrary function.

An extensive study of the conservation laws for the potential KP equation was conducted by Infeld and Frycz [41]. They give a “generating formula” for conservation laws where $f(t) = t^n$ with n a positive integer. By letting $f(t) = t^n$, equations (8.22), (8.23), and (8.24) produce the same set of conservation laws given in [41]. Rosenhaus computes several non-vanishing densities for the potential KP equation [59]. His densities also restrict $f(t)$ to powers of t . Several densities in [59] contain terms that can be moved into the flux of the generalized conservation laws. This makes it difficult to verify the results of Rosenhaus.

8.2 The Fluid Dynamics Equations

Three systems of PDEs proposed by Dellar [23] are based on a model by Ripa [58] that describes geophysical fluid dynamics in a structure of inhomogeneous layers. The first system models shallow water hydrodynamics, the second system considers the effects of a magnetic field in shallow water hydrodynamics, and the third system accounts for the effects of temperature gradients and magnetic fields. All three are systems of evolution equations and several conservation laws exist for each case.

8.2.1 The Shallow Water Equations

The first system is composed of shallow water equations with forcing due to a horizontally varying potential temperature field. This case describes the motion of water in an inhomogeneous layer over a flat bottom [23]. The evolution system is given as

$$\begin{aligned} \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + 2\boldsymbol{\Omega} \times \mathbf{u} &= -\nabla(h\theta) + \frac{1}{2}h\nabla\theta, \\ \theta_t + \mathbf{u} \cdot (\nabla\theta) &= 0, \quad h_t + \nabla \cdot (h\mathbf{u}) = 0, \end{aligned} \tag{8.25}$$

where $\mathbf{u} = (u(x, y, t), v(x, y, t), 0)$ is the horizontal fluid velocity, $\boldsymbol{\Omega} = (0, 0, \Omega)$ is the angular velocity of the Coriolis force, $\theta(x, y, t)$ is the horizontally varying potential temperature field, and $h(x, y, t)$ is the layer depth. The scalar equations for (8.25) are

$$\begin{aligned} u_t + uu_x + vu_y - 2\Omega v + \frac{1}{2}h\theta_x + \theta h_x &= 0, \\ v_t + uv_x + vv_y + 2\Omega u + \frac{1}{2}h\theta_y + \theta h_y &= 0, \\ \theta_t + u\theta_x + v\theta_y &= 0, \quad h_t + uh_x + u_x h + vh_y + v_y h = 0. \end{aligned} \tag{8.26}$$

Like the KP equation, the shallow water equations (8.25) were used as a test case for `ConservationLawsMD.m`. The polynomial conservation laws given below are reported by Dellar [23]. `ConservationLawsMD.m` is able to verify all of the conservation laws given by Dellar, and also find conservation laws with terms multiplied by x , y , and t explicitly.

The ranks for the conservation laws are calculated using the weights

$$\begin{aligned} W(u) = 1, \quad W(v) = 1, \quad W(\theta) = 1, \quad W(h) = 1, \quad W(\Omega) = 2, \\ W(\mathbf{D}_t) = 2, \quad W(\mathbf{D}_x) = 1, \quad W(\mathbf{D}_y) = 1, \end{aligned} \tag{8.27}$$

where Ω is a constant parameter that carries weight. Conservation laws at Rank 1 are

$$\mathbf{D}_t(h) + \mathbf{D}_x(uh) + \mathbf{D}_y(vh) = 0, \quad (8.28)$$

$$\begin{aligned} \mathbf{D}_t\left(h(\Omega(x^2 + y^2) - yu + xv)\right) + \mathbf{D}_x\left(h(u(\Omega(x^2 + y^2) - yu + xv) - \frac{1}{2}y\theta h)\right) \\ + \mathbf{D}_y\left(h(v(\Omega(x^2 + y^2) - yu + xv) + \frac{1}{2}x\theta h)\right) = 0. \end{aligned} \quad (8.29)$$

Conservation laws at Rank 2 are

$$\mathbf{D}_t(\theta h) + \mathbf{D}_x(u\theta h) + \mathbf{D}_y(v\theta h) = 0, \quad (8.30)$$

$$\mathbf{D}_t\left(h(v + 2\Omega x)\right) + \mathbf{D}_x\left(uh(v + 2\Omega x)\right) + \mathbf{D}_y\left(h(v^2 + \frac{1}{2}\theta h + 2\Omega xv)\right) = 0, \quad (8.31)$$

$$\mathbf{D}_t\left(h(u - 2\Omega y)\right) + \mathbf{D}_x\left(h(u^2 + \frac{1}{2}\theta h - 2\Omega yu)\right) + \mathbf{D}_y\left(vh(u - 2\Omega y)\right) = 0. \quad (8.32)$$

Conservation laws at Rank 3 are

$$\mathbf{D}_t(\theta^2 h) + \mathbf{D}_x(u\theta^2 h) + \mathbf{D}_y(v\theta^2 h) = 0, \quad (8.33)$$

$$\mathbf{D}_t\left(h(u^2 + v^2 + \theta h)\right) + \mathbf{D}_x\left(uh(u^2 + v^2 + 2\theta h)\right) + \mathbf{D}_y\left(vh(u^2 + v^2 + 2\theta h)\right) = 0, \quad (8.34)$$

$$\begin{aligned} \mathbf{D}_t\left(\theta(2\Omega - u_y + v_x)\right) + \mathbf{D}_x\left(\theta(u(2\Omega - u_y + v_x) - \frac{1}{2}\theta_y h)\right) \\ + \mathbf{D}_y\left(\theta(v(2\Omega - u_y + v_x) + \frac{1}{2}\theta_x h)\right) = 0. \end{aligned} \quad (8.35)$$

Conservation laws at Rank 4 are

$$\mathbf{D}_t(\theta^3 h) + \mathbf{D}_x(u\theta^3 h) + \mathbf{D}_y(v\theta^3 h) = 0, \quad (8.36)$$

$$\begin{aligned} \mathbf{D}_t\left(\theta^2(2\Omega - u_y + v_x)\right) + \mathbf{D}_x\left(\theta^2(u(2\Omega - u_y + v_x) - \frac{1}{2}\theta_y h)\right) \\ + \mathbf{D}_y\left(\theta^2(v(2\Omega - u_y + v_x) + \frac{1}{2}\theta_x h)\right) = 0. \end{aligned} \quad (8.37)$$

Conservation laws at Rank k , where integer $k > 5$ are

$$\mathbf{D}_t(\theta^{k-1} h) + \mathbf{D}_x(u\theta^{k-1} h) + \mathbf{D}_y(v\theta^{k-1} h) = 0, \quad (8.38)$$

$$\begin{aligned} \mathbf{D}_t\left(\theta^{k-2}(2\Omega - u_y + v_x)\right) + \mathbf{D}_x\left(\theta^{k-2}(u(2\Omega - u_y + v_x) - \frac{1}{2}\theta_y h)\right) \\ + \mathbf{D}_y\left(\theta^{k-2}(v(2\Omega - u_y + v_x) + \frac{1}{2}\theta_x h)\right) = 0. \end{aligned} \quad (8.39)$$

All of the conservation laws listed above have physical meanings. Equation (8.29) is the conservation of angular momentum, while (8.31) and (8.32) are the conservation of linear momentum [58]. Conservation of mass is shown in (8.28) and conservation of total energy corresponds to (8.34) [23].

Using the methods shown for the KP equation, conservation laws (8.28), (8.30), (8.33), (8.36), and (8.38) can be generalized to

$$\mathbf{D}_t \left(f(\theta)h \right) + \mathbf{D}_x \left(f(\theta)uh \right) + \mathbf{D}_y \left(f(\theta)vh \right) = 0, \quad (8.40)$$

for an arbitrary function $f(\theta)$. Conservation law (8.40) represents the Casimir integral of motion [58], a conserved integral on the Poisson bracket. Similarly, conservation laws (8.35), (8.37), and (8.39) can be generalized to

$$\begin{aligned} \mathbf{D}_t \left(f(\theta)(2\Omega - u_y + v_x) \right) + \mathbf{D}_x \left(f(\theta)(u(2\Omega - u_y + v_x) - \frac{1}{2}\theta_y h) \right) \\ + \mathbf{D}_y \left(f(\theta)(v(2\Omega - u_y + v_x) + \frac{1}{2}\theta_x h) \right) = 0, \end{aligned} \quad (8.41)$$

for any function $f(\theta)$. This law is the conservation of vorticity on a contour of constant θ [23]. In general, vorticity is not conserved in (8.25). With the arbitrary function $f(\theta)$, `ConservationLawsMD.m` cannot directly calculate the densities in (8.40) and (8.41) but it can verify these densities. The homotopy operator computed the flux for (8.40), but was unable to do so for (8.41) because f is a function of a dependent variable. The flux for (8.41) is obtained based on the pattern that is revealed in the conservation laws computed for ranks 2 to k .

8.2.2 The Shallow Water Magnetohydrodynamics Equations

The second system of equations are shallow water magnetohydrodynamic equations [23]. The system is given as

$$\begin{aligned} \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + 2\boldsymbol{\Omega} \times \mathbf{u} &= -g\nabla h + (\mathbf{B} \cdot \nabla)\mathbf{B}, \\ \psi_t + \mathbf{u} \cdot (\nabla\psi) &= 0, \quad h_t + \nabla \cdot (h\mathbf{u}) = 0. \end{aligned} \quad (8.42)$$

The corresponding scalar equations for (8.42) are

$$\begin{aligned} u_t + uu_x + vu_y - 2\Omega v + gh_x + \frac{\psi_y}{h^3} (\psi_y h_x - \psi_{xy} h) + \frac{\psi_x}{h^3} (\psi_{2y} h - \psi_y h_y) &= 0, \\ v_t + uv_x + vv_y + 2\Omega u + gh_y + \frac{\psi_y}{h^3} (\psi_{2x} h - \psi_x h_x) + \frac{\psi_x}{h^3} (\psi_x h_y - \psi_{xy} h) &= 0, \\ \psi_t + u\psi_x + v\psi_y &= 0, \quad h_t + uh_x + u_x h + vh_y + v_y h &= 0, \end{aligned} \quad (8.43)$$

where \mathbf{u} and h are the same as in (8.25), ψ is a flux function for the horizontal magnetic field, \mathbf{B} is the vertically averaged Lorentz force due to the magnetic field, and g is acceleration due to gravity. The equations can be written in terms of \mathbf{u} , ψ , and h

taking $\mathbf{B} = \frac{1}{h}(-\psi_y, \psi_x, 0)$. This representation for \mathbf{B} satisfies the constraint $\nabla(h\mathbf{B}) = 0$ in conventional magnetohydrodynamics [23].

Conservation laws computed by `ConservationLawsMD.m` for the magnetohydrodynamic equations are listed by rank as with previous cases. The ranks are based on the weights for (8.43),

$$\begin{aligned} W(u) = 1, \quad W(v) = 1, \quad W(\psi) = 2, \quad W(h) = 2, \quad W(\Omega) = 2, \\ W(D_t) = 2, \quad W(D_x) = 1, \quad W(D_y) = 1, \end{aligned} \quad (8.44)$$

where Ω is a weighted parameter. Conservation laws at Rank 2 are

$$D_t(h) + D_x(uh) + D_y(vh) = 0, \quad (8.45)$$

$$\begin{aligned} D_t\left(h(xv - yu + \Omega(x^2 + y^2))\right) + D_x\left(h(u(\Omega(x^2 + y^2) - yu + xv) - \frac{1}{2}gyh)\right) \\ + \frac{\psi_y}{h}(x\psi_x + y\psi_y) + D_y\left(h(v(\Omega(x^2 + y^2) - yu + xv) + \frac{1}{2}gxh)\right) \\ - \frac{\psi_x}{h}(x\psi_x + y\psi_y) = 0. \end{aligned} \quad (8.46)$$

Conservation laws at Rank 3 are

$$D_t\left(h(u - 2\Omega y)\right) + D_x\left(h(u(u - 2\Omega y) + \frac{1}{2}gh) - \frac{\psi_y^2}{h}\right) + D_y\left(hv(u - 2\Omega y) + \frac{\psi_x\psi_y}{h}\right) = 0, \quad (8.47)$$

$$D_t\left(h(v + 2\Omega x)\right) + D_x\left(hv(v + 2\Omega x) + \frac{\psi_x\psi_y}{h}\right) + D_y\left(h(v(v + 2\Omega x) + \frac{1}{2}gh) - \frac{\psi_x^2}{h}\right) = 0. \quad (8.48)$$

Conservation laws at Rank 4 are

$$D_t(\psi h) + D_x(u\psi h) + D_y(v\psi h) = 0, \quad (8.49)$$

$$\begin{aligned} D_t\left(h(u^2 + v^2 + gh) + \frac{1}{h}(\psi_x^2 + \psi_y^2)\right) + D_x\left(uh(u^2 + v^2 + 2gh)\right) \\ + \frac{\psi}{h^2}(u\psi_{2y}h - u\psi_y h_y + u_y\psi_y h - v\psi_{xy}h) + \frac{\psi_x}{h^2}(u\psi_x h + v\psi_y h + v\psi h_y - v_y\psi h) \\ + D_y\left(vh(u^2 + v^2 + 2gh) - \frac{\psi}{h^2}(u\psi_{xy}h + v\psi_x h_x - v\psi_{2x}h - v_x\psi_x h)\right) \\ + \frac{\psi_y}{h^2}(u\psi h_x + u\psi_x h - u_x\psi h + v\psi_y h) = 0, \end{aligned} \quad (8.50)$$

$$\begin{aligned} D_t\left(\psi(2\Omega - u_y + v_x)\right) + D_x\left(\psi(u(2\Omega - u_y + v_x) + \frac{\psi_y}{h^3}(\psi_{2x}h + \psi_{2y}h - \psi_x h_x - \psi_y h_y))\right) \\ + D_y\left(\psi(v(2\Omega - u_y + v_x) - \frac{\psi_x}{h^3}(\psi_{2x}h + \psi_{2y}h - \psi_x h_x - \psi_y h_y))\right) = 0. \end{aligned} \quad (8.51)$$

Although the density in equation (8.50) has rational terms and cannot be computed directly by `ConservationLawsMD.m`, the program has verified the density and calculated the flux.

Again, (8.46) is the conservation of angular momentum while both (8.47) and (8.48) express conservation of linear momentum. Conservation law (8.45) is the conservation of mass, and (8.50) is the conservation of total energy [23].

Like the shallow water system (8.25), two generalized conservation laws can be deduced from the laws calculated by `ConservationLawsMD.m`. The first describes the Casimir integral of motion,

$$\mathbf{D}_t(f(\psi)h) + \mathbf{D}_x(f(\psi)uh) + \mathbf{D}_y(f(\psi)vh) = 0, \quad (8.52)$$

where $f(\psi)$ is any function. The second generalization is the conservation of vorticity on a contour of constant ψ ,

$$\begin{aligned} \mathbf{D}_t\left(f(\psi)(2\Omega - u_y + v_x)\right) + \mathbf{D}_x\left(f(\psi)(u(2\Omega - u_y + v_x) + \frac{\psi_y}{h^3}(\psi_{2x}h + \psi_{2y}h - \psi_x h_x \right. \\ \left. - \psi_y h_y))\right) + \mathbf{D}_y\left(f(\psi)(v(2\Omega - u_y + v_x) - \frac{\psi_x}{h^3}(\psi_{2x}h + \psi_{2y}h - \psi_x h_x - \psi_y h_y))\right) = 0. \end{aligned} \quad (8.53)$$

Equation (8.52) has been confirmed by `ConservationLawsMD.m`, however, only the density of (8.53) can be confirmed by the program. The flux was determined by an analysis of conservation laws with $f(\psi) = \psi^n$ since the homotopy operator was unable to compute the flux involving an arbitrary function of dependent variable ψ .

8.2.3 The Thermal Shallow Water Magnetohydrodynamics Equations

The third system combines the properties of horizontal temperature fields and magnetic fields from the first two systems. The thermal shallow water magnetohydrodynamics system [23] is given by

$$\begin{aligned} \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + 2\mathbf{\Omega} \times \mathbf{u} &= -\nabla(\theta h) + \frac{1}{2}h\nabla\theta + (\mathbf{B} \cdot \nabla)\mathbf{B}, \\ \theta_t + \mathbf{u} \cdot (\nabla\theta) &= 0, \quad \psi_t + \mathbf{u} \cdot (\nabla\psi) = 0, \quad h_t + \nabla \cdot (h\mathbf{u}) = 0, \end{aligned} \quad (8.54)$$

where \mathbf{u} , θ , ψ , and h are as described in (8.25) and (8.42), and $\mathbf{B} = \frac{1}{h}(-\psi_y, \psi_x, 0)$. The

scalar equations for (8.54) are

$$\begin{aligned}
u_t + uu_x + vu_y - 2\Omega v + \theta h_x + \frac{1}{2}\theta_x h + \frac{\psi_y}{h^3}(\psi_y h_x - \psi_{xy} h) + \frac{\psi_x}{h^3}(\psi_{2y} h - \psi_y h_y) &= 0, \\
v_t + uv_x + vv_y + 2\Omega u + \theta h_y + \frac{1}{2}\theta_y h + \frac{\psi_y}{h^3}(\psi_{2x} h - \psi_x h_x) + \frac{\psi_x}{h^3}(\psi_x h_y - \psi_{xy} h) &= 0, \\
\theta_t + u\theta_x + v\theta_y = 0, \quad \psi_t + u\psi_x + v\psi_y = 0, \quad h_t + uh_x + u_x h + vh_y + v_y h &= 0.
\end{aligned} \tag{8.55}$$

`ConservationLawsMD.m` finds several conservation laws for (8.55). An investigation into (8.54) was not carried out in [23], thus none of these conservation laws have been stated before. The ranks for the conservation laws are determined by the weights,

$$\begin{aligned}
W(u) = 1, \quad W(v) = 1, \quad W(\theta) = 1, \quad W\psi = 1, \quad W(h) = 1, \\
W(\Omega) = 2 \quad W(D_t) = 2, \quad W(D_x) = 1, \quad W(D_y) = 1.
\end{aligned} \tag{8.56}$$

A conservation law for the combined system at rank 1 is

$$D_t(h) + D_x(uh) + D_y(vh) = 0. \tag{8.57}$$

Conservation laws at rank 2 are

$$D_t(\theta h) + D_x(u\theta h) + D_y(v\theta h) = 0, \tag{8.58}$$

$$D_t(\psi h) + D_x(u\psi h) + D_y(v\psi h) = 0, \tag{8.59}$$

$$D_t\left(h(u - 2\Omega y)\right) + D_x\left(h(u^2 + \frac{1}{2}\theta h - 2\Omega y u) - \frac{1}{h}\psi_y^2\right) + D_y\left(hv(u - 2\Omega y) + \frac{1}{h}\psi_x\psi_y\right) = 0, \tag{8.60}$$

$$D_t\left(h(v + 2\Omega x)\right) + D_x\left(hu(v + 2\Omega x) + \frac{1}{h}\psi_x\psi_y\right) + D_y\left(h(v^2 + \frac{1}{2}\theta h + 2\Omega x v) - \frac{1}{h}\psi_x^2\right) = 0. \tag{8.61}$$

Conservation laws at rank 3 are

$$D_t(\theta^2 h) + D_x(u\theta^2 h) + D_y(v\theta^2 h) = 0, \tag{8.62}$$

$$D_t(\psi^2 h) + D_x(u\psi^2 h) + D_y(v\psi^2 h) = 0, \tag{8.63}$$

$$D_t(\theta\psi h) + D_x(u\theta\psi h) + D_y(v\theta\psi h) = 0, \tag{8.64}$$

$$\begin{aligned}
D_t\left(h(u^2 + v^2 + \theta h) + \frac{1}{h}(\psi_x^2 + \psi_y^2)\right) + D_x\left(uh(u^2 + v^2 + 2\theta h) \right. \\
+ \frac{\psi}{h^2}(u\psi_{2y}h - u\psi_y h_y + u_y\psi_y h - v\psi_{xy}h) + \frac{\psi_x}{h^2}(u\psi_x h + v\psi_y h + v\psi h_y - v_y\psi h) \\
+ D_y\left(vh(u^2 + v^2 + 2\theta h) - \frac{\psi}{h^2}(u\psi_{xy}h + v\psi_x h_x - v\psi_{2x}h - v_x\psi_x h) \right. \\
\left. + \frac{\psi_y}{h^2}(u\psi h_x + u\psi_x h - u_x\psi h + v\psi_y h)\right) = 0.
\end{aligned} \tag{8.65}$$

Conservation law (8.57) is conservation of mass, (8.60) and (8.61) express conservation of linear momentum, and conservation of total energy is given by (8.65). An analysis of higher ranks leads to the general conservation laws of Casimirs,

$$D_t\left(f(\theta, \psi)h\right) + D_x\left(f(\theta, \psi)uh\right) + D_y\left(f(\theta, \psi)vh\right) = 0, \quad (8.66)$$

where $f(\theta, \psi)$ is an arbitrary function. Again, `ConservationLawsMD.m` verifies (8.66). For (8.55), vorticity is not conserved.

8.3 The Khokhlov-Zabolotskaya Equation

The Khokhlov-Zabolotskaya (KZ) equation, sometimes called the dispersionless KP equation, describes the propagation of sound in non-linear media in two or three space dimensions [57]. Conservation laws for both the (2+1)-dimensional and (3+1)-dimensional versions of the KZ equation are given in this section.

8.3.1 The (2+1)-Dimensional Khokhlov-Zabolotskaya Equation

In (2+1) dimensions, the KZ equation is

$$(u_t - uu_x)_x - u_{2y} = 0. \quad (8.67)$$

Like the KP equation, by interchanging y and t in (8.67), then setting v equal to u_t , a system of evolution equations for the KZ equation are found to be

$$\begin{aligned} u_t &= v, \\ v_t &= u_{xy} - u_x^2 - uu_{2x}. \end{aligned} \quad (8.68)$$

The KZ equation is a conservation law itself,

$$D_t\left(u_x\right) - D_x\left(uu_x\right) - D_y\left(u_y\right) = 0, \quad (8.69)$$

representing the conservation of momentum.

The conservation laws shown for the (2+1)-dimensional KZ equation were discovered by applying the method used for the KP equation. `ConservationLawsMD.m` computes densities tu , tyu , t^2u , t^2yu , t^3u , t^3yu , and so on to large powers of t , and these are the only densities found. It is easy to discover two general densities, $t^n u$ and $t^n yu$ by observing the list produced by the program. It is conjectured that t^n can be

replaced by arbitrary $f = f(t)$. The forms of the densities, fu and fyu , are submitted to `ConservationLawsMD.m` for testing. The program confirms that these proposed densities are correct densities for the KZ equation, and computes the fluxes. The conservation laws returned by the program are

$$\mathbf{D}_t(fu) - \mathbf{D}_x\left(\frac{1}{2}fu^2 + (fx + \frac{1}{2}f'y^2)(u_t - uu_x)\right) + \mathbf{D}_y\left((fx + \frac{1}{2}f'y^2)u_y - f'yu\right) = 0, \quad (8.70)$$

and

$$\begin{aligned} \mathbf{D}_t(fyu) - \mathbf{D}_x\left(\frac{1}{2}fyu^2 + (fxy + \frac{1}{6}f'y^3)(u_t - uu_x)\right) + \mathbf{D}_y\left((fxy + \frac{1}{6}f'y^3)u_y \right. \\ \left. - (fx + \frac{1}{2}f'y^2)u\right) = 0, \end{aligned} \quad (8.71)$$

where $f = f(t)$ is arbitrary. No other conservation laws were found for this equation.

Sharomet [65] reports four conservation laws for (8.67), of which two are similar to (8.70) and (8.71), namely,

$$\begin{aligned} \mathbf{D}_t(bu) - \mathbf{D}_x\left((bx + \frac{1}{2}b'y^2)(u_t - uu_x) + \frac{1}{2}bu^2\right) + \mathbf{D}_y\left((bx + \frac{1}{2}b'y^2)u_y \right. \\ \left. + (\frac{1}{y}bx - \frac{1}{2}b'y)u\right) = 0, \end{aligned} \quad (8.72)$$

and

$$\begin{aligned} \mathbf{D}_t(ayu) - \mathbf{D}_x\left((axy + \frac{1}{6}a'y^3)(u_t - uu_x) + \frac{1}{2}ayu^2\right) + \mathbf{D}_y\left((axy + \frac{1}{6}a'y^2)u_y \right. \\ \left. - \frac{1}{3}\alpha'y^2u\right) = 0, \end{aligned} \quad (8.73)$$

where $a = a(t)$, $b = b(t)$, and $\alpha = \alpha(t)$. An investigation and comparison of results found discrepancies in Sharomet's results. Neither (8.73) nor (8.72) law reduces to zero on the left-hand side when the total derivatives are applied and substitutions are made using (8.67). In the y -component of (8.73), no description for α is given and the term $\frac{1}{3}\alpha'y^2u$ is incomplete. It is possible that Sharomet meant to write $\alpha' = a'$, yet one still cannot obtain equality in the continuity equation. Three changes are needed; the term α' needs to be $\frac{3}{2}a'$, the term axu needs to be added to the y -component of the flux, and the term $(axy + \frac{1}{6}a_t y^2)u_y$ needs to be changed to $(axy + \frac{1}{6}a'y^3)u_y$. In (8.72), the term $(\frac{1}{y}b(t)x - \frac{1}{2}b'y)u$ needs to be replaced by $b'yu$. After the changes, (8.72) and (8.73) match the results in (8.70) and (8.71). The remaining two conservation laws given by Sharomet are trivial using the definition in Section 5.2.

8.3.2 The (3+1)-Dimensional Khokhlov-Zabolotskaya Equation

The (3+1)-dimensional KZ equation,

$$(u_t - uu_x)_x - u_{2y} - u_{2z} = 0, \quad (8.74)$$

has conservation laws of a different nature than those for the (2+1)-dimensional KZ equation. By interchanging t and z , and introducing a new variable $v = u_t$, an evolution system for the KZ equation is

$$\begin{aligned} u_t &= v, \\ v_t &= u_{xz} - u_x^2 - uu_{2x} - u_{2y}. \end{aligned} \quad (8.75)$$

Like the (2+1)-dimensional version, the (3+1)-dimensional KZ equation is itself a conservation law,

$$D_t(u_x) - D_x(uu_x) - D_y(u_y) - D_z(u_z) = 0. \quad (8.76)$$

`ConservationLawsMD.m` produces a variety of conservation laws for this equation, all with densities of the form $P(t, y, z)u$ or $Q(t, y, z)xu_x$, where P and Q are polynomials. Table 8.2 shows a list of densities calculated by the program. Ranks for the densities are determined by the weight system for (8.74) where

$$W(u) = 2, \quad W(D_t) = 3, \quad W(D_x) = 1, \quad W(D_y) = 2, \quad W(D_z) = 2. \quad (8.77)$$

The density $\rho_1 = xu_x$ from Table 8.2 is from the conservation law

$$D_t(xu_x) + D_x\left(\frac{1}{2}u^2 - xuu_x\right) - D_y(xu_y) - D_z(xu_z) = 0,$$

which can be rewritten as

$$D_t(u) - D_x\left(\frac{1}{2}u^2 + x(u_t - uu_x)\right) + D_y(xu_y) + D_z(xu_z) = 0$$

by swapping terms in the density and the x -component of the flux. A similar swap can be made for all of the densities with the factor xu_x in Table 8.2 so that every density in the table has the form $\rho = P(t, y, z)u$. Introducing arbitrary functions $f = f(t, y, z)$ and $g = g(t, y, z)$, the conservation laws corresponding to the densities in Table 8.2 can be generalized to a single equation,

$$\begin{aligned} D_t(fu) - D_x\left(\frac{1}{2}fu^2 + (fx + g)(u_t - uu_x)\right) - D_y\left((f_yx + g_y)u - (fx + g)u_y\right) \\ - D_z\left((f_zx + g_z)u - (fx + g)u_z\right) = -(x(f_{2z} + f_{2y}) + g_{2y} + g_{2z} - f_t)u. \end{aligned} \quad (8.78)$$

Table 8.2: Table of densities calculated by `ConservationLawsMD.m` for the (3+1)-dimensional Khokhlov-Zabolotskaya (8.74) equation.

Rank	Densities
2	$\rho_1 = xu_x$
0	$\rho_2 = xyu_x, \quad \rho_3 = xzu_x$
-1	$\rho_4 = tu$
-2	$\rho_5 = xyzu_x, \quad \rho_6 = (y^2 - z^2)xu_x$
-3	$\rho_7 = tyu, \quad \rho_8 = tzu$
-4	$\rho_9 = t^2u, \quad \rho_{10} = (y^3 - 3yz^2)xu_x, \quad \rho_{11} = (3yz^2 - z^3)xu_x$
-5	$\rho_{12} = tyzu, \quad \rho_{13} = (y^2 - z^2)tu$
-6	$\rho_{14} = t^2yu, \quad \rho_{15} = t^2xzu, \quad \rho_{16} = (y^3z - yz^3)xu_x,$ $\rho_{17} = (y^4 - 6y^2z^2 + z^4)xu_x$
-7	$\rho_{18} = t^3u, \quad \rho_{19} = (y^3 - 3yz^2)tu, \quad \rho_{20} = (3y^2z - z^3)tu$
-8	$\rho_{21} = t^2yzu, \quad \rho_{22} = (y^2 - z^2)t^2u, \quad \rho_{23} = (y^5 - 10y^3z^2 + 5yz^4)xu_x,$ $\rho_{24} = (5y^4z - 10y^2z^3 + z^5)xu_x$
-9	$\rho_{25} = t^3yu, \quad \rho_{26} = t^3xzu, \quad \rho_{27} = (y^3z - yz^3)tu, \quad \rho_{28} = t(y^4 - 6y^2z^2 + z^4)u$
-10	$\rho_{29} = t^4u, \quad \rho_{30} = (y^3 - 3yz^2)t^2u, \quad \rho_{31} = (3y^2z - z^3)t^2u$
-11	$\rho_{32} = t^3yzu, \quad \rho_{33} = (y^2 - z^2)t^3u, \quad \rho_{34} = (y^5 - 10y^3z^2 + 5yz^4)tu,$ $\rho_{35} = (5y^4z - 10y^2z^3 + z^5)tu$

Equation (8.78) becomes a conservation law when the constraints

$$\Delta f = 0, \tag{8.79}$$

$$\Delta g = f_t \tag{8.80}$$

are satisfied. Combining the two-dimensional Laplace equation (8.79) and the two-dimensional Poisson equation (8.80) produces a biharmonic equation in g . Indeed, taking the partial derivative of (8.79) with respect to t ,

$$\Delta f_t = 0, \tag{8.81}$$

then, applying the Laplacian to both sides of (8.80) produces

$$\Delta(\Delta g) = 0. \tag{8.82}$$

General solutions [69] for (8.82) are

$$g = yg_1(y, z) + g_2(y, z),$$

$$g = zg_1(y, z) + g_2(y, z),$$

where g_1 and g_2 are harmonic functions, that is, $\Delta g_1 = 0$ and $\Delta g_2 = 0$. Taking t as a parameter, four solutions for $g(t, y, z)$ are

$$g(t, y, z) = \frac{1}{2}y\partial_y^{-1}f_t(t, y, z), \quad (8.83)$$

$$g(t, y, z) = \frac{1}{2}\partial_y^{-1}(yf_t) = \frac{1}{2}(y\partial_y^{-1}f_t(t, y, z) - \partial_y^{-2}f_t(t, y, z)), \quad (8.84)$$

$$g(t, y, z) = \frac{1}{2}z\partial_z^{-1}f_t(t, y, z), \quad (8.85)$$

$$g(t, y, z) = \frac{1}{2}\partial_z^{-1}(zf_t) = \frac{1}{2}(z\partial_z^{-1}f_t(t, y, z) - \partial_z^{-2}f_t(t, y, z)), \quad (8.86)$$

giving a direct relationship between f and g in (8.78). For every conservation law corresponding to the densities in Table 8.2, g can be found using one of (8.83) - (8.86).

Example 8.1. *The full conservation law for ρ_{13} in Table 8.2 is*

$$\begin{aligned} & \mathbf{D}_t\left(tu(y^2 - z^2)\right) - \mathbf{D}_x\left(\frac{1}{2}tu^2(y^2 - z^2) + (tx(y^2 - z^2) + \frac{1}{2}z^2(y^2 - \frac{1}{3}z^2))(u_t - uu_x)\right) \\ & - \mathbf{D}_y\left(yu(2tx + z^2) - u_y(tx(y^2 - z^2) + \frac{1}{2}z^2(y^2 - \frac{1}{3}z^2))\right) \\ & + \mathbf{D}_z\left(zu(2tx - y^2 + \frac{2}{3}z^2) + u_z(tx(y^2 - z^2) + \frac{1}{2}z^2(y^2 - \frac{1}{3}z^2))\right) = 0. \end{aligned}$$

This conservation law is the same as (8.78) with $f = t(y^2 - z^2)$ and $g = \frac{1}{2}z^2(y^2 - \frac{1}{3}z^2)$. In this case, g is related to f using (8.85), as follows,

$$\begin{aligned} g &= \frac{1}{2}z\partial_z^{-1}(t(y^2 - z^2))_t \\ &= \frac{1}{2}z\partial_z^{-1}(y^2 - z^2) \\ &= \frac{1}{2}z(y^2z - \frac{1}{3}z^3) \\ &= \frac{1}{2}z^2(y^2 - \frac{1}{3}z^2). \end{aligned}$$

A generalized conservation law for the KZ equation (8.74) similar to (8.78) is reported by Sharomet [65], however, the term $g(u_t - uu_x)$ is missing from the x -component of the flux. This is easily detected by finding the total derivatives in the continuity equation, then making substitutions using (8.74). The result is nonzero after applying the constraints (8.79) and (8.80). A different result was given by Sanders [61], which also contained errors in the density and the x -component of the flux. Sanders has since corrected his conservation law [62], which is now equivalent to (8.78).

Results were also obtained using Wolf's CONLAW2 program [76, 77]. In this case, the complex transformations $y = \frac{1}{2i}(q - p)$ and $z = \frac{1}{2}(q + p)$ were applied to (8.74) and

the program was run on the transformed equation,

$$u_{tx} = 4u_{pq} + u_x^2 + uu_x.$$

The transformation removes the constraints, producing

$$D_t(hu) - D_x\left(\frac{1}{2}hu^2 - (h_q x + \frac{1}{4}h_t p)(uu_x - u_t)\right) + D_p\left(u_q(4h_q x + h_t p)\right) - D_q(h_t u) = 0,$$

where $h = h(q, t)$ is an arbitrary function without constraints. The program also produces a complex conjugate of the conservation law and two simpler forms corresponding to special choices for f and g in (8.78). All four equations are equivalent to (8.78) when the inverse transformation is applied.

8.4 The Coupled Korteweg-de Vries Equations

The coupled KdV equation is a (2+1)-dimensional integrable extension of the well-known KdV equation (3.8). The (2+1)-dimensional coupling of the KdV equation was produced using a perturbation expansion of u with two independent space variables, x and y [49]. The first order perturbation produces the system

$$\begin{aligned} u_t &= u_{3x} + 6uu_x, \\ v_t &= v_{3x} + 3u_{2xy} + 6(uv)_x + 6uu_y. \end{aligned} \tag{8.87}$$

The system is considered to be integrable and possesses the Painlevé property⁵ [80].

Conservation laws for (8.87) are given below. Ranks are computed using the weights

$$W(u) = 2, \quad W(v) = 2, \quad W(D_t) = 3, \quad W(D_x) = 1, \quad W(D_y) = 1. \tag{8.88}$$

The conservation law for rank 0 is

$$\begin{aligned} D_t\left(tu(xu - 2yv) + \frac{1}{3}x(xu - yv)\right) - D_x\left(tx(4u^3 - u_x^2 + 2uu_{2x}) - 2ty(6u^2v + uv_{2x}) \right. \\ \left. + 2uu_{xy} - u_x u_y - u_x v_x + u_{2x}v\right) + x^2\left(u^2 + \frac{1}{3}u_{2x}\right) - xy\left(2uv + u_{xy} + \frac{1}{3}v_{2x}\right) - \frac{2}{3}xu_x \\ \left. + \frac{1}{3}y(u_y + v_x)\right) + D_y\left(ty(4u^3 - u_x^2 + 2uu_{2x}) + xyu^2 - \frac{2}{3}yu_x\right) = 0. \end{aligned} \tag{8.89}$$

Conservation laws for rank 1 are

$$D_t\left(u\left(tu + \frac{1}{3}x\right)\right) - D_x\left(t(4u^3 - u_x^2 + 2uu_{2x}) + x\left(u^2 + \frac{1}{3}u_{2x}\right) - \frac{1}{3}u_x\right) + D_y(0) = 0, \tag{8.90}$$

⁵See the footnote in Chapter 3 for a description.

$$\mathbf{D}_t(xu - yv) - \mathbf{D}_x(x(3u^2 + u_{2x}) - y(6uv + 2u_{xy} + v_{2x})) + \mathbf{D}_y(y(3u^2 + u_{2x})) = 0, \quad (8.91)$$

$$\begin{aligned} \mathbf{D}_t\left(v(tu + \frac{1}{6}x)\right) - \mathbf{D}_x\left(t(6u^2v + 3uu_{xy} - u_xv_x + uv_{2x} + u_{2x}v) - xy(2uv + u_{xy} + \frac{1}{3}v_{2x}) \right. \\ \left. + x(uv + \frac{1}{6}v_{2x}) - \frac{1}{6}v_x\right) - \mathbf{D}_y\left(t(2u^3 - \frac{3}{2}u_x^2) + \frac{1}{2}x(u^2 + u_{2x})\right) = 0. \end{aligned} \quad (8.92)$$

Conservation laws for rank 2 are

$$\mathbf{D}_t(u) - \mathbf{D}_x(3u^2 + u_{2x}) + \mathbf{D}_y(0) = 0, \quad (8.93)$$

$$\mathbf{D}_t(v) - \mathbf{D}_x(6uv + v_{2x}) + \mathbf{D}_y(-3u^2 - 3u_{2x}) = 0, \quad (8.94)$$

$$\begin{aligned} \mathbf{D}_t(yu(xu - yv)) - \mathbf{D}_x(xy(4u^3 - u_x^2 + 2uu_{2x}) - y^2(6u^2v + 2uu_{xy} + uv_{2x} - u_xu_y \\ - u_xv_x + u_{2x}v)) + \mathbf{D}_y(y^2(2u^3 - \frac{1}{2}u_x^2 + uu_{2x})) = 0. \end{aligned} \quad (8.95)$$

The first two conservation laws at rank 2 are a reformulation of (8.87). Conservation laws for rank 4 are

$$\mathbf{D}_t(u^2) - \mathbf{D}_x(4u^3 - u_x^2 + 2uu_{2x}) + \mathbf{D}_y(0) = 0, \quad (8.96)$$

$$\mathbf{D}_t(uv) - \mathbf{D}_x(6u^2v + 3uu_{xy} + uv_{2x} + vu_{2x} - u_xv_x) - \mathbf{D}_y(2u^3 - \frac{3}{2}u_x^2) = 0. \quad (8.97)$$

Conservation laws for rank 6 are

$$\mathbf{D}_t\left(u^3 - \frac{1}{2}u_x^2\right) - \mathbf{D}_x\left(\frac{9}{2}u^4 + 3u^2u_{2x} - 6uu_x^2 + \frac{1}{2}u_{2x}^2 - u_xu_{3x}\right) + \mathbf{D}_y(0) = 0, \quad (8.98)$$

$$\begin{aligned} \mathbf{D}_t\left(u^2v - \frac{1}{3}u_xu_y - \frac{1}{3}u_xv_x\right) - \mathbf{D}_x\left(6u^3v + u^2(4u_{xy} + v_{2x}) - 2u(2u_xv_x - u_{2x}v) - 2u_x^2v \right. \\ \left. - \frac{1}{3}u_x(u_{2xy} + v_{3x}) + \frac{1}{3}u_{2x}(u_{xy} + v_{2x}) - \frac{1}{3}u_{3x}(u_y + v_x)\right) + \mathbf{D}_y\left(-\frac{3}{2}u^4 + u^2u_{2x} \right. \\ \left. + 6uu_x^2 - \frac{1}{2}u_{2x}^2\right) = 0. \end{aligned} \quad (8.99)$$

`ConservationLawsMD.m` continues to find conservation laws at higher even numbered ranks. However, the number of terms in higher rank conservation laws makes them impractical to report.

Note that the first equation of (8.87) is a scaled version of the (1+1)-dimensional KdV equation (3.8). Conservation laws (8.90), (8.93), (8.96), and (8.98) are also conservation laws for the KdV equation. None of the conservation laws given here have been found in a search of literature.

8.5 The Manakov-Santini System

The (2+1)-dimensional Manakov-Santini (MS) system [50] given as

$$\begin{aligned} u_{tx} + u_{2y} + (uu_x)_x + v_x u_{xy} - u_{2x} v_y &= 0, \\ v_{tx} + v_{2y} + uv_{2x} + v_x v_{xy} - v_y v_{2x} &= 0, \end{aligned} \quad (8.100)$$

is an integrable system of PDEs. Note that when $v = 0$ the MS system reduces to the (2+1)-dimensional KZ equation (8.67). The MS system can be put into evolution form by first interchanging t with y then setting \tilde{u} and \tilde{v} equal to u_t and v_t , respectively, to get

$$\begin{aligned} u_t &= \tilde{u}, \\ v_t &= \tilde{v}, \\ \tilde{u}_t &= -u_{xy} - (uu_x)_x - \tilde{u}_x v_x + u_{2x} \tilde{v}, \\ \tilde{v}_t &= -v_{xy} - uv_{2x} - v_x \tilde{v}_x + \tilde{v} v_{2x}. \end{aligned} \quad (8.101)$$

`ConservationLawsMD.m` finds a number of conservation laws for the Manakov-Santini system, most of which have explicit t , x and y in the density and flux. Again, by analyzing the patterns in the densities, the conservation laws computed by the program can be reduced to five general equations. In each case, the general density was submitted to `ConservationLawsMD.m` for verification. After verifying the density, the program returned the conservation laws as follows. Taking $f = f(t)$ as an arbitrary function, three conservation laws are

$$\begin{aligned} D_t \left(f u_x v_x \right) + D_x \left(f (u u_x v_x - u_x v_x v_y - u_y v_y) - f' y (u_t + u u_x - u_x v_y) \right) \\ + D_y \left(f (u_x v_y + u_y v_x + u_x v_x^2) + f' (u - y u_y - y u_x v_x) \right) = 0, \end{aligned} \quad (8.102)$$

$$\begin{aligned} D_t \left(f (2u + v_x^2 - y u_x v_x) \right) + D_x \left(f (u^2 + u v_x^2 + u_y v - v_y^2 - v_x^2 v_y) \right. \\ \left. - y (u u_x v_x - u_x v_x v_y - u_y v_y) \right) - f' y (v_t + u v_x - v_x v_y) \\ - (2fx - f') y^2 (u_t + u u_x - u_x v_y) - D_y \left(f (u_x v - 2v_x v_y - v_x^3) \right. \\ \left. + y (u_x v_x^2 + u_x v_y + u_y v_x) \right) - f' (v - y (2u + v_y + v_x^2)) \\ \left. + (2fx - f' y^2) (u_x v_x + u_y) \right) = 0, \end{aligned} \quad (8.103)$$

$$\begin{aligned}
& D_t \left(f(u_x v_y + u_x v_x^2) + f'(u - y u_x v_x) \right) + D_x \left(f(u u_x v_y + u u_x v_x^2 + u u_y v_x \right. \\
& \quad \left. - u_x v_y^2 - u_x v_x^2 v_y + u_y v_t - u_y v_x v_y) + f' \left(\frac{1}{2} u^2 + u_y v \right. \right. \\
& \quad \left. \left. - y(u u_x v_x - u_x v_x v_y - u_y v_y) \right) - \left(f' x - \frac{1}{2} f'' y^2 \right) (u_t + u u_x - u_x v_y) \right) \\
& \quad - D_y \left(f(u u_x v_x + u_x v_t - u_x v_x^3 - 2 u_x v_x v_y - u_y v_y - u_y v_x^2) + f'(u_x v \right. \\
& \quad \left. + y(u_x v_x^2 + u_x v_y + u_y v_x)) - f'' y u - \left(f' x - \frac{1}{2} f'' y^2 \right) (u_x v_x + u_y) \right) = 0. \quad (8.104)
\end{aligned}$$

Since the fluxes for the last two conservation laws have a large number of terms, only the densities are given. The densities are

$$\begin{aligned}
\rho_4 &= f(3u_x v - v_x^3 - v_x v_y + y u_x (v_x^2 + v_y)) + (2f' x - f' y^2) u_x v_x \\
&\quad - f'(v - y(3u + v_x^2)), \quad (8.105)
\end{aligned}$$

$$\begin{aligned}
\rho_5 &= f(u u_{2x} v + u_x (u_x v - v_x v_y - v_x^3) + u_y (v_x^2 + v_y)) \\
&\quad - f'(u_x (v + y(v_x^2 + v_y))) + \frac{1}{2} (2f' x - f'' y^2) u_x v_x + f'' y u. \quad (8.106)
\end{aligned}$$

Like the KP equation, the existence of large families of conservation laws supports the fact that the MS equation is integrable. The MS equation has only recently appeared in literature [50, 51], and a study of conservation laws for (8.100) has not been found.

8.6 The Camassa-Holm Equation

The (2+1)-dimensional Camassa-Holm (CH) equation given in this section was derived in a study of water waves [42] as an extension of the (1+1)-dimensional CH equation. Although the (1+1)-dimensional CH equation [13] was derived as a completely integrable equation, a study by Gordoia et al. [31] suggests that the (2+1)-dimensional version may be non-integrable.

The (2+1)-dimensional CH equation,

$$(u_t + \kappa u_x - u_{t2x} + 3u u_x - 2u_x u_{2x} - u u_{3x})_x + u_{2y} = 0, \quad (8.107)$$

is related to the (1+1)-dimensional version in a way that is similar to the relationship of the KP equation to the KdV equation.

An evolution system of equations for the CH equation are found by interchanging t with y in (8.107), then setting $v = u_t$ to get

$$\begin{aligned}
u_t &= v, \\
v_t &= -u_{xy} - \kappa u_{2x} + u_{3xy} - 3u_x^2 - 3u u_{2x} + 2u_{2x}^2 + 3u_x u_{3x} + u u_{4x}. \quad (8.108)
\end{aligned}$$

However, (8.108) does not have a scaling symmetry. When `ConservationLawsMD.m` tries to compute the weights to determine the scaling symmetry for (8.108), the program finds

$$W(u) = W(D_y), \quad W(v) = \frac{3}{2}W(D_y), \quad W(D_t) = W(D_y), \quad W(D_x) = 0. \quad (8.109)$$

With $W(D_x) = 0$, the program cannot form a candidate density. To induce a scaling symmetry, parameters α and β must be introduced into (8.108) such that α and β carry weight. The existing parameter κ , must also carry weight. The balance equations for (8.108), with κ as a weighted parameter,

$$W(v) = W(u) + W(D_t), \quad (8.110)$$

$$W(v) + W(D_t) = W(\kappa) + W(u) + 2W(D_x) \quad (8.111)$$

$$= W(u) + W(D_x) + W(D_y) \quad (8.112)$$

$$= W(u) + 3W(D_x) + W(D_y) \quad (8.113)$$

$$= 2W(u) + 2W(D_x) \quad (8.114)$$

$$= 2W(u) + 4W(D_x), \quad (8.115)$$

can be used to determine where α and β should be placed. Clearly, the combination of (8.112) and (8.113) as well as (8.114) and (8.115) require $W(D_x) = 0$. To be able to build a candidate density, the weights must be positive. The balance equations will yield positive weights if $W(\alpha)$ is added to (8.112) and $W(\beta)$ is added to (8.114). The CH evolution equations, with weighted parameters α , β , and κ ,

$$u_t = v, \quad (8.116)$$

$$v_t = -\alpha u_{xy} - \kappa u_{2x} + u_{3xy} - 3\beta u_x^2 - 3\beta u u_{2x} + 2u_{2x}^2 + 3u_x u_{3x} + u u_{4x},$$

have a scaling symmetry with weights

$$\begin{aligned} W(u) = 1, \quad W(v) = \frac{7}{2}, \quad W(\alpha) = 2, \quad W(\beta) = 2, \\ W(\kappa) = 3, \quad W(D_t) = \frac{5}{2}, \quad W(D_x) = 1 \quad W(D_y) = 2. \end{aligned} \quad (8.117)$$

The CH equation with the additional weighted parameters,

$$(u_t + \kappa u_x - \alpha u_{t2x} + 3\beta u u_x - 2u_x u_{2x} - u u_{3x})_x + u_{2y} = 0, \quad (8.118)$$

exhibits conservation laws with densities similar to the densities found for the KP equation (8.1). The weighted parameters are shown in the conservation laws computed by

`ConservationLawsMD.m`. They can be removed by setting $\alpha = \beta = 1$. Like several other PDEs in this chapter, equation (8.118) is a conservation law itself,

$$D_t(\alpha u_x - u_{3x}) + D_x(\kappa u_x + 3\beta u u_x - 2u_x u_{2x} - u u_{3x}) + D_y(u_y) = 0. \quad (8.119)$$

Again, the procedure used to find generalizations for the KP equation are repeated here. The conservation laws produced by `ConservationLawsMD.m` can be generalized to two equations. `ConservationLawsMD.m` is able to verify these conservation laws testing and recomputing the coefficients for each density. The conservation laws are

$$D_t(fu) + D_x\left(\frac{1}{\alpha}f\left(\frac{3}{2}\beta u^2 + \kappa u - \frac{1}{2}u_x^2 - uu_{2x} - u_{tx}\right) - \left(\frac{1}{\alpha}fx - \frac{1}{2}f'y^2\right)(\alpha u_t + \kappa u_x + 3\beta uu_x - 2u_x u_{2x} - uu_{3x} - u_{t2x})\right) - D_y\left(\left(\frac{1}{\alpha}fx - \frac{1}{2}f'y^2\right)u_y + f'yu\right) = 0, \quad (8.120)$$

$$D_t(fyu) + D_x\left(\frac{1}{\alpha}fy\left(\frac{3}{2}\beta u^2 + \kappa u - \frac{1}{2}u_x^2 - uu_{2x} - u_{tx}\right) - y\left(\frac{1}{\alpha}fx - \frac{1}{6}f'y^2\right)(\alpha u_t + \kappa u_x + 3\beta uu_x - 2u_x u_{2x} - uu_{3x} - u_{t2x})\right) - D_y\left(yu_y\left(\frac{1}{\alpha}fx - \frac{1}{6}f'y^2\right) - u\left(\frac{1}{\alpha}fx - \frac{1}{2}f'y^2\right)\right) = 0, \quad (8.121)$$

where $f(t)$ is an arbitrary function.

Like the MS equation, the (2+1)-dimensional CH equation has only recently appeared in literature [31, 42]. Although a number of articles have been found for the (1+1)-dimensional CH equation, little appeared for the (2+1)-dimensional version. A study of conservation laws for (8.107) has not been found. The existence of families of conservation laws would contradict the claim by Gordoia et al., that the CH equation is not integrable.

8.7 Navier's Equation

Navier's equation is a (3+1)-dimensional linear PDE which describes wave motion in elastic solids. In vector form, the equation is given as

$$\hat{\rho} \frac{\partial^2 \mathbf{u}}{\partial t^2} = (\hat{\lambda} + \hat{\mu}) \nabla(\nabla \cdot \mathbf{u}) + \hat{\mu} \Delta \mathbf{u}, \quad (8.122)$$

where $\mathbf{u} = (u, v, w)$, $\hat{\lambda}$ and $\hat{\mu}$ are the Lamé constants, and $\hat{\rho}$ is the constant undisturbed density of the elastic solid [10]. In component form, (8.122) becomes the system

$$\hat{\rho} u_{2t} = (\hat{\lambda} + \hat{\mu})(u_{2x} + v_{xy} + w_{xz}) + \hat{\mu}(u_{2x} + u_{2y} + u_{2z}), \quad (8.123)$$

$$\hat{\rho} v_{2t} = (\hat{\lambda} + \hat{\mu})(u_{xy} + v_{2y} + w_{yz}) + \hat{\mu}(v_{2x} + v_{2y} + v_{2z}), \quad (8.124)$$

$$\hat{\rho}w_{2t} = (\hat{\lambda} + \hat{\mu})(u_{xz} + v_{yz} + w_{2z}) + \hat{\mu}(w_{2x} + w_{2y} + w_{2z}), \quad (8.125)$$

which can be easily written into evolution form by forming three new equations, $\tilde{u} = u_t$, $\tilde{v} = v_t$, and $\tilde{w} = w_t$ and replacing u_{2t} , v_{2t} , and w_{2t} with \tilde{u}_t , \tilde{v}_t , and \tilde{w}_t , respectively.

`ConservationLawsMD.m` found several conservation laws for Navier's equation. The ranks are based on the weights,

$$W(u) = W(v) = W(w) = W(D_t) = W(D_x) = W(D_y) = W(D_z) = 1. \quad (8.126)$$

The three component equations (8.124) are first order conservation laws representing the conservation in the stress tensor associated with the deformation of \mathbf{u} [55]. There are no constraints on the parameters $\hat{\lambda}$, $\hat{\mu}$, and $\hat{\rho}$ in any of the conservation laws that follow. Conservation laws at rank 2 are

$$D_t(\hat{\rho}u_t) - D_x((\hat{\lambda} + 2\hat{\mu})u_x) - D_y(\hat{\mu}u_y + (\hat{\lambda} + \hat{\mu})v_x) - D_z(\hat{\mu}u_z + (\hat{\lambda} + \hat{\mu})w_x) = 0, \quad (8.127)$$

$$D_t(\hat{\rho}v_t) - D_x(\hat{\mu}v_x + (\hat{\lambda} + \hat{\mu})u_y) - D_y((\hat{\lambda} + 2\hat{\mu})v_y) - D_z(\hat{\mu}v_z + (\hat{\lambda} + \hat{\mu})w_y) = 0, \quad (8.128)$$

$$D_t(\hat{\rho}w_t) - D_x(\hat{\mu}w_x + (\hat{\lambda} + \hat{\mu})u_z) - D_y(\hat{\mu}w_y + (\hat{\lambda} + \hat{\mu})v_z) - D_z((\hat{\lambda} + 2\hat{\mu})w_z) = 0, \quad (8.129)$$

a reformulation of (8.123) - (8.125). Conservation laws at rank 4 are

$$\begin{aligned} D_t & \left(\hat{\rho}(u_t u_x + v_t v_x + w_t w_x) \right) - D_x \left(\frac{1}{2}(\hat{\rho}(u_t^2 + v_t^2 + w_t^2) + (\hat{\lambda} + 2\hat{\mu})(u_x^2 + v v_{2y} \right. \\ & \quad \left. + w w_{2z}) + \hat{\mu}(u u_{2y} + u u_{2z} + v_x^2 + v v_{2z} + w_x^2 + w w_{2y}) + 2(\hat{\lambda} + \hat{\mu})w v_{yz}) \right) \\ & - D_y \left(\frac{1}{2}((\hat{\lambda} + 2\hat{\mu})(v_x v_y - v v_{xy}) + \hat{\mu}(u_x u_y - u u_{xy} + w_x w_y - w w_{xy}) \right. \\ & \quad \left. + 2(\hat{\lambda} + \hat{\mu})(u_x v_x - w v_{xz})) \right) - D_z \left(\frac{1}{2}((\hat{\lambda} + 2\hat{\mu})(w_x w_z - w w_{xz}) \right. \\ & \quad \left. + \hat{\mu}(u_x u_z - u u_{xz} + v_x v_z - v v_{xz}) + 2(\hat{\lambda} + \hat{\mu})(u_x w_x + v_x w_y)) \right) = 0, \end{aligned} \quad (8.130)$$

$$\begin{aligned} D_t & \left(\hat{\rho}(u_t u_y + v_t v_y + w_t w_y) \right) - D_x \left(\frac{1}{2}((\hat{\lambda} + 2\hat{\mu})(u_x u_y - u u_{xy}) + \hat{\mu}(v_x v_y - v v_{xy} \right. \\ & \quad \left. + w_x w_y - w w_{xy}) + 2(\hat{\lambda} + \hat{\mu})(u_y v_y - u w_{yz})) \right) - D_y \left(\frac{1}{2}(\hat{\rho}(u_t^2 + v_t^2 + w_t^2) \right. \\ & \quad \left. + (\hat{\lambda} + 2\hat{\mu})(u u_{2x} + v_y^2 + w w_{2z}) + \hat{\mu}(u_y^2 + u u_{2z} + v v_{2x} + v v_{2z} + w w_{2x} + w_y^2) \right. \\ & \quad \left. + 2(\hat{\lambda} + \hat{\mu})u w_{xz}) \right) - D_z \left(\frac{1}{2}((\hat{\lambda} + 2\hat{\mu})(w_y w_z - w w_{yz}) + \hat{\mu}(u_y u_z - u u_{yz} + v_y v_z \right. \\ & \quad \left. - v v_{yz}) + 2(\hat{\lambda} + \hat{\mu})(u_x w_y + v_y w_y)) \right) = 0, \end{aligned} \quad (8.131)$$

$$\begin{aligned}
& \mathbf{D}_t \left(\hat{\rho}(u_t u_z + v_t v_z + w_t w_z) \right) - \mathbf{D}_x \left(\frac{1}{2}((\hat{\lambda} + 2\hat{\mu})(u_x u_z - u u_{xz}) + \hat{\mu}(v_x v_z - v v_{xz}) \right. \\
& \quad \left. + w_x w_z - w w_{xz}) + 2(\hat{\lambda} + \hat{\mu})(u_z w_z - v u_{yz}) \right) - \mathbf{D}_y \left(\frac{1}{2}((\hat{\lambda} + 2\hat{\mu})(v_y v_z - v v_{yz}) \right. \\
& \quad \left. + \hat{\mu}(u_y u_z - u u_{yz} + w_y w_z - w w_{yz}) + 2(\hat{\lambda} + \hat{\mu})(u_z v_x + v_z w_x) \right) \\
& \quad - \mathbf{D}_z \left(\frac{1}{2}(\hat{\rho}(u_t^2 + v_t^2 + w_t^2) + (\hat{\lambda} + 2\hat{\mu})(u u_{2x} + v v_{2y} + w_z^2) + \hat{\mu}(u u_{2y} + u_z^2 \right. \\
& \quad \left. + v v_{2x} + v_z^2 + w w_{2x} + w w_{2y}) + 2(\hat{\lambda} + \hat{\mu})v u_{xy} \right) = 0, \tag{8.132}
\end{aligned}$$

$$\begin{aligned}
& \mathbf{D}_t \left(\frac{1}{2}(\hat{\rho}(u_t^2 + v_t^2 + w_t^2) + (\hat{\lambda} + 2\hat{\mu})(u_x^2 + v_y^2 + w_z^2) + \hat{\mu}(u_y^2 + u_z^2 + v_x^2 + v_z^2 + w_x^2 + w_y^2) \right. \\
& \quad \left. + 2(\hat{\lambda} + \hat{\mu})(u_x v_y + u_x w_z + v_y w_z) \right) + \mathbf{D}_x \left((\hat{\lambda} + 2\hat{\mu})u_t u_x - (\hat{\lambda} + \hat{\mu})(u_t v_y + u_t w_z) \right. \\
& \quad \left. - \hat{\mu}(v_t v_x + w_t w_x) \right) + \mathbf{D}_y \left((\hat{\lambda} + 2\hat{\mu})v_t v_y - (\hat{\lambda} + \hat{\mu})(v_t u_x + v_t w_z) - \hat{\mu}(u_t u_y + w_t w_y) \right) \\
& \quad + \mathbf{D}_z \left((\hat{\lambda} + 2\hat{\mu})w_t w_z - (\hat{\lambda} + \hat{\mu})(w_t u_x + w_t v_y) - \hat{\mu}(u_t u_z + v_t v_z) \right) = 0. \tag{8.133}
\end{aligned}$$

A list of polynomial densities found at rank 5 is given in Table 8.3. The fluxes are lengthy and therefore are not shown. They can be obtained using `ConservationLawsMD.m`.

Table 8.3: Table of densities at rank 5 calculated by `ConservationLawsMD.m` for the (3+1)-dimensional Navier equation (8.122).

$\rho_1 = u_x(v_{tz} - w_{ty}) - v_x(u_{tz} - w_{tx}) + w_x(u_{ty} - v_{tx})$
$\rho_2 = u_y(v_{tz} - w_{ty}) + v_x w_{ty} - v_y u_{tz} - w_x v_{ty} + w_y u_{ty}$
$\rho_3 = (u_y - v_x)w_{tz} - (u_z - w_x)v_{tz} + (v_z - w_y)u_{tz}$
$\rho_4 = \hat{\rho}(v_t u_{tz} - w_t(u_{ty} - v_{tx})) + \hat{\mu}(u_y w_{zz} + v_x(u_{xz} - w_{yy} - w_{zz}) + v_y u_{yz} + v_z u_{zz} \right. \\ \left. + w_x(v_{xx} - u_{xy}) - w_y u_{yy})$

Navier's equation also admits conservation laws with explicit independent variables, all of which reduce to a single general form. Let $f = f(t, x, y, z)$, $g = g(t, x, y, z)$, and $h = h(t, x, y, z)$, then the general conservation law is

$$\begin{aligned}
& \mathbf{D}_t \left(f u_t + g v_t + h w_t - f_t u - g_t v - h_t w \right) + \mathbf{D}_x \left(\frac{\hat{\lambda} + 2\hat{\mu}}{\hat{\rho}}(f_x u - f u_x) + \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(f_y v \right. \\
& \quad \left. + f_z w - g u_y - h u_z) + \frac{\hat{\mu}}{\hat{\rho}}(g_x v - g v_x + h_x w - h w_x) \right) + \mathbf{D}_y \left(\frac{\hat{\lambda} + 2\hat{\mu}}{\hat{\rho}}(g_y v - g v_y) \right. \\
& \quad \left. + \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(-f v_x + g_x u + g_z w - h v_z) + \frac{\hat{\mu}}{\hat{\rho}}(f_y u - f u_y + h_y w - h w_y) \right) \\
& \quad + \mathbf{D}_z \left(\frac{\hat{\lambda} + 2\hat{\mu}}{\hat{\rho}}(h_z w - h w_z) - \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(f w_x + g w_y - h_x u - h_y v) \right. \\
& \quad \left. + \frac{\hat{\mu}}{\hat{\rho}}(f_z u - f u_z + g_z v - g v_z) \right) = 0. \tag{8.134}
\end{aligned}$$

There are no constraints on the parameters $\hat{\lambda}$, $\hat{\mu}$, and $\hat{\rho}$, however, the following constraints on (8.134) occur for f , g , and h ,

$$f_{2t} = \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(f_{2x} + g_{xy} + h_{xz}) + \frac{\hat{\mu}}{\hat{\rho}}(f_{2x} + f_{2y} + f_{2z}), \quad (8.135)$$

$$g_{2t} = \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(f_{xy} + g_{2y} + h_{yz}) + \frac{\hat{\mu}}{\hat{\rho}}(g_{2x} + g_{2y} + g_{2z}), \quad (8.136)$$

$$h_{2t} = \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(f_{xz} + g_{yz} + h_{2z}) + \frac{\hat{\mu}}{\hat{\rho}}(h_{2x} + h_{2y} + h_{2z}). \quad (8.137)$$

These constraints state that f , g , and h must also satisfy Navier's equation. Every conservation law containing explicit t , x , y , and z computed by `ConservationLawsMD.m` is a special case of (8.134).

Example 8.2. `ConservationLawsMD.m` computes the following conservation law with a rank of -1 ,

$$\begin{aligned} & \mathbf{D}_t \left((t^2 y + \frac{\hat{\rho}}{\hat{\mu}} x^2 y) w_x - 2tyw \right) - \mathbf{D}_x \left(\frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}} t^2 y u_z - 2xyw + (\frac{\hat{\mu}}{\hat{\rho}} t^2 y + x^2 y) w_x \right) \\ & + \mathbf{D}_y \left((\frac{\hat{\mu}}{\hat{\rho}} t^2 + x^2) w - (\frac{\hat{\mu}}{\hat{\rho}} t^2 y + x^2 y) w_y \right) - \mathbf{D}_z \left(\frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}} x^2 y u_x + (\frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}} t^2 y + \frac{\hat{\lambda} + \hat{\mu}}{\hat{\mu}} x^2 y) v_y \right. \\ & \left. + (\frac{\hat{\lambda} + 2\hat{\mu}}{\hat{\rho}} t^2 y + \frac{\hat{\lambda} + 2\hat{\mu}}{\hat{\mu}} x^2 y) w_z \right) = 0. \end{aligned}$$

For this conservation law, $f = 0$, $g = 0$ and $h = t^2 y + \frac{\hat{\rho}}{\hat{\mu}} x^2 y$. Clearly, this choice for h holds for constraints (8.135) and (8.136) since any partial derivative of h with respect to z is 0. Constraint (8.137) also holds since $h_{2t} = 2y$ and

$$\frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(f_{xz} + g_{yz} + h_{2z}) + \frac{\hat{\mu}}{\hat{\rho}}(h_{2x} + h_{2y} + h_{2z}) = \frac{\hat{\lambda} + \hat{\mu}}{\hat{\rho}}(0) + \frac{\hat{\mu}}{\hat{\rho}}(\frac{\hat{\rho}}{\hat{\mu}}(2y)) = 2y.$$

In addition, `ConservationLawsMD.m` reports conservation laws which hold only under the condition $\hat{\lambda} = -\hat{\mu}$. Obviously, when $\hat{\lambda} + \hat{\mu} = 0$, one of the terms in (8.122) drops out. Therefore, these conservation laws are ignored.

First order conservation laws for a form of Navier's equation without time dependencies were analyzed by Olver. Conservation laws given in [55] match the fluxes in (8.127)-(8.129) and (8.130)-(8.132). The conservation law (8.133) does not have a counterpart since it is of second order.

CHAPTER 9

SOFTWARE PACKAGES

In previous chapters, the algorithmic methods used in `ConservationLawsMD.m` to compute conservation laws have been explained. In addition, the techniques and methods used in `HomotopyIntegrator.m` to invert divergences and `IndependenceTest.m` to verify independence of conservation laws were covered. All three programs are written in *Mathematica* syntax [79] and require *Mathematica* to run. This chapter addresses how to run each program and provides details on special features in the programs. Each program is part of a software package which includes a *Mathematica* notebook to get started and several examples.

All three packages were designed and thoroughly tested in versions 5, 5.1, and 5.2 of *Mathematica*. No compatibility problems have been discovered with versions 4, 6, or 7, but testing has not been as thorough on these versions. The user should be aware that a significant number of changes have been made to many standard *Mathematica* functions in versions 6 and 7, so unanticipated problems may appear. Also, if the user wants the program to print a large amount of information to the screen, the newer versions will run much slower.

9.1 The Software Package `ConservationLawsMD.m`

The package `ConservationLawsMD.m` contains the key algorithm for computing conserved densities, the algorithm from `HomotopyIntegrator.m` to compute fluxes and the algorithm from `IndependenceTest.m` for checking densities for independence. The package is menu driven, offering thirty-nine (1+1)-dimensional examples and forty (2+1)- and (3+1)-dimensional examples. The program has two user modes. It will either prompt the user for information about how the density is to be constructed or it can be set to run automatically, without requiring any interaction from the user. If a user wishes to investigate a PDE not listed in the menu, the information can be entered into the program via a separate data file.

Before running the program, the user needs to place the program file and all data

files provided with the program into one directory. Any additional data files created by the user must also be placed in the same directory. Since all code and data are stored in files separate from the notebook, *Mathematica* needs to be instructed where to look. Details about how a data file for a PDE is constructed will be given in Section 9.1.2.

To start `ConservationLawsMD.m`, open a *Mathematica* notebook. First set the directory with the command,

```
SetDirectory["c:\\ set your directory path here "].
```

After each command, *Mathematica* requires the user to press shift-enter to accept the entry. Next, load the program by executing the command

```
Get["ConservationLawsMD.m"].
```

To start the program, type

```
ConservationLawsMD[].
```

A menu will appear containing the four items shown in Figure 9.1. If the user chooses option 1, a new menu will appear listing examples in (1+1)-dimensions. Option 2 has examples in (2+1)- and (3+1)-dimensions. Option `tt` lets the user supply a data file to the program. Option `qq` immediately aborts the program.

```
-----  
1) Cases in (1+1) Dimensions  
2) Cases in (2+1) and (3+1) Dimensions  
tt) Take Equation or System from a File  
qq) Exit the Program  
-----
```

Figure 9.1: The menu that appears when `ConservationLawsMD[]` is initiated.

9.1.1 Running a Test Example

A demonstration of the program can be given by running any one of the test examples provided with the program. The procedure for running the algorithm on the

(2+1)-dimensional Zakharov-Kuznetsov equation (3.7) is shown here so that the user can see the steps involved.

To begin, execute the command **ConservationLawsMD[]** in a *Mathematica* notebook after the program has been loaded. When the menu in Figure 9.1 appears, respond with 2. A second menu giving a list of PDEs will appear as shown in Figure 9.2. To run the (2+1)-dimensional ZK equation, choose option 1. The program will

```

-----
*** MENU INTERFACE *** (page: 1)
1) (2+1) Zakharov-Kuznetsov (ZK) Equation (d_zk2d.m)
2) (3+1) Zakharov-Kuznetsov (ZK) Equation (d_zk3d.m)
3) (2+1) Kadomtsev-Petviashvili (KP) Equation (d_kp2d.m)
4) (2+1) Potential KP Equation (d_kp2dpo.m)
5) (2+1) KP Equation with extra y-derivative (d_kp2ddy.m)
6) (2+1) KP-B Equation (d_kpb2d.m)
7) (2+1) Alternative KP Equation (d_kp2dal.m)
8) (3+1) KP Equation (d_kp3d.m)
9) (3+1) Navier's Equation (d_navi3d.m)
10) (2+1) Gardner Equation (Konopelchenko & Dubrovsky)
      (d_kd2d.m)
nn) Next Page
qq) Exit the Program
-----

```

Figure 9.2: Page 1 of the menu listing the first ten multi-dimensional PDEs.

show the ZK equation as

$$u_{1,t} + \alpha(u_1)(u_{1,x}) + \beta(u_{1,xxx} + u_{1,xyy}) = 0.$$

Since the program is designed for systems of PDEs, all dependent variables carry a subscript. Thus, u in (3.7) is shown in the program as u_1 . The *Mathematica* notation for u_1 is `u[1][x,y,t]`.

After the equation is displayed, the program will provide information as it runs. First, the program computes weights for the variables according to the scaling symmetry of the PDE. The program will display a table of weights, then ask the user to choose a rank for calculating the density. For the ZK equation, the list of weights is shown in Figure 9.3. Directly after the table of weights, a pop-up dialog box will appear with

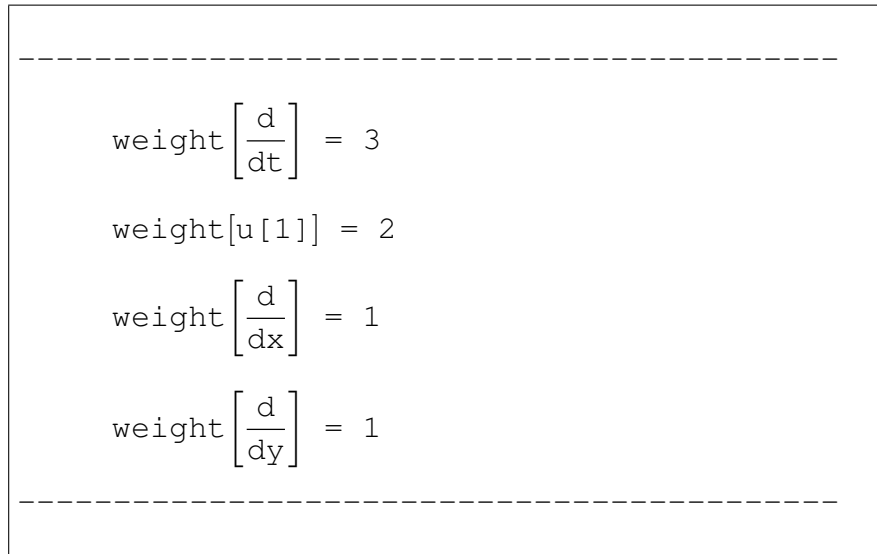


Figure 9.3: A list of weights for the (2+1)-dimensional Zakharov-Kuznetsov equation.

the statement:

If the user wishes to calculate densities without explicit dependence on $\{x, y, t\}$, enter 1. If the user wishes to calculate densities explicitly dependent on $\{x, y, t\}$, enter 2.

Choose 1 to calculate a polynomial density. Next, another dialog box will appear and the program will request the user to:

Enter the RANK to use for calculating the density.
 The rank for the density should be an integer multiple of the lowest weight of the DEPENDENT variable(s). Fractional ranks are allowed.

Since the only dependent variable in the ZK equation is u and the list in Figure 9.3 gives $W(u) = 2$, choose, for example, 4 for the rank. Once the program receives the rank, it runs automatically. If the rank is sufficiently high (for the ZK equation, above 16), the program may take several minutes to perform the computations. At rank 4, the program will give the output shown in Figure 9.4. The output shows the density along with any compatibility conditions on parameters in the PDE, followed by the flux. An explicit verification of each density-flux pair using the continuity equation (3.5) follows the flux.


```

Attempting to find conservation law(s) for RANK 4.

*****
***** RESULTS LISTED BY RANK *****
*****

Conservation Laws for:  Generalized (2+1)-dimensional
Zakharov-Kuznetsov Equation

*****

A normalized density with the rank of 4 is

(u1)2

with no constraints on the parameters.

The corresponding flux is

{
  2/3 alpha (u1)3 - beta (u1,x)2 + beta (u1,y)2 + 2 beta (u1) (u1,xx)
  + 2 beta (u1) (u1,yy), -2 beta (u1,x) (u1,y) }

Result of explicit verification:  ρt + Div J = 0

*****

```

Figure 9.4: Information provided by `ConservationLawsMD.m` when the program terminates. A conservation law for the (2+1)-dimensional Zakharov-Kuznetsov equation is shown.

If more than one density is found, a pop-up box will appear asking the user whether to run an independence test on the densities. In general, the program will only report independent densities. However, when a transformation is required to write a PDE in evolution form, the program may report trivial densities. Furthermore, when a weighted parameter is needed to give the PDE a scaling symmetry, the program may report equivalent densities if calculations are made over several ranks at one time. Running the independence test gives a concise synopsis of independent densities.

When the program finishes, the user has access to the conservation laws in *Mathematica* notation by entering the command `conservationlaws`. If the independence test is run at the end, the command `independentconservationlaws` will produce a list of conservation laws that are fully independent.

If the user wishes to see details of intermediate calculations while the program runs, the command **ConservationLawsMD[Verbose -> All]** should be used to start the program. When running the program at high ranks, `Verbose -> All` will print very large expressions and systems of equations to the screen. The setting `Verbose -> Minimal` provides less information, but avoids printing large expressions to the screen. `ConservationLawsMD.m` requires that the PDE be an evolution equation (3.1) in one of the independent variables before it can run. In a system of PDEs, every equation must be an evolution equation with respect to one independent variable in the system. If the PDE given to `ConservationLawsMD.m` is not a evolution equation or system, the program will automatically apply several transformations, attempting to find an evolution form. If the program is successful, it will continue to run. However, if `Verbose -> Minimal` or `Verbose -> All` is being used, all intermediate data will be in terms of the transformed PDE, which could include a change of variables. Hence, the intermediate results may not appear to fit the PDE as it was originally given. All transformations are removed before any conservation laws are reported. Final results are given in terms of the PDE as it was originally given by the user.

9.1.2 The Data File

`ConservationLawsMD.m` draws the necessary information about a PDE from a data file. The data filename is given in the menu with the PDE name. From Figure 9.2, the data filename for the (2+1)-dimensional ZK equation is `d_zk2d.m`. Figure 9.5 shows the contents of this data file. To test a PDE not given among the menu items, a user must to construct a data file containing the items shown in Figure 9.5. A description of each item in the data file now follows.

The first item the date file is the PDE itself. To enter the PDE, or system of PDEs, rewrite each equation with all terms on one side and enter it as an expression with the label `eq[1]`. If there is a system of PDEs, label successive equations with `eq[1]`, `eq[2]`, etc. For example, the ZK equation is entered as

$$\begin{aligned} \text{eq}[1] = & D[u[1][x,y,t],t] + \\ & \alpha*u[1][x,y,t]^n*D[u[1][x,y,t],x] + \\ & \text{beta}*(D[u[1][x,y,t],x,3] + D[D[u[1][x,y,t],y,2],x]); \end{aligned}$$

```

(* d_zk2d.m *)
(* Menu item 2-1 *)

(* Last Updated: 6 August, 2008, 16:12 by DP at CSM *)

(** GENERALIZED (2+1)-DIMENSIONAL ZAKHAROV-KUZNETSOV EQUATION **)

n = 1;

eq[1] = D[u[1][x,y,t],t] +
        alpha*u[1][x,y,t]^n*D[u[1][x,y,t],x] +
        beta*( D[u[1][x,y,t],x,3] + D[u[1][x,y,t],x,{y,2}] );

diffFunctionListINPUT = {eq[1]};
numDependentVariablesINPUT = 1;
independentVariableListINPUT = {x,y};
nameINPUT = "Generalized (2+1)-dimensional Zakharov-Kuznetsov
Equation";
noteINPUT = "The exponent in the term "<>
ToString[Superscript[u, "n"], FormatType -> TraditionalForm]<>
ToString[Subscript[u, x], FormatType -> TraditionalForm]<>
" is set to n = "<>ToString[n]<>.";

parametersINPUT = {alpha,beta};
weightedParametersINPUT = {};

userWeightRulesINPUT = {};
rankRhoINPUT = Null;
explicitIndependentVariablesInDensitiesINPUT = Null;
formRhoINPUT = {};

(* Reference: Zakharov, V. E. and Kuznetsov, E. A., 1974, *)
(* Three-dimensional solitons, Sov. Phys. JETP 39 (2), 285-286. *)

(* d_zk2d.m *)
(* end of file *)

```

Figure 9.5: The contents of the data file, d_zk2d.m, for the (2+1)-dimensional Zakharov-Kuznetsov equation.

Dependent variables must have the head u with an index attached. If $u(x, y, t)$ is the only dependent variable in the PDE, it must be entered as $u[1][x, y, t]$. If a system of PDEs has dependent variables $u(x, t)$ and $v(x, t)$, one must use $u[1][x, t]$ and $u[2][x, t]$, respectively. Independent variables must be given in the order $\{space\ variables, t\}$. If the equation is put on multiple lines, operation symbols must be at the end of the line so that *Mathematica* will read the next line.

In the following list of input variables, items marked with * are required for the program to run. The other items add versatility to the program and give the user additional options for investigating PDEs. The data file for the (1+1)-dimensional KdV equation, `d_kdv.m`, also contains a description of each item with examples that the user can use to test the program.

- `diffFunctionListINPUT*` is a list of PDE(s), $\{\text{eq}[1], \text{eq}[2], \dots\}$. All lists in *Mathematica* are set in braces `{}`.
- `numDependentVariablesINPUT*` is the number of dependent variables in the PDE or system of PDEs.
- `independentVariableListINPUT*` is a list of the independent space variables in the PDE. The program accepts only x , y , and z as space variables. The time variable t , does not need to be placed in this list.
- `nameINPUT*` is the name of the PDE or a title the user wishes to give the problem. All text must be placed inside quotes `" "`.
- `noteINPUT` is used for any other information the user wishes to include. `noteINPUT` is optional and can be left out of the data file.
- `parametersINPUT*` is a list of constant parameters in the PDE. If parameters are not included in this list, the program may not produce correct results. If there are no parameters, set `parametersINPUT = {}`.
- `weightedParametersINPUT*` is a list of parameters that must carry weight so that the program can compute a scaling symmetry. Indeed, if the program is unable to compute a scaling symmetry for the PDE, it may ask the user to move one or more parameters from the list `parametersINPUT` to the list `weightedParametersINPUT` or to place a weighted parameter into the PDE. If there are no weighted parameters, set `weightedParametersINPUT = {}`.
- `userWeightRulesINPUT` is an optional list of weight rules to be applied to the scaling symmetry of the PDE. In cases where the PDE is multi-uniform in rank (see

Definition 7.3) the user may wish to adjust the weights provided by the program. The weights must be given as a list of rules, for example,

$$\{\text{weight}[u[1]] \rightarrow 1, \text{weight}[d/dx] \rightarrow 1, \text{weight}[d/dy] \rightarrow 2\}.$$

When no weights are given, set `userWeightRulesINPUT = {}`.

- `rankRhoINPUT` is the rank of the (candidate) density to be calculated. If the user wishes to calculate densities at several ranks all at once, `rankRhoINPUT` will take a list of such values. If `rankRhoINPUT = Null`, the program will prompt the user for a rank during execution. `Null` is the best setting the first time a PDE is being run.
- `explicitIndependentVariablesInDensitiesINPUT` tells the program whether the user wishes to calculate densities containing explicit independent variables. Enter 0 if the program is to calculate densities in polynomial form only. Otherwise enter a positive integer for the highest degree an independent variable can have. If `explicitIndependentVariablesInDensitiesINPUT = Null`, the program will prompt the user for a response.
- `formRhoINPUT` allows the user to provide an expression representing a density. The program will check and, if necessary, update coefficients in the expression. After the check is finished the program will report the part of the expression that is density. If the program finds a density based on the given expression, it will also report the flux. See Section 9.1.3 for more details.

When values are entered for `rankRhoINPUT` and `explicitIndependentVariablesInDensitiesINPUT`, `ConservationLawsMD.m` will run automatically after the file for the PDE has been chosen. The program will run over multiple ranks, however, the user should have some feel about the complexity and possible run times before attempting this. Attempting to compute a conservation law for a complex PDE at a high rank may quickly use up the computer's memory.

9.1.3 Testing a Conservation Law

`ConservationLawsMD.m` has a feature that allows a user to test a possible conservation law against a PDE. This feature will work on conservation laws that the program

cannot normally compute, such as conservation laws with rational, transcendental or arbitrary terms. The program takes an expression given as a density, applies undetermined coefficients to each term, then calculates what the undetermined coefficients should be. If a term does not belong, its coefficient will be zero and it will be eliminated. If coefficients provided by the user are incorrect, the program will alter them and report the correct coefficients. If the expression provided is a linear combination of densities, the program will separate them. Once the program is finished analyzing the density, it will compute the flux and report a full conservation law.

The manner in which the densities are given to `ConservationLawsMD.m` is different depending on whether the PDE is a evolution equation (system) or not. Examples include testing densities for the evolution Riemann system, and for the KP equation (8.1), which is not an evolution equation.

9.1.3.1 Testing Densities of Evolution PDEs

If a user wishes to test a conservation law for a PDE that is an evolution equation (system), all that is needed is the density. If a user has a conjecture about a density for a PDE, `ConservationLawsMD.m` can verify or reject the conjecture. For example, `ConservationLawsMD.m` can compute several polynomial conservation laws for the (1+1)-dimensional Chaplygin gas dynamics equations in Riemann coordinates (the Riemann system) [60],

$$\begin{aligned} u_t &= v u_x, \\ v_t &= u v_x, \end{aligned} \tag{9.1}$$

but it cannot generate rational terms or terms with transcendental functions.

An investigation into the Riemann system suggested that the rational expression,

$$\rho = \frac{u + v}{u - v}, \tag{9.2}$$

is a density for (9.1). To test (9.2), open data file `d_riemab.m`, and set

$$\text{formRhoINPUT} = (u[1][x,t] + u[2][x,t]) / (u[1][x,t] - u[2][x,t]);,$$

where $u = u[1][x,y,t]$ and $v = u[2][x,y,t]$. Then start `ConservationLawsMD[]` in a *Mathematica* notebook. The program will ask the user to specify a PDE, so reply 1

for the first menu and 31 for the second menu to get the Riemann System (9.1). Once the program knows which data file to load, the program will automatically take the form of the density and start the verification process. The program will first check all coefficients not given as parameters and, if necessary, ask the user to allow the program to assign undetermined coefficients. If the user says no, the program will stop. Since undetermined coefficients were not given in `formRhoINPUT`, answer yes. From this point, the program will run automatically. The output for this case is shown in Figure 9.6.

The density (9.2) given to the program is actually a linear combination of two independent densities. Two new conservation laws,

$$D_t \left(\frac{u}{u-v} \right) - D_x \left(\frac{uv}{u-v} \right) = 0, \quad (9.3)$$

$$D_t \left(\frac{v}{u-v} \right) - D_x \left(\frac{uv}{u-v} \right) = 0, \quad (9.4)$$

have been computed and verified by `ConservationLawsMD.m` for the Riemann system.

The test density given to `formRhoINPUT` can have a number of forms. The user can assign undetermined coefficients,

$$\begin{aligned} \text{formRhoINPUT} = & c[1]*u[1][x,t]/(u[1][x,t] - u[2][x,t]) \\ & + c[2]*u[2][x,t]/(u[1][x,t] - u[2][x,t]);, \end{aligned}$$

so that the program will run automatically. The user can also give a list of terms instead of an expression,

$$\begin{aligned} \text{formRhoINPUT} = & \{u[1][x,t]/(u[1][x,t] - u[2][x,t]), \\ & u[2][x,t]/(u[1][x,t] - u[2][x,t])\};. \end{aligned}$$

The program will automatically create an expression, using only terms that belong to the density.

9.1.3.2 Testing Densities of Non-evolution PDEs

To use `ConservationLawsMD.m` to compute or test conservation laws for non-evolution PDEs, the PDE must be transformed into an evolution equation or system. While `ConservationLawsMD.m` can automatically transform many PDEs, it cannot transform a density to be tested. When testing a conservation law for a non-evolution

```

*****
The form of the density given by the user is
u1 + u2
u1 - u2
From the form given by the user, the program can form the
normalized density
u1
u1 - u2
The corresponding flux is
(u1)(u2)
u1 - u2
Result of explicit verification: ρt + DxJ = 0
*****
The form of the density given by the user is
u1 + u2
u1 - u2
From the form given by the user, the program can form the
normalized density
u2
u1 - u2
The corresponding flux is
(u1)(u2)
u1 - u2
Result of explicit verification: ρt + DxJ = 0
*****

```

Figure 9.6: The result for testing the density of (9.3) for the Riemann System (9.1).

PDE, the conservation law must undergo the same transformations used to get an evolution equation before being provided to the program. If the transformation requires a variable interchange, both the density and the flux must be transformed. The test will be shown using conservation law (8.13) for the KP equation (8.1).

`ConservationLawsMD.m` can generate densities tu , t^2u , and t^3u , etc. for the KP equation (8.1), but it cannot generate fu for arbitrary $f = f(t)$ as shown in (8.13).

However, the program can test and verify that fu is indeed a density for the KP equation.

Since the KP equation is not an evolution equation, it is necessary to transform (8.1) by interchanging t with y , then setting new dependent variable v equal to u_t to get the evolution equations (8.2) before running `ConservationLawsMD.m`. The evolution equations must be given to the program in the data file, instead of the original equation. The same transformation must be applied to the conservation law being tested.

Example 9.1. Consider the conservation law (8.13) for the KP equation,

$$\begin{aligned} & D_t(fu) + D_x(f(\frac{1}{2}\alpha u^2 + u_{2x}) + (\sigma^2 f' y^2 - fx)(u_t + \alpha u u_x + u_{3x})) \\ & - D_y(2f' y u - (f' y^2 - \sigma^2 fx)u_y) = 0, \end{aligned} \quad (9.5)$$

where $f = f(t)$ is an arbitrary function. To test this law, it must undergo the same transformations used to get the KP evolution equations (8.2). First interchange t with y to get

$$\begin{aligned} & D_y(fu) + D_x(f(\frac{1}{2}\alpha u^2 + u_{2x}) + (\sigma^2 f' t^2 - fx)(u_y + \alpha u u_x + u_{3x})) \\ & - D_t(2f' t u - (f' t^2 - \sigma^2 fx)u_t) = 0, \end{aligned}$$

where now $f = f(y)$. Note that with the interchange of t and y , the density and the y -component of the flux have also been interchanged. Thus, the density to be tested is different from the actually density. Next, replace u_t with v . The conservation law corresponding to the evolution equations is

$$\begin{aligned} & D_t(-2f' t u + (f' t^2 - \sigma^2 fx)v) + D_x(f(\frac{1}{2}\alpha u^2 + u_{2x}) \\ & + (\sigma^2 f' t^2 - fx)(u_y + \alpha u u_x + u_{3x})) + D_y(fu) = 0. \end{aligned} \quad (9.6)$$

The density to be tested is $\rho = -2f' t u + (f' t^2 - \sigma^2 fx)v$.

Once the transformation of the conservation law is complete, open the data file, `d_kp2d.m`, and set

```
formRhoINPUT = -2*f[y]*t*u[1][x,y,t] + (D[f[y],y]* t^2 -
sigma^f[y]*x)*u[2][x,y,t];
```

where $u = u[1][x,y,t]$ and $v = u[2][x,y,t]$. Start `ConservationLawsMD` in a *Mathematica* notebook. When the program asks the user to specify a PDE, reply 2

for the first menu and 3 for the second menu to get the KP equation (8.1). Now the program can start the verification process. The program will first ask the user if it should automatically multiply each term by an undetermined coefficient. Again, no coefficients have been given, so state yes. The results from the program are shown in Figure 9.7. The conservation law reported in Figure 9.7 is precisely (9.6). Therefore, `ConservationLawsMD.m` has verified (9.6) is indeed a conservation law. Conservation law (8.13) is obtained by inverting the transformations used to get (9.6).

```

*****
Conservation Laws for:  the Kadomtsev-Petvishvili Equation
*****

The form of the density given by the user is
t (u1) f'[y] + (u2) (-sigma^2 x f[y] + t^2 f'[y])

From the form given by the user, the program can form the
normalized density
-2 sigma^2 x f[y] (u2) - 2 t (u1) f'[y] + t^2 (u2) f'[y]

  with no constraints on the parameters.

The corresponding flux is
{ alpha f[y] (u1)^2 - 2 alpha x f[y] (u1) (u1, x) - 2 x f[y] (u1, y)
  + 2 f[y] (u1, xx) - 2 x f[y] (u1, xxx) + (alpha t^2 (u1) (u1, x) f'[y]) /
  sigma^2
  + (t^2 (u1, y) f'[y]) / sigma^2 + (t^2 (u1, xxx) f'[y]) / sigma^2, 2 f[y] (u1) }

Result of explicit verification rho_t + Div J: 0
*****

```

Figure 9.7: The result from `ConservationLawsMD.m` verifying the density for the Kadomtsev-Petvishvili evolution equations (8.2).

9.2 The Software Package `HomotopyIntegrator.m`

The code for `HomotopyIntegrator.m` is based on the homotopy operator described in Chapter 6. The algorithm was designed to work together with *Mathematica*'s `Integrate` function. The algorithm works only on differential functions $f(\mathbf{x}, u^{(M)}(\mathbf{x}))$ in one, two, or three dimensions. When $\mathbf{x} = x$, the algorithm will integrate an exact differential function with respect to x . When $\mathbf{x} = (x, y)$ or $\mathbf{x} = (x, y, z)$, it will invert a total divergence.

To run the homotopy operator, open a *Mathematica* notebook, set the directory to the file where `HomotopyIntegrator.m` resides with

```
SetDirectory["c:\ \ set your directory path here  "]
```

then load the program with

```
Get["HomotopyIntegrator.m"].
```

To start the program, enter the command

```
HomotopyIntegrator[differential function, list of independent variables,  
list of parameters ].
```

The function `HomotopyIntegrator` accepts three arguments.

- The first argument is an integrable differential function. There are no restrictions on the types of variables that can be used in the differential function, as long as they conform to variables that *Mathematica* will recognize.
- The second argument is a list of independent variables. The list of independent variables contains only variables for which the operation is to be carried out; any independent variables acting as parameters must be excluded from this list. The independent variable list cannot contain more than three items.
- The third argument, a list of parameters, is optional. If the algorithm determines that the differential function is not integrable, it will look for special cases on parameters placed in the list of parameters where the function might be integrable. If no such list is given, the algorithm will stop.

Several examples show how `HomotopyIntegrator.m` works. The first example shows integration with respect to a single variable.

Example 9.2. Let $F = uv$. Then $f = D_x F = u_x v + uv_x$. To find $D_x^{-1} f$, execute the command

```
f[x_] := D[u[x],x]*v[x] + u[x]*D[v[x],x];
HomotopyIntegrator[f[x], x]
```

Output from HomotopyIntegrator is

```
Check: Given Expression - D[Homotopy Result] = 0
```

```
u[x] v[x].
```

In the second example, the homotopy operator will evaluate conditions on a parameter so that a non-integrable expression becomes integrable.

Example 9.3. Let $g = u_x v - uv_x + auv_x$. An attempt to find $D_x^{-1} g$ using the homotopy operator without specifying that the equation contains the parameter a ,

```
g[x_] := D[u[x],x]*v[x] - u[x]*D[v[x],x]
          + a*u[x]*D[v[x],x];
HomotopyIntegrator[g[x], x]
```

gives

```
The zeroth-Euler operator applied to the expression
has not produced zero for all dependent variables. The
expression is not integrable. The integration process
will not be activated.
```

However, if a is declared in the list of parameters, the algorithm will try to find values of a that make the expression integrable.

```
HomotopyIntegrator[g[x], x {a}]
```

will produce

In the result that follows, parameters have been reassigned under the rules given by: {a->2}

Check: Given Expression - D[Homotopy Result] = 0

u[x] v[x].

By assigning parameters to each term in an expression, the homotopy operator can separate the expression into an integrable part and a non-integrable part, then integrate or invert the divergence on the integrable part.

Example 9.4. *Let*

$$f(x) = (u_x^2 v - v_{2x}) \sin u - (2u_x v_x + u_{2x} v) \cos u + u_x^2 + uu_{2x} + uv_{2x} - u_x v_{2x}.$$

The homotopy integrator shows that the expression is not exact. That is,

```
f[x_] := (D[u[x], x]^2*v[x] - D[v[x], {x, 2}])*Sin[u[x]]
- (2*D[u[x], x]*D[v[x], x]
+ D[u[x], {x, 2}]*v[x])*Cos[u[x]]
+ D[u[x], x]^2 + u[x]*D[u[x], {x, 2}]
+ u[x]*D[v[x], {x, 2}] - D[u[x], x]*D[v[x], {x, 2}];
HomotopyIntegrator[f[x], x]
```

produces

The zeroth-Euler operator applied to the expression has not produced zero for all dependent variables. The expression is not integrable. The integration process will not be activated.

If an undetermined parameter is attached to each term of f, so that

$$f(x) = (b_1 u_x^2 v - b_2 v_{2x}) \sin u - (2b_3 u_x v_x + b_4 u_{2x} v) \cos u + b_5 u_x^2 + b_6 uu_{2x} + b_7 uv_{2x} - b_8 u_x v_{2x},$$

the homotopy operator can determine if any part of this expression is integrable. Executing the commands

```

f[x_] := (b[1]*D[u[x],x]^2*v[x]
  -b[2]*D[v[x],{x,2}])*Sin[u[x]]
  - (2*b[3]*D[u[x],x]*D[v[x],x]
  +b[4]*D[u[x],{x,2}]*v[x])*Cos[u[x]]
  +b[5]*D[u[x],x]^2+b[6]*u[x]*D[u[x],{x,2}]
  +b[7]*u[x]*D[v[x],{x,2}]
  -b[8]*D[u[x],x]*D[v[x],{x,2}];
HomotopyIntegrator[f[x], x, Table[b[i], {i, 1, 8}]]

```

produces the response

The zeroth-Euler operator applied to the expression has not produced zero for all dependent variables. The expression may become integrable if the parameters are set to the value(s) given in the following list:

```

{b[1] → b[4], b[2] → 2b[3] - b[4], b[5] → b[6], b[7] → 0,
 b[8] → 0 }

```

The parameters {b[3], b[4], b[6]} should not be changed.

The program now asks the user to rename each of the parameters given to the program in the parameter list. If the parameters are renamed as directed, the homotopy operator can integrate the expression. The result given is

```

In the result that follows, parameters have been
reassigned under the rules given by: {b[1] → b[4],
b[2] → 2b[3] - b[4], b[3] → b[3], b[4] → b[4], b[5] → b[6],
b[6] → b[6], b[7] → 0, b[8] → 0}

```

Check: Given Expression - D[Homotopy Result] = 0

```

(b[6]u[x] - b[4]v[x]Cos[u[x]])u'[x]
+ (-2b[3] + b[4])v'[x]Sin[u[x]].

```

Note that $b_7 = b_8 = 0$. The terms attached to these parameters are not integrable. Thus $f(x)$ can be partially integrated,

$$\int f(x) dx = (u - v \cos u)u_x - v_x \sin u + \int (uv_{2x} - u_x v_{2x}) dx.$$

Separating integrable and non-integrable parts is discussed in detail in [22].

The next example shows the inversion of a divergence on a function with three independent variables. The algorithm first inverts the divergence, then removes divergence free terms from the homotopy result.

Example 9.5. Let $f = 2uu_x + 3v_x^2v_{xy} + 4w_{xy}^3w_{xyz}$. Applying the homotopy operator,

```
f[x_, y_, z_] := 2*u[x, y, z]*D[u[x, y, z], x]
+ 3*D[v[x, y, z], x]^2*D[v[x, y, z], x, y]
+ 4*D[w[x, y, z], x, y]^3*D[x[x, y, z], x, y, z];
HomotopyIntegrator[f[x, y, z], {x, y, z}]
```

gives the result,

Check: Given Expression - Div[Homotopy Result] = 0

{ u[x, y, z]², v^(1, 0, 0)[x, y, z]³, w^(1, 1, 0)[x, y, z]⁴}

Translating the Mathematica notation, $F = \text{Div}^{-1}f = (u^2, v_x^3, w_{xy}^4)$.

The last example shows a case where the homotopy operator is capable of integrating a function that *Mathematica*'s Integrate cannot.

Example 9.6. Let $f = \frac{u_{2x}u_{6x} + u_x u_{7x}}{\sqrt{u_x u_{6x}}}$ which is entered into Mathematica as

```
f[x_] := (D[u[x], x, 2]*D[u[x], {x, 6}]
+ D[u[x], x]*D[u[x], {x, 7}]) /
(2*Sqrt[D[u[x], x]*D[u[x], {x, 6}]]) ;
```

When Mathematica's Integrate is used, the following result occurs,

```
Integrate[f[x], x]
```

$$\frac{1}{2} \int \frac{u''[x] u^{(6)}[x] + u'[x] u^{(7)}[x]}{\sqrt{u'[x] u^{(6)}[x]}} dx.$$

When the homotopy operator is applied, the correct solution is returned,

```
HomotopyIntegrator[f[x], x]
```

$$\sqrt{u'[x] u^{(6)}[x]}.$$

9.3 The Software Package `IndependenceTest.m`

`IndependenceTest.m` is an algorithm that takes a list of conserved densities and verifies that every density in the list is independent. The algorithm sorts the densities by complexity, checks each density for divergences or divergence-equivalent terms as described in Section 5.1, then takes the densities one at a time, checking for equivalent densities and linear combinations of densities as described in Sections 5.3 and 5.4. As each density is evaluated, the algorithm prints a report to the screen either stating what the dependence is or, alternatively, that the density is independent.

To run `IndependenceTest.m`, a data file containing the list of densities to be evaluated must be constructed first. A sample data file for densities on the potential KP equation (8.15) is shown in Figure 9.8. The first item in the file is the PDE or system of PDEs being evaluated. The PDE is entered using the same rules as in `ConservationLawsMD.m`, however, the PDE does not need to be in evolution form. Each equation is labeled with `eq[1]`, `eq[2]`, etc. Dependent variables must have the head `u` and an index. If a solution to a PDE is $u(x, y, t)$, the dependent variable must be entered as `u[1][x,y,t]`. If a system of PDEs has the solutions $u(x, t)$ and $v(x, t)$, they must be entered as `u[1][x,t]` and `u[2][x,t]`, respectively. Independent space variables are x , y , and z , and the independent time variable is t , and independent variables must be given in the order $\{space\ variables, t\}$. When the PDE information has been entered, build a list of densities. Each density must be labeled `density[1]`, `density[2]`, etc, and the densities must be entered in the same format as the PDEs. Compatibility conditions must accompany each density and must be labeled `compatibility[1]`, `compatibility[2]`, etc. If a density does not have compatibility conditions, set `compatibility[i] = {}`.

After the densities have been entered into the file, the program needs the following information.

- `totalDensities` is the number of densities in the file.
- `numberOfEquations` is the number of equations representing the given PDE.
- The following three statements must be included as they are shown.

```
pdeINPUT = Table[eq[i], {i, 1, numberOfEquations}];
```



```

(* tstkpp1.m *)
(* Last Updated: 21 February, 2008, 11:34 by DP at CSM *)

(* The potential KP equation *)
eq[1] = D[u[1][x,y,t],t,x] + sigma^2*D[u[1][x,y,t],{y,2}] +
      D[u[1][x,y,t],{x,4}] + alpha*D[u[1][x,y,t],x]*D[u[1][x,y,t],{x,2}];

(* density[1] is independent and in simplest form. *)
density[1] = D[u[1][x,y,t],x]^2;
compatibility[1] = {};

(* density[2] is independent and in simplest form. *)
density[2] = D[u[1][x,y,t],x]*D[u[1][x,y,t],y];
compatibility[2] = {};

(* density[3] is independent and in simplest form. *)
density[3] = -3*sigma^2*D[u[1][x,y,t],y]^2 +
      2*alpha*u[1][x,y,t]*D[u[1][x,y,t],x]*D[u[1][x,y,t],{x,2}] +
      3*u[1][x,y,t]*D[u[1][x,y,t],{x,4}] ;
compatibility[3] = {};

(* density[4] contains a linear combinatiton of density[1], *)
(* density[2], and a density equivalent to density[3]. *)
density[4] = 2*density[1] - 3*density[2] -
      (-alpha*D[u[1][x,y,t],x]^3 + 3*D[u[1][x,y,t],{x,2}]^2 -
      3*sigma^2*D[u[1][x,y,t],y]^2);
compatibility[4] = {};

totalDensities = 4;
numberOfEquations = 1;
pdeINPUT = Table[eq[i], {i, 1, numberOfEquations}];
densityListINPUT = Table[density[i], {i, 1, totalDensities}];
compatibilityINPUT = Table[compatibility[i], {i, 1, totalDensities}];

independentVariableListINPUT = {x,y};
numberDependentVariablesINPUT = 1;
listOfParametersINPUT = {alpha, sigma};
nameINPUT = "Potential Kadomtsev-Petviashvili (KP) Equation";

(* tstkpp1.m *)
(* end of file *)

```

Figure 9.8: A sample data file of densities for the potential Kadomtsev-Petviashvili equation. Note that three densities in this file are independent and one is not.

```
densityListINPUT = Table[density[i], {i, 1, totalDensities}];  
compatibilityINPUT = Table[compatibility[i], {i, 1, totalDensities}];
```

- `independentVariablesList` is a list of independent space variables.
- `numberDependentVariables` is the number of dependent variables.
- `nameInput` is the name of the PDE or system of PDEs. Any information about the PDE can be given here, however, all information must be placed in quotes, "".

Once the data file is ready, open a *Mathematica* notebook, set the directory to the file where `IndependenceTest.m` has been located using

```
SetDirectory["c:\\ set your directory path here "],
```

then load the program with

```
Get ["IndependenceTest.m"].
```

To start the program, give the command

```
AnalyzeDensitiesForIndependence[].
```

The algorithm will produce a dialog box asking for the name of the data file. Once it has the name of the data file, it will run automatically. If the program is started with the command

```
AnalyzeDensitiesForIndependence[Verbose -> True],
```

it will provide detailed calculations and suggest methods for moving terms from the density to the flux when needed. The output for `density[4]` in the data file of Figure 9.8, which is clearly dependent on the first three densities, is shown in Figure 9.9. The algorithm will make a report for each density in the file.

DENSITY NOW BEING EVALUATED:

Density no. 4 from the original list of densities:

$$2 (u_{1, x})^2 + \alpha (u_{1, x})^3 - 3 (u_{1, x}) (u_{1, y}) + 3 \sigma^2 (u_{1, y})^2 - 3 (u_{1, xx})^2$$

A check for direct linear dependence on other densities yields the following information.

* Density no. 4 is partially dependent. The density contains a linear combination of:

The density $(u_{1, x})^2$ multiplied by 2
 The density $(u_{1, x}) (u_{1, y})$ multiplied by -3

* All linearly dependent terms are removed. Density no. 4 now consists of

$$\alpha (u_{1, x})^3 + 3 \sigma^2 (u_{1, y})^2 - 3 (u_{1, xx})^2$$

A check for equivalence to other densities yields the following information.

* Density no. 4 is equivalent to the established independent density

$$-\frac{3 \sigma^2 (u_{1, y})^2}{2 \alpha} + (u_1) (u_{1, x}) (u_{1, xx}) + \frac{3 (u_1) (u_{1, xxxx})}{-2 \alpha}$$

multiplied by -2α

The difference between Density no. 4 and the equivalent established independent density multiplied by the factor given is

$$\begin{aligned} & \alpha (u_{1, x})^3 + 2 \alpha (u_1) (u_{1, x}) (u_{1, xx}) - 3 (u_{1, xx})^2 \\ & + 3 (u_{1, x}) (u_{1, xxxx}) \\ & = \text{Div} \{ [\alpha (u_1) (u_{1, x})^2 - 3 (u_{1, x}) (u_{1, xx}) + 3 (u_1) (u_{1, xxxx}), 0] \}. \end{aligned}$$

Since the difference is a divergence, the two densities are equivalent.

The evaluation of Density no. 4 is complete. It has no independent part.

Figure 9.9: The evaluation of a density for the potential Kadomtsev-Petviashvili equation based on previously established independent densities.

CHAPTER 10

CONCLUSIONS

The purpose of this dissertation was to show that conservation laws for multi-dimensional PDEs can be computed using a direct method that relies on tools from calculus, linear algebra, and the calculus of variations. Written in *Mathematica* syntax, the software package `ConservationLawsMD.m` uses this direct method to compute conservation laws in an automated fashion. Conservation laws have been computed for a variety of multi-dimensional PDEs, demonstrating the versatility of the software package. The software package computes both a density and a flux, giving a complete conservation law.

To compute a conservation law, a candidate density is constructed by choosing a rank and constructing a list of terms invariant under the scaling symmetry of the PDE. All divergences and divergence-equivalent terms are removed from the list so that trivial and equivalent densities are not computed. A linear combination of the terms left in the list forms the candidate density. To evaluate the undetermined coefficients in the candidate density, the zeroth-Euler operator is applied to the t -derivative of the candidate density after all time-derivatives have been replaced using the PDE. By the continuity equation, the time-derivative of the candidate density must be a divergence, so the result of applying the zeroth-Euler operator must be zero. Thus, it is a matter of solving a linear system to compute the coefficients for the density. Once the coefficients are known, the homotopy operator is applied to the time-derivative of the density to compute the flux.

The program `ConservationLawsMD.m` is fast and easy to use. The package offers a variety of PDEs, all of which have been added to the menu of the program, including many PDEs that have not been mentioned in this dissertation. The program allows the user to test computations done independently or test reformulations of densities that the program was unable to compute, as was done for the KP equation (8.1). The program will run automatically, or allow the user to interact with it. The drawbacks to using `ConservationLawsMD.m` are twofold: (i) that the PDE must have an evolution

form, perhaps after some transformations, and (ii) the program cannot generate non-polynomial densities, although it can test them. A future version of the program will include methods for computing conservation laws for PDEs with transcendental nonlinearities. Therefore, it will be able to compute conservation laws for a larger class of multi-dimensional non-evolution PDEs.

Using `ConservationLawsMD.m`, conservation laws were computed for the Zakharov-Kuznetsov equation, the non-stationary transonic gas flow equation, the Kadomtsev-Petviashvili equation, the Khoklov-Zabolotskaya equation, and the fluid dynamics equations as test cases. For each of these cases, the conservation laws computed by the program `ConservationLawsMD.m` were compared to conservation laws given in literature. This process not only verified that the program is computing what it is supposed to compute, but also identified a number of errors in refereed publications. For other PDEs, new conservation laws were found. Indeed, conservation laws for the Gardner equation, the Manakov-Santini system, and the Camassa-Holm equation were not found in literature. Conservation laws for Navier's equation without time dependencies are given in [55], whereas the conservation laws reported in this dissertation are for Navier's equation with time dependency. Many other examples of PDEs are provided in the package `ConservationLawsMD.m` for the user to explore.

The homotopy operator is a tool that integrates an exact one-dimensional differential expression, or inverts a total divergence on an exact two- or three-dimensional expression. The homotopy operator adds functionality to *Mathematica's* `Integrate` function, by integrating differential functions that `Integrate` cannot do and by adding a function that inverts total divergences of differential vectors. Since it has many independent applications, the software package `HomotopyIntegrator.m` is a stand alone package for applications of the homotopy operator. The package `HomotopyIntegrator.m` can also determine if a differential expression is exact, and evaluate any parameters in a non-exact differential expression to determine if the expression can be made integrable. In a future version, the homotopy operator will be able to separate a differential function into an exact part and a non-exact part, then invert the divergence on the exact part. A second method for inverting divergences that involves Hodge decomposition [61] remains to be investigated.

While computing conservation laws for multi-dimensional PDEs, it became necessary to verify that the densities being computed were independent. The issue of independence occurred again when analyzing conservation laws and when making comparisons with results reported in literature. For these purposes, the package `IndependenceTest.m` was created. The package `IndependenceTest.m` works independently and can be used on a list of densities calculated using any method.

A large amount of research has been done in the study of the integrability of PDEs, of which the study of conservation laws is an intrinsic part. However, there is a lot to be done in the study of differential-difference equations (DDEs), integro-differential equations (IDEs) and delay difference equations. The methods developed for the computation of conservation laws have applications in DDEs and have yet to be investigated for IDEs and delay DEs. Future research will include the study of integrability of these types of equations with the expectation that symbolic algorithms can be constructed as an aid to analyze nonlinear difference equations. Such studies may lead to techniques, algorithms, and software to compute conservation laws, generalized symmetries, recursion operators, and symplectic and cosymplectic operators for DDEs, IDEs and delay DEs.

REFERENCES CITED

- [1] Ablowitz, M. J. and Clarkson, P. A., 1991, *Solitons, Nonlinear Evolution Equations and Inverse Scattering*, Cambridge University Press, Cambridge, UK.
- [2] Adams, P. J., 2003, *Symbolic computation of conserved densities and fluxes for systems of partial differential equations with transcendental nonlinearities*. Master of Science Thesis, Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO.
- [3] Adams, P. J. and Hereman, W., 2002, TransPDEDensityFlux.m: Symbolic computation of conserved densities and fluxes for systems of partial differential equations with transcendental nonlinearities, <http://inside.mines.edu/~whereman>, under Scientific Software.
- [4] Anco, S. C. and Bluman, G., 2002, Direct construction method for conservation laws of partial differential equations, Part I: Examples of conservation law classifications, *Euro. J. Appl. Math.* **13**: 545-566.
- [5] Anco, S. C. and Bluman, G., 2002, Direct construction method for conservation laws of partial differential equations, Part II: General treatment, *Euro. J. Appl. Math.* **13**: 567-585.
- [6] Anderson, I. M., 2004, *The Vessiot package*, Department of Mathematics, Utah State University, Logan UT, http://www.math.usu.edu/~fg_mp/Pages/SymbolicsPage/VessiotDownloads.html.
- [7] Anderson, I. M., and Cheb-Terrab, E. S., 2009, Differential geometry package, *Maple Online Help*, <http://www.maplesoft.com/support/help/view.aspx?sid=26040>.
- [8] Baldwin, D. and Hereman, W., 2007, PainleveTest.m: A Mathematica package for the Painleve test of systems of nonlinear ordinary and partial differential equations, <http://inside.mines.edu/~whereman>, under Scientific Software.
- [9] Baldwin, D. and Hereman, W., 2009, A symbolic algorithm for computing recursion operators of nonlinear PDEs, *Int. J. Comp. Math.* Submitted for publication.
- [10] Billingham, J. and King, A. C., 2000 *Wave Motion*, Cambridge University Press, Cambridge, U.K.
- [11] Bishop, R. L. and Goldberg, S. I., 1968, *Tensor Analysis on Manifolds*, Dover Publications, New York.
- [12] Bluman, G. W. and Kumei, S., 1989, *Symmetries and Differential Equations*, Springer Verlag, New York.
- [13] Camassa, R. and Holm, D. D., 1993, An integrable shallow water equation with peaked solutions, *Phys. Rev. Lett.* **71**: 1661-1664.
- [14] Cantwell, B. J., 2002, *Introduction to Symmetry Analysis*, Cambridge University Press, Cambridge, U.K.

- [15] Cheviakov, A. F., 2007, GeM software package for computation of symmetries and conservation laws of differential equations, *Comp. Phys. Comm.* **76** (1): 48-61.
- [16] Cheviakov, A. F., 2009, Computation of fluxes of conservation laws, *J. Eng. Math.*, Submitted for publication.
- [17] Cohen, M. M., 1973 *A Course in Simple-Homotopy Theory*, Grad. Texts in Math. **10**, Springer-Verlag, New York.
- [18] Cole, J. D., 1951, On a quasi-linear parabolic equation occurring in aerodynamics, *Quart. Appl. Math.*, **9**: 225-236.
- [19] Colley, S. J., 2002, *Vector Calculus*, Prentice Hall, Upper Saddle River, NJ.
- [20] Deconinck, B., 2005, Personal notes on the one-dimensional homotopy operator
- [21] Deconinck, B. and Nivala, M., 2005, INTSUM, Maple software for the symbolic integration and summation of expressions that may or may not be exact, <http://www.amath.washington.edu/~bernard/software/intsum>.
- [22] Deconinck, B. and Nivala, M., 2008, Symbolic integration using homotopy methods, *Math. Comput. Simul.*, Submitted for publication.
- [23] Dellar, P., 2003 Common Hamiltonian structure of the shallow water equations with horizontal temperature gradients and magnetic fields, *Phys. Fluids* **15** (2): 292-297.
- [24] Fermi, E., Pasta, J. and Ulam, S., 1955, Studies in nonlinear problems, I, in *Nonlinear Wave Motion* (1970), A. C. Newell, Ed., AMS, Providence Rhode Island.
- [25] Flanders, H., 1963 *Differential Forms with Applications to the Physical Sciences*, Academic Press, London.
- [26] Flaschka, H., Newell, A. C., and Tabor, M., 1991, Integrability, in *What is Integrability?*, V. E. Zakharov, Ed., Springer-Verlag, Berlin.
- [27] Gazizov, R. K., 1994, Hydrodynamics and gasdynamics, in *CRC Handbook of Lie Group Analysis of Differential Equations, V. 1*, N. H. Ibragimov, Ed., CRC Press, Boca Raton, FL.
- [28] Göktaş, Ü., 1996, *Symbolic computation of conserved densities for systems of nonlinear evolution equations*, Master of Science Thesis, Colorado School of Mines, Golden CO.
- [29] Göktaş, Ü. and Hereman, W., 1996, CONDENS.M: A Mathematica program for the symbolic computation of conserved densities for systems of nonlinear evolution equations, <http://inside.mines.edu/~wherman>, under Scientific Software.
- [30] Göktaş, Ü. and Hereman, W., 1997, Symbolic computation of conserved densities for systems of nonlinear evolution equations, *J. Symb. Comput.* **24**: 591-621.

- [31] Gordoia, P. G., Pickering, A. and Senthilvelan, M., 2004, Evidence for the nonintegrability of a water wave equation in 2+1 dimensions, *Zeitschrift für Naturforschung* **59a**: 640-644.
- [32] Graham, R., Knuth, R., and Patashnik, O., 1994 *Concrete Mathematics*, 2nd ed., Addison-Wesley, Boston.
- [33] Hereman, W., 2006, Symbolic computation of conservation laws of nonlinear partial differential equations in multi-dimensions, *J. Quant. Chem.* **106**: 278-299.
- [34] Hereman, W., Adams, P. J., Eklund, H. L., Hickman, M. S. and Herbst, B. M., 2009, Direct methods and symbolic software for conservation laws of nonlinear equations, in: *Advances in Nonlinear Waves and Symbolic Computation*, Z. Yan, Ed., Nova Science Publishers, New York: 19-79.
- [35] Hereman, W., Colagrosso, M., Sayers, R., Ringler, A., Deconinck, B., Nivala, M., and Hickman, M. S., 2005, Continuous and discrete homotopy operators and the computation of conservation laws, in *Differential Equations with Symbolic Computation*, D. Wang and Z. Zheng, Eds., Birkhäuser Verlag, Basel: 249-285.
- [36] Hereman, W., Deconinck, B., and Poole, L. D., 2006, Continuous and discrete homotopy operators: a theoretical approach made concrete, *Math. Comput. Simul.* **74**: 352-360.
- [37] Hopf, E., 1950 The partial differential equation $u_t + uu_x = mu_{xx}$, *Comm. Pure Appl. Math.* **3**: 201-230.
- [38] Hydon, P. E., 2000, *Symmetry Methods for Differential Equations*, Cambridge University Press, Cambridge, U.K.
- [39] Ibragimov, N. H., Ed., 1994, *CRC Handbook of Lie Group Analysis of Differential Equations, V. 1,2,3*, CRC Press, Boca Raton, FL.
- [40] Infeld, E., 1985 Self-focusing nonlinear waves, *J. Plasma Phys.* **33**: 171-182.
- [41] Infeld, E. and Frycz, P., 1983, Infinite sequences of conservation laws for the Kadomtsev-Petviashvili equation, *Acta Phys. Polon.* **B14** (2): 129-132.
- [42] Johnson, R. S., 2002, Camassa-Holm, Korteweg-de Vries and related models for water waves, *J. Fluid Mech.* **455**: 63-82.
- [43] Kadomtsev, B. B. and Petviashvili, V. I., 1970, On the stability of solitary waves in weakly dispersive media, *Sov. Phys. Dokl.* **15**: 539-541.
- [44] Khamitova, R. S., 1982, Group structure and the basis of conservation laws, *Theor. Math. Phys.* **52** (2): 777-781.
- [45] Konopelchenko, B. G. and Dubrovsky, V. G., 1984, Some new integrable nonlinear evolution equations in 2+1 dimensions, *Phys. Lett.* **102A** (1): 15-17.

- [46] Korteweg, D. J. and de Vries, G., 1895, On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves *Phil. Mag.* **39**: 422-443.
- [47] Kruskal, M. D., Miura, R. M., Gardner, C. S., and Zabusky, N. J., 1970, Korteweg-de Vries equation and generalizations V. Uniqueness and nonexistence of polynomial conservation laws, *J. Math. Phys.* **11** (3): 952-960.
- [48] Lin, C. and Zhang, X., 2007, The formally variable separation approach for the modified Zakharov-Kuznetsov equation, *Commun. Nonl. Sci. Numer. Simul.* **12**: 636-642.
- [49] Ma, W. X. and Fuchssteiner, B., 1996, The bi-Hamiltonian structure of the perturbation equations of the KdV hierarchy, *Phys. Lett. A* **213**: 49-55.
- [50] Manakov, S. V. and Santini, P. M., 2006, Cauchy problem on the plane for the dispersionless Kadomtsev-Petviashvili equation, *JETP Lett.* **83** (10): 462-466.
- [51] Manakov, S. V. and Santini, P. M., 2007, A hierarchy of integrable PDEs in 2+1 dimensions associated with 2-dimensional vector fields, *Theor. Math. Phys.* **152** (1): 1004-1011.
- [52] Miura, R. M., Gardner, C. S., and Kruskal, M. D., 1968, Korteweg-de Vries equation and generalizations II. Existence of conservation laws and constants of motion, *J. Math. Phys.*, **9** (8): 1204-1209.
- [53] Naz, R., 2008, *Symmetry solutions and conservation laws for some partial differential equations in field mechanics*, Doctor of Philosophy Thesis, University of the Witswatersrand, Johannesburg.
- [54] Noether, E., 1918, Invariant variation problems, Translated by M. A. Tavel, *Transport Theory and Stat. Phys.* **1**: 186-207.
- [55] Olver, P. J., 1984, Conservation laws in elasticity II: linear homogenous isotropic elastostatics, *Arch. Rat. Mech. Anal.* **85**: 131-160.
- [56] Olver, P. J., 1993, *Applications of Lie Groups to Differential Equations*, 2nd ed., Grad. Texts in Math., **107**, Springer, New York.
- [57] Polyanin, A. D. and Zaitsev, V. F., 2004, *Handbook of Nonlinear Partial Differential Equations*, Chapman and Hall/CRC, Boca Raton, FL.
- [58] Ripa, P., 1991, General stability conditions for zonal flows in a one-layer model on the beta-plane or the sphere, *J. Fluid Mech.* **222**: 119-137.
- [59] Rosenhaus, V., 2006, On conserved densities and asymptotic behaviour for the potential Kadomtsev-Petviashvili equation, *J. Phys. A: Math. Gen.* **39**: 7693-7703.
- [60] Rozdestvenskii, B. and Janenko, N. N., 1983, *Systems of Quasilinear Equations and their Applications to Gas Dynamics*, Translations Mathematical Monographs, **55**, AMS, Providence, RI.

- [61] Sanders, J. and Wang, J. P., 1997, Hodge decomposition and conservation laws, *Math. Comput. Simul.* **44**: 483-493.
- [62] Sanders, J. and Wang, J. P., 1997, Hodge decomposition and conservation laws, <http://www.math.vu.nl/~jansa/#research>.
- [63] Sanz-Serna, J. M., 1982, An explicit finite-difference scheme with exact conservation properties, *J. Comput. Phys.* **47**: 199-210.
- [64] Schouten, J. A. and Struik, D. A., 1935, *Einführung in die Neueren Methoden der Differentialgeometrie*, V. 1, P. Noordhoff N. V., Groningen-Batavia.
- [65] Sharomet, N. O., 1989, Symmetries, invariant solutions and conservation laws of the nonlinear acoustics equation, *Acta Appl. Math.* **15**: 83-120.
- [66] Shivamoggi, B. K., 1990, The Painlevé analysis of the Zakharov-Kuznetsov equation, *Phys. Scr.* **42**: 641-642.
- [67] Shivamoggi, B. K., 2009, private communication.
- [68] Shivamoggi, B. K., Rollins, D. K. and Fanjul, R., 1993, Analytic aspects of the Zakharov-Kuznetsov equation, *Phys. Scr.* **47**: 15-17.
- [69] Tikhonov, A. N. and Samarskii, A. A., 1963, *Equations of Mathematical Physics*, Dover Publications, New York.
- [70] Verheest, F. and Hereman, W., 1994, Conservation laws and solitary wave solutions for generalized Schamel equations, *Phys. Scr.* **50**: 611-614.
- [71] Volterra, V., 1913, *Leçons sur les Fonctions de Lignes*, Gauthier-Villars, Paris.
- [72] Wazwaz, A. M., 2005, Exact solutions with solitons and periodic structures for the Zakharov-Kuznetsov (ZK) equation and its modified form, *Commun. Nonl. Sci. Numer. Simul.* **10**: 597-606.
- [73] Weinstock, R., 1974, *Calculus of Variations*, Dover Publications, Mineola NY.
- [74] Whitham, G. B., 1965, Non-linear dispersive waves, *Proc. Roy. Soc. Lond.* **A 283**: 238-261.
- [75] Wolf, T., 2002, A comparison of four approaches to the calculation of conservation laws, *Euro. J. Appl. Math.* **13**: 129-152.
- [76] Wolf, T., 2008, *Crack interactive tutorial*, <http://lie.math.brocku.ca/crack/demo>.
- [77] Wolf, T., 2008, private communication.
- [78] Wolf, T., Brand, A., and Mohammadzadeh, M., 1999, Computer algebra algorithms and routines for the computation of conservation laws and fixing of gauge in differential expressions, *J. Symb. Comp.* **27**: 221-238.
- [79] Wolfram, S., 2003, *The Mathematica Book*, 5th Edition for Mathematica Version 5, Wolfram Media Inc., Champaign, IL.

- [80] Yan, Z., 2005, The (2+1)-dimensional integrable coupling of KdV equation: Auto-Bäcklund transformation and new non-traveling wave profiles, *Phys. Lett. A* **345**: 363-377.
- [81] Zabusky, N. J. and Kruskal, M. D., 1965, Interaction of ‘solitons’ in a collisionless plasma and the recurrence of initial states, *Phys. Rev. Lett.* **15**: 240-243.
- [82] Zakharov, V. E., 1991, *What Is Integrability?*, V. E. Zakharov, Ed., Springer-Verlag, Berlin.
- [83] Zakharov, V. E. and Kuznetsov, E. A., 1974, Three-dimensional solitons, *Sov. Phys. JETP* **39** (2): 285-286.
- [84] Zhi, H., 2009, Symmetry reductions of the Lax pairs of 2+1-dimensional Konopelchenko-Dubrovsky equation, *Appl. Math. Comput.* **210** (2): 530-535.

APPENDIX

ADDITIONAL PROOFS FOR THE HOMOTOPY OPERATOR

This appendix contains several lemmas with proofs to support proofs of theorems in Chapter 6. The first section contains the proof of a lemma used in Theorem 6.4, and the second section has a detailed proof of Theorem 6.5. The third section covers a proof for Theorem 6.6, which was not shown in Chapter 6.

A.1 Lemmas for the Proof of Theorem 6.4

The proof of Theorem 6.4 is completed using the identity (6.31), which is not trivial. Lemma A.3 will show that (6.31) holds. Two preceding lemmas are needed to establish well known identities [32].

Lemma A.1 (Pascal's First Identity). *For any positive integers i and k ,*

$$\binom{k+1}{i+1} = \binom{k}{i+1} + \binom{k}{i}. \quad (\text{A.1})$$

PROOF:

$$\begin{aligned} \binom{k+1}{i+1} &= \frac{(k+1)!}{(k-i)!(i+1)!} \\ &= \frac{(k-i+i+1)k!}{(k-i)!(i+1)!} \\ &= \frac{(k-i)k!}{(k-i)!(i+1)!} + \frac{(i+1)k!}{(k-i)!(i+1)!} \\ &= \frac{k!}{(k-i-1)!(i+1)!} + \frac{k!}{(k-i)!i!} \\ &= \binom{k}{i+1} + \binom{k}{i}. \end{aligned} \quad \blacksquare$$

Lemma A.2. *For any positive integers k and l , with $k \geq l$,*

$$\sum_{i=l}^k (-1)^{k-i} \binom{i}{l} \binom{k}{i} = \delta_{k,l}, \quad (\text{A.2})$$

where $\delta_{k,l} = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases}$ is the Kronecker delta.

PROOF: The binomial expansion for $(x - y)^n$ is $(x - y)^n = \sum_{i=0}^n (-1)^i \binom{n}{i} x^{n-i} y^i$. Let $x = y = 1$. Then

$$0 = (1 - 1)^n = \sum_{i=0}^n (-1)^i \binom{n}{i}, \quad (\text{A.3})$$

for $n > 0$. For the case where $n = 0$, choose $x \neq y$. Then

$$1 = (x - y)^0 = \sum_{i=0}^0 (-1)^i \binom{0}{i}. \quad (\text{A.4})$$

The combination of (A.3) and (A.4) gives the identity

$$\sum_{i=0}^n (-1)^i \binom{n}{i} = \delta_{0,n}. \quad (\text{A.5})$$

Now,

$$\begin{aligned} \sum_{i=l}^k (-1)^{k-i} \binom{i}{l} \binom{k}{i} &= \sum_{i=l}^k (-1)^{k-i} \frac{i!}{(i-l)! l!} \frac{k!}{(k-i)! i!} \\ &= \frac{k!}{l!(k-l)!} \sum_{i=l}^k (-1)^{k-i} \frac{(k-l)!}{(i-l)! (k-i)!} \\ &= \binom{k}{l} \sum_{i=l}^k (-1)^{k-i} \binom{k-l}{i-l}. \end{aligned}$$

Setting $j = i - l$,

$$\begin{aligned} \binom{k}{l} \sum_{i=l}^k (-1)^{k-i} \binom{k-l}{i-l} &= \binom{k}{l} \sum_{j=0}^{k-l} (-1)^{k-j-l} \binom{k-l}{j} \\ &= (-1)^{k-l} \binom{k}{l} \delta_{0,k-l} \quad \text{by (A.5)} \\ &= \delta_{k,l}. \quad \blacksquare \end{aligned}$$

Lemma A.3. For any positive integers m and n , where $n \geq m + 1$,

$$\sum_{i=m}^{n-1} (-1)^{i-m} \binom{i}{m} \binom{n}{i+1} = 1. \quad (\text{A.6})$$

PROOF: The proof is done by induction on n [20]. Let $n = m + 1$, the smallest value for n . Then

$$\sum_{i=m}^m (-1)^{m-1} \binom{i}{m} \binom{m+1}{i+1} = 1. \quad (\text{A.7})$$

Thus, (A.6) holds for the case where $n = m + 1$. Assume that (A.6) holds for any $k > m + 1$, that is,

$$\sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \binom{k}{i+1} = 1. \quad (\text{A.8})$$

Then, by the induction principle, it is necessary to show that the identity (A.6) holds for $k + 1$, that is, to show that

$$\sum_{i=m}^k (-1)^{i-m} \binom{i}{m} \binom{k+1}{i+1} = 1 \quad (\text{A.9})$$

holds. Simplifying the left-hand side of (A.9),

$$\begin{aligned} & \sum_{i=m}^k (-1)^{i-m} \binom{i}{m} \binom{k+1}{i+1} \\ &= \sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \binom{k+1}{i+1} + (-1)^{k-m} \binom{k}{m} \binom{k+1}{k+1} \\ &= \sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \left(\binom{k}{i+1} + \binom{k}{i} \right) + (-1)^{k-m} \binom{k}{m} \quad \text{by Lemma A.1} \\ &= \sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \binom{k}{i+1} + \sum_{i=m}^{k-1} (-1)^{i-m} \binom{i}{m} \binom{k}{i} + (-1)^{k-m} \binom{k}{m} \\ &= 1 + \sum_{i=m}^k (-1)^{i-m} \binom{i}{m} \binom{k}{i} \quad \text{by (A.8)} \\ &= 1 + \delta_{m,k} \quad \text{by lemma A.2} \\ &= 1 \quad \text{since } m < k. \end{aligned}$$

Therefore, (A.6) holds. ■

A.2 Extended Proof of Theorem 6.5

Theorem 6.5 states that if $f(x, y, \mathbf{u}^{(M)}(x, y))$ is exact, then

$$\text{Div}^{-1} f = \left(\mathcal{H}_{\mathbf{u}(x,y)}^{(x)}, \mathcal{H}_{\mathbf{u}(x,y)}^{(y)} \right).$$

The proof shown here gives the details for $f(x, y, \mathbf{u}^{(2)}(x, y))$ following the outline of the proof given in Section 6.3.1 for the one-dimensional case. The steps for splitting the initial expression are clearly shown with the order 2 case. To show the same steps for order M would require numerous pages and would not add to the understanding of the proof technique.

PROOF FOR THEOREM 6.5 FOR THE CASE $f(x, y, \mathbf{u}^{(2)}(x, y))$: To start, working with component u^j , multiply $\mathcal{L}_{u^j(x,y)} f$ by u^j to restore the degree. Following the description given in Section 6.3.2, $u^j \mathcal{L}_{u^j(x,y)} f$ will be split into three parts, one under the \mathbf{M} operator, one under the \mathbf{D}_x operator, and one under the \mathbf{D}_y operator. To begin with, take the

$u \frac{\partial f}{\partial u}$ term out of the sum, then separate the remaining sum into D_x and D_y parts where possible. Next, integrate by parts to split off $\sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j}$ and $\sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j}$. Thus,

$$\begin{aligned}
u^j \mathcal{L}_{u^j(x,y)} f &= u^j \sum_{k_1=0}^2 \sum_{k_2=0}^2 (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&= u^j \frac{\partial f}{\partial u^j} + u^j \sum_{k_1=1}^2 (-D_x)^{k_1} \frac{\partial f}{\partial u_{k_1 x}^j} + u^j \sum_{k_2=1}^2 (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_2 y}^j} \\
&\quad + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&= u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} \\
&\quad - D_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \quad (\text{A.10})
\end{aligned}$$

$$- D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \quad (\text{A.11})$$

$$+ u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 (-D_x)^{k_1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \quad (\text{A.12})$$

Lines (A.10) and (A.11) result from the procedure used in Theorem 6.3. Next, split (A.12) into two pieces, one with $\frac{k_1}{k_1+k_2}$ x -derivatives and one with $\frac{k_2}{k_1+k_2}$ y -derivatives, then integrate these pieces by parts. This gives

$$\begin{aligned}
u^j \mathcal{L}_{u^j(x,y)} f &= u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} \\
&\quad - D_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
&\quad \left. + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
&\quad - D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
&\quad \left. + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
&\quad + u_x^j \sum_{k_1=1}^1 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.13})
\end{aligned}$$

$$+ u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1 + k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.14})$$

$$+ u_y^j \sum_{k_1=1}^2 \sum_{k_2=1}^1 \left(\frac{k_2}{k_1 + k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.15})$$

$$+ u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1 + k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \quad (\text{A.16})$$

The terms in the expression above have been reorganized into a D_x part, a D_y part and left over terms. Note that the “outside” terms (A.13) - (A.16) result from splitting (A.12), followed by integration by parts. The pieces will be grouped in different places after the next integration. In the next step, (A.13) and (A.16) are split into fractions of x - and y -derivatives, then again integrated by parts. For example, in (A.14), there are $\frac{k_1 - 1}{k_1 + k_2 - 1}$ x -derivatives and $\frac{k_2}{k_1 + k_2 - 1}$ y -derivatives, so the term is split accordingly. Therefore,

$$u^j \mathcal{L}_{u^j(x,y)} f = u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 x}^j \frac{\partial f}{\partial u_{k_2 x}^j} \quad (\text{A.17})$$

$$\begin{aligned} & - D_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\ & \left. + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1 + k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\ & - D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\ & \left. + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_2}{k_1 + k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\ & + \sum_{k_2=1}^2 \left(\frac{1}{k_2 + 1} \right) u_{x k_2 y}^j \frac{\partial f}{\partial u_{x k_2 y}^j} \end{aligned} \quad (\text{A.18})$$

$$\begin{aligned} & - D_y \left[\sum_{i_2=0}^1 u_{x i_2 y}^j \sum_{k_1=1}^1 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1 + k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\ & + \sum_{k_1=1}^2 \left(\frac{1}{k_1 + 1} \right) u_{k_1 x y}^j \frac{\partial f}{\partial u_{k_1 x y}^j} \quad (\text{A.19}) \\ & - D_x \left[\sum_{i_1=0}^1 u_{i_1 x y}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=1}^1 \left(\frac{k_2}{k_1 + k_2} \right) (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \end{aligned}$$

$$\begin{aligned}
& -D_x \left[u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& -D_y \left[u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& -D_x \left[u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& -D_y \left[u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + u_{2x}^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.20}) \\
& + u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.21}) \\
& + u_{xy}^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.22}) \\
& + u_{2y}^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \quad (\text{A.23})
\end{aligned}$$

In the next step, the terms (A.18) and (A.19) are moved up with the split off terms in (A.17) since they no longer contain any x - or y -derivatives. All terms under D_x or D_y are brought together. In (A.21) as well as (A.22) a piece is split off of each and the two pieces are combined with (A.17). No integration is done in this step, giving

$$\begin{aligned}
u^j \mathcal{L}_{w^j(x,y)} f &= u^j \frac{\partial f}{\partial w^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} \\
&+ \sum_{k_1=1}^2 \left(\frac{1}{k_1+1} \right) u_{k_1 xy}^j \frac{\partial f}{\partial u_{k_1 xy}^j} + \sum_{k_2=1}^2 \left(\frac{1}{k_2+1} \right) u_{x k_2 y}^j \frac{\partial f}{\partial u_{x k_2 y}^j} \quad (\text{A.24}) \\
&- D_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
&+ \sum_{i_1=0}^1 u_{i_1 xy}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=1}^1 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}
\end{aligned}$$

$$\begin{aligned}
& + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \Big] \\
& - D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_2=0}^1 u_{x i_2 y}^j \sum_{k_1=1}^1 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& \left. + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + u_{2x}^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.25}) \\
& + u_{2y}^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.26}) \\
& + u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=1}^1 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.27}) \\
& + u_{xy}^j \sum_{k_1=1}^1 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \quad (\text{A.28}) \\
& + \binom{1+1}{1} u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \\
& \quad \quad \quad (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \quad (\text{A.29})
\end{aligned}$$

In the next step, integration by parts is applied immediately to (A.25) - (A.28), as only a D_x or a D_y can be removed from each term. The last term (A.29) still contains both D_x and D_y , so it must be split according to the fraction $\frac{k_1-1}{k_1+k_2-2}$ of x -derivatives and the fraction $\frac{k_2-1}{k_1+k_2-2}$ of y -derivatives. The terms in (A.24) are written in expanded form. This yields

$$\begin{aligned}
u^j \mathcal{L}_{u^j(x,y)} f & = u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} + \frac{1}{2} u_{xy}^j \frac{\partial f}{\partial u_{xy}^j} + \frac{1}{3} u_{2xy}^j \frac{\partial f}{\partial u_{2xy}^j} \\
& + \frac{1}{2} u_{xy}^j \frac{\partial f}{\partial u_{xy}^j} + \frac{1}{3} u_{x2y}^j \frac{\partial f}{\partial u_{x2y}^j} \quad (\text{A.30})
\end{aligned}$$

$$\begin{aligned}
& -D_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_1=0}^1 u_{i_1 x y}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=1}^1 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& \left. + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& -D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_2=0}^1 u_{x i_2 y}^j \sum_{k_1=1}^1 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& \left. + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + u_{x2y}^j \left(\frac{2}{3} \right) \left(\frac{1}{2} \right) \frac{\partial f}{\partial u_{x2y}^j} + u_{2x2y}^j \left(\frac{2}{4} \right) \left(\frac{1}{3} \right) \frac{\partial f}{\partial u_{2x2y}^j} \tag{A.31}
\end{aligned}$$

$$\begin{aligned}
& -D_x \left[\sum_{i_1=0}^1 u_{i_1 x 2y}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=2}^2 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + u_{2xy}^j \left(\frac{2}{3} \right) \left(\frac{1}{2} \right) \frac{\partial f}{\partial u_{x2y}^j} + u_{2x2y}^j \left(\frac{2}{4} \right) \left(\frac{1}{3} \right) \frac{\partial f}{\partial u_{2x2y}^j} \tag{A.32}
\end{aligned}$$

$$\begin{aligned}
& -D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=2}^2 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1-2} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + u_{2xy}^j \left(\frac{2}{3} \right) \left(\frac{1}{2} \right) \frac{\partial f}{\partial u_{x2y}^j} + u_{x2y}^j \left(\frac{2}{3} \right) \left(\frac{1}{2} \right) \frac{\partial f}{\partial u_{x2y}^j} \tag{A.33}
\end{aligned}$$

$$\begin{aligned}
& -D_x \left[u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=1}^1 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& -D_y \left[u_{xy}^j \sum_{k_1=1}^1 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right]
\end{aligned}$$

$$\begin{aligned}
& + \binom{1+1}{1} u_{2xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_1-1}{k_1+k_2-2} \right) \\
& \quad (-D_x)^{k_1-2} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \tag{A.34}
\end{aligned}$$

$$\begin{aligned}
& + \binom{1+1}{1} u_{x2y}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_2-1}{k_1+k_2-2} \right) \\
& \quad (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \tag{A.35}
\end{aligned}$$

$$\begin{aligned}
& - D_x \left[\binom{1+1}{1} u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_1-1}{k_1+k_2-2} \right) \right. \\
& \quad \left. (-D_x)^{k_1-2} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& - D_y \left[\binom{1+1}{1} u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_2-1}{k_1+k_2-2} \right) \right. \\
& \quad \left. (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right].
\end{aligned}$$

Next, the terms (A.31) - (A.33) are combined with (A.30) as all D_x and D_y have been removed. Then all D_x and D_y parts are brought together and rearranged. Again, no integration is done in this step. Hence,

$$\begin{aligned}
& u^j \mathcal{L}_{u^j(x,y)} f \\
& = u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} + u_{xy}^j \frac{\partial f}{\partial u_{xy}^j} + \frac{1}{3} u_{2xy}^j \frac{\partial f}{\partial u_{2xy}^j} + \frac{1}{3} u_{x2y}^j \frac{\partial f}{\partial u_{x2y}^j} \\
& + \frac{2}{3} u_{x2y}^j \frac{\partial f}{\partial u_{x2y}^j} + \frac{2}{3} u_{2xy}^j \frac{\partial f}{\partial u_{2xy}^j} + \frac{1}{3} u_{2x2y}^j \frac{\partial f}{\partial u_{2x2y}^j} \\
& - D_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_1=0}^1 u_{i_1 xy}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=1}^1 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=1}^1 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}
\end{aligned}$$

$$\begin{aligned}
& + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + \left(\begin{matrix} 1+1 \\ 1 \end{matrix} \right) u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_1-1}{k_1+k_2-2} \right) \\
& \quad \left. (-D_x)^{k_1-2} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& - D_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-D_x)^{k_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_2=0}^1 u_{x i_2 y}^j \sum_{k_1=1}^1 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + \sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=2}^2 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-D_x)^{k_1-2} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_{xy}^j \sum_{k_1=1}^1 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + \left(\begin{matrix} 1+1 \\ 1 \end{matrix} \right) u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_2-1}{k_1+k_2-2} \right) \\
& \quad \left. (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + \left(\begin{matrix} 1+1 \\ 1 \end{matrix} \right) u_{2xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_1-1}{k_1+k_2-2} \right) \\
& \quad \left. (-D_x)^{k_1-2} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \tag{A.36}
\end{aligned}$$

$$\begin{aligned}
& + \left(\begin{matrix} 1+1 \\ 1 \end{matrix} \right) u_{x2y}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_2-1}{k_1+k_2-2} \right) \\
& \quad \left. (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right]. \tag{A.37}
\end{aligned}$$

The last integration by parts is applied to (A.36) and (A.37). No splitting is necessary as only a D_y can be removed from (A.36) and only a D_x can be removed from (A.37).

Integrating by parts, one obtains

$$\begin{aligned}
u^j \mathcal{L}_{u^j(x,y)} f &= u^j \frac{\partial f}{\partial u^j} + \sum_{k_1=1}^2 u_{k_1 x}^j \frac{\partial f}{\partial u_{k_1 x}^j} + \sum_{k_2=1}^2 u_{k_2 y}^j \frac{\partial f}{\partial u_{k_2 y}^j} + u_{xy}^j \frac{\partial f}{\partial u_{xy}^j} + u_{2xy}^j \frac{\partial f}{\partial u_{2xy}^j} \\
&+ u_{x2y}^j \frac{\partial f}{\partial u_{x2y}^j} + \frac{1}{3} u_{2x2y}^j \frac{\partial f}{\partial u_{2x2y}^j} \\
&- \mathbf{D}_x \left[\sum_{i_1=0}^1 u_{i_1 x}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=0}^0 (-\mathbf{D}_x)^{k_1-(i_1+1)} (-\mathbf{D}_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
&+ \sum_{i_1=0}^1 u_{i_1 xy}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=1}^1 \left(\frac{k_2}{k_1+k_2} \right) (-\mathbf{D}_x)^{k_1-(i_1+1)} (-\mathbf{D}_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ \sum_{i_1=0}^1 u_{i_1 x 2y}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) \\
&\quad \left. (-\mathbf{D}_x)^{k_1-(i_1+1)} (-\mathbf{D}_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
&+ u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=1}^1 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-\mathbf{D}_x)^{k_1-2} (-\mathbf{D}_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-\mathbf{D}_x)^{k_1-1} (-\mathbf{D}_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) (-\mathbf{D}_x)^{k_1-2} (-\mathbf{D}_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-\mathbf{D}_x)^{k_1-1} (-\mathbf{D}_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ \left(\begin{matrix} 1+1 \\ 1 \end{matrix} \right) u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_1-1}{k_1+k_2-2} \right) \\
&\quad \left. (-\mathbf{D}_x)^{k_1-2} (-\mathbf{D}_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
&- \mathbf{D}_y \left[\sum_{i_2=0}^1 u_{i_2 y}^j \sum_{k_1=0}^0 \sum_{k_2=i_2+1}^2 (-\mathbf{D}_x)^{k_1} (-\mathbf{D}_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
&+ \sum_{i_2=0}^1 u_{x i_2 y}^j \sum_{k_1=1}^1 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1+k_2} \right) (-\mathbf{D}_x)^{k_1-1} (-\mathbf{D}_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&+ \sum_{i_2=0}^1 u_{2x i_2 y}^j \sum_{k_1=2}^2 \sum_{k_2=i_2+1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_1-1}{k_1+k_2-1} \right) \\
&\quad \left. (-\mathbf{D}_x)^{k_1-2} (-\mathbf{D}_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right]
\end{aligned} \tag{A.38}$$

$$\begin{aligned}
& + u_{xy}^j \sum_{k_1=1}^1 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u^j \sum_{k_1=1}^2 \sum_{k_2=1}^2 \left(\frac{k_2}{k_1+k_2} \right) (-D_x)^{k_1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_x^j \sum_{k_1=2}^2 \sum_{k_2=1}^2 \left(\frac{k_1}{k_1+k_2} \right) \left(\frac{k_2}{k_1+k_2-1} \right) (-D_x)^{k_1-1} (-D_y)^{k_2-1} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + u_y^j \sum_{k_1=1}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_2-1}{k_1+k_2-1} \right) (-D_x)^{k_1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& + \binom{1+1}{1} u_{xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_2-1}{k_1+k_2-2} \right) \\
& \quad \left. (-D_x)^{k_1-1} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& + 2u_{2x2y}^j \binom{2}{4} \binom{2}{3} \binom{1}{2} \frac{\partial f}{\partial u_{2x2y}^j} + 2u_{2x2y}^j \binom{2}{4} \binom{2}{3} \binom{1}{2} \frac{\partial f}{\partial u_{2x2y}^j} \quad (A.39) \\
& - D_x \left[\binom{1+1}{1} u_{x2y}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_2-1}{k_1+k_2-2} \right) \right. \\
& \quad \left. (-D_x)^{k_1-2} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& - D_y \left[\binom{1+1}{1} u_{2xy}^j \sum_{k_1=2}^2 \sum_{k_2=2}^2 \left(\frac{k_2}{k_1+k_2} \right) \left(\frac{k_1}{k_1+k_2-1} \right) \left(\frac{k_1-1}{k_1+k_2-2} \right) \right. \\
& \quad \left. (-D_x)^{k_1-2} (-D_y)^{k_2-2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right].
\end{aligned}$$

Next, (A.39) and (A.38), are combined into a single sum. Likewise, terms under D_x and D_y are combined into single sums, and coefficients are presented in factorial form. Thus,

$$\begin{aligned}
u^j \mathcal{L}_{u^j(x,y)} f & = \sum_{k_1=0}^2 \sum_{k_2=0}^2 u_{k_1 x k_2 y}^j \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& - D_x \left[\sum_{i_1=0}^1 \sum_{i_2=0}^2 u_{i_1 x i_2 y}^j \binom{i_1+i_2}{i_1} \sum_{k_1=i_1+1}^2 \sum_{k_2=i_2}^{i_2} \frac{(k_1+k_2-i_1-i_2-1)! k_1! k_2!}{(k_1-i_1-1)! (k_1+k_2)! (k_2-i_2)!} \right. \\
& \quad \left. (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_1=0}^1 \sum_{i_2=0}^1 u_{i_1 x i_2 y}^j \binom{i_1+i_2}{i_1} \sum_{k_1=i_1+1}^2 \sum_{k_2=i_2+1}^2 \frac{(k_1+k_2-i_1-i_2-1)! k_1! k_2!}{(k_1-i_1-1)! (k_1+k_2)! (k_1-i_1)!} \\
& \quad \left. (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right]
\end{aligned}$$

$$\begin{aligned}
& -D_y \left[\sum_{i_1=0}^2 \sum_{i_2=0}^1 u_{i_1 x i_2 y}^j \binom{i_1+i_2}{i_2} \sum_{k_1=i_1}^{i_1} \sum_{k_2=i_2+1}^2 \frac{(k_1+k_2-i_1-i_2-1)! k_1! k_2!}{(k_2-i_2-1)! (k_1+k_2)! (k_1-i_1)!} \right. \\
& \quad \left. (-D_x)^{k_1-i_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right. \\
& + \sum_{i_1=0}^1 \sum_{i_2=0}^1 u_{i_1 x i_2 y}^j \binom{i_1+i_2}{i_1} \sum_{k_1=i_1+1}^2 \sum_{k_2=i_2+1}^2 \frac{(k_1+k_2-i_1-i_2-1)! k_1! k_2!}{(k_2-i_2-1)! (k_1+k_2)! (k_1-i_1)!} \\
& \quad \left. (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& = \sum_{k_1=0}^2 \sum_{k_2=0}^2 u_{k_1 x k_2 y}^j \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
& - D_x \left[\sum_{i_1=0}^1 \sum_{i_2=0}^2 u_{i_1 x i_2 y}^j \sum_{k_1=i_1+1}^2 \sum_{k_2=i_2}^2 \frac{\binom{i_1+i_2}{i_1} \binom{k_1+k_2-i_1-i_2-1}{k_1-i_1-1}}{\binom{k_1+k_2}{k_1}} \right. \\
& \quad \left. (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right] \\
& - D_y \left[\sum_{i_1=0}^2 \sum_{i_2=0}^1 u_{i_1 x i_2 y}^j \sum_{k_1=i_1}^2 \sum_{k_2=i_2+1}^2 \frac{\binom{i_1+i_2}{i_2} \binom{k_1+k_2-i_1-i_2-1}{k_2-i_2-1}}{\binom{k_1+k_2}{k_2}} \right. \\
& \quad \left. (-D_x)^{k_1-i_1} (-D_y)^{k_2-(i_2+1)} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right]. \tag{A.40}
\end{aligned}$$

The full form (A.40) is equivalent to (6.40) in Theorem 6.5. The completion of the proof is identical to the completion of Theorem 6.5 after (6.40). \blacksquare

A.3 Proof of Theorem 6.6

Theorem 6.6 states that the x -integrand (6.8) for the two-dimensional homotopy operator (6.5) - (6.7) given in [35] and the two-dimensional x -integrand (6.32) given in this dissertation are the same. That is,

$$\begin{aligned}
& \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \left(\frac{1+i_1}{1+i_1+i_2} \right) D_x^{i_1} D_y^{i_2} \left(u_{i_1 x i_2 y}^j \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-D_x)^{k_1-(i_1+1)} (-D_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \\
& = \sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \left(\sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} \frac{\binom{i_1+i_2}{i_1} \binom{k_1+k_2-i_1-i_2-1}{k_1-(i_1+1)}}{\binom{k_1+k_2}{k_1}} u_{i_1 x i_2 y}^j (-D_x)^{k_1-i_1-1} (-D_y)^{k_2-i_2} \right) \frac{\partial f}{\partial u_{k_1 x k_2 y}^j}. \tag{A.41}
\end{aligned}$$

The proof of Theorem 6.6 requires two lemmas. Both lemmas are combinatoric identities. The proof for the first lemma is based on a partial proof given in [32]. The second lemma was easily established using *Mathematica*, however, the lemma was not found in literature.

Lemma A.4. *For any positive integer n and $x \notin \{0, -1, \dots, -n\}$,*

$$\sum_{k=0}^n \binom{n}{k} \frac{(-1)^k}{x+k} = \frac{1}{x \binom{x+n}{n}}. \quad (\text{A.42})$$

PROOF: Define $\Delta f(x) = f(x+1) - f(x)$ and $Ef(x) = f(x+1)$. Then

$$\Delta f(x) = (E - 1)f(x). \quad (\text{A.43})$$

Apply the binomial theorem to (A.43) to get

$$\begin{aligned} \Delta^n f(x) &= \sum_{k=0}^n \binom{k}{n} (E^k (-1)^{n-k}) f(x) \\ &= \sum_{k=0}^n \binom{k}{n} (-1)^{n-k} E^k f(x) \\ &= \sum_{k=0}^n \binom{k}{n} (-1)^{n-k} f(x+k). \end{aligned} \quad (\text{A.44})$$

Now, define

$$x^{\overline{m}} = x(x-1)(x-2)\cdots(x-(m-1)) = \frac{x!}{(x-m)!}, \quad m > 0. \quad (\text{A.45})$$

From (A.45), $(-1)^{\overline{-m}} = (-1)(-2)\cdots(-m) = (-1)^m m!$ and $(x-1)^{\overline{-m}} = \frac{(x-1)!}{(x-1+m)!}$, where x is a positive integer. Next, take $f(x) = (x-1)^{\overline{-1}}$. Using (A.44) and (A.45),

$$\begin{aligned} \Delta^n f(x) &= \Delta^n (x-1)^{\overline{-1}} \\ &= \Delta^n \frac{(x-1)!}{x!} \\ &= \Delta^n \frac{1}{x} \\ &= \sum_{k=0}^n \binom{k}{n} (-1)^{n-k} \frac{1}{x+k}, \end{aligned} \quad (\text{A.46})$$

which is the left-hand side of (A.42) multiplied by the factor $(-1)^n$. Using (A.45), the right-hand side of (A.42) becomes

$$\begin{aligned} \frac{1}{x \binom{x+n}{n}} &= \frac{n! x!}{x(x+n)!} = \frac{n! (x-1)!}{(x+n)!} \\ &= n! (x-1)^{\overline{-n-1}}. \end{aligned} \quad (\text{A.47})$$

Thus, to show that (A.42) holds, it is necessary to show that

$$\Delta^n(x-1)^{-1} = (-1)^n n! (x-1)^{-n-1}. \quad (\text{A.48})$$

The proof of (A.48) is shown by induction on n since n must be a positive integer by (A.45). For the first induction step, set $n = 1$,

$$\begin{aligned} \Delta(x-1)^{-1} &= \Delta \frac{1}{x} \\ &= \frac{1}{x+1} - \frac{1}{x} \\ &= \frac{x - (x+1)}{x(x+1)} \\ &= -\frac{1}{x(x+1)} \\ &= -\frac{(x-1)!}{(x+1)!} \\ &= (-1)^1 (x-1)^{-2}, \end{aligned}$$

which is the right-hand side of (A.48) with $n = 1$. For the second induction step, assume that (A.48) holds for $k \geq 1$, that is,

$$\Delta^k(x-1)^{-1} = (-1)^k k! (x-1)^{-k-1}. \quad (\text{A.49})$$

Next, it is necessary to show that (A.49) holds for $k+1$. The proof proceeds as follows,

$$\begin{aligned} \Delta^{k+1}(x-1)^{-1} &= \Delta \Delta^k(x-1)^{-1} \\ &= \Delta(-1)^k k! (x-1)^{-k-1} \quad \text{by (A.49)} \\ &= \Delta(-1)^k k! \frac{(x-1)!}{(x+k)!} \\ &= (-1)^k k! \left[\frac{x!}{(x+k+1)!} - \frac{(x-1)!}{(x+k)!} \right] \\ &= (-1)^k k! \left[\frac{x(x-1)! - (x-1)!(x+1+k)}{(x+1+k)!} \right] \\ &= (-1)^k k! \frac{(-k-1)(x-1)!}{(x+1-k)!} \\ &= (-1)^{k+1} (k+1)! \frac{(x-1)!}{(x+1-k)!} \\ &= (-1)^{k+1} (k+1)! (x-1)^{-k-2}, \end{aligned}$$

which is the right-hand side of (A.48) with $n = k+1$. Dividing both sides of (A.48) by $(-1)^n$ gives (A.42). ■

Lemma A.5. For any positive integer n , and real a and b ,

$$\sum_{i=0}^n (-1)^i \binom{i+a}{i} \binom{b}{n-i} = \binom{b-a-1}{n}. \quad (\text{A.50})$$

PROOF BY INDUCTION ON n : For the first induction step, let $n = 0$. Then

$$\begin{aligned} \sum_{i=0}^n (-1)^i \binom{i+a}{i} \binom{b}{-i} &= (-1)^0 \binom{a}{0} \binom{b}{0} \\ &= 1 \\ &= \binom{b-a-1}{0}. \end{aligned}$$

For the second induction step, let $n = k$ and assume that

$$\sum_{i=0}^k (-1)^i \binom{i+a}{i} \binom{b}{k-i} = \binom{b-a-1}{k}, \quad k \geq 0. \quad (\text{A.51})$$

For $n = k + 1$,

$$\begin{aligned} &\sum_{i=0}^{k+1} (-1)^i \binom{i+a}{i} \binom{b}{(k+1)-i} \\ &= \sum_{i=0}^{k+1} (-1)^i \frac{(i+a)!}{i! a!} \frac{b!}{(b-k+i-1)! (k-i+1)!} \\ &= \frac{a! b!}{a! (b-k-1)! (k+1)!} + \frac{(-1)^k (a+k+1)! b! (k+1)!}{(k+1)! a! (b! (k+1)!)} \\ &\quad + \sum_{i=1}^k (-1)^i \frac{(i+a)! b! \frac{(k+1)!}{(k-i+1)!}}{i! a! (b-k+i-1)! (k+1)!} \\ &= \frac{a! b!}{a! (b-k-1)! (k+1)!} + \frac{(-1)^k (a+k+1)! b! (k+1)!}{(k+1)! a! (b! (k+1)!)} \\ &\quad + \sum_{i=1}^k (-1)^i \left[\frac{(i+a)! b! \frac{k!}{(k-i+1)!} i}{i! a! (b-k+i-1)! (k+1)!} + \frac{(i+a)! b! \frac{k!}{(k-i)!}}{i! a! (b-k+i-1)! (k+1)!} \right] \\ &= \sum_{i=0}^k (-1)^i \left[\frac{(i+a)! b! \frac{k!}{(k-i)!}}{i! a! (b-k+i-1)! (k+1)!} - \frac{(a+i+1)! b! (i+1) \frac{k!}{(k-i)!}}{(i+1)! a! (b-k+i)! (k+1)!} \right] \\ &= \sum_{i=0}^k (-1)^i \left[\frac{(i+a)!}{i! a!} \frac{b!}{(b-k+1)! (k-i)!} \frac{b-k+i}{k+1} \right. \\ &\quad \left. - \frac{i+1}{i+1} \frac{(i+a)!}{i! a!} \frac{b!}{(b-k+1)! (k-i)!} \frac{a+i+1}{k+1} \right] \frac{k!}{(k-i)!} \\ &= \sum_{i=0}^k (-1)^i \binom{i+a}{i} \binom{b}{k-i} \frac{b-a-k-1}{k} \end{aligned}$$

$$\begin{aligned}
&= \binom{b-a-1}{k} \frac{b-a-k-1}{k} \quad \text{by (A.51)} \\
&= \frac{(b-a-1)!(b-a-k-1)}{k!(b-a-k-1)!(k+1)} \\
&= \frac{(b-a-1)!}{(k+1)!(b-a-k-2)!} \\
&= \binom{b-a-1}{k+1},
\end{aligned}$$

which is the right-hand side of (A.50). ■

Now, the proof for Theorem 6.6 can be shown. Theorem 6.6 shows the equivalence of the x -integrands. The y -integrands, (6.9) and (6.33), are also equivalent. The proof for the y -integrands is analogous to the proof for the x -integrands and is not shown.

PROOF OF THEOREM 6.6:

$$\begin{aligned}
\mathcal{I}_{u^j(x,y)}^{(x)} f &= \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \left(\frac{1+i_1}{1+i_1+i_2} \right) \\
&\quad \mathbb{D}_x^{i_1} \mathbb{D}_y^{i_2} \left(u^j \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-\mathbb{D}_x)^{k_1-(i_1+1)} (-\mathbb{D}_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \\
&= \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \left(\frac{1+i_1}{1+i_1+i_2} \right) \mathbb{D}_x^{i_1} \left(\sum_{m_2=0}^{i_2} \binom{i_2}{m_2} u_{m_2 y}^j \mathbb{D}_y^{i_2-m_2} \right. \\
&\quad \left. \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-\mathbb{D}_x)^{k_1-i_1-1} (-\mathbb{D}_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \\
&= \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \left(\frac{1+i_1}{1+i_1+i_2} \right) \left(\sum_{m_1=0}^{i_1} \binom{i_1}{m_1} \sum_{m_2=0}^{i_2} \binom{i_2}{m_2} u_{m_1 x m_2 y}^j \mathbb{D}_x^{i_1-m_1} \mathbb{D}_y^{i_2-m_2} \right. \\
&\quad \left. \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-\mathbb{D}_x)^{k_1-i_1-1} (-\mathbb{D}_y)^{k_2-i_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \right) \\
&= \sum_{i_1=0}^{M_1^j-1} \sum_{i_2=0}^{M_2^j} \sum_{m_1=0}^{i_1} \sum_{m_2=0}^{i_2} \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} u_{m_1 x m_2 y}^j \mathbb{D}_x^{k_1-m_1-1} \mathbb{D}_y^{k_2-m_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&\quad \left(\frac{1+i_1}{1+i_1+i_2} \right) \binom{i_1}{m_1} \binom{i_2}{m_2} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-1)^{k_1-i_1-1} (-1)^{k_2-i_2} \\
&= \sum_{m_1=0}^{M_1^j-1} \sum_{m_2=0}^{M_2^j} \sum_{i_1=m_1}^{M_1^j-1} \sum_{i_2=m_2}^{M_2^j} \sum_{k_1=i_1+1}^{M_1^j} \sum_{k_2=i_2}^{M_2^j} u_{m_1 x m_2 y}^j \mathbb{D}_x^{k_1-m_1-1} \mathbb{D}_y^{k_2-m_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&\quad \left(\frac{1+i_1}{1+i_1+i_2} \right) \binom{i_1}{m_1} \binom{i_2}{m_2} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-1)^{k_1+k_2-i_1-i_2-1}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{m_1=0}^{M_1^j-1} \sum_{m_2=0}^{M_2^j} \sum_{k_1=m_1+1}^{M_1^j} \sum_{k_2=m_2}^{M_2^j} \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} u_{m_1 x m_2 y}^j D_x^{k_1-m_1-1} D_y^{k_2-m_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&\quad \left(\frac{1+i_1}{1+i_1+i_2} \right) \binom{i_1}{m_1} \binom{i_2}{m_2} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-1)^{k_1+k_2-i_1-i_2-1} \\
&= \sum_{m_1=0}^{M_1^j-1} \sum_{m_2=0}^{M_2^j} u_{m_1 x m_2 y}^j \sum_{k_1=m_1+1}^{M_1^j} \sum_{k_2=m_2}^{M_2^j} D_x^{k_1-m_1-1} D_y^{k_2-m_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}^j} \\
&\quad \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \left(\frac{1+i_1}{1+i_1+i_2} \right) \binom{i_1}{m_1} \binom{i_2}{m_2} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-1)^{k_1+k_2-i_1-i_2-1}.
\end{aligned} \tag{A.52}$$

In (A.52), all total derivatives have been propagated to the left and combined. Next, the inner sum with the combinatoric coefficient will be simplified. Taking the innermost double sum from (A.52),

$$\begin{aligned}
&\sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \left(\frac{1+i_1}{1+i_1+i_2} \right) \binom{i_1}{m_1} \binom{i_2}{m_2} \binom{k_1}{i_1+1} \binom{k_2}{i_2} (-1)^{k_1+k_2-i_1-i_2-1} \\
&= \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \frac{1+i_1}{1+i_1+i_2} \frac{i_1!}{m_1!(i_1-m_1)!} \frac{k_1!}{(i_1+1)!(k_1-i_1-1)!} \frac{i_2!}{m_2!(i_2-m_2)!} \\
&\quad \frac{k_2!}{i_2!(k_2-i_2)!} (-1)^{k_1+k_2-i_1-i_2-1} \\
&= (-1)^{k_1+k_2-1} \frac{k_1!k_2!}{m_1!m_2!} \\
&\quad \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \frac{(-1)^{-(i_1+i_2)}}{(1+i_1+i_2)(i_1-m_1)!(k_1-i_1-1)!(i_2-m_2)!(k_2-i_2)!} \\
&= \frac{(-1)^{k_1+k_2-1} k_1! k_2!}{m_1! m_2! (k_1-m_1)! (k_2-m_2)!} \\
&\quad \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \frac{(-1)^{-(i_1+i_2)} (k_1-m_1)! (k_2-m_2)!}{(1+i_1+i_2)(i_1-m_1)!(k_1-i_1-1)!(i_2-m_2)!(k_2-i_2)!} \\
&= (-1)^{k_1+k_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \binom{k_1-m_1}{i_1-m_1} \binom{k_2-m_2}{i_2-m_2} \frac{(-1)^{-(i_1+i_2)} (k_1-i_1)}{1+i_1+i_2}.
\end{aligned}$$

Let $n_1 = i_1 - m_1$ and $n_2 = i_2 - m_2$, Then

$$\begin{aligned}
&(-1)^{k_1+k_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \sum_{i_1=m_1}^{k_1-1} \sum_{i_2=m_2}^{k_2} \binom{k_1-m_1}{i_1-m_1} \binom{k_2-m_2}{i_2-m_2} \frac{(-1)^{-(i_1+i_2)} (k_1-i_1)}{1+i_1+i_2} \\
&= (-1)^{k_1+k_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \\
&\quad \sum_{n_1=0}^{k_1-m_1-1} \sum_{n_2=0}^{k_2-m_2} \binom{k_1-m_1}{n_1} \binom{k_2-m_2}{n_2} \frac{(-1)^{-(n_1+m_1+n_2+m_2)} (k_1-n_1-m_1)}{1+n_1+m_1+n_2+m_2}
\end{aligned}$$

$$\begin{aligned}
&= (-1)^{k_1+k_2-m_1-m_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \sum_{n_1=0}^{k_1-m_1-1} \binom{k_1-m_1}{n_1} (k_1-n_1-m_1) (-1)^{-n_1} \\
&\quad \sum_{n_2=0}^{k_2-m_2} \binom{k_2-m_2}{n_2} \frac{(-1)^{-n_2}}{1+m_1+m_2+n_1+n_2} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \\
&\quad \sum_{n_1=0}^{k_1-m_1-1} \binom{k_1-m_1}{n_1} (k_1-n_1-m_1) \frac{(-1)^{-n_1}}{(1+n_1+m_1+m_2) \binom{1+n_1+m_1+k_2}{k_2-m_2}}
\end{aligned}$$

by Lemma A.4

$$\begin{aligned}
&= (-1)^{k_1+k_2-m_1-m_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \sum_{n_1=0}^{k_1-m_1-1} \frac{(-1)^{-n_1} (k_1-n_1-m_1) \binom{k_1-m_1}{n_1}}{(1+n_1+m_1+m_2) \binom{1+n_1+m_1+k_2}{k_2-m_2}} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \\
&\quad \sum_{n_1=0}^{k_1-m_1-1} \frac{(-1)^{-n_1} (k_1-n_1-m_1)}{(1+n_1+m_1+m_2)} \frac{(k_1-m_1)! (1+n_1+m_1+m_2)! (k_2-m_2)!}{(k_1-m_1-n_1)! n_1! (1+n_1+m_1+k_2)!} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \binom{k_1}{m_1} \binom{k_2}{m_2} \\
&\quad \sum_{n_1=0}^{k_1-m_1-1} \frac{(-1)^{-n_1} (k_1-m_1)! (n_1+m_1+m_2)! (k_2-m_2)!}{(k_1-m_1-n_1-1)! n_1! (1+n_1+m_1+k_2)!} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \frac{k_1! k_2!}{m_1! m_2!} \sum_{n_1=0}^{k_1-m_1-1} \frac{(-1)^{-n_1} (n_1+m_1+m_2)!}{(k_1-m_1-n_1-1)! n_1! (1+n_1+m_1+k_2)!} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \frac{k_1! k_2! (m_1+m_2)!}{m_1! m_2! (k_1+k_2)!} \\
&\quad \sum_{n_1=0}^{k_1-m_1-1} \frac{(-1)^{-n_1} (n_1+m_1+m_2)! (k_1+k_2)!}{(k_1-m_1-n_1-1)! n_1! (1+n_1+m_1+k_2)! (m_1+m_2)!} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \frac{\binom{m_1+m_2}{m_1}}{\binom{k_1+k_2}{k_1}} \sum_{n_1=0}^{k_1-m_1-1} (-1)^{-n_1} \binom{n_1+m_1+m_2}{n_1} \binom{k_1+k_2}{k_1-m_1-n_1-1} \\
&= (-1)^{k_1+k_2-m_1-m_2-1} \frac{\binom{m_1+m_2}{m_1}}{\binom{k_1+k_2}{k_1}} \binom{k_1+k_2-m_1-m_2-1}{k_1-m_1-1} \quad \text{by Lemma A.5.}
\end{aligned}$$

(A.53)

Replacing the inner sum of (A.52) with (A.53) gives the right-hand side of (A.41). \blacksquare