Applications of Symbolic Computation to the Calculus of Moving Surfaces

A Thesis

Submitted to the Faculty

of

Drexel University

by

Mark W. Boady

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

June 2016

© Copyright 2016
Mark W. Boady. All Rights Reserved.

Dedications

To my lovely wife, Petrina. Without your endless kindness, love, and patience, this thesis never would have been completed.

To my parents, Bill and Charlene, you have loved and supported me in every endeavor. I am eternally grateful to be your son.

To my in-laws, Jerry and Annelie, your endless optimism and assistance helped keep me motivated throughout this project.

Acknowledgments

I would like to express my sincere gratitude to my supervisors Dr. Jeremy Johnson and Dr. Pavel Grinfeld.

Dr. Johnson's patience and insight provided a safe harbor during the seemingly endless days of debugging and testing code. I cannot thank him enough for improvements in my writing, coding, and presenting skills.

This thesis could not have existed without Dr. Grinfeld's pioneering research. His vast knowledge and endless enthusiasm propelled us forward at every roadblock.

A special thanks also goes out to the members of my thesis committee: Dr. Bruce Char, Dr. Dario Salvucci, and Dr. Hoon Hong. Without their valuable input, this work would not have been fully realized. They consistently helped to keep this project focused while I was buried in the details of development.

All the members of the Drexel Computer Science Department have provided endless help in my development as a teacher and researcher. I am especially grateful to the other members of the High Performance Computing Lab: Lingchuan Meng and Gavin Harrison.

Two members of the Drexel Faculty, Dr. Werner Krandick and Dr. Krzysztof Nowak, need to be highlighted for providing valuable sources for discussion and insight. I am very grateful for their interest in both this project and my development as a student.

All the members of the Computer Science faculty have been a great help to me, and I am thankful to them all.

Contents

List of Figures

List of Tables

Abstract
Applications of Symbolic Computation to the Calculus of Moving Surfaces

Mark W. Boady
Advisors: Dr. Jeremy Johnson and Dr. Pavel Grinfeld

In the physical world, objects change shape over time. A soap bubble blowing in the wind changes shape and density as it floats through the air. Red blood cells change shape to carry oxygen through our veins. Modeling these problems requires deforming manifolds.

The Calculus of Moving Surfaces (CMS) is an analytical framework for studying deforming manifolds. The CMS is an extension of tensor calculus. Both approach problems from a geometric perspective, without reference to specific coordinate systems. To evaluate a specific realization of a problem, a coordinate system is chosen and a CMS expression is converted to a series of n-dimensional array calculations using standard calculus.

This generality has many costs. The length of expressions grows quickly, in many cases exponentially. Although it is applicable to a wide range of problems, calculations quickly become intractable. The expressions generated are not only long and difficult to work with, evaluating them on a specific coordinate system introduces an entirely different set of challenges.

We present the first compute algebra system designed specifically for the CMS. Our system, the Symbolic Computation of Moving Surfaces (SCMS) supports the derivation of CMS expressions and the evaluation of expressions on specific coordinate systems. Although large expressions are inherent in the framework, computer automation allows for the application of the CMS to significantly larger problems then can be done by hand and allows the CMS to be applied in an error free way to non-trivial problems.

We have developed two libraries making up the SCMS. The first is a term rewrite system, CMSTRS, developed in Java. This library automates the analytic framework of the CMS. Expressions are kept at a high level, retaining the generality of the CMS. The second, CMSTensor, is for evaluation on specific coordinate systems. It is implemented using the Maple computer algebra system. It leverages the power of this computer algebra system to evaluate CMS expressions as a combination of n-dimensional array manipulations and standard calculus operations.

We have applied our system to a non-trivial boundary variation problem: the symbolic series expansion of the Laplace Eigenvalues on the N-sided regular polygon under Dirichlet boundary conditions. This series is computed up to N^{-6}, two orders higher then previous results. Our calculations confirm previous hand calculations and extend the series beyond what was previously known.

1. Introduction

The Calculus of Moving Surfaces (CMS) is an analytic framework extending tensor calculus to deforming manifolds. Deforming manifolds appear in a wide range of physical systems as well as mathematical structures. The CMS gives a geometric representation of a surface, its properties, and the forces acting on it. Tensor calculus and the CMS give expressions that can be evaluated in any coordinate system. This approach leads to generalized expressions which can give insight into the inherent geometric structure. This generality also leads to additional complexity. None of the simplifications that may be present in specific coordinate spaces or on particular realizations of a problem will be present. This leads to a trade-off between generality and expression complexity.

The tensor calculus was proposed in 1900 by Gregorio Ricci-Curbastro and Tullio Levi-Civita [93]. This famous paper laid the foundations of a general and compact notation for absolute differential calculus. Tensor calculus has become a key tool in the study of Reimannian geometry, general relativity, euclidean geometry, and Newtonian mechanics [104]. Albert Einstein's classic works on General Relativity depend on tensor calculus for their generality and beauty [29].

The tensor calculus was created for the study of stationary surfaces. Jacques Hadamard created the CMS in 1903 to allow for the study of surfaces with evolving shapes [52]. Work to merge Hadamard's ideas with tensor calculus culminated in 1991 when the CMS was defined as an extension of tensor calculus [37]. This greatly extended the reach of an already powerful calculus. An improved version of the framework was presented in 2012 by Pavel Grinfeld [44].

Moving surfaces are an intrinsic part of the physical world, like stationary surfaces they require a unique representation [45]. The CMS is applicable to a wide breadth problems from physical simulations, fluid film dynamics and blood cell modeling, to shape optimization and boundary variation problems. The power and generality inherent in the CMS comes at a cost. For many problems, working with complex models can quickly lead to intractable problems. The cost of generality is expression swell.

Expression swell is an important problem in using the CMS. Taking repeated derivatives in the CMS can cause exponential expression swell. Simple expressions can reach thousands of terms after only a few derivatives. Dealing with an expression this long by hand is error prone if not impossible. Even the seemingly simple problem of understanding the change in surface area as a shape changes runs 94 terms at the 4th derivative. Each of these terms is the product of 5-7 values. An additional problem is attempting to simplify these expressions by combining terms. Just comparing each of these expressions by hand would be a long and arduous task.

Once the CMS expression has be generated, a different type of complexity is introduced. To evaluate the expression, each term must be computed as an N-dimensional array on a particular surface. With hundreds or thousands of terms, this may already be intractable. Even if the expressions just need to be translated to a computer system, doing so without introducing error would be difficult. Evaluation of these expressions would also be different for every surface. This means restarting from the CMS expression for each new surface. Once these expressions are converted into algebra using n-dimensional arrays, the value at each possible may still be an algebraic expression running hundreds of terms.

Our system tackles all these problems, without giving up the generality of the CMS. Although expressions can always grow past the limits of computer hardware, the system presented here reaches far past existing methods. These are the same problems that have long motivated the automation of mathematical calculations.

Computer Algebra Systems (CAS) have a long history of advancing analytic methods through automation [34]. By automating the manipulation and derivation of large symbolic formulas, CAS have allowed for numerous advancements. Specialized systems for tensor calculus [86], Groebner basis [78], Fourier Transforms [33], and Quantum Mechanics [100] have all been successful. Computers have long come to the rescue of mathematical problems that were outside the reach of human abilities.

The tensor calculus contains all same problems as the CMS. Its importance in general relativity has motivated a number of CASs [69, 98, 110, 75, 89, 81]. All these systems have advanced research in their fields [102, 91, 63]. These systems continue to push the limits of physics research [111]. Without an automated system, all these problems would have become intractable long ago. Our system provides the same advantages to a new class of problems.

The CMS has reached the limits of the tools currently available. The framework can be applied to the problems it was built to model, but those computations fail due to human limitations. Although computers can never erase the burden of complexity, they can exceed a human's ability to work with large expressions. The CMS has thus far lacked a specialized tool set. Piecemeal frameworks built on existing systems have worked in some cases, but a general purpose system will greatly improve the usability of the CMS. It does not require users to be able to develop or modify existing systems. It is also designed to be used in the same way as a person doing pen and paper calculations. If a user is familiar with the CMS, our system replicates the experience they are used to.

We have provided a general purpose system called The Symbolic Computation of Moving Surfaces (SCMS). It is a software system that fills this gap. The software contains two libraries which can be used to create custom programs. Each of our libraries views the CMS from a different perspective. When combined they allow for general symbolic computation within the CMS. Depending on the type of work being done, each of the libraries may provide all the functionality needed on its own.

The Calculus of Moving Surfaces Term Rewriting System (CMSTRS) is a library developed for the Java programming language. It treats the CMS as a purely analytical framework. Expressions are written symbolically and computations replicate the algebraic manipulations performed during pen and paper calculations. Expressions derived using this library retain the geometric generality of the CMS. They are true for all coordinate systems. This library handles the expressions swell caused by taking repeated derivatives. It also automates the process of determine if expressions can be simplified using equivalence rules and provides automated export of expressions, eliminating more potential errors and tedious work.

The CMSTensor library is developed for the Maple computer algebra system [78]. The second perspective on the CMS is as a recipe book. CMS expressions can be evaluated in any coordinate system, but how this evaluation takes place is dependent on the coordinate system. A tensor represents a geometric property, such as curvature, but the actual value of the property is determined by the surface coordinate system.

The CMSTensor library can evaluate CMS expressions once a coordinate system is selected. The library transforms the expression into a coordinate specific set of algebraic computations. The library is implemented in Maple to allow access to a plethora of existing computer algebra tools. Specifically, we take advantage of its ability to take derivatives, integrate, and work with special functions, such as the Bessel J functions and trigonometric functions.

This system can deal with the large expressions generated by the CMS. It can also handle the complexity of correctly computing the values of the CMS objects in a specified coordinate system. Finally, this library leverages the power of existing CASs to evaluate the CMS expressions on the specified surface.

The SCMS was motivated by a question posed by Pavel Grinfeld and Gilbert Strang in 2004 [46]. *What is the series in $1/N$ for the simple Laplace eigenvalues λ_N on a regular polygon with N sides under Dirichlet boundary conditions?* Initial results generated interest in many fields from quantum mechanics to pattern recognition [4, 50, 64]. Investigation into the exact series was hindered by the lack of available tools. This problem embodies the flaws and triumphs of the CMS. General expressions for how the Laplacian Eigenvalues change as the surface they are on deforms can be found, but these expressions became so large and complex that they become nearly impossible to work with.

We proposed the SCMS in 2011 [10]. It confirmed the results of the first partial series answer to Grinfeld and Strang's question in 2012 [46]. The first version of the system was presented in 2013 [11]. The version of the system presented here extends the series from [46] by two orders in N. The size and complexity of the expressions required for these new terms had previously hindered their computation.

The SCMS provides advantages over current state of the art systems. First and foremost, this is the first symbolic computation system designed specifically for the CMS. Current tools are a combination of extensions of software built for other purposes and custom development to fill gaps. This system is designed with applied mathematics researchers in mind.

The CMSTRS gave the first rewrite rule set for the CMS and also showed the importance of combining TRS with object oriented design [11]. Comparison of CMS expressions to determine equivalence is closely related to some of the hardest problems in computer science.

The CMSTensor library translates the rules of the CMS into an algorithmic framework. This library gives algorithms to redefine the expressions of the CMS into a collection of operations on multi-dimensional arrays. This library supports spatial fields, static surfaces, and deforming surfaces. Current Tensor libraries are designed to handle spatial calculations, with a few extended to static surfaces. Support for deforming manifolds is at best partially implemented in existing libraries.

The SCMS also contributes to the study of Laplacian Eigenvalues with the results it has generated. The series for the Laplace-Dirichlet Eigenvalues on the regular polygon with N sides is the most extensive series published to date. It has confirmed recent high accuracy numerical results [58]. The extended series also provides advances in the study of boundary variation problems and can be applied to many fields.

We have used the SCMS to give the series for λ_N up to N^{-6}. This is the first time the values for N^{-5} and N^{-6} have been computed. We have collected supporting evidence for these calculations using multiple methods. The CMS expressions have been evaluated on two other surfaces. One with known answers and one that can be numerically approximated to very high accuracy. These two alternative problems provide evidence for the correctness of the CMS expressions. The solution on the polygon is computed both numerically and algebraically to provide strong evidence supporting the final results. This is the first time derivatives of these orders have been taken for this problem.

This thesis begins by laying out the historical and practical importance of the CMS. This builds a foundation for the CMSTRS package where the mathematical description of the CMS is translated into a collection of objects and rules. Next, the CMSTensor library describes how the CMS can be translated into a series of algorithms. With this groundwork laid, an extended example is given. The contour length (parameter) of a shape is studied with respect to its deformation.

The second half of the thesis looks at a class of problems related to the motivational question. The Laplace-Dirichlet Eigenvalues are introduced and their computation on multiple surfaces is described. This evaluation led to the need to extend the functionality of Maple by creating a new library. A special framework for finding closed forms of convolutions is described, which is required to deal with the eigenfunctions present on this surface.

Shapes with known solutions are used to test the correctness of the system. The N-sided polygon is computed symbolically and numerically to provide evidence of correctness. The results of applying the SCMS to these problems shows the power, accuracy, and importance of this system.

2. The Calculus of Moving Surfaces

2.1 Introduction

The CMS finds its roots in tensor calculus. The roots of tensor calculus are found in geometry. An understanding of the CMS must find its foundation in geometric reasoning. This chapter begins by building up the CMS from its geometric roots. First ambient space and static surfaces are examined. This leads the way for deforming surfaces. Some of the mechanics will be left for the implementation of the SCMS in Chapters 3 and 4.

Euclid's Elements, written in the 3rd century BCE, provides the basis for modern geometry [30]. In this work, geometry is studied from the perspective of the physical world. The axioms described here are based the simple ideas of lines and measurements.

The introduction of algebra to geometry by Rene Descartes created the cartesian coordinate system and lead the way for modern calculus [16]. Many coordinate systems have been used since Descartes' original work. Based on the problem to be tackled, some of these coordinate systems work better than others. For example, polar coordinates simplify calculations on a circle. This reliance on coordinate systems has allowed for pure algebraic computation, which may not reflect the geometric underpinnings. This algebraic representation allows for geometrically meaningless computations [74].

Tensor calculus returns to Euclid's geometric foundations. It is a full calculus that frees itself from specific coordinate systems by writing expressions that are true in any coordinate system [104]. This system returns to an abstraction of geometric concepts.

The CMS retains all the advantages of the tensor calculus, but adds additional properties necessary to represent deforming manifolds. This allows for geometric axioms, but in a formal calculus on a wide range of surfaces.

2.2 Elements of Tensor Calculus

Geometric features exist regardless of a coordinate system. The tensor calculus quantifies these features. Given any coordinate system, the values of these features can be computed.

A Euclidean space is a space that is essentially flat [30]. This means that the space allows for straight lines. This is the everyday experience of geometry. If a space allows for straight lines, then the notion of a vector can be defined. From some arbitrary position, a straight line can be draw of a certain length at a certain angle. Selecting an arbitrary origin point, every position in space can be defined by a vector starting at the origin point.

Given this general definition of a space, the elements of the tensor calculus can be defined. The position vector R is a geometric object [45]. It represents the position of a point from an arbitrarily selected origin. It can also be thought of as a function. Given some coordinate system $\mathbb{Z}^i = \{z_1, z_2, \cdots\}$, the function $R(\mathbb{Z}^i)$ returns the position within the coordinate system $\mathbb{Z}^i$. An example position vector for a cartesian space is shown in Figure 2.1.

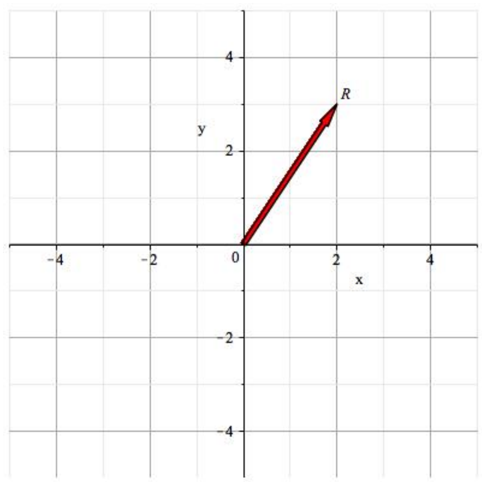

Figure 2.1: Position Vector for (2,3) in 2D Cartesian Coordinates.

The superscript in the expression $\mathbb{Z}^i$ is used to iterate through the coordinate system. For example, $\mathbb{Z}^2 = z_2$. Given two coordinate systems, a tensor can be described by how it changes relative to them. The covariant or contravariant property of a tensor describes how it changes with respect to two coordinate systems $\mathbb{Z}^i$ and $\mathbb{Z}^k$ [104].

A tensor is a geometric field defining a linear and homogeneous transformation [104]. A linear transformation is a mapping that preserves addition and scalar multiplication [95]. All tensors transform from a tensor to another tensor. They are homogeneous because the result of a transform is always a tensor.

A simple linear transform is rotation. If a vector is rotated 90° and then doubled in size, its length is the same as if it was doubled then rotated 90°. Additionally, when the vector is rotated, it is still a vector. All operations on a tensor retain the tensor property. If the derivative is taken of a tensor, the result is still a tensor. This means that it can be evaluated on any coordinate system.

The set of quantities T^i is a contravariant vector. It is a vector with a position indexed by i. For example, a vector with three positions T^i has values T^1, T^2, and T^3. When the vector is transformed to a new coordinate system, indexed by k, the relationship between the two vectors is given by

$$T^k = T^i \frac{\partial \mathbb{Z}^k}{\partial \mathbb{Z}^i} \tag{2.1}$$

If it is a covariant vector then it transforms according to

$$T_k = T_i \frac{\partial \mathbb{Z}^i}{\partial \mathbb{Z}^k} \tag{2.2}$$

.

The repeated index i denotes a contraction, a summation over the index.

These transforms give the tensor property and allow tensor expressions to be evaluated in any coordinate system. Given any expression, in any coordinate system, these formula give a recipe to transform it into any other coordinate system.

The position vector is used to create the covariant basis for the space. The partial derivative of each element in R is taken with respect to each coordinate. This produces an n by n array, for an n-dimensional coordinate space. The covariant basis Z_i is a set of vectors that define the basis imposed by the coordinate system.

$$Z_i = \frac{\partial R}{\partial \mathbb{Z}^i} \tag{2.3}$$

The covariant basis can be used to generate the covariant metric tensor using a pairwise dot product [45].

$$Z_{ij} = Z_i \bullet Z_j \tag{2.4}$$

This is the first tensor created for the space. This tensor can be used to measure lengths, areas, and volumes [45]. It contains all the information about the dot product. There is also a contravariant basis Z^i. The contravariant metric tensor Z^{ij} is the inverse of the covariant metric tensor.

The two basis are orthonormal and their product is the Kronecker delta.

$$Z_i \bullet Z^j = \delta^i_j \tag{2.5}$$

The Kronecker delta is an identity, it is 1 when $i = j$ and zero otherwise [104].

Objects which can be obtained by the measurement of distances and derivatives of distances are intrinsic [45]. The study of geometric properties using only intrinsic objects is called Riemannian geometry.

The Christoffel Symbol, Γ_{abc}, defines parallel transportation. When taking a partial derivative, the result may not retain the tensor property. When taking derivatives the Christoffel symbol is combined with the partial derivative so that the result is a tensor. This will be used in Equation 2.31.

The Levi-Civita symbol, $\epsilon_{ij\ldots}$, is the permutation tensor. It has the square root of the determinant of the covariant metric times -1 for odd permutations of the indexes and 1 for even permutations. All other spaces are 0. The covariant metric is a square matrix, so the determinant uses the standard linear algebra formula. The contravariant version is 1 over the square root of the determinant. Derivatives can be taken with respect to the space using the covariant derivative, ∇_i, and contravariant derivative, ∇^i. The metric tensor may also be used to switch a tensor index between the covariant and contravariant property.

If a surface is embedded into an ambient space, it has its own metric tensors, $S^{\alpha\beta}$ and $S_{\alpha\beta}$. To differentiate the two spaces, greek letters are used for the indexes on the surface coordinates and latin letters are used for those defined in the ambient space.

An object defined in the entire space also exists on the surface, since this is just a subset of space. The shift tensor, Z^i_α, gives the relationship between the ambient space and the surface.

The shift tensor and surface metrics define the surface's tangent plane. The normal is perpendicular to the surface and of unit length. The tensor representing the surface normal is N^i. Figure 2.2 shows a curved surface with a normal and tangent plane.

Figure 2.2 shows a curved plane. If a straight line is drawn on a sphere, two parallel lines may intersect. This is not true on a flat plane. This curvature needs to be taken into account when working on a surface. The curvature tensor B^α_β encapsulates the curvature of the manifold.

Forces on the surface have derivatives, just like those in the ambient space. The surface has its

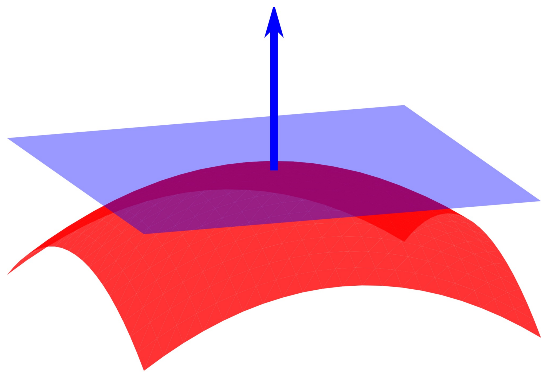

Figure 2.2: The surface normal and tangent plane at a point on a surface [3].

own versions of the covariant and contravariant derivatives ∇_α and ∇^α.

2.3 Example: Mean Curvature

The mean or average curvature of a surface captures geometrically the stretching of the plane [73]. In tensor calculus, the mean curvature is represented by B^α_α.

The repeated index α is a contraction. A contraction is a summation over the repeated indexes. In this case there are only two indexes, meaning it is the diagonal. The tensor B^α_α can be treated as a 2-dimensional array. The array is summed along matching indexes. The number of positions in the two dimensions are the same, and determined by the number of dimensions on the space S.

$$B^\alpha_\alpha = \sum_{\alpha \in S} B[\alpha][\alpha] \tag{2.6}$$

A contraction can only take place between two indexes one contravariant and one covariant. The indexes must both be spatial or surface indexes. The contraction is shorthand for a summation, which means the letter representing the index can change without changing the expression. The indexes can also be juggled, flipping the covariant and contravariant property of both.

The curvature tensor is given as

$$B_{\alpha\beta} = -1 Z^i_\alpha \nabla_\beta (N_i) \tag{2.7}$$

$$B^\beta_\alpha = B_{\alpha\gamma} S^{\gamma\beta} \tag{2.8}$$

Computation of the mean curvature starts by defining the position vector. On a circle, the position vector in polar coordinates is given by

$$R = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix} \tag{2.9}$$

The covariant basis is the partial derivative of the position vector.

$$Z_i = \frac{\partial R}{\partial \{r, \theta\}} \tag{2.10}$$

$$= \begin{bmatrix} \frac{\partial(r\cos\theta)}{\partial r} & \frac{\partial(r\sin\theta)}{\partial r} \\ \frac{\partial(r\cos\theta)}{\partial\theta} & \frac{\partial(r\sin\theta)}{\partial\theta} \end{bmatrix} \tag{2.11}$$

$$= \begin{bmatrix} \cos\theta & \sin\theta \\ -r\sin\theta & r\cos\theta \end{bmatrix} \tag{2.12}$$

The covariant metric tensor is product of Z_i with itself. Since the indexes are not being contracted, each copy needs its own index letter.

$$Z_{ij} = Z_i \bullet Z_j \tag{2.13}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix} \tag{2.14}$$

The restriction of this ambient space to the circle requires fixing the radius r. The shift tensor for this surface restriction is created by taking the partial derivative with respect to the surface coordinates of the restricted position vector.

$$Z^i_\alpha = \begin{bmatrix} [\frac{\partial r}{\partial \theta}] \\ [\frac{\partial \theta}{\partial \theta}] \end{bmatrix} \tag{2.15}$$

$$= \begin{bmatrix} [0] \\ [1] \end{bmatrix} \tag{2.16}$$

Notice the double brackets in the shift tensor's definition. The tensor has two indexes i and α. The i index has 2 positions, because the space was 2D. The α index only has one space because the radius has been fixed. The brackets are used to clarify nested arrays.

The surface normal is computed as

$$N^i = \epsilon^{ij} \epsilon_\alpha Z^\alpha_j \tag{2.17}$$

$$N_i = Z_{ij} N^j \tag{2.18}$$

On the circle, the Levi-Civita symbol is

$$\epsilon^{ij} = \begin{bmatrix} 0 & \frac{1}{r} \\ -\frac{1}{r} & 0 \end{bmatrix} \tag{2.19}$$

$$\epsilon_\alpha = [r] \tag{2.20}$$

The shift tensor has indexes in different locations then defined above. The metric tensors is used to fix this.

$$Z^\alpha_i = Z_{ij} Z^j_\beta S^{\beta\alpha} \tag{2.21}$$

The surface metric is

$$S^{\beta\alpha} = \left[\left[\frac{1}{r^2} \right] \right] \tag{2.22}$$

The tensor product multiplies every element by every other element. The number of dimensions in the matrix is the sum of the original dimensions.

$$Z^j_\beta S^{\beta\alpha} = \sum_\beta \begin{bmatrix} [0] \\ [1] \end{bmatrix} \left[\left[\frac{1}{r^2}\right]\right] \tag{2.23}$$

$$= \sum_\beta \begin{bmatrix} [[[0]]] \\ [[[\frac{1}{r^2}]]] \end{bmatrix} \tag{2.24}$$

$$= \begin{bmatrix} [0] \\ [\frac{1}{r^2}] \end{bmatrix} \tag{2.25}$$

This is now plugged into Equation 2.21.

$$Z^\alpha_i = Z_{ij} Z^j_\beta S^{\beta\alpha} \tag{2.26}$$

$$= \sum_j \begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix} \begin{bmatrix} [0] \\ [\frac{1}{r^2}] \end{bmatrix} \tag{2.27}$$

$$= \begin{bmatrix} [0] \\ [1] \end{bmatrix} \tag{2.28}$$

The surface normal is now evaluated.

$$N_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.29}$$

$$N^i = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.30}$$

To complete the evaluation of the curvature tensor, the covariant derivative is needed. The formula is given in [45] is

$$\nabla_\alpha V^i = \frac{\partial V^i}{\partial \mathbf{S}^\alpha} + \Gamma^i_{jm} V^m \tag{2.31}$$

The Γ^i_{jm} term is repeated for every index in the input tensor.

The curvature tensor is now given

$$B_{\alpha\beta} = -1 Z^i_\alpha \nabla_\beta (N_i) \tag{2.32}$$

$$= -1 \begin{bmatrix} [0] \\ [1] \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 \\ r \end{bmatrix} \end{bmatrix} \tag{2.33}$$

$$= [[-r]] \tag{2.34}$$

This is contracted with the surface metric to change the indexes to be non-matching.

$$B^\alpha_\beta = \left[\left[-\frac{1}{r}\right]\right] \tag{2.35}$$

The mean curvature is the trace of this array.

$$B^\alpha_\alpha = \sum_{\alpha=1}^{1} B[\alpha][\alpha] = -\frac{1}{r} \tag{2.36}$$

This is the mean curvature for a circle of radius r. The definition of the normal, positive or negative, will determine the sign of this solution. The mean curvature of any surface is B^α_α. Tensor calculus allows this general formula to be written and evaluated for any surface.

2.4 The Calculus of Moving Surfaces

What happens if the coordinate system defining the surface changes over time? This is the question that inspired the CMS.

A deforming manifold is a surface that has a coordinate system with a time parameter t. The instantaneous velocity of the manifold in the direction of the normal is called the surface velocity

C. This tensor encapsulated the transformation of the surface.

If the surface is changing over time, it is important to study this change. How does the deformation of the surface effect those tensors defined on it? The invariant time derivative $\dot{\nabla}$ gives the derivative with respect to change over time while keeping the tensor property for all objects [44]. The tensor property means that the expression can be evaluated on any coordinate system. Take the derivative of a tensor will also produce a tensor and all tensors can be evaluated on any coordinate system.

The invariant time derivative was originally conceived by Jacques Hadamard in 1903 while studying discontinuities in continuous media [52]. The first attempt to apply this concept to tensors was by T. Thomas in 1957 [105]. Another attempt was made in 1960 by C. Truesdell and R. Toupin [108]. None of these attempts ensured that for any tensor, application of the derivative resulted in a tensor. This meant that taking the derivative could potentially lead to expressions that could not be transformed between coordinate systems.

It wasn't until the work of M. Grinfeld in 1991 that a definition was given that preserved the tensor property for all operands, finally making the CMS an algebraically complete calculus [37]. Although the calculus was complete, it was hindered by a large rule set. The $\frac{\delta}{\delta t}$ derivative was dependent on the covariant or contravariant position of indexes. This means that for every permutation of index positions, it was possible a new rule was needed.

A revised invariant time derivative, $\dot{\nabla}$, was proposed by Pavel Grinfeld in 2012 [44]. This new definition retains the tensor property but is independent of the covariant/contravariant property of the input, drastically reducing the number and complexity of rules.

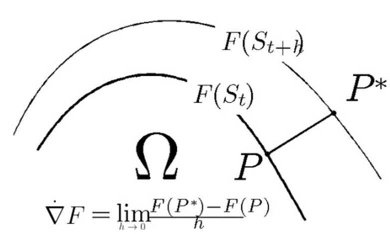

Figure 2.3: The invariant time derivative as time goes from t to $t+h$ [45].

The CMS is described in more detail in [49, 41]. The roots of the CMS can be found in [71, 82, 106]. A historical review of the CMS can be found in [40].

2.5 Applications of the CMS

The foundation of tensor calculus is geometry, which makes its reach extremely broad. The most popular application is the study of general relativity, where Einstein's contributions to the field are well known. The tensor calculus has been used in fields such as linear algebra, differential geometry, the calculus of variations, and continuum mechanics [45].

The CMS takes this power and expands it to surfaces that change shape over time. Deforming manifolds have applications in theoretical and physical modeling. The motivation for this symbolic computation system lies in boundary variation problems, specifically the solution to partial differential equations on bounded surfaces. The CMS, and by extension the SCMS, is by no means limited to only these fields.

Boundary variation problems examine fields defined on bounded surfaces and how those fields change when the boundary is deformed. A problem of interest is, what is the series in $1/N$ for the Laplace-Dirichlet Eigenvalues on a regular polygon with N-sides [46]? The solution can be found by deforming a known value, the eigenvalues on the unit circle, and changing the boundary into an N-sided regular polygon. The series has been shown up to $1/N^4$ in [47]. This was accomplished by a combination of hand calculations and computer algebra. The current series has already proven useful in applied mathematics, quantum physics, and pattern recognition [50, 4, 64]. This problem is described in detail beginning Chapter 6. The series is computed up to $1/N^6$ using the SCSM.

A common theme in CMS problems is that higher order variations provide greater accuracy as well as additional quantitative results. Computation of these variations is reliant on taking the invariant time derivative, $\dot{\nabla}$. In shape optimization determining the stability of a stationary configuration requires the second variation [40]. Even for some simple shape optimization problems, the second variation can be to difficult to calculate. Calculation of the minimal surface with a cavity of a given perimeter has only recently had any success due to the difficultly of determining variations [8]. A simple minimal surface would be a soap film with a hole cut in it by a string. A more complex minimal surface is shown in Figure 2.4.

Many problems require even higher order variations, the isoperimetric problem on surfaces of revolution of decreasing gauss curvature requires calculation of the fourth order variation [55]. In

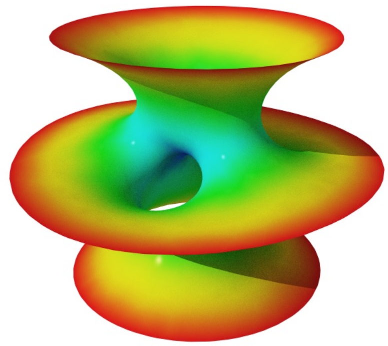

Figure 2.4: A minimal surface with a spherical hole [96, 22].

this problem, the goal is to enclose a prescribed area with a plane while minimizing the perimeter. The SCMS has determined that this variation contains 1380 terms and its computation requires a symbolic tool.

A similar problem with a physical application is the wobble of a slightly ellipsoidal inner core inside a slightly ellipsoidal outer core and mantle [48]. The planet Mercury has a large core, as seen in Figure 2.5, which creates an interesting wobble.

The fourth variation is also crucial in some biological applications. The fourth variation of Helfrich energy is essential to analysis of a red blood cell's shape [72]. Even under simplifying assumptions some of the expressions run longer then a page and remain largely intractable.

An even more general problem is the complete dynamics of a red blood cell. An exact nonlinear model has been proposed [41]. The blood cell is only one application of the CMS to fluid film dynamics. The CMS introduces a great deal of analytical order to these systems [39, 41, 43, 42]. One specific example of fluid film dynamics is the surface tension of a soap bubble. Again, these problems have remained largely intractable for hand calculations due to complexity and rapid expression swell. The SCMS will advance these fields through automation.

In all these applications, the CMS presents an analytical method to study the deformation. In each case, the computation of higher order variations limits research. By automating the CMS, computations can be done with expressions that are larger than can be manipulated by hand. Solutions to these problems are not limited by the theoretical foundations of the CMS, but by the technology available to apply it.

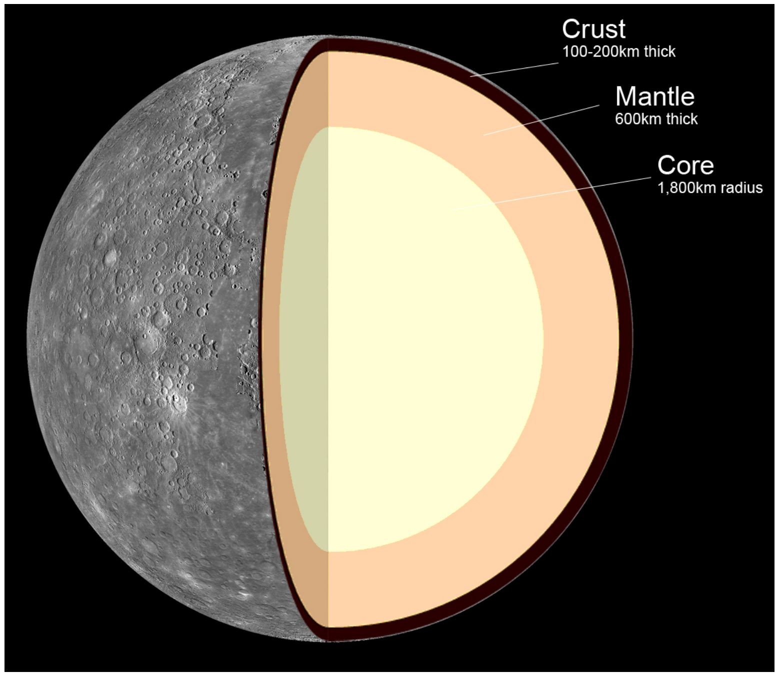

Figure 2.5: Mercury's large core plays an important role when modeling its motion [84].

2.6 Previous Work Automating Tensor Calculus

No existing symbolic packages are designed specifically for the CMS. A number of systems exist for applying the tensor calculus to stationary geometries. These packages have been successful in their respective fields, although they cannot be easily extended to the CMS. The complexity and breadth of applications is far greater when applying the CMS. The packages described below have proven successful within their fields, primarily General Relativity. The SCMS will provide the same advantages on an even wider range of problems. In these fields, calculations become intractable for many of the same reasons as in the CMS. These problems can become intractable both because of human limitations or computational ones. For example, computing derivatives in the CMS may become intractable because of the limits of hand calculations. On the other hand, computing all terms with a time parameter and taking the derivatives with a computer algebra system may be computationally difficult using available software.

Symbolic packages have helped overcome these obstacles on stationary geometries [109, 98, 86].

The CMS is an extension of tensor calculus, it contains all its complexities and more. The analysis of moving surfaces is a more complex analytical challenge then stationary geometries. These manifolds are Riemannian surfaces with a time-dependent metric.

General Purpose systems, such as Maple, Mathematica, and Sage are extremely successful, but force a reduction to specific analytical problems and lack the kind of high level view given by the CMS [101, 77, 112]. During the evaluation of CMS expressions, these general purpose tools prove extremely important. For example, computing the fourth variation of the perimeter of a circle stretching into an ellipse can be computed in polar coordinates with Maple. A more general questions is, "how does the curvature of a surface affect its perimeter during deformation?" This question cannot be answered on specific coordinate system, but can be with the CMS.

Although many of these packages implement the tensor notation, none implement the CMS. The majority of these packages originated in Relativity and lack even stationary embedded manifolds. Some packages, such as Cadabra and MathTensor, perform symbolic index manipulations in the tradition of classical tensor calculus, which is a challenging problem itself [79]. Other packages, such as FTensor [69], GRTensorII [110], the Maple tensor package [75], focus on the translation of expressions into literal multidimensional arrays and efficiently implementing contractions.

Cadabra [89, 88, 17, 18, 87] is closest to the system presented here. MathTensor [81, 86] is the oldest and perhaps most advanced but also a commercial package. Other symbolic manipulation packages focused on relativity include Ricci [70], Cartan [99], Maxima [27], RGCT [12], xAct [80], and TeLa [1].

These packages have proven quite successful their fields.. MathTensor has supported research in general relativity, such as [102], [91], and [63]. It has also been successful in statistical mechanics [7] and quantum field theory [51]. Cadabra is used in general relativity [19, 28, 13, 17, 83]. The xAct package has also been used in relativity [24, 25, 114]. In many of these works, the packages have ensured the correctness of proofs by replicating the proof and proving a rule trace. These packages were based in the theory of relativity and therefore closely match the research.

The SCMS handles both abstract formalism and symbolic evaluation. Although these systems share some very high level similarities with the presented system, the underlying calculus is drastically different. Modifying one of these systems to handle the CMS would be as difficult, if not more so, than building a new system.

3. Term Rewriting Systems

3.1 Introduction

Term Rewriting Systems (TRSs) are a method of automating equational logic. Due to its direct relation with equational reasoning, it has proven important in computer algebra, specifically algebraic simplification [5]. Surveys of TRS can be found in [26], [57], and [65].

A TRS is a pair $T = (\Sigma, R)$ where, Σ is signature defining the language, and R is the set of reduction rules. The signature is a set of function symbols and their arity. Constants are function symbols with arity zero. All valid terms in the language are produced through applying functions in this signature to each other.

The set of reduction (or rewrite) rules R is a collection of expressions with the form $l \to r$. The patterns l and r are terms made from the signature $\Sigma \cup V$ where V is a set of variables. The TRS is applied to an input term generated from Σ. The term is matched on the l side of a rewrite rule by matching variables to specific terms in the input. The term is rewritten to the expression defined in r, replacing variables with their values. The TRS repeatedly applies the rules from R until the there exist no matching patterns on the l side of any rules. The final value is called a normal form, a term for which no rewrite rules can be applied. The TRS terminates when the input term has been rewritten to a normal form.

One important application of TRSs is the uniform word problem. Given a set of equations, E, and two valid expressions t and s, determine if $t = s$ under E. Given two expressions s and t, determine if there exists any path using the equations of E thats makes them equivalent. A TRS can be created by giving a direction to each equation in E to produce rewrite rules R.

To solve the uniform word problem two important properties are required of the TRS. The first is termination, for any input term the rewrite system will terminate and produce an expression in normal form. A normal form is a term that does not match any rewrite rules. The TRS determines that it is at a normal form when no rules match the current term.

The second property is confluence, if multiple rules match a term which rule is applied will not change the final result. All paths will lead to the same normal form. If both these properties hold, the uniform word problem is solved by reducing both t and s to their normal forms. If the normal forms are exact matches, there exists a path to apply the equations which makes the original expressions equal.

This method is the basis for the Knuth-Bendix algorithm. The Knuth-Bendix algorithm shows that the existence of a terminating and confluent TRS for an equational system is not guaranteed [66]. The algorithm is nonterminating, it will either find a TRS that solves the uniform word problem, determine it is impossible and fail, or loop infinitely. Not all TRS are terminating [56]. It is possible to create nonterminating TRS from equational specifications. It is also impossible in some cases to determine if a TRS is terminating [56].

Although the general problem may be impossible, for many systems proving termination and confluence is possible [66]. These properties may also be proven modulo an equational system, such as associativity and commutativity [6, 61, 60]. This means that a single rule can match multiple expressions because it takes into account how commutativity can reorder values around operators. The normal forms of two equivalent terms may not be the same, but they will only differ by parenthesis and term orders. These are both simple equivalences to deal with by sorting. The CMSTRS presented here works modulo associativity and commutativity.

TRSs are popular for algebraic simplification. Maude, a standalone system for building TRS, includes an implementation of the Knuth-Bendix algorithm [20]. Rewriting is also included in computer algebra systems Maple and Mathematica [77, 112]. The TRS that can be built in many of these systems are first-order TRS, meaning that there can be no bound variables. A bound variable is when a variable appears multiple times in the same term and the relationship needs to be retained. For example, in an integral $\int f(x)dx$ we would need to track that the integration variable is x when simplifying. The CMS requires bounded variables.

The full source code is open source and available for download from `https://www.cs.drexel.edu/SCMS`.

3.2 A TRS for the Calculus of Moving Surfaces

The signature of the TRS is derived from the formalism of the CMS. This has already been described in Chapter 2. This section will focus symbolic definitions of the rules and objects.

The basic object of the CMS is the tensor. In the CMSTRS library, a tensor is a named value and has a set of properties. The tensor can exist in the ambient space or be restricted to a surface. The tensor also has an ordered list of named indexes. Each index can also be defined in space or on the surface. The index is either a contravariant or covariant index.

Table 3.1: TRS Signature

Symbol	Description
C	Surface Velocity
N^i	Surface Normal
$B^{\alpha}_{.\beta}$	Curvature
$Z^i_{.\alpha}$	Shift Tensor
$R^{\gamma}_{.\alpha\beta}$	Commutator
$+$	Addition
Juxtaposition	Multiplication
Repeated Indexes	Contraction
Integer Superscript	Exponent
∇_i	Covariant Space Derivative
∇_α	Covariant Surface Derivative
$\dot{\nabla}$	$\dot{\nabla}$-derivative
$\frac{\partial}{\partial t}$	Partial Time Derivative

The primary objects of the CMS are described in Chapter 2. The set of objects implement as part of the CMSTRS library are given in Table 3.1. Transformations of the objects by index juggling are also implemented.

A special symbol, the commutator tensor, is created to simplify implementation of the CMS rule set. The commutator tensor facilitates switching the order of $\dot{\nabla}$ and ∇_α [40]. It is a shorthand for the following:

$$R^{\gamma}_{.\alpha\beta} = \nabla^{\gamma}(CB_{\alpha\beta}) - \nabla^{\alpha}(CB^{\gamma}_{\beta}) - \nabla_{\beta}(CB^{\gamma}_{\alpha}) \quad (3.1)$$

The rewrite rules for the CMS require the introduction of variables for pattern matching. Index names are always considered variables. Unless explicitly noted, all other properties of the index must match exactly. F and G are variables for general tensors. In the CMS any valid expression is a tensor. If no indexes are attached to the variables F and G, then any combination of indexes is valid. Constant integers and rationals are given by $c_1, c_2, \cdots, c_n$.

The covariant and contravariant derivatives are defined by rules (3.2) to (3.5). These rules are true for any index of the derivative. Only the rules for ∇_α are shown, but these rules are repeated for other indexes. The summation symbol is used explicitly when need to clarify contractions.

$$\nabla_\alpha(FG) \to G\nabla_\alpha(F) + F\nabla_\alpha(G) \tag{3.2}$$

$$\nabla_\alpha(c_1) \to 0 \tag{3.3}$$

$$\nabla_\alpha(F+G) \to \nabla_\alpha(F) + \nabla_\alpha(G) \tag{3.4}$$

$$\nabla_\alpha(\sum_i F^{\cdots i \cdots}_{\cdots i \cdots}) \to \sum_i \nabla_\alpha(F^{\cdots i \cdots}_{\cdots i \cdots}) \tag{3.5}$$

The $\dot{\nabla}$-derivative is at the heart of the CMS. Calculating this derivative is the key to finding higher order variations. The differentiation table for specific tensor objects is given. These rules are independent of index juggling.

$$\dot{\nabla} Z^i_\alpha \to N^i \nabla_\alpha C \tag{3.6}$$

$$\dot{\nabla} N^i \to Z^i_\alpha \nabla^\alpha C \tag{3.7}$$

$$\dot{\nabla} B^\alpha_\beta \to \nabla^\alpha \nabla_\beta C + C B^\alpha_\gamma B^\gamma_\beta \tag{3.8}$$

$$\dot{\nabla} C^{c_1} \to c_1 C^{c_1 - 1} \dot{\nabla} C \tag{3.9}$$

$$\dot{\nabla} c_1 \to 0 \tag{3.10}$$

$$\dot{\nabla} f \to \frac{\partial f}{\partial t} + C N^i \nabla_i f \tag{3.11}$$

In the last rule 3.11, the value f is a scalar field defined in space.

The $\dot{\nabla}$-derivative commutes with contraction and satisfies the sum and the product rules.

$$\dot{\nabla} \sum_i F^{\cdots i \cdots}_{\cdots i \cdots} \to \sum_i \dot{\nabla} F^{\cdots i \cdots}_{\cdots i \cdots} \tag{3.12}$$

$$\dot{\nabla} FG \to G\dot{\nabla} F + F\dot{\nabla} G \tag{3.13}$$

$$\dot{\nabla}(F+G) \to \dot{\nabla} F + \dot{\nabla} G \tag{3.14}$$

Reordering $\dot{\nabla}$ and surface derivative introduces a collection of rules. These rules are given for the variable tensor with no indexes A. For each index in A, an additional term is added to the sum. Examples for all variations of A with one index are shown.

$$\dot{\nabla}\nabla_\alpha A \to \nabla_\alpha \dot{\nabla} A + C B^\gamma_\alpha \nabla_\gamma A \tag{3.15}$$

$$\dot{\nabla}\nabla_\alpha A^\beta \to \nabla_\alpha \dot{\nabla} A^\beta + C B^\gamma_\alpha \nabla_\gamma A^\beta + \dot{R}^{\beta}_{\cdot\gamma\alpha} A^\gamma \tag{3.16}$$

$$\dot{\nabla}\nabla_\alpha A_\beta \to \nabla_\alpha \dot{\nabla} A_\beta + C B^\gamma_\alpha \nabla_\gamma A_\beta - \dot{R}^{\gamma}_{\cdot\beta\alpha} A_\gamma \tag{3.17}$$

$$\tag{3.18}$$

For each index in the A tensor, a new $R^{\cdots}_{\cdots} A^{\cdots}_{\cdots}$ term is added. All possible terms are shown, they are just repeated as need for the indexes of A. Note that to add these values the tensors must be permuted to align the index orders.

The partial derivative $\frac{\partial}{\partial t}$ is defined by the following rules.

$$\frac{\partial FG}{\partial t} \to F\frac{\partial G}{\partial t} + G\frac{\partial F}{\partial t} \tag{3.19}$$

$$\frac{\partial c_1}{\partial t} \to 0 \tag{3.20}$$

$$\frac{\partial (F+G)}{\partial t} \to \frac{\partial F}{\partial t} + \frac{\partial G}{\partial t} \tag{3.21}$$

$$\frac{\partial \sum_i F^{\cdots i \cdots}_{\cdots i \cdots}}{\partial t} \to \sum_i \frac{\partial F^{\cdots i \cdots}_{\cdots i \cdots}}{\partial t} \tag{3.22}$$

$$\frac{\partial \nabla_\alpha F}{\partial t} \to \nabla_\alpha \frac{\partial F}{\partial t} \tag{3.23}$$

To complete the TRS, we add a few additional rules for simplification.

$$A^{c_1} A^{c_2} \to A^{c_1+c_2} \tag{3.24}$$

$$A + 0 \to A \tag{3.25}$$

$$A(F+G) \to AF + AG \tag{3.26}$$

$$0\, A \to 0 \tag{3.27}$$

Expressions with rationals are calculated immediately. After reaching a normal form, like terms are combined to decrease the size of the result term. This is handled by a separate routine.

3.3 Implementation

The CMSTRS is implemented as a Java Library. The rules and objects of the CMS have properties that can be modeled using an object oriented language. One of our goals was to make a system that replicated how the CMS is used by hand. There are many generalized rules, for example rules that match on any derivative. These are naturally implemented using inheritance. The result of performing operations on tensors always results in tensors, which is again easily replicated using inheritance. Additionally, features outside of a TRS are important to make the system more functional. Additional components for combining like terms and exporting to special file formats add to the power of this library.

3.3.1 Signature

All objects in the signature are tensors. The top level interface object is `Tensor`. A tensor is an object is an ordered list of free indexes. A tensor also has a boolean denoting if it is restricted to spatial coordinates or not.

The `Tensor` object has a minimum set of methods all symbols in the signature must implement. Each object has three closely related views. The first is a tensor. The second view is a collection of functions. Each object represents a mathematical function that takes input, or a mathematical function of arity zero. The arity of a function is the number of inputs, a function with arity zero is a constant or variable. The last view is as an expression tree. In this view, functions have their inputs as children.

The basic interface for the `Tensor` object has the following methods. These are implemented for every object in Table 3.1.

`toString`: prints the tensor out as a latex expression.

`numIndexes`: returns the number of indexes of the tensor.

`arity`: return the arity of the function, this is equivalent to the number of children.

`getInput`: on input i, returns the i-th function input if it exists.

`getIndex`: on input i, returns the i-th index to the tensor.

`replaceInput`: replace input at position i with tensor T.

`replaceIndex`: replace index at position i with new index T.

`copy`: makes a copy of the tensor.

`order`: return the derivative order with respect to $\dot{\nabla}$.

`equals`: check for exact equivalence. This means all attributes of the two objects being compared match exactly.

The `TensorAbstract` class implements two methods for `Tensor`. The `order` function defaults to returning zero. A helper method, `printIndexList`, is provided to display the latex code of an index list. This class gives a general interface for working with all possible inputs and outputs of the rewrite system.

Tensor objects, those objects with arity 0, generally have one protected property. If the object can have indexes, then it has either an array or fixed number of index positions. The following are the basic objects of the signature: `CommuatorTensor`, `CurvatureTensor`, `Normal`, `ShiftTensor`, and `SurfaceVelocity`.

There are also two special objects. The `ScalarRational` object allows for multiplication by a scalar and can perform basic rational arithmetic. The `NamedTensor` object allows the user to create a generic name and define it on either the space or the surface. The `TensorSpace` enum gives constants for `Spacial` and `Surface`.

The application of mathematical functions always results in a tensor, any tensor expression is itself a tensor. The `CDerivative` object covers the four derivatives ∇_i, ∇^i, ∇_α, and ∇^α. The constructor takes two inputs, the tensor to take the derivative of and an index denoting the type of derivative.

The `Contraction` function takes a tensor and array indexes of the two tensor indexes to contract. Addition and multiplication are handled by the `TensorSum` and `TensorProduct` classes respectively. These are representations of mathematical functions with a fixed arity of two. When the constructors are called with more then two inputs, a tree structure is built through recursive calls to the constructor. The `Exponent` takes a tensor and raises it to a rational power.

Two types of integrals are supported, both are arity one. The `IntegralSpace` is for integration in the ambient space and `IntegralSurface` is for integration on the surface.

The CMS introduces the `InvariantTimeDerivative` and `PartialInvariantTimeDerivative`. The `InvariantTimeDerivative` provides the main function for the ruleset of the CMS.

Tensor indexes are supported by the `TensorIndex` object. It takes two inputs, an `IndexFlavor` and `TensorSpace`. The `IndexFlavor` is an enum containing `Covariant` and `Contravariant`. When new indexes are generated, they are automatically named. If the number of indexes generated surpasses the default letters, spatial names ri and gk are used. The values of k and i are iterated as needed to avoid name repetitions.

This functionality allows for tensor expressions to be written using the library. To simplify an expression, a set or rewrite rules and a strategy is needed.

3.3.2 Reduction

All reduction rules must implement the `Rule` interface. The two key methods are `matches`, which takes a tensor and returns true if the rule can be applied to it, and `reduce`, which applies the rule. The interface also includes `example`, `toString`, and `name` for debugging.

The `RuleAbstract` class provides helper methods for matching. This allows a general pattern to be matched, for example matching index types but not positions.

The majority of rules are implemented in a straightforward manner. The `matches` function checks against either exact values or a general pattern. The `reduce` method copies needed objects and returns a new tensor expression with the reduced term. Inheritance is used to allow for general rule creation. All the rules in Chapter 3.2 are implemented as their own classes. Rules 3.15 - 3.17 can be easily handled by a single rule that uses a while loop to iterate over all indexes.

The rules

$$\dot{\nabla} N^i \to -Z^i_\alpha \nabla^\alpha C \tag{3.28}$$

$$\dot{\nabla} N_i \to -Z_{i\alpha} \nabla^\alpha C \tag{3.29}$$

Are implemented with a match that checks for the invariant time derivative and normal.

```
public boolean matches(Tensor T) throws Exception {
if(this.className(T,
"drexel.cs.cmstrs.signature.InvariantTimeDerivative"))
        {
                if(this.className(T.getInput(0),
                 "drexel.cs.cmstrs.signature.Normal"))
                {
                        return true;
                }
}
return false;
```

```
}
```

If the input term matches, then the reduce command get the original index attached to the normal and uses it in the new expression.

```
public Tensor reduce(Tensor T) throws Exception {
        TensorIndex i = T.getIndex(0);


        TensorIndex alpha = new TensorIndex(
                IndexFlavors.Covariant,
                TensorSpace.Surface);
        TensorIndex Alpha = new TensorIndex(
                IndexFlavors.Contravariant,
                TensorSpace.Surface);
        Tensor answer= new Contraction(
                new TensorProduct(
                        new ScalarRational(-1),
                        new ShiftTensor(i,alpha),
                        new CDerivative(
                                new SurfaceVelocity()
                        ,Alpha)
                ),1,2);
        return answer;
}
```

New objects need to be created during the reduction because an attribute, like a free index, might appear in another part of the expression where it is not affected by the current rule.

The reduction strategy is implemented in the class `Reduction`. To reduce a term, first a new `Reduction` object is created. This object has an array with one copy of each reduction rule. New rules can be added by implementing the interface and adding them to the array in `Reduction`. The reduction class has a counter for how many rules have been applied in any one pass.

A reduction pass does an inorder tree walk of the entire input expression. The reduction pass first looks at the current function symbol in the tree. If the expression matches a reduction pattern, then the `reduce` function is called to return the new tensor. This new tensor is then placed into the tree at the correct node, replacing the old tree with a new one. Once this is completed, the reduction pass is called recursively on each input to the function. This continues until all nodes have been checked against all rules.

It is possible that multiple rules may need to be applied to each node, or that new rules will only match after other reductions have taken place. After one full inorder reduction walk of the expression tree, the count of rules that matched is checked. If no rules match, completeing another tree walk will not change the result. If the number of matches was nonzero, then another inorder reduction walk is performed. This process repeats until no more matches are found. When a complete tree walk is performed with no matches found, the expression is in normal form and returned. The normal form is a sum of products. The normal formal has one additional property, all values may be evaluated at time $t = 0$. This will be the most important property in evaluation because it can be used to ensure all values need to be computed on a simple surface.

3.3.3 Combining Like Terms

It is convenient to combine equivalent terms to decrease the size of the expression. The `CombineLikeTerms` object provides the method `combine` for this purpose. The goal is to simplify expressions like

$$B^{\alpha}_{\beta}\nabla^{\beta}u + 5B^{\beta\alpha}\nabla_{\alpha}u = 6B^{\alpha}_{\beta}\nabla^{\beta}u \tag{3.30}$$

The combine method only works on tensors in the sum of products form. This is the normal form produced by reduce. The reduce command can always be applied first to put expressions in the correct form.

Given an expression in sum of products form, the first task is to create an array of all the products. Any two products in the sum may combine. The addition operator is only binary, but this comparison is easier once flattened into a single array. Once the array is created, all unique pairs of products are compared. If any two products are found to match under the following properties, then they are combined and their constant multipliers are added. Before a comparison takes place, the expressions will need to be sorted.

The rules used to match are communtativity, associativity, index juggling, and index renaming. Index juggling means that the type of any two indexes can be switched if they are contracted. This can be thought of as a seesaw motion between the two indexes. Associativity is handled for both products and sums but converting the binary functions into lists and allowing them to be reordered.

$$A^{i}_{.i} = A^{.i}_{i.} \tag{3.31}$$

Index juggling is handled by making all contracted indexes covariant for the comparison. If two expressions have no contracted indexes, this action has no effect. If there is exactly one contracted index, then the name must appear exact two times, one lower and once upper. Since the indexes can be juggled indefinitely, the comparison test can be done with both indexes lowered. If the comparison matches, then the original expressions can be made equivalent by index juggling. This can be extended to an infinite number of indexes since each contraction only links two unique indexes. All contracted indexes are lowered for the match. If a match is found, the indexes are arbitrarily placed in upper and lower positions on the combined expression. If no match is found, the indexes are returned to their original positions for the next compare.

Any two contracted indexes many be renamed as long as they remain spacial or surface indexes. To handle this property, the lists of all names for both tensors being matches are created. All possible permutations of the renaming are then generated. Each renaming is attempted. If any one renaming provides a match, then that renaming is selected as the correct one. If no renamings can lead to a match, then the two expression cannot be matched. Once a renaming is picked, the expressions need to be sorted before matching.

The final property is commutativity. After lowering all indexes and picking a renaming, communtative property is handled by sorting. The product is sorted by class name followed by index names and any special properties. Any scalar multiples are removed. If the expressions match after the sorting, then they are the same under commutativity. If they do not match then they cannot be matched.

After two matching tensors are found, their scalar multiples are added together. Both expressions are removed from the sum list and a new expression is added with the combined value. This process repeated until no new combinations are found that can be combined. The order in which terms are selected and compared in the addition does not matter. Within multiplications, sorting will remove an ordering differences between two terms.

One additional property is not implemented because of the added complexity of comparison. In tensor calculus it is legal to reorder derivatives.

$$\nabla_\alpha \nabla_\beta u = \nabla_\beta \nabla_\alpha u \tag{3.32}$$

This complexity will be discussed in the Section 3.3.5.

3.3.4 Complexity of Reduction

It is important to understand the growth rate and complexity of expressions. The rapid increase in the size of expressions causes many problems in the CMS to become intractable.

The largest growth comes from the chain rules, Equations 3.2 and 3.13. Given a product of n tensors, the chain rule creates a sum with n terms. Each term in the sum contains an additional n products. An asymptotic upper bound can be given on the size of the expression using O-notation. The classic definition of O from [21] is given below.

$$\begin{aligned} O(g(n)) = \{f(n) : &\text{ there exists positive constants } c \text{ and } n_0 \\ &\text{such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\} \end{aligned} \tag{3.33}$$

Taking the k-th derivative of an expression with the product of n terms using only the chain rules causes the number of terms to increase. A recursive formula for this growth is given.

$$T(n,k) = nT(n,k-1) \tag{3.34}$$

$$T(n,1) = n^2 \tag{3.35}$$

The closed form for this recursion is

$$T(n,k) = n^{k+1} \tag{3.36}$$

The chain rules cause the majority of the growth in expression size. The expression size can be bounded by $O\left(n^{k+1}\right)$. This means taking four derivatives of an expression with five terms leads to $3125 * c$ terms for some c. The growth rate for expressions is exponential in the number of derivatives. The number of derivatives that need to be taken is problem dependent, so this will be an important limitation. For example, in our Laplace-Eigenvalue problem described in Chapter 6 the number of terms that can be found is linearly related to the number of derivatives that can be taken.

To help with this exponential growth rate, terms can be combined to decrease the input size before the next derivative is taken. Unfortunately, this also leads to an exponential problem. When comparing possible index renamings, the worst case it to check all possible renamings. This is a less critical problem because in practice the number of indexes is small and many permutations can be easily eliminated by comparing the object types.

3.3.5 Relationship between Equivalence and Graph Isomorphism

There are two important groups of algorithms P and NP. Problems in P can be solved in polynomial time, for an input size n there exists some constant c such that the problem can be solved in $O(n^c)$ [21]. The group NP contains all problems that can be solved by a nondeterministic Turing Machine in polynomial time [21]. A nondeterministic Turing Machine is an imaginary computer that can run an infinite number of computations in parallel. For a problem in NP, at least one of these infinite machines would find a solution in polynomial time and the rest can be stopped. All problems in P are also in NP.

The hardest problems in NP are in a set called NP-complete. These problems are thought to be hard, there exists no constant c such that the problem can always be solved in $O(n^c)$. The problem of Graph Isomorphism is known to be in NP but has not be proven to be in either P or NP [97]. There exists no known polynomial time algorithm to decide Graph Isomorphism. Unfortunately, Tensor equivalence is at least as hard as Graph Isomorphism.

The Graph Isomorphism problem asks, given two Graphs G and H, does a bijection exist between the vertex sets. A bijection means there is a renaming that will make the two graphs exactly the same. The graphs shown in Figure 3.1 and Figure 3.2 show the same graph with different labels applied to the nodes. These two graphs are isomorphic.

For any graph, a tensor expression can be written. If the tensors are equivalent under associativity, commutativity, index renaming, index juggling and derivative reordering, then the graphs are isomorphic.

To create a tensor expression for graph G in Figure 3.1, create a value u for each node. The u values are temporarily given subscripts to make the connection to the graph more visible.

$$u_a u_b u_c u_d u_g u_h u_i u_j \tag{3.37}$$

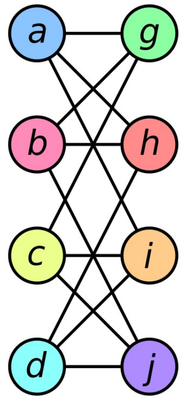

Figure 3.1: Graph G, shown above, is isomorphic to Graph H [14].

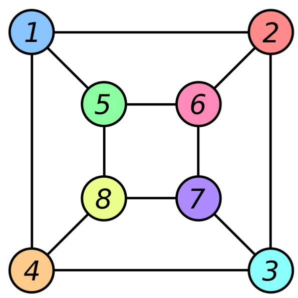

Figure 3.2: Graph H, shown above, is isomorphic to Graph G [15].

For every edge, add a covariant derivative to one side and a contravariant derivative to the other. The indexes are given with tuples to show the relationship to the edges.

$$\begin{aligned}&\nabla_{(a,g)}\nabla_{(a,h)}\nabla_{(a,i)}u_a\nabla_{(b,g)}\nabla_{(b,h)}\nabla_{(b,j)}u_b\nabla_{(c,g)}\nabla_{(c,i)}\nabla_{(c,j)}u_c\\&\nabla_{(d,h)}\nabla_{(d,i)}\nabla_{(d,j)}u_d\nabla^{(a,g)}\nabla^{(b,g)}\nabla^{(c,g)}u_g\nabla^{(a,h)}\nabla^{(b,h)}\nabla^{(d,h)}u_h\\&\nabla^{(a,i)}\nabla^{(c,i)}\nabla^{(d,i)}u_i\nabla^{(b,j)}\nabla^{(c,j)}\nabla^{(d,j)}u_j\end{aligned} \tag{3.38}$$

To make this a legal tensor expression, the subscripts on u are dropped and the indexes are given single variable names.

$$G = \nabla_a \nabla_b \nabla_c u \nabla_d \nabla_e \nabla_f u \nabla_g \nabla_h \nabla_i u \nabla_j \nabla_k \nabla_m u$$
$$\nabla^a \nabla^d \nabla^g u \nabla^b \nabla^e \nabla^j u \nabla^c \nabla^h \nabla^k u \nabla^f \nabla^i \nabla^m u \quad (3.39)$$

The same approach can be used on Graph H from Figure 3.2.

$$H = \nabla_n \nabla_o \nabla_p u \nabla^n \nabla_q \nabla_r u \nabla^r \nabla_s \nabla_t u \nabla^p \nabla^t \nabla_v u$$
$$\nabla^o \nabla_w \nabla_x u \nabla^q \nabla^x \nabla_y u \nabla^s \nabla^y \nabla_z u \nabla^v \nabla^w \nabla^z u \quad (3.40)$$

The creation of these tensor expressions can be done in linear time, each edge must be examined once. If $H = G$ under tensor equivalence then the graphs are isomorphic. This shows that tensor equivalence can be used to solve Graph Isomorphism. It is at least as hard as Graph Isomorphism. The ability to reorder derivatives is not implemented to decrease the search space in the CMSTRS library. This leads to a system that can only simplify equivalence with relation to index juggling, index renaming, commutativity, and associativity, but not complete mathematical equivalence.

3.3.6 Output Methods

Output methods are provided to generate Maple worksheets and Maple scripts. These objects are called `MapleFileGenerator` and `MapleScriptGenerator`. They generate code for the Maple evaluation library described in Chapter 4. Both these objects take a tensor expression and perform a tree walk. At each node, a definition for how to output the class needs to be given. A single inorder walk can output the entire object structure. Additional methods are given to meet the formating requirements of each type of file.

Additional output methods can be easily created by extending the `OutputMethod` interface. The interface has four methods. The `closeFile` method closes any and all files opened by the constructor. The `addEquation` method takes a tensor expression, title string, and comments string. It outputs the tensor expression is the target language with a comment section starting with the title.

The `appendRawCode` command allows raw code in the target language to be written directly to the file. The `addCode` command also writes raw code, but in a template section with a title and comment area to match the one made by `addEquation`.

3.4 Example: Contour Length

The contour length is the outer perimeter of a surface. In the case of a one dimensional surface embedded in a two dimensional ambient space, this is the classic definition of perimeter. For a two dimensional surface, it is the surface area.

The contour length can be described as the integral of 1 over the surface.

$$L = \int_S 1 dS \tag{3.41}$$

If a surface is changing shape over time, then its contour length will also be changing in time. Taking the $\dot{\nabla}$-derivative of this expression explores how this change affects the contour length. In this section, only the CMS expressions for the first three variations will be given. In Chapter 5, a specific deformation from unit circle to ellipse will be evaluated.

In Java, a new class is created that imports the TRS signature and reduction rules.

```
import drexel.cs.cmstrs.signature.*;
import drexel.cs.cmstrs.strategies.Reduction;


public class Contour1 {
    public static void main(String[] args)
    {
```

The integral is created using the signature.

```
        Tensor first_variation
                = new IntegralSurface(new ScalarRational(1));
```

The integral is printed out as latex code using `System.out.println`.

```
System.out.println("Variation 1");
System.out.println(first_variation);
```

The exact output from the system is

```
\int_{S} \left( 1 \right) dS
```

This renders as

$$\int_S \left(1 \right) dS \tag{3.42}$$

An unlimited number of derivatives can be taken. A for loop is used to generate the next derivative from the simplified form of the previous derivative. This saves on recomputing the same simplified expressions. A variable is created for the previous order found and a reduction object is created to perform simplification.

```
Tensor last_variation = first_variation;
Reduction TRS = new Reduction();
```

The for loop iterates i from 1 to 3. A try block is added to catch any exceptions thrown due to malformed expressions. As long as only the reduction rules are used, no invalid expressions can be created. Inside the loop, the $\dot{\nabla}$ operator is added one additional time to the previous result and then printed.

```
Tensor next_variation =
          new InvariantTimeDerivative(last_variation);
next_variation = TRS.reduce(next_variation);
System.out.println("Variation "+i);
System.out.println(next_variation);


last_variation = next_variation;
```

The output of the next two variations are

$$\dot{\nabla} L = \int_S (-1)\, C B^{\gamma \cdot}_{\cdot \gamma} dS \tag{3.43}$$

and

$$\dot{\nabla}^2 = L \int_S (1)\, C^2 B^{\gamma \cdot}_{\cdot \gamma} B^{g93 \cdot}_{\cdot g93} dS + \int_S (-1)\, \dot{\nabla} [C]\, B^{\gamma \cdot}_{\cdot \gamma} dS \tag{3.44}$$

$$+ \int_S (-1)\, C^2 B^{\gamma \cdot}_{\cdot g141} B^{\cdot g141}_{\gamma \cdot} dS + \int_S (-1)\, C \nabla^\gamma \nabla_\gamma C dS \tag{3.45}$$

Notice that numerous new index names were automatically created during this process.

These give geometric expressions for how the contour length is changing with respect to time. In the case of $\dot{\nabla} L$, it can be seen that the first derivative is the integral over the surface of C multiplied by the mean curvature.

These expressions give a high level geometric view on the deformation of an arbitrary surface. To create a more real world example, a specific surface deformation must be selected and the expressions must be evaluated. Evaluation is described in Chapter 4. After evaluation has been detailed, this example will be discussed further in Chapter 5. At that point, all the tools needed to evaluate these expressions will have been discussed. In Chapter 5, the change in contour length between the unit circle and an ellipse with semi-axis $A = 1$ and $B = 1 + \epsilon$ will be examined. A series solution describing the contour length on the ellipse in terms of ϵ will be found using the above expressions.

4. Evaluation of CMS Expressions

4.1 Package Overview

The CMSTensor Library is a Maple Library that implements the CMS as a collection of algebraic and array manipulations. The calculations required for any specific index in an array is performed by Maple's built-in symbolic computation algorithms. The primary functions define the basic manipulation of arrays. These are used to build the objects and advanced functions. Global information about the surface and space in which the calculations are being completed must be given before any features can be used.

The full source code is available for download from `https://www.cs.drexel.edu/SCMS`

4.2 Global Settings

The ambient space is defined by three global variables.

`ambient_coordinates` - A list with the variables used for the coordinate system.

`time_coordinate` - A variable name representing the time depended coordinate.

`ambient_mapping` - A list of functions mapping the coordinate system to the Cartesian plane.

The function `initialize_ambient_space` is used to set these globals. It takes values in the order given above. To define an ambient space in polar coordinates, the code would be

```
initialize_ambient_space([r,theta],t,[r*cos(theta),r*sin(theta)]);
```

The second global setting is for a surface restriction. If no surface is being defined, this initialization is not required. Likewise, if the time dependent coordinates are not being used, the variable set for time is not required. The surface manifold must have exactly 1 fewer coordinates then the ambient space. There are two globals set by this function.

`surface_coordinates` - A list containing the variable names defining the surface.

`ambient_to_surface` - A list of mappings from the ambient space to the surface.

To define the unit circle in the above initialized ambient space, restrict r to a fixed value.

```
initialize_surface(phi,[1,phi]);
```

Once the required globals have been initialized, the library is functional. The size of the arrays needed to define the components of a tensor are depended on the number of coordinates. Additionally, these globals are used to define the derivative operators.

4.3 CMS Object

There is only one object introduced by the library. `CMSObject` is an object that defines a tensor. The object has four local variables.

`indexes` - A list of the indexes and their position.

`coordinates` - The coordinates used by this object, either the surface or ambient from the global settings.

`components` - A multidimensional array of algebraic values.

`is_surface` - A boolean defining if the object has been restricted to the surface.

When a new `CMSObject` is constructed, only two inputs are required. The constructor must be given the indexes and components. The object is assumed to be in the ambient space unless the third argument, `restricted` is set to true. This argument defaults to `false` if not set. A spatial object can be restricted to the surface at any point.

The list of indexes is realized as a list of integer values. There are only four valid atoms that can be in this list. They are given in Table 4.1. The `components` holds the values at each point in the tensor as a multidimensional array. This array holds the actual values while the other variables store ancillary information.

Table 4.1: Possible Atoms for Index List

Index Values	
Integer Value	Meaning
1	Ambient Space index in contravariant position
-1	Ambient Space index in covariant position
2	Surface Space index in contravariant position
-2	Surface Space index in covariant position

The `CMSObject` has a small set of public methods. This object is primarily designed to be manipulated by the library functions and not directly accessed. The constructor is defined by `ModuleApply` and `ModuleCopy`. The inputs are given in the order indexes, components, then surface restriction, which defaults to false. Two other overloaded operators are `ModulePrint`, for displaying the object, and `*`, which maps to the `prod` function.

There are four accessors, `getCompts`, `getIndexes`, `isSurf`, and `getRank`. The rank of a tensor is the number of indexes, which is the length of the `indexes` list. The most interesting components of the `CMSObject` are its two mutators. Both are used to restrict the object. If the object is defined in space it is possible to call `restrictSurface`. This function loops through each element in the components and replaces the spacial coordinates with the surface coordinates using the functions defined in the global `ambient_to_surface`. The `restrictTime` function preforms a similar action on the components, but instead replaces the variable name set in `time_coordinate` with a given value. The default is to restrict the time to $t = 0$. Both these functions simplify the component values after substitution.

4.4 Primary Functions

The following functions give the basic mathematical operations that can be performed on tensors. These are independent of the deformation of the surfaces. The majority of this section is spent on multiplication and contraction. Addition and exponents are also defined. These basic tools will then be used to create objects and derivative functions.

4.4.1 Multiplication

The key component for multiplication is the `TensorProduct` function. This function defines what is means to multiply the components of two tensors A and B. The components of these two tensors are multi-dimensional arrays. Let $A[i_1, i_2, \cdots]$ access an expression in the A tensor and $B[j_1, j_2, \cdots]$ access an expression in B. Then the value in the product $C = AB$ is

$$C[i_1, i_2, \cdots, j_1, j_2, \cdots] = A[i_1, i_2, \cdots] * B[j_1, j_2, \cdots] \tag{4.1}$$

The following helper functions are needed to accomplish this task. First `array_sizes` determines the positions of the elements in each array. Next, `create_array` creates a new array to store the value of the product. To make implementation of the loops more straightforward, `array_pos` creates a list with all the access positions for a given array size. Once all the access positions are converted into flat lists, the multiplications can be completed at all positions. It is important to note that these are all intended as helper functions and do not do error checking. The user accessible function is `prod`, which does error checking before the calculations.

The `prod` function multiplies two tensors. If both tensors are spatial or surface, then the multiplication can proceed. If the product is a mix, then any spacial tensors must first be restricted to the surface. This is done using the `restrictSurface` method of `CMSObject`. The indexes of the new tensor are a concatenation of the indexes of the two input tensors. Finally, if either input is a scalar, then `ScalarProduct` is used instead of TensorProduct.

When multiplying a scalar by a tensor, the size of the tensor does not change. The new tensor has the same size as the old one and each component position is multiplied by the scalar. This is done by `ScalarProduct`. A scalar appears as an array with a single element, but this dimension is not added to the final tensor. The array storing the scalar value is an artifact of the implementation and not the same as the meaningful dimensions for non-scalar tensors.

4.4.2 Contraction

The public interface to contraction is the `contract` function. This function takes a `CMSObject` and a list of index positions. Each pair of index positions in the list will be contracted. The new tensor's index list will be the original list with all contracted indexes removed. The number of indexes to be contracted must be even and cannot contain duplicates. This function takes each pair of indexes to contract on and calls `contractij`. After the two indexes are contracted, any remaining index positions are shifted to account for the newly removed indexes. Index positions are counted starting at one.

The `contractij` function contracts two indexes of a tensor, at positions i and j. First, additional error checking is performed. The indexes must be different values. One index must be covariant and the other contravariant. The indexes must both be either spatial or surface. Lastly, the array dimensions must match for both indexes.

When it is confirmed the contraction is possible, the array must be permuted. Let $a = \min(i, j)$ and $b = \max(i, j)$, wherever the indexes appear in the original tensor, `permute_indices` is used to reorder the indexes so that a is first and b is second. Next, create a new zero-filled array that is two less then the original. Let A be the tensor being contracted and B be the new tensor, then for all i compute

$$B[k_1, k_2, \cdots] = B[k_1, k_2, \cdots] + A[i, i, k_1, k_2, \cdots] \tag{4.2}$$

In the event that there are no free indexes remaining, the tensor's components must be flattened one additional time because the array will have an extra level. This happens because a tensor with one index and a scalar appear the same using an array representation. The component attribute of the object is always an array, this is allows for a consistent interface. A one-index, one-dimensional array is an array with only one position. This is also how a constant is stored. Most scalar arithmetic works using this representation. In this case, it adds an extra dimension that appears is not be present mathematically.

4.4.3 Addition

The `lin_com` function adds two tensors. The indexes and array sizes on both tensors must match exactly. Additionally, both tensors must be spatial or surface. If only one is on the surface, then the spatial tensor is restricted to the surface before addition takes place. The `array_pos` function is again used to flatten the multi-dimensions array access into a single loop where

$$C[k_1, k_2, \cdots] = A[k_1, k_2, \cdots] + B[k_1, k_2, \cdots] \tag{4.3}$$

The indexes of the sum are the same as the indexes of either input term. A local function, `permute_indices` may be used to permute the array to add non-matching index dimensions. This is only used as a helper function in some methods. For standard addition using + or `lin_com` it is required that the indexes match.

4.4.4 Additional Methods

The `exponent` function takes a tensor and an integer. It repeatedly calls the prod function. The `inverse` function is used to invert a Tensor. It is only possible to invert a scalar or 2 by 2 array. Inverting a 2 by 2 array is exactly the same as the method for a matrix.

4.5 Object Constructors

Construction of the objects of the CMS starts with the covariant basis. First, a `CMSObject` R is created with one contravariant surface index and the value of `ambient_mapping` as its components. Next, the partial derivative is computed with respect to the coordinate space.

$$Z_i^{\wp} = \frac{\partial}{\partial Z_i} R^{\wp} \tag{4.4}$$

This is done using the `ddZiPartial`. Note that this object has two indexes, but should really be a vector. This is a artifact of the `CMSObject` structure. This is only required for handling the basis vectors. This allows it to be used directly with the entire library. The special symbol $\wp$ is used to show this is not truly a spacial or surface index. This value is not a tensor, the first two tensors are created next. The $Z_i^{\wp}$ object is just used to get from a vector to a tensor. It should not be used for later calculations because it is not a tensor.

The `covariant_metric`, Z_{ij} and `contravariant_metric`, Z^{ij} can now be defined. Z^{ij} is the inverse of Z_{ij}.

$$Z_{ij} = Z_i^{\wp} Z_{j\wp} \tag{4.5}$$

In the library this math is defined as

```
contravariant_basis:=proc()
        return contract(
                prod(contravariant_metric(),
                covariant_basis())
                ,[2,3]);
end proc;
```

New objects can continue to be built up from their definitions using these tools. The `contravariant_basis`, $Z^{i\wp}$, is given as

$$Z^{i\wp} = Z^{ij} Z_j^{\wp} \tag{4.6}$$

The metric tensors define the space, next the metrics for the surface are defined. To do this, the shift tensor, Z_i^{α} must be created. The first index must be a space index and the second must be a surface index, but a shift tensor is defined for each combination of positions. The function `shift_tensor` takes two inputs to denote the desired indexes and returns the appropriate object.

Creation of the first shift tensor, Z_α^i requires creating a new R_{amb} `CMSObject` using the global `ambient_to_surface`. The function first creates Z_α^i using the code below

```
R_cms := CMSObject([1],
                Array(ambient_to_surface),true);
ZIa:=permute_indices(ddSaPartial(R_cms),[2,1]);
```

The remaining three variations can be easily generated.

$$Z_{i\alpha} = Z_{ij} Z^i_\alpha \tag{4.7}$$

$$Z^{i\alpha} = Z^i_\alpha S^{\alpha\beta} \tag{4.8}$$

$$Z^\alpha_i = Z_{i\alpha} S^{\alpha\beta} \tag{4.9}$$

$S^{\alpha\beta}$ is the `contravariant_surface_metric`, which is the inverse of $S_{\alpha\beta}$, the `covariant_surface_metric`. The shift tensor is used to compute the covariant surface metric.

$$S_{\alpha\beta} = Z^i_\alpha Z_{i\beta} \tag{4.10}$$

Taking the partial derivative of a Tensor does not always yield another tensor. The Christoffel symbol is used to ensure that the covariant derivative produces a tensor. The Christoffel symbol is defined both in space, `christoffel_space`, and on the surface, `christoffel_surface`. When computing the value in space, the index of $\frac{\partial Z^\wp_i}{\partial S_\alpha}$ must be pivoted with `pivot_index`. This is not a true index lowering since $Z^\wp_i$ is not a true index. This is a mechanical requirement to keep these computations in line with library. Mathematically, there are no tensors until Z_{ij} created, but building it requires all the same algorithms as tensor calculations.

```
christoffel_space:=proc()
        return contract(prod(
                contravariant_basis(),
                pivot_index(ddZiPartial(covariant_basis()),3))
                ,[2,5]);
end proc;
```

The surface Christoffel is computed purely algebraically.

$$\Gamma^\alpha_{\beta\gamma} = Z^\alpha_i \frac{\partial Z^I_\beta}{\partial S^\gamma} + \Gamma^I_{jk} Z^\alpha_i Z^j_\beta Z^k_\gamma \tag{4.11}$$

To create the Levi-Civita Tensor, first create a `LeviCivitaMatrix`. This function takes one input with the target index position. All indexes will be of the same type. It then creates a square array with the correct number of dimensions. For each position $\epsilon[i_1, i_2, \cdots]$ the value is either the sign of the permutation of $1, 2, \cdots, n$ related to $i_1, i_2, \cdots$ or zero. The `LeviCivita` tensor is computed by multiplying the array by the determinant.

$$E^{i,j,\cdots} = \frac{1}{\sqrt{\det(Z_{ij})}}\epsilon^{i,j,\cdots} \tag{4.12}$$

$$E_{i,j,\cdots} = \sqrt{\det(Z_{ij})}\epsilon_{i,j,\cdots} \tag{4.13}$$

$$E^{\alpha,\beta,\cdots} = \frac{1}{\sqrt{\det(S_{ij})}}\epsilon^{\alpha,\beta,\cdots} \tag{4.14}$$

$$E_{\alpha,\beta,\cdots} = \sqrt{\det(S_{ij})}\epsilon_{\alpha,\beta,\cdots} \tag{4.15}$$

The determinant of a tensor is found using `determinant`. This is only valid on 1x1 and 2x2 dimensional arrays.

The surface curvature is given by the curvature tensor $B_{\alpha\beta}$. It is created using the function `curvature_tensor` which takes two indexes determining the index positions. These must be surface indexes.

$$B_{\alpha\beta} = -Z^i_\alpha \nabla_\beta N_i \tag{4.16}$$

$$B^\alpha_\beta = S^{\alpha\gamma} B_{\gamma\beta} \tag{4.17}$$

$$B^{\alpha\beta} = S^{\alpha\gamma} B^\beta_\gamma \tag{4.18}$$

$$B^\beta_\alpha = B_{\alpha\gamma} S^{\gamma\beta} \tag{4.19}$$

Creation of the surface normal depends on the number of dimensions. It is only defined for one and two dimensions. The surface normals definition is dependent on the number of dimensions, for higher orders it needs to be solved for. The general approach to finding the normal can be found in [45]. For a one dimensional surface, it is defined as

$$N^i = E^{i,j} E_\alpha Z^\alpha_j \tag{4.20}$$

$$N_i = Z_{i,j} N^j \tag{4.21}$$

For a two dimensional manifold, the definitions are

$$N_i = \frac{1}{2} E_{ijk} E^{\alpha\beta} Z^i_\alpha Z^j_\beta \tag{4.22}$$

$$N^i = Z^{ij} N_j \tag{4.23}$$

Scalars are created using the `CMSScalar` function. The remaining functions are related to deforming manifolds.

The `contravariant_space_velocity` is created by taking the `ddtPartial` derivative of `ambient_to_surface`. This is represented by v^i. The related tensors `covariant_space_velocity`, `contravariant_surface_velocity`, and `covariant_surface_velocity` are defined by application of the shift or metric tensor.

$$v_i = Z_{ij} v^j \tag{4.24}$$

$$v^\alpha = Z_{i\alpha} v^i \tag{4.25}$$

$$v_\alpha = Z^\alpha_i v^i \tag{4.26}$$

This leads to the definition of `surface_velocity`.

$$C = v_i N^i \tag{4.27}$$

The last object constructor function is the Grinfeld Commutor, which simplifies the relationship between $\dot{\nabla}$ and ∇_α. This function takes three surface index positions. Indexes are permuted using `permute_indices` to make both sides match. The left side is shown before permutation is used to line the indexes up for addition.

$$R^{\alpha}_{\beta\gamma} = \nabla^{\alpha}(CB_{\beta\gamma}) - \nabla_{\beta}(CB^{\alpha}_{\gamma}) \tag{4.28}$$

$$R^{\alpha\gamma}_{\beta} = S^{\gamma\zeta} R^{\alpha}_{\beta\zeta} \tag{4.29}$$

$$R^{\alpha\beta}_{\gamma} = S^{\beta\zeta} R^{\alpha}_{\zeta\gamma} \tag{4.30}$$

$$R^{\gamma}_{\alpha\beta} = \nabla^{\gamma}(CB_{\beta\alpha}) - \nabla_{\beta}(CB^{\gamma}_{\alpha}) \tag{4.31}$$

$$R^{\beta}_{\alpha\gamma} = \nabla^{\beta}(CB_{\alpha\gamma}) - \nabla_{\gamma}(CB^{\beta}_{\alpha}) \tag{4.32}$$

Index positions not defined above are not supported by the constructor. They can be created by contraction with the surface metric.

The remaining functions define how derivatives and integrals are taken.

Permuations are done using `permute_indices`. This function takes two inputs, a tensor and a list of integers. The list of integers gives the reordering. This function is based on the Maple function of the same name [76]. The n-th position of the original tensor is moved to position n in the input list. For example, $[2, 3, 1]$ means place old index 2 into new position 1, old index 3 into new position 2, and old index 1 into new position 3.

In the above equations, this is represented by $S^{\gamma\zeta} R^{\alpha}_{\beta\zeta}$ having γ in a position that does not match the right side of the equation. Note that contracted indexes are not accounted for, the contraction will take place before the permutation.

4.6 Advanced Functions

Derivatives of elements in a tensor are taken with the `ddZiPartial`, `ddSaPartial`, and `ddtPartial` functions. The `ddZiPartial` function adds a new dimension to the array, with a covariant spacial index. For each dimension on the space, the derivative is take with respect to the dimension. Given an array with two positions, the array looks like

$$\begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix} \tag{4.33}$$

Taking the derivative of this array with respect to a two dimensions space, $\{x, y\}$, results in the array

$$\begin{pmatrix} \frac{df(x,y)}{dx} & \frac{df(x,y)}{dy} \\ \frac{dg(x,y)}{dx} & \frac{dg(x,y)}{dy} \end{pmatrix} \tag{4.34}$$

The partial surface derivative, `ddSaPartial`, does the same thing but on the surface coordinates. There is only one time coordinate, which means `ddtPartial` does not add a new index. For all these functions, when applied to a scalar one extra array dimension is added and must be removed. This is because scalars are stored as arrays with one element.

The covariant derivative, ∇_i, is given by `ddZi`. If the tensor this function is applied to has no indexes then

$$\nabla_i T = \frac{\partial}{\partial Z_i} T \tag{4.35}$$

For every index, an adjustment must be made. This is shown by example for two indexes.

$$\nabla_i T_j^k = \frac{\partial}{\partial Z_i} T_j^k - \Gamma_{ij}^m T_m^k + \Gamma_{im}^k T_j^m \tag{4.36}$$

The `ddZi` function has a local helper function adj to handle creating this sum. Given a tensor T with dim dimensions, it will give the adjustment on index n. The `ddZi` function loops through all indexes. The `adj` function for `ddZi` is given below.

```
adj:=proc(T,n,dim)
          local tp,swap,ind_char, ChristoffelIjk;
          ChristoffelIjk:=christoffel_space();
          ind_char := getIndexes(T);
          if ind_char[n]=-1 then
                    return prod(CMSScalar(-1),
                            permute_indices(
                            contract(prod(
     ChristoffelIjk,T),
     [1,n+getRank(ChristoffelIjk)]),
                            [1,`$`(3 .. n+1),2,`$`(n+2 .. dim+1)])
          );
```

```
elif ind_char[n]=1 then
         return permute_indices(
                  contract(
                           prod(ChristoffelIjk,T)
                  ,[3,n+getRank(ChristoffelIjk)])
                  ,
         [2,`$`(3 .. n+1),1,`$`(n+2 .. dim+1)]);
else
         error sprintf(
"ddZi on index with character \%a"
,ind_char[n]);
end if;
end proc;
```

The contravariant version `ddZI` is created using contraction with the metric.

$$\nabla^i T^{\cdots}_{\cdots} = Z^{ji} \nabla_j T^{\cdots}_{\cdots} \tag{4.37}$$

The surface versions of both functions follow the same pattern. Their are now four possible index types for the input.

$$\nabla_\alpha T^{\beta i}_{\gamma j} = \frac{\partial T^{\beta i}_{\gamma j}}{\partial S_\alpha} + \Gamma^{\beta}_{\alpha\zeta} T^{\zeta i}_{\gamma j} - \Gamma^{\zeta}_{\alpha\gamma} T^{\beta i}_{\zeta j} + Z^n_\alpha \Gamma^i_{nm} T^{\beta m}_{\gamma j} - Z^n_\alpha \Gamma^m_{nj} T^{\beta i}_{\gamma m} \tag{4.38}$$

$$\nabla^\alpha T^{\cdots}_{\cdots} = S^{\alpha\beta} \nabla_\beta T^{\cdots}_{\cdots} \tag{4.39}$$

The ddt function follows a similar pattern. Again, the index possibilities are shown by example.

$$\begin{aligned}\dot{\nabla} T^{\alpha i}_{\beta j} = &\frac{\partial T^{\alpha i}_{\beta j}}{\partial t} + \left(\nabla_\alpha (v^\zeta) - C B^\zeta_\alpha\right) T^{\zeta i}_{\beta j} - \left(\nabla_\beta (v^\zeta) - C B^\zeta_\beta\right) T^{\alpha i}_{\zeta j} \\ &+ v^n \Gamma^i_{nm} T^{\alpha m}_{\beta j} - v^n \Gamma^m_{nj} T^{\alpha i}_{\beta m}\end{aligned} \tag{4.40}$$

The `integrate` function is just a shell for Maple's integrate command. The function accepts scalars, tensors with no indexes, and a list of variables with ranges to integrate over. It calls the built-in integrate function with the input ranges as given. If the integral needs any special treatment, for example multiplication by a variable, this will not be handled automatically. The `integrate` function does not take the manifold into account.

4.7 Example: Poisson's Equation

A brief example from [10] is now given. The key components of evaluation will be discussed. For a more complete explanation see [10].

This example analyzes Poisson's equation under arbitrary smooth deformations of the domain. The function u where $\triangle u = 1$ on the regular N-sided polygon will be examined. The Poisson energy E_N will be computed as a partial series. This series explores the asymptotic behavior of E_N as $N \to \infty$.

To use the library, it must be included into the Maple worksheet. In this example, the library is in a folder CMSTensors in the same directory as the Maple worksheet. The library functions are imported to the namespace using `with`.

```
restart;
libname := "./CMSTensors", libname;
with(CMSTensors);
```

The surface deformation for this problem is from the unit circle to a regular N-sided polygon. In this case it is easiest to use polar coordinates. The ambient space is defined as

```
initialize_ambient_space(
        [r,theta],t,
        [r*cos(theta),r*sin(theta)]):
```

The surface restriction only has one variable, ψ, for the angle. The radius is computed based on the angle and time, t.

```
initialize_surface([psi],[
        1-t*(1-cos(Pi/N)/cos(psi))
        ,psi]):
```

The solution to $\triangle u = 1$ on the unit circle under Dirichlet boundary conditions is

$$u(r,\theta) = \frac{1}{4}\left(r^2 - 1\right) \tag{4.41}$$

This is defined in the library as

```
u:=CMSObject([], Array([1/4*(r^2-1)]));
```

The first term in the series for E_N, is the solution on the unit circle to

$$E_0 = \int_\Omega \left(\frac{1}{2}\nabla_i u \nabla^i u + u\right) d\Omega \tag{4.42}$$

The integrand is computed as

```
E0_integrand:=lin_com(
        contract(CMSScalar(1/2)*ddZi(u)*ddZI(u),[1,2])
        ,u
        );
```

To integrate $f(r,\theta)$ over Ω, the expression expands to

$$\int_\Omega f(r,\theta) d\Omega = \int_0^1 \int_{-\pi}^{\pi} r f(r,\theta) d\theta dr \tag{4.43}$$

This is done in two steps, first multiplication by r then integration.

```
E0_integrand:=CMSScalar(r)*E0_integrand;
E0:=integrate(E0_integrand,[r=0..1,theta=-Pi..Pi]);
```

The value for the first term in the partial series is stored in a list.

```
energy[0]:=getCompts(E0)[1];
```

The next term is computed as

$$E_1 = -\frac{1}{2}\int_S C \nabla_i u \nabla^i u dS \tag{4.44}$$

The surface velocity, C, is computed. The expression is evaluated at $t = 0$. The `restrictTime` function is called to simplify C.

```
C0:=surface_velocity();
C0:=restrictTime(C0);
```

The integrand can be computed easily.

```
E1_integrand:=contract(
        CMSScalar(-1/2)*C0*ddZi(u)*ddZI(u)
,[1,2]);
```

Integration becomes more difficult with this term. The value of C is defined from $-Pi/N$ to Pi/N and repeated N times. Integration of $C = -1 + \cos(\pi/N)/\cos\theta$ is also difficult.

First, the integrand is evaluated at $t = 0$ and removed from the `CMSObject` structure.

```
E1_integrand:=getCompts(restrictTime(E1_integrand))[1];
```

This sets `E1_integrand` to be

$$f(\psi) = -\frac{1}{8}\frac{-\cos(\psi) + \cos\left(\frac{\pi}{N}\right)}{\cos(\psi)} \tag{4.45}$$

The target integral is

$$\int_S f(\psi)d\psi = N\int_{-\pi/N}^{\pi/N} f(\psi)d\psi \tag{4.46}$$

This is simplified with a change of variables.

```
E1_integrand:=eval(E1_integrand,psi=theta/N);
```

The expression is converted into a series at $N = \infty$ for integration.

```
E1_integrand:=convert(expand(
        series(E1_integrand,N=infinity)
),`polynom`);
```

A much simpler integral is now taken.

```
energy[1]:=expand(
        int(E1_integrand,theta=-Pi..Pi)
);
```

This results in the second term in the series

$$\frac{1}{12}\frac{\pi^3}{N^2} \tag{4.47}$$

Computing the second variation requires dealing with infinite series. This cannot be handled by

Maple automatically. The CMSTensor library relies on Maple for algebraic simplification. Continuing the series would require specialized simplification code for the algebraic expressions.

The first two terms in the series are

$$E_N = -\frac{\pi}{16} + \frac{\pi\zeta(2)}{2N^2} + \cdots \tag{4.48}$$

This shows how a general CMS expression can be evaluated on a specific coordinate system. Maple is used to evaluate Poisson's equation. The CMS expression are true any deforming surface, but we evaluate a specific realization of the problem.

5. Example: Contour Length

The SCMS combines both the CMSTRS and Maple CMSTensor library. A problem can be simplified and evaluated by using both libraries together. This problem can also be easily calculated using alternative means. This allows the answer to be easily confirmed.

Consider the contour length of an ellipse with semi-axis $1+\epsilon$ and 1. Find a series for the contour length in terms of ϵ. The approach to this problem is to consider a smooth evolution of the boundary from the unit circle at $t=0$ to the ellipse at $t=1$. The evolution is shown in Figure 5.1.

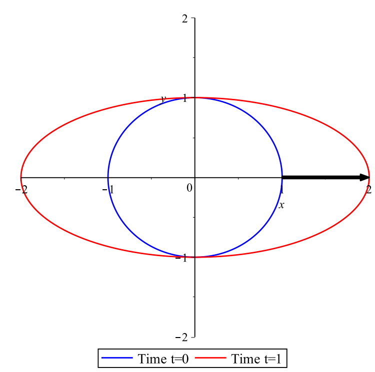

Figure 5.1: The unit circle being stretched into an ellipse.

The evolution is parameterized as

$$x(t,\theta) = (1+\epsilon t)\cos\theta \tag{5.1}$$

$$y(t,\theta) = \sin\theta \tag{5.2}$$

The contour length at time t is denoted by

$$L(t) = \int_{S(t)} 1 dS \tag{5.3}$$

To find the contour length at $t=1$, use a Taylor series. The series is in terms of the derivatives of L, $L^n = \dot{\nabla}^n L$.

$$L(1) = L(0) + L^1(0) + \frac{1}{2!}L^2(0) + \frac{1}{3!}L^3(0) + \cdots \tag{5.4}$$

The first term in the series is trivial,

$$L(0) = \int_{S(t=0)} 1dS = \int_0^{2\pi} 1d\theta = 2\pi \tag{5.5}$$

The first derivative is found by applying the $\dot{\nabla}$-derivative. This is simplified by the TRS.

$$L'(t) = \dot{\nabla} \int_S 1dS \tag{5.6}$$

$$= \int_S \left(\dot{\nabla}(1) - CB^\alpha_\alpha\right) dS \tag{5.7}$$

$$= -\int_S CB^\alpha_\alpha dS \tag{5.8}$$

To evaluate this expression, the coordinate system is defined. The code below initializes the coordinate system from equations 5.1 and 5.2.

```
initialize_ambient_space([x,y],t,[x,y]):
initialize_surface([theta],[(1+epsilon*t)*cos(theta),sin(theta)]):
```

Next, the needed tensors are generated for the coordinate system.

```
C0:=surface_velocity():
BAb:=curvature_tensor(2,-2):
```

Expression 5.8 is evaluated.

```
SurfIntegral(contract(prod(CMSScalar(-1),prod(C0 ,BAb)),[1,2])):
```

This returns $\epsilon\pi$.

Since this is a simple expression, It can also be evaluated directly. The value of the tensors are

$$C|_{t=0} = \epsilon\cos^2\theta \tag{5.9}$$

$$B^\alpha_\alpha|_{t=0} = -1 \tag{5.10}$$

Plugging in this values computes the same answer.

$$L'(t=0) = -\int_S CB_\alpha^\alpha dS|_{t=0} \tag{5.11}$$
$$= \int_S CdS \tag{5.12}$$
$$= \int_0^{2\pi} \epsilon \cos^2\theta d\theta \tag{5.13}$$
$$= \epsilon\pi \tag{5.14}$$

To find the next value, $L^2(t=0)$, take an additional $\dot{\nabla}$-derivative. The general pattern for these expressions is given. It is split up into the integrand and integral. The majority of the simplification takes place in the integrand.

$$M_n = \dot{\nabla}(M_{n-1}) - CB_\alpha^\alpha M_{n-1} \tag{5.15}$$
$$L^n(0) = \int_S M_n|_{t=0} dS \tag{5.16}$$

With M_1 known, M_2 and L^2 can be determined.

$$M_1 = -CB_\alpha^\alpha \tag{5.17}$$
$$M_2 = -\dot{\nabla}(CB_\alpha^\alpha) + C^2 B_\alpha^\alpha B_\beta^\beta \tag{5.18}$$

The TRS reduces this expression to a normal form. The total reduction takes 31 rewrites including structure changes like rule (3.12) and simple reductions like rule (3.25). Some key reductions are highlighted below. Subscripts are attached to the arrow symbol referencing the rule list in Section 3.2.

$$\begin{aligned} -\dot{\nabla}(CB_\alpha^\alpha) + C^2 B_\alpha^\alpha B_\beta^\beta &\to_{3.13} -B_\alpha^\alpha \dot{\nabla} C - C\dot{\nabla} B_\alpha^\alpha + C^2 B_\alpha^\alpha B_\beta^\beta \\ &\to_{3.8} -B_\alpha^\alpha \dot{\nabla} C - C(\nabla^\alpha \nabla_\alpha C + CB_\gamma^\alpha B_\alpha^\gamma) + C^2 B_\alpha^\alpha B_\beta^\beta \\ &\to_{3.26} -B_\alpha^\alpha \dot{\nabla} C - C\nabla^\alpha \nabla_\alpha C - C^2 B_\gamma^\alpha B_\alpha^\gamma + C^2 B_\alpha^\alpha B_\beta^\beta \end{aligned} \tag{5.19}$$

The normal form for M_2 is given by equation (5.19). This expression is true for the contour length of any deforming manifold. Code is generated to evaluate the expression for this realization of the problem.

```
tempsum:=CMSScalar(0):

Term1 := SurfIntegral(contract(prod( CMSScalar(1/1),
         prod(exponent(C0, 2), prod(BAb, BAb))),[ 1, 2, 3, 4]) ) :
Term1 := restrictTime(Term1):
tempsum:=lin_com(tempsum,Term1):

Term2 := SurfIntegral(contract(prod(CMSScalar(-1/1),
         prod(C1, BAb)),[ 1, 2]) ) :
Term2 := restrictTime(Term2):
tempsum:=lin_com(tempsum,Term2):

Term3 := SurfIntegral(contract(prod(CMSScalar(-1/1),
         prod(exponent(C0, 2), prod(BAb, BaB))),[ 2, 4, 1, 3]) ) :
Term3 := restrictTime(Term3):
tempsum:=lin_com(tempsum,Term3):

Term4 := SurfIntegral(contract(prod(CMSScalar(-1/1),
         prod(C0, ddSA(ddSa(C0)))),[ 1, 2]) ) :
Term4 := restrictTime(Term4):
tempsum:=lin_com(tempsum,Term4):
```

The final value is determined to be $\frac{1}{4}\epsilon^2\pi$.

$$M_2|_{t=0} = \epsilon^2(7\cos^2\theta - 5)\cos^2\theta \tag{5.20}$$

$$L^2(0) = \int_0^{2\pi} (M_2|_{t=0})\, d\theta = \frac{1}{4}\epsilon^2\pi \tag{5.21}$$

The TRS repeats this process and determines the normal form of M_3 which requires 118 rewrites.

In addition, terms are combined to shorten the expression.

$$
\begin{aligned}
M_3 = &- C^3 B^\alpha_\alpha B^\beta_\beta B^\gamma_\gamma + 3C^3 B^\alpha_\beta B^\beta_\alpha B^\gamma_\gamma - 2C^3 B^\alpha_\beta B^\gamma_\alpha B^\beta_\gamma \\
&+ 3C^2 B^\alpha_\alpha \nabla^\beta \nabla_\beta C - 4C^2 B^{\alpha\beta} \nabla_\beta \nabla_\alpha C \\
&+ 3C\dot{\nabla}(C) B^\alpha_\alpha B^\beta_\beta - 3C\dot{\nabla}(C) B^\alpha_\beta B^\beta_\alpha - 2\dot{\nabla}(C) \nabla^\alpha \nabla_\alpha C \\
&- \dot{\nabla}^2(C) B^\alpha_\alpha - C\nabla^\alpha \nabla_\alpha \dot{\nabla}(C) + CR^{\alpha\beta}_{\beta} \nabla_\alpha C
\end{aligned} \tag{5.22}
$$

Equation (5.22) already shows the rapid growth of expressions in the CMS. The challenge of calculating M_4 without an automated system is obvious. M_4 is the sum of 94 products and requires 595 rewrites. An important feature of the CMS remains in M_3, this expression is valid for any surface deformation. The library can easily determine that $L^3(0) = -\frac{-3}{8}\epsilon^3\pi$ and $L^4(0) = \frac{51}{64}\pi\epsilon^3$.

The Taylor series is now created.

$$
\frac{1}{2!}L^2(0) = \frac{1}{2}\left(\frac{1}{4}\epsilon^2\pi\right) = \frac{1}{8}\epsilon^2\pi \tag{5.23}
$$

$$
\frac{1}{3!}L^3(0) = \frac{1}{6}\left(-\frac{3}{8}\pi\epsilon^3\right) = -\frac{1}{16}\epsilon^3\pi \tag{5.24}
$$

$$
\frac{1}{4!}L^4(0) = \frac{1}{24}\left(\frac{51}{64}\pi\epsilon^3\right) = \frac{17}{512}\epsilon^4\pi \tag{5.25}
$$

This is compared with a series computed using an entirely different method. This is used to verify the results.

$$
\begin{aligned}
L(1) &= \int_0^{2\pi} \left(\sqrt{(1+\epsilon)^2\cos^2(\theta) + \sin^2(\theta)}\right) d\theta \\
&= \left(2 + \epsilon + \frac{1}{8}\epsilon^2 - \frac{1}{16}\epsilon^3 + \frac{17}{512}\epsilon^4 - \frac{19}{1024}\epsilon^5 + \cdots\right)\pi
\end{aligned} \tag{5.26}
$$

The library has been verified and correctly generates the series up for L^7.

6. Laplace-Dirichlet Eigenvalues

6.1 Introduction

The study of eigenvalues touches a wide range of fields. Most people's first interaction with the eigenvalue comes from linear algebra. In this context, there exists some matrix A. When multiplying a vector x by the matrix, some vectors will change direction but a rare few will not. These vectors are eigenvectors. In these cases, there exists some number λ, called the eigenvalue such that

$$Ax = \lambda x \tag{6.1}$$

Although this is the first version of eigenvalues seen by many students, it is not the only version of the concept. A more detailed examination of eigenvalues in linear algebra can be found in [103].

The foundations of eigenvalues can be found in the structural mechanics of the 18th century. One key example is Euler's 1751 work *Du mouvement dun corps solide quelconque lorsquil tourne autour dun axe mobile* [31]. In this paper, Euler examines the problem of rotating a rigid structure. Any rotation of a rigid body such that some point remains fixed is equivalent to a rotation around the fixed point. This work would be continued by Joseph-Louis Lagrange and Augustin-Louis Cauchy in their study of celestial bodies [54]. These works would move the study of eigenvalues toward more traditional linear algebra concepts. This would lead to the field of spectral theory when David Hilbert began his study of operators and spaces [53].

The Laplacian eigenvalue also came out of this work. In this case, there is an eigenvalue, λ, and an eigenfunction, u, such that taking the Laplacian, $\triangle$, the function only changes by scalar multiplication. The Laplacian operator is a second order differential operator defined on the manifold [85].

$$\triangle u = \lambda u \tag{6.2}$$

This classical eigenvalue problem is still open on many bounded domains. In 1877, Lord Rayleigh examined eigenvalues on a drum and their relationship to the sounds produced by the drum [92]. By minimization of this eigenvalue, Lord Rayleigh proposed that *among all drums of a given area, the circular drum is the one which produces the deepest bass note* [2]. It would take almost 30 years before this conjecture was proven [2]. This result is the basis for the solution on the unit circle.

For these eigenvalues, the boundary of the domain is under the Dirichlet boundary condition. This condition means that the eigenfunction u must be equal to zero on the boundary. This is one of three common boundary conditions, along with Neumann and Robin conditions [36].

The Laplacian eigenvalues appear in numerous fields outside of acoustics including electron wave functions, the theory of diffusion, and study of dynamic systems [36].

Eigenvalues are one of the most successful tools in applied mathematics [107]. A broad range of topics where eigenvalues are used in presented in [107]. Some of the applications are

- acoustics
- ecology
- fluid mechanics
- Markov chains
- partial differential equations
- quantum mechanics
- structural analysis
- functional analysis
- physics of music
- vibration analysis

6.2 Eigenvalues on the Polygon

One question of current interest in the field of Laplacian eigenvalues was proposed in 2004, *what is the series in* $1/N$ *for the simple Laplace eigenvalues on the* N *sided regular polygon under Dirichlet boundary conditions* [46]?

In this paper, the spectrum analysis of a regular polygon is proposed. At the time, the closed form was known for only the square and three special triangles [46]. This attempt uses two approaches, the finite element method and a Taylor series approach. The Taylor series approach will be the one implemented in the SCMS.

The finite element method is shown to have difficulties in this problem space. The size of mesh needed to create an estimate grows quadratically with respect to the number of nodes and the number of nodes is strictly defined in terms of the number of edges on the polygon. Additionally the error grows rapidly and cannot be improved by series acceleration tools [46]. Finally, this approach is applicable for fixed values of N and does not lead to a general solution. Recent work has been done by Robert Jones improving these numerical approximations [58].

The Taylor series approach provides an algebraic solution that is shown to be more accurate than the finite method approach. It has been used to find numerical estimates for the first 10 simple eigenvalues on the regular 128-sided polygon [46].

Although this question is interesting in and of itself, it has a number of applications. An algebraic series in terms of the number of sides N gives all eigenvalues for all possible $N > 2$. These can be used for both numerical and exact computations. The relationship between eigenvalues and shapes is a rich and important field [36].

Many approaches to finite elements and the level set method approximate original domains by replacing them with polygonal meshes. The error created by the replacement of smooth boundaries with polygonal boundaries can be further examined through study of these eigenvalues [46].

The results of [46] drew interest in a number of fields. It provided new insight into the study of eigenvalues on general 2 dimensional domains [50]. The study of quantum billiards investigates the movement of particles bouncing in a bounded domain [36]. The eigenvalues on the polygon can be applied to quantum systems with polygonal domains as well as their spectroscopy [4]. Returning to the relationship between shapes and eigenvalues, knowledge of the eigenvalues on the polygon can be used to improve shape recognition techniques [64].

The next significant advance was an exact series up to N^{-4} which was presented in [47]. This solution no longer required numerical approximation. The truncated series is now given for the general N and for any order eigenvalue. An early version of the CMSTRS presented here was used to find and correct errors in these hand calculations. The results promoted more investigation [9, 68]. The results were immediately found useful in a variety of fields.

The approach presented in [47] is extended and generalized below. The methods described are recursive and algorithmic. Given the starting cases, the series can be theoretically extended to any number of terms. The approach shows both that the presented system works, but also that it improves on previous research. This approach is only limited by the algebraic simplification tools and computational resources.

In the remainder of this chapter, the mathematics of computing eigenvalues will be examined. The solution on the unit circle is used as both the basis for boundary variation and an example domain. The first term in the series on the regular N sided polygon is found. Additionally, the first term on the ellipse with semi-axis $A = 1$ and $B = 1 + \epsilon$ is found. The second deformation uses the same CMS expressions, since these expressions are true for all coordinate systems, but can be more easily numerically approximated. This problem provides both interesting results and an additional method of error checking.

6.3 Solution on Unit Circle

The eigenvalues on the unit circle can be visualized using an ideal drum. A drum, such as the one in Figure 6.1, is a circle with a rigid frame. When any point on the drum is hit, the drum vibrates. These vibrations stop when they reach the edge of the drum. The different vibrations of the drum are related to the sounds that can be produced. The eigenvalues of the surface will determine these vibrations, and therefore sounds, that the drum can produce. More about the relationship between eigenvalues and sounds produced by drums can be found in [62], [90], [113], and [35].

Figure 6.1: Drum used by 40th Regiment New York Veteran Volunteer Infantry Mozart Regiment

The idealized drum is a circle of radius 1 centered at $(0, 0)$. The eigenfunction $u(r, \theta)$ and the eigenvalues, λ, are defined by a system of three equations.

The Laplacian provides a relationship between the derivatives of u and the eigenvalues [32].

$$\triangle u + \lambda u = 0 \tag{6.3}$$

The Laplacian is defined as

$$\triangle u = \sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2} \tag{6.4}$$

When restricted to polar coordinates, this is

$$\triangle u = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} \tag{6.5}$$

The Dirichlet boundary condition states that the eigenfunction is zero along the boundary of the surface [32]. This can be visualized as the rigid frame of the drum. S is the boundary restriction of space Ω.

$$u|_S = 0 \tag{6.6}$$

The eigenfunction is normalized over the entire space Ω.

$$\int_\Omega u^2 d\Omega = 1 \tag{6.7}$$

To find the eigenvalues, the following system must be solved.

$$\triangle u = -\lambda u \tag{6.8}$$

$$u|_S = 0 \tag{6.9}$$

$$\int_\Omega u^2 d\Omega = 1 \tag{6.10}$$

The Bessel Function solves Equation 6.8 [2]. The Bessel function is related to the second derivative operations in cylindrical coordinates [67]. The Bessel function of the first kind is given in [67].

$$J_v(z) = \left(\frac{z}{2}\right) \sum_{m=0}^{\infty} \frac{(-1)^m (z/2)^{2m}}{m!\Gamma(v+m+1)} \tag{6.11}$$

A solution for the eigenfunction, u, is given with two unknowns, x and c.

$$u(r, \theta) = cJ_0(xr) \tag{6.12}$$

The Laplacian of the zeroth Bessel function meets the requirements of Equation 6.8.

$$\triangle cJ_0(xr) = -x^2 cJ_0(x\rho) \tag{6.13}$$

The eigenvalue is $\lambda = x^2$. The Dirichlet boundary condition, Equation 6.9 determines the value of x.

$$
\begin{aligned}
u|_S =& u(r = 1, \theta) && (6.14)\\
=& cJ_0(x) && (6.15)\\
=& 0 && (6.16)
\end{aligned}
$$

For $c \neq 0$, select x to be a zero of the Bessel function. Let $x = \rho$ where ρ is the n-th zero of the zeroth Bessel J function.

$$
\lambda = \rho^2 \qquad (6.17)
$$

The normalization condition determines the value of the unknown c.

$$
\begin{aligned}
\int_\Omega u^2 d\Omega =& 1 && (6.18)\\
\int_0^1 r \int_{-\pi}^{\pi} \left(cJ_0(\rho r)\right)^2 d\theta dr =& 1 && (6.19)\\
c^2 2\pi \int_0^1 rJ_0(\rho r)^2 dr =& 1 && (6.20)\\
c^2 \pi J_1(\rho)^2 =& 1 && (6.21)\\
c^2 =& \frac{1}{\pi J_1(\rho)^2} && (6.22)\\
c =& \frac{1}{\sqrt{\pi} J_1(\rho)} && (6.23)
\end{aligned}
$$

This gives a final solution for the system

$$
\begin{aligned}
\lambda =& \rho^2 && (6.24)\\
u(r, \theta) =& \frac{J_0(\rho r)}{\sqrt{\pi} J_1(\rho)} && (6.25)
\end{aligned}
$$

There are infinitely many real zeros of the Bessel J function and no complex zeros [67]. The pattern of zeros can be seen in Figure 6.2.

Each of the zeros of this function produces a different vibration on the drum. The zeros close to the y-axis are very distinct. As the function approaches infinity, the difference between two zeros

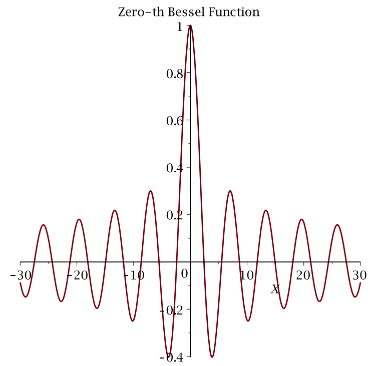

Figure 6.2: Plot of J_0 from -30 to 30 showing multiple zeros.

becomes to small to produce distinct sounds.

6.4 Regular Polygon

The series in $1/N$ for the Laplace-eigenvalues on the regular N-sided polygon under Dirichlet boundary conditions is now examined.

The approach presented here applies the CMS. A surface deformation is defined that starts with the unit circle, at time $t = 0$, and ends with the regular N-sided polygon, at time $t = 1$ [47]. An example deformation with $N = 8$ sides with shown in Figure 6.3.

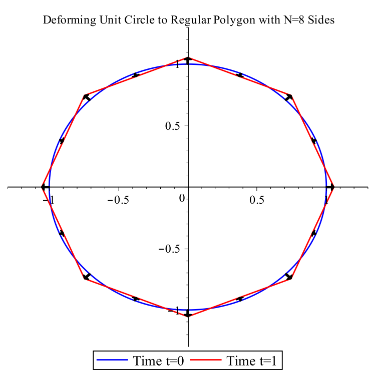

Figure 6.3: Boundary deformation from circle to regular polygon.

The answer is a Taylor series

$$\lambda_N = \lambda_0 + \lambda_1 + \frac{1}{2}\lambda_2 + \cdots + \frac{1}{k!}\lambda_k + \cdots \tag{6.26}$$

$$\lambda_N = \lambda(t = 1) \tag{6.27}$$

$$\lambda_0 = \lambda(t = 0) = \rho^2 \tag{6.28}$$

$$\lambda_k = \frac{d^k \lambda}{dt^k}|_{t=0} \tag{6.29}$$

Calculation of the variations of λ will be dependent on the deformation. The first derivative is given by Hadamard's formula [40].

$$\lambda_1 = -\int_S C \nabla_i u \nabla^i u dS \tag{6.30}$$

Evaluation of this expression requires the surface velocity C. The coordinate system is restricted to a fixed r on the boundary.

$$r(\phi) = 1 - t\left(1 - \frac{\cos\left(\frac{\pi}{N}\right)}{\cos\phi}\right) \tag{6.31}$$

$$\theta(\phi) = \phi \tag{6.32}$$

The surface velocity, C, is fully derived in [47]

$$D(\theta) = 1 - \frac{\cos(\pi/N)}{\cos\theta} \tag{6.33}$$

$$C(t,\theta) = \frac{D(\theta) + tD^2(\theta)}{\sqrt{(1-t)^2 + 2t(1-t)(1+D(\theta)) + t^2(1+D(\theta))^2\cos^{-2}\theta}} \tag{6.34}$$

At the initial time, $t = 0$, this is equal to

$$C_{t=0} = \frac{\cos(\pi/N)}{\cos\theta} - 1 \tag{6.35}$$

This expression holds for $-\pi/N < \theta < \pi/N$ and it extends around the circle in N periods [47]. The Fourier decomposition of $C_{t=0}$ is used as C.

$$C = \sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta} \tag{6.36}$$

The coefficient c_0 is defined as

$$c_0(k) = \begin{cases} -\frac{1}{3}\frac{\pi^2}{N^2} + O(\frac{1}{N^6}) & k = 0 \\ \frac{(-1)^k}{N^2k^2} + \frac{(-1)^k\pi^2}{3N^4k^2} - \frac{5(-1)^k}{N^4k^4} + O(\frac{1}{N^6}) & k \neq 0 \end{cases} \tag{6.37}$$

Hadamard's equation can now be evaluated

$$\nabla_i u \nabla^i u = \begin{pmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial \theta} \end{pmatrix}_i \begin{pmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial \theta} \end{pmatrix}^i = \sum_{i=1}^{2} \begin{pmatrix} \frac{\partial u}{\partial r}^2 & \frac{\partial u}{\partial r}\frac{\partial u}{\partial \theta} \\ \frac{\partial u}{\partial r}\frac{\partial u}{\partial \theta} & \frac{\partial u}{\partial \theta}^2 \end{pmatrix}_i^i \tag{6.38}$$

$$= \frac{\partial u}{\partial r}^2 + \frac{\partial u}{\partial \theta}^2 = \left(\frac{\partial}{\partial r}\frac{J_0(\rho r)}{\sqrt{\pi}J_1(\rho)}\right)^2 \tag{6.39}$$

$$= \left(\frac{-J_1(\rho r)\rho}{\sqrt{\pi}J_1(\rho)}\right)^2 = \frac{J_1(\rho r)^2\rho^2}{\pi J_1(\rho)^2} \tag{6.40}$$

$$\nabla_i u \nabla^i u|_{r=1} = \frac{\rho^2}{\pi} \tag{6.41}$$

This leads to

$$\lambda_1 = -\frac{\rho^2}{\pi}\int_S C dS \tag{6.42}$$

This integral is solved using a change of variables.

$$\int_S C dS = \int_{-\pi/N}^{\pi/N} \sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta} d\theta \tag{6.43}$$

$$= \int_{-\pi}^{\pi} \sum_{k=-\infty}^{\infty} c_0(k) e^{ik\phi} d\phi \tag{6.44}$$

$$= 2\pi c_0(0) \tag{6.45}$$

The solution for λ_1 can now be computed.

$$\lambda_1 = -2\rho^2 c_0 \tag{6.46}$$

$$= \frac{2}{3}\frac{\pi^2\rho^2}{N^2} + \frac{2}{315}\frac{\pi^2\rho^2}{N^6} + O\left(\frac{1}{N^8}\right) \tag{6.47}$$

$$= \frac{4\zeta(2)\lambda}{N^2} + \frac{6\zeta(6)\lambda}{N^6} + O\left(\frac{1}{N^8}\right) \tag{6.48}$$

Only the value of C was dependent on the boundary deformation. The solution $\lambda_1 = -2\lambda c_0$ is general. By finding a different C, the respective λ can be found.

Deforming the unit circle into an ellipse with semi-axis, $A = 1$ and $B = 1 + \epsilon$, gives a different value. This deformation is shown for a fixed ϵ in Figure 6.4.

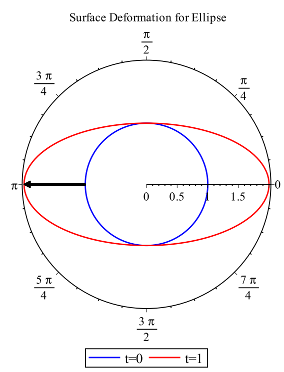

Figure 6.4: Boundary deformation from circle to ellipse.

The coefficient of the Fourier series for C, deforming to the ellipse is

$$c_{0,\text{ellipse}} = \begin{cases} \frac{1}{4} & k = -2 \\ \frac{1}{2} & k = 0 \\ \frac{1}{4} & k = -2 \\ 0 & \text{otherwise} \end{cases} \tag{6.49}$$

This gives the solution for λ_1 on the ellipse.

$$\lambda_{1,\text{ellipse}} = -2\rho^2 c_{0,\text{ellipse}} = -\lambda \tag{6.50}$$

This is an important feature that will be applied to the remaining variations. Expressions can be confirmed against the ellipse where numerical approximations can computed independently.

6.5 Second Variation of λ_2

The second variation of λ is found by applying the $\dot{\nabla}$-derivative to the first variation.

$$\lambda_2 = \dot{\nabla}\lambda_1(t) \tag{6.51}$$

$$\lambda_2 = \dot{\nabla}\left(-\int_S C\nabla_i u \nabla^i u dS\right) \tag{6.52}$$

More generally, any variation can be found by

$$\lambda_k = \dot{\nabla}^{k-1}\left(-\int_S C\nabla_i u \nabla^i u dS\right) \tag{6.53}$$

The rules of the CMS allow for the expansion of λ_2.

$$\lambda_2 = \dot{\nabla}\left(-\int_S C\nabla_i u \nabla^i u dS\right) \tag{6.54}$$

$$= -\int_S \dot{\nabla}\left(C\nabla_i u \nabla^i u\right) - C^2 B^\alpha_\alpha \nabla_i u \nabla^i u dS \tag{6.55}$$

$$= -\int_S C_1 \nabla_i u \nabla^i u + 2C\nabla_i u \nabla^i u_1 - C^2 B^\alpha_\alpha \nabla_i u \nabla^i u dS \tag{6.56}$$

The surface integral is evaluated when $t = 0$ and $r = 0$. When $t = 0$, the first variation of C vanishes.

$$C_1 = \dot{\nabla}(C)|_{t=0} = 0 \tag{6.57}$$

To continue with this computation the value of u_1 is required. u_1 is the first partial derivative of u evaluated at $t = 1$.

$$u_1 = \left. \frac{\partial u}{\partial t} \right|_{t=0} \tag{6.58}$$

The next chapter will solve u_1 and give a general method for solving u_k as a recursive function of $u_0 \cdots u_{k-1}$ and $\lambda \cdots \lambda_k$. After establishing u_1, computation of the λ values will be revisited.

7. Determining Partial Derivatives of u

7.1 Introduction

It has been shown that partial derivatives of the eigenfunction u are required to solve for the variations of lambda. Computation of these derivatives begins with the eigenfunction on the unit circle.

$$u(r,\theta) = \frac{J_0(\rho r)}{\sqrt{\pi} J_1(\rho)} \tag{7.1}$$

To evaluate the variations of λ, only the values of $\frac{\partial^i u}{\partial t^i}$ at $t = 0$ are needed. The method for determining these values is given in this chapter. A general method for computing u_1, will be given where

$$u_i = \left(\frac{\partial^i u}{\partial t^i} \right) \Bigg|_{t=0} \tag{7.2}$$

The chapter begins with the solution for u_1. Many simplifications appear in u_1, which are not true in the general case. Next, a general method will be given for any u_i. A requirement to find u_i will be the values $u_0 \cdots u_{i-1}$ and $\lambda \cdots \lambda_i$. It is convenient to treat these as symbolic values whenever possible.

7.2 Solving u_1

The system of equations used to find $u(r,\theta)$ will also determine u_1.

$$\triangle u = -\lambda u \tag{7.3}$$

$$u|_S = 0 \tag{7.4}$$

$$\int_\Omega u^2 d\Omega = 1 \tag{7.5}$$

They are referred to as the boundary condition, equation 7.4, the kernel condition, equation 7.3, and the normalization condition, equation 7.5.

7.2.1 Boundary Condition

The solution to u_1 requires that all these conditions remain fulfilled. The boundary condition, Equation 7.4, is the most straightforward to satisfy. By definition, the boundary condition is met when $r = 1$. The $\dot{\nabla}$-operator is applied to both sides, then the expression is solved for u_1.

$$\dot{\nabla}\left(u|_S\right) = \dot{\nabla}(0) \tag{7.6}$$

$$\dot{\nabla}\left(u\right)|_S = 0 \tag{7.7}$$

$$\left(\frac{\partial u}{\partial t} + CN^i \nabla_i u\right)\bigg|_S = 0 \tag{7.8}$$

$$\frac{\partial u}{\partial t}\bigg|_S + \left(CN^i \nabla_i u\right)\big|_S = 0 \tag{7.9}$$

$$u_1 + \left(CN^i \nabla_i u\right)\big|_S = 0 \tag{7.10}$$

$$u_1 = -\left(CN^i \nabla_i u\right)\big|_S \tag{7.11}$$

This is solved using the same method as algebraic manipulations for the variations of λ.

$$u_1 = -\left(\sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta}\right) \frac{du}{dr}\bigg|_{r=1} \tag{7.12}$$

$$= -\left(\sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta}\right) \left(\frac{J_0'(r\rho)\rho}{\sqrt{\pi} J_1(\rho)}\right)\bigg|_{r=1} \tag{7.13}$$

$$= -\left(\sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta}\right) \left(\frac{-J_1(r\rho)\rho}{\sqrt{\pi} J_1(\rho)}\bigg|_{r=1}\right) \tag{7.14}$$

$$= \frac{\rho}{\sqrt{\pi}} \sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta} \tag{7.15}$$

The final expression is a Fourier series. The coefficient of $e^{ik\theta}$ is the most important part. It is given a name

$$f_{1,\mathrm{surface}}(k) = \frac{\rho}{\sqrt{\pi}} c_0(k) \tag{7.16}$$

This condition must be true when u_1 is evaluated at $t = 0$. Without inserting the values of c_0, this expression cannot be simplified further. This final step is delayed to keep the expressions general.

7.2.2 Kernel Condition

Next, the $\dot{\nabla}$-operator is applied to Equation 7.3 to find the kernel condition for u_1.

$$\dot{\nabla}\triangle u = \dot{\nabla}\left(-\lambda u\right) \tag{7.17}$$

$$\triangle\dot{\nabla}u = -\lambda_1 u - \lambda u_1 \tag{7.18}$$

$$\triangle u_1 = -\lambda_1 u - \lambda u_1 \tag{7.19}$$

$$\left(\triangle + \lambda\right) u_1 = -\lambda_1 u \tag{7.20}$$

$$\tag{7.21}$$

The Helmholtz operator, $(\triangle + \lambda)$, is applied to u_1 [23]. An inverse operator is created to solve for u_1. There is only one specific pattern that this operator will be applied to.

$$B(x) = \left(\triangle + \lambda\right)(x) \tag{7.22}$$

$$B^{-1}(B(x)) = x \tag{7.23}$$

The inverse of the Helmholtz operator will allow for a solution to this condition of u_1.

$$u_1 = B^{-1}\left(-\lambda_1 u\right) \tag{7.24}$$

$$= -\frac{\lambda_1}{\sqrt{\pi}J_1(\rho)}B^{-1}\left(J_0(\rho r)\right) \tag{7.25}$$

The general formula for the Helmholtz operator is

$$B\left(\frac{1}{2np}r^n J_{|m|+n}(\rho r)e^{im\theta}\right) = r^{n-1}J_{|m|+n-1}(\rho r)e^{im\theta} \tag{7.26}$$

This is used to create an inverse pattern, $r^a J_b(\rho r)e^{im\theta}$. This is the only pattern that will appear as input to the inverse operator.

$$r^a J_b(\rho r)e^{im\theta} = r^{n-1}J_{|m|+n-1}(\rho r)e^{im\theta} \tag{7.27}$$

$$a = n-1 \tag{7.28}$$

$$b = |m|+n-1 = |m|+a \tag{7.29}$$

The general pattern for the inverse operator is given, this is only valid for positive subscripts of of the Bessel J function.

$$B\left(\frac{1}{2(a+1)\rho}r^{a+1}J_{|m|+a+1}(\rho r)e^{im\theta}\right) = r^a J_{|m|+a}(\rho r)e^{im\theta} \tag{7.30}$$

$$B^{-1}\left(r^a J_{|m|+a}(\rho r)e^{im\theta}\right) = \frac{1}{2(a+1)\rho}r^{a+1}J_{|m|+a+1}(\rho r)e^{im\theta} \tag{7.31}$$

This is applied to the specific case for u_1.

$$-\frac{\lambda_1}{\sqrt{\pi}J_1(\rho)}B^{-1}\left(J_0(\rho r)\right) = -\frac{\lambda_1}{2\rho\sqrt{\pi}J_1(\rho)}rJ_1(\rho r) \tag{7.32}$$

For consistency with the general form, this is treated as a coefficient to a Fourier series.

$$f_{1,\text{kernal}}(k,r) = \begin{cases} -\frac{\lambda_1}{2\rho\sqrt{\pi}J_1(\rho)}rJ_1(\rho r) & k=0 \\ 0 & k \neq 0 \end{cases} \tag{7.33}$$

The following expression will meet the kernel condition.

$$(\triangle+\lambda)\sum_{k=-\infty}^{\infty} f_{1,\text{kernel}}(k,r)e^{ik\theta} = -\lambda_1 u \tag{7.34}$$

7.2.3 Partial Solution to u_1

Before solving for the normalization, a partial solution for u_1 is generated. This will be required to work with the normalization condition. The expression must be equal to the surface condition when $r = 1$. Additionally, the expression must correctly evaluate under the Helmholtz operator. The nullspace of B is used to add terms to the series without changing the outcome of the Helmholtz operator.

$$B\left(J_{|m|}(\rho r)e^{im\theta}\right) = 0 \tag{7.35}$$

To ensure these values act correctly under surface restriction, they are divided by their value at $r = 1$.

$$B\left(\frac{J_{|m|}(\rho r)}{J_{|m|}(\rho)}e^{im\theta}\right) = 0 \tag{7.36}$$

The first part of u_1 meets the boundary condition.

$$u_1 = \sum_{k\neq 0} f_{1,\text{surface}}(k)\frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)}e^{ikN\theta} + \cdots \tag{7.37}$$

This works as long as $k \neq 0$. In that cases, $J_0(\rho)$ is 0 in the denominator of the fraction. This will be handled by the normalization condition. For $k \neq 0$, this equation meets the boundary condition. Any terms added to meet the kernel condition must therefore cancel out at $r = 1$.

Under the Helmholtz operator, all these terms will evaluate to 0. The coefficients needed to meet the kernel condition are added next.

$$u_1 = \sum_{k\neq 0} f_{1,\text{surface}}(k)\frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)}e^{ikN\theta} + \sum_{k=-\infty}^{\infty} f_{1,\text{kernal}}(k,r)e^{ikN\theta} + \cdots \tag{7.38}$$

This changes the value on the surface, but that change can be easily accounted for.

$$\begin{aligned} u_1 = &\sum_{k\neq 0} f_{1,\text{surface}}(k)\frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)}e^{ikN\theta} \\ &+ \sum_{k=-\infty}^{\infty} f_{1,\text{bulk}}(k,r)e^{ikN\theta} - \sum_{k\neq 0} f_{1,\text{bulk}}(k,1)\frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)}e^{ikN\theta} + \cdots \end{aligned} \tag{7.39}$$

There is still one unknown term, $S_{1,0}$, at J_0.

$$u_1 = \sum_{k \neq 0} f_{1,\text{surface}}(k) \frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)} e^{ikN\theta} + S_{1,0} J_0(\rho r)$$
$$+ \sum_{k=-\infty}^{\infty} f_{1,\text{bulk}}(k, r) e^{ik\theta} - \sum_{k \neq 0} f_{1,\text{bulk}}(k, 1) \frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)} e^{ik\theta} \tag{7.40}$$

The normalization condition, Equation 7.5, will determine the final unknown. First, u_1 is simplified.

$$u_1 = \sum_{k=-\infty}^{\infty} f_1(r, k) e^{ikN\theta} \tag{7.41}$$
$$f_1(k, r) = \begin{cases} S_{1,0} J_0(\rho r) - \frac{\lambda_1}{2\rho\sqrt{\pi} J_1(\rho)} r J_1(\rho r) & k = 0 \\ \frac{\rho c_0(k)}{\sqrt{\pi}} \frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)} & k \neq 0 \end{cases} \tag{7.42}$$

7.2.4 Normalization Condition

The final condition that needs to be solved is the normalization condition, given by Equation 7.5. The $\dot{\nabla}$-operator is used to find the condition for u_1.

$$\dot{\nabla} \left(\int_\Omega u^2 d\Omega \right) = \dot{\nabla}(1) \tag{7.43}$$
$$\int_\Omega \frac{\partial u^2}{\partial t} d\Omega + \int_S C u^2 dS = 0 \tag{7.44}$$
$$\int_\Omega u_1 \, u + u \, u_1 d\Omega = - \int_S C u^2 dS \tag{7.45}$$
$$2 \int_\Omega u \, u_1 d\Omega = 0 \tag{7.46}$$
$$\int_\Omega u \, u_1 d\Omega = 0 \tag{7.47}$$

The spatial integral on the circle is defined as

$$\int_\Omega f(r, \theta) d\Omega = \int_0^1 r \int_{-\pi}^{\pi} f(r, \theta) d\theta dr \tag{7.48}$$

The majority of terms immediately go to zero because of the following identity, which is true for integer k and $k \neq 0$.

$$\int_{-\pi}^{\pi} e^{ik\theta} d\theta = 0 \tag{7.49}$$

This leaves only

$$\int_0^1 r \left(\frac{J_0(\rho r)}{\sqrt{\pi} J_1(\rho)} \right) \left(S_{1,0} J_0(\rho r) - \frac{\lambda_1}{2\rho\sqrt{\pi} J_1(\rho)} r J_1(\rho r) \right) dr = 0 \tag{7.50}$$

This is solved algebraically for $S_{1,0}$.

$$0 = 2\pi \left(\frac{S_{1,0}}{\sqrt{\pi} J_1(\rho)} \int_0^1 r J_0(r\rho)^2 dr - \frac{\lambda_1}{2\rho\pi J_1(\rho)^2} \int_0^1 r^2 J_0(\rho r) J_1(\rho r) dr \right) \tag{7.51}$$

$$0 = \frac{S_{1,0}}{\sqrt{\pi} J_1(\rho)} \left(\frac{1}{2} J_1(\rho)^2 \right) - \frac{\lambda_1}{2\rho\pi J_1(\rho)^2} \left(\frac{1}{2} \frac{J_1(\rho)^2}{\rho} \right) \tag{7.52}$$

$$0 = \frac{S_{1,0}}{2\sqrt{\pi}} J_1(\rho) - \frac{\lambda_1}{4\rho^2\pi} \tag{7.53}$$

$$-\frac{S_{1,0}}{2\sqrt{\pi}} J_1(\rho) = -\frac{\lambda_1}{4\rho^2\pi} \tag{7.54}$$

$$S_{1,0} = \frac{2\sqrt{\pi}}{J_1(\rho)} \frac{\lambda_1}{4\rho^2\pi} \tag{7.55}$$

$$S_{1,0} = \frac{\lambda_1}{2\sqrt{\pi}\rho^2 J_1(\rho)} \tag{7.56}$$

The complete expression for u_1 can now be given.

$$u_1 = \sum_{k=-\infty}^{\infty} f_1(r,k) e^{ikN\theta} \tag{7.57}$$

$$f_1(k,r) = \begin{cases} \frac{1}{2} \frac{\lambda_1 (J_0(\rho r) - r J_1(\rho r)\rho)}{\rho^2 \sqrt{\pi} J_1(\rho)} & k = 0 \\ \frac{c_0(k)\rho}{\sqrt{\pi}} \frac{J_{|kN|}(\rho r)}{J_{|kN|}(\rho)} & k \neq 0 \end{cases} \tag{7.58}$$

7.2.5 Justification of u_1

To show that the value of u_1 meets the boundary condition, we first evaluate f_1 when $r = 1$.

$$f_1(k,1) = \begin{cases} -\frac{1}{2} \frac{\lambda_1}{\rho\sqrt{\pi}} & k = 0 \\ \frac{c_0(k)\rho}{\sqrt{\pi}} & k \neq 0 \end{cases} \tag{7.59}$$

We next show this meets the boundary condition from Equation 7.15.

$$u_1|_S = \frac{\rho}{\sqrt{\pi}} \sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta} \tag{7.60}$$

$$\sum_{k=-\infty}^{\infty} \left(\begin{cases} -\frac{1}{2}\frac{\lambda_1}{\rho\sqrt{\pi}} & k = 0 \\ \frac{c_0(k)\rho}{\sqrt{\pi}} & k \neq 0 \end{cases} \right) e^{ikN\theta} = \frac{\rho}{\sqrt{\pi}} \sum_{k=-\infty}^{\infty} c_0(k) e^{ikN\theta} \tag{7.61}$$

This is true as long as

$$-\frac{1}{2}\frac{\lambda_1}{\rho\sqrt{\pi}} = \frac{\rho}{\sqrt{\pi}} c_0(0) \tag{7.62}$$

The value for λ_1 was given in Chapter 6.4. After plugging this in, the two expressions are equivalent.

$$-\frac{1}{2}\frac{-2\rho^2 c_0(0)}{\rho\sqrt{\pi}} = \frac{\rho}{\sqrt{\pi}} c_0(0) \tag{7.63}$$

$$\frac{\rho c_0(0)}{\sqrt{\pi}} = \frac{\rho c_0(0)}{\sqrt{\pi}} \tag{7.64}$$

This shows that u_1 meets the boundary condition. Next, the kernel condition is checked, Equation 7.34.

The majority of these calculation will come from evaluating the Laplacian on u_1.

$$(\triangle + \lambda)\, u_1 = -\lambda_1 u \tag{7.65}$$

$$\triangle u_1 + \lambda u_1 = -\lambda_1 u \tag{7.66}$$

$$\triangle u_1 = -\lambda_1 u - \lambda u_1 \tag{7.67}$$

First, evaluate when $k \neq 0$ in the Fourier series.

$$-\lambda_1 u - \lambda u_1 = -\lambda u_1 \tag{7.68}$$
$$= \triangle\left(\sum_{k\neq 0} \frac{c_0(k)\rho}{\sqrt{\pi}} \frac{J_k(\rho r)}{J_k(\rho)} e^{ikN\theta}\right) \tag{7.69}$$
$$= -\frac{c_0(k)\rho^3 J_{|kN}(\rho r)}{\sqrt{\pi} J_{|kN|}(\rho)} e^{ikN\theta} \tag{7.70}$$
$$= -\lambda \sum_{k\neq 0} \frac{\rho c_0(k) J_{|kN|}(\rho r)}{\sqrt{\pi} J_{|kN}(\rho)} e^{ikN\theta} \tag{7.71}$$
$$= -\lambda u_1 \tag{7.72}$$

Next, we see if the expression is true for the case $k = 0$.

$$-\lambda_1 u - \lambda u_1 = \triangle\left(\frac{1}{2}\frac{\lambda_1\left(J_0(\rho r) - rJ(1,\rho r)\rho\right)}{\rho^2\sqrt{\pi}J_1(\rho)}\right) \tag{7.73}$$
$$= \frac{1}{2}\frac{\lambda_1 r J_1(\rho r)\rho - 3\lambda_1 J_0(\rho r)}{\sqrt{\pi}J_1(\rho)} \tag{7.74}$$
$$= \frac{1}{2}\frac{\lambda_1 r J_1(\rho r)\rho}{\sqrt{\pi}J_1(\rho)} - \frac{1}{2}\frac{\lambda_1 J_0(\rho r)}{\sqrt{\pi}J_1(\rho)} - \lambda_1\frac{J_0(\rho r)}{\sqrt{\pi}J_1(\rho)} \tag{7.75}$$
$$= -\lambda\left(\frac{1}{2}\frac{\lambda_1\left(J_0(\rho r) - rJ_1(\rho r)\rho\right)}{\rho^2\sqrt{\pi}J_1(\rho)}\right) - \lambda_1 u \tag{7.76}$$
$$= -\lambda u_1 - \lambda_1 u \tag{7.77}$$

These two cases combined prove that our expression for u_1 meets the kernel condition. The final condition is normalization.

$$\int_{\Omega} uu_1 d\Omega = 0 \tag{7.78}$$

$$2\pi \int_0^1 ruf_1(0)dr = 0 \tag{7.79}$$

$$2\pi \int_0^1 r \left(\frac{J_0(\rho r)}{\sqrt{\pi} J_1(\rho)} \right) \left(\frac{1}{2} \frac{\lambda_1 \left(J_0(\rho r) - rJ(1, \rho r)\rho \right)}{\rho^2 \sqrt{\pi} J_1(\rho)} \right) dr = 0 \tag{7.80}$$

$$\int_0^1 \left(\frac{\lambda_1 r J_0(\rho r)^2}{\rho^2 J_1(\rho)^2} - \frac{\lambda_1 r^2 J(1, \rho r) J_0(\rho r)\rho}{\rho^2 J_1(\rho)^2} \right) dr = 0 \tag{7.81}$$

$$\frac{\lambda_1}{\rho^2 J_1(\rho)^2} \int_0^1 \left(rJ_0(\rho r)^2 - r^2 J(1, \rho r) J_0(\rho r)\rho \right) dr = 0 \tag{7.82}$$

$$\int_0^1 rJ_0(\rho r)^2 dr - \int_0^1 r^2 J(1, \rho r) J_0(\rho r)\rho dr = 0 \tag{7.83}$$

$$\frac{1}{2} J_1(\rho)^2 - \frac{1}{2} J_1(\rho)^2 = 0 \tag{7.84}$$

This proves that u_1 is a correct solution. In the next section, this method will generalize u_m.

7.3 General Solution for u_m

To solve for the general u_m, the same pattern is followed with repeated application of the $\dot{\nabla}$-operator. All components will be solved as Fourier series. The boundary condition is straightforward to define.

$$\dot{\nabla}^m \left(u|_S \right) = 0 \tag{7.85}$$

$$\sum_{k=-\infty}^{\infty} f_{m,\text{surface}}(k) e^{ikN\theta} = \dot{\nabla}^m \left(u|_S \right) \tag{7.86}$$

There is a closed form for finding the m-th application of the $\dot{\nabla}$-operator for the kernel condition.

$$\dot{\nabla}^m \triangle u = \dot{\nabla}^i \left(-\lambda u\right) \tag{7.87}$$

$$\left(\triangle + \lambda\right) u_m = -\sum_{j=1}^{m} \binom{m}{j} \lambda_j u_{m-j} \tag{7.88}$$

$$u_m = B^{-1} \left(-\sum_{j=1}^{m} \binom{m}{j} \lambda_j u_{m-j} \right) \tag{7.89}$$

$$\sum_{k=-\infty}^{\infty} f_{m,\text{bulk}}(k,r) e^{ikN\theta} = B^{-1} \left(-\sum_{j=1}^{m} \binom{m}{j} \lambda_j u_{m-j} \right) \tag{7.90}$$

A general formula for the $S_{0,m}$ coefficient can be found using the normalization condition. A partial solution to u_m is put into this equation with an unknown m,

$$\dot{\nabla}^i \left(\int_\Omega u^2 d\Omega \right) = 0 \tag{7.91}$$

After taking repeated $\dot{\nabla}$-derivatives, the expression is solved for $S_{0,m}$.

Combining all these parts gives a general solution u_m.

$$u_m = \sum_{k=-\infty}^{\infty} f_m(k,r) e^{ikN\theta} \tag{7.92}$$

$$f_m(k,r) = \begin{cases} S_{0,m} J_0(\rho r) + f_{m,\text{bulk}}(0,r) & k = 0 \\ \frac{(f_{m,\text{surface}}(k) - f_{m,\text{bulk}}(k,1)) J_{|kN|}(\rho r)}{J_{|kN|}(\rho)} + f_{m,\text{bulk}}(k,r) & k \neq 0 \end{cases} \tag{7.93}$$

The value of u_1 can now be used to find λ_2. Additional values of u_m can be derived when needed.

During computation of the Laplace eigenvalues, these partial derivatives will appear. The values are needed to for evaluation. The process described above is implemented using the SCMS. This allows for the automated computation of all partial derivatives need to our eigenvalue expressions.

8. Second Variation of λ

An expression for λ_2 has already been established. It has been shown that all u_m functions will have the form of a Fourier Series. This expression can now be simplified into convolutions of these coefficients.

$$\lambda_2 = -\int_S C_1 \nabla_i u \nabla^i u + 2C \nabla_i u \nabla^i u_1 - C^2 B^\alpha_\alpha \nabla_i u \nabla^i u dS \tag{8.1}$$

$$= 2\text{conv}(c_0, c_0)(0)\rho^2 + 4\sqrt{\pi}\text{conv}(c_0, \frac{df_1}{dr}|_{r=1})(0)\rho \tag{8.2}$$

$$= 2\text{conv}(c_0, c_0)(0)\rho^2 + 4\sqrt{\pi}\text{conv}(c_0, f_{1,dr})(0)\rho \tag{8.3}$$

For convenience, a new subscript notation is introduced to denote the derivatives of the eigenfunction coefficients. This is for the reader's convenience, it is implemented as explicit calls to inert functions.

$$\frac{df(k)}{dr}|_{r=1} = f_{1,dr} \tag{8.4}$$

$$= \begin{cases} \frac{c_0(0)\rho}{\sqrt{\pi}} & k = 0 \\ \frac{c_0(k)\rho}{\sqrt{\pi}} \frac{\rho J'_{|kN|}(\rho)}{J_{|kN|}(\rho)} & k \neq 0 \end{cases} \tag{8.5}$$

A series is given for $\frac{\rho J'_{|kN|}(\rho)}{J_{|kN|}(\rho)}$. This comes from the continued fraction decomposition of the Bessel Function.

$$\frac{\rho J'_{|kN|}(\rho)}{J_{|kN|}(\rho)} = |kN| - \frac{1}{2}\frac{\rho^2}{|kN|} + \frac{1}{2}\frac{\rho^2}{|kN|^2} - \frac{\rho^2}{|kN|^3}\left(\frac{1}{8}\rho^2 + \frac{1}{2}\right) + O\left(\frac{1}{|kN|^4}\right) \tag{8.6}$$

8.1 First Convolution

The first convolution in λ_2 contains only c_0.

$$2\text{conv}(c_0, c_0)(0)\rho^2 \tag{8.7}$$

The convolution is defined as

$$T_1 = 2\rho^2 \sum_{k=-\infty}^{\infty} c_0(k)c_0(-k) \tag{8.8}$$

This can be solved as a sum with three parts, $k = 0$, $k < 0$, and $k > 0$.

When $k \neq 0$

$$c_0(k)c_0(-k) = \frac{1}{k^4 N^4} + \frac{2}{3}\frac{\pi^2}{k^4 N^6} - \frac{10}{k^6 N^6} \tag{8.9}$$

Since k is the only variable in the infinite sum, each part can be simplified individually.

$$\frac{1}{N^4}\sum_{k=1}^{\infty}\frac{1}{k^4} = \frac{1}{N^4}\zeta(4) \tag{8.10}$$

$$\frac{2\pi^2}{3N^6}\sum_{k=1}^{\infty}\frac{1}{k^4} = \frac{2\pi^2}{3N^6}\zeta(4) = \frac{4\zeta(2)\zeta(4)}{N^6} \tag{8.11}$$

$$\frac{-10}{N^6}\sum_{k=1}^{\infty}\frac{1}{k^6} = \frac{-10}{N^6}\zeta(6) \tag{8.12}$$

The same is result is derived for $k = -\infty \cdots - 1$ because multiplication is commutative.

These are combined to give the value of the summation with $k \neq 0$.

$$\sum_{k\neq 0} c_0(k)c_0(-k) = 2\left(\frac{\zeta(4)}{N^4} + \frac{4\zeta(2)\zeta(4)}{N^6} - \frac{10\zeta(6)}{N^6}\right) \tag{8.13}$$

$$= \frac{2\zeta(4)}{N^4} - \frac{6\zeta(6)}{N^6} \tag{8.14}$$

The value when $k = 0$ is trivial to calculate.

$$c_0(0)^2 = \frac{1}{9}\frac{\pi^4}{N^4} = 4\frac{\zeta(2)^2}{N^4} = 10\frac{\zeta(4)}{N^4} \tag{8.15}$$

This gives the total for the convolution

$$\mathrm{conv}(c_0,c_0)(0) = \frac{2\zeta(4)}{N^4} - 6\frac{\zeta(6)}{N^6} + 10\frac{\zeta(4)}{N^4} \tag{8.16}$$

$$= 12\frac{\zeta(4)}{N^4} - 6\frac{\zeta(6)}{N^6} \tag{8.17}$$

It remains to multiply by $2\rho^2$.

$$T_1 = 2\mathrm{conv}(c_0,c_0)(0)\rho^2 \tag{8.18}$$

$$= \left(24\frac{\zeta(4)}{N^4} - 12\frac{\zeta(6)}{N^6}\right)\rho^2 \tag{8.19}$$

8.2 Second Convolution

The second convolution requires $f_{1,dr}$, which is already known.

$$T_2 = 4\sqrt{\pi}\mathrm{conv}(c_0, f_{1,dr})(0)\rho \tag{8.20}$$

Again, the value when $k = 0$ is straightforward to compute.

$$c_0(0)f_{1,dr}(0) = c_0(0)^2\frac{\rho}{\sqrt{\pi}} \tag{8.21}$$

$$= 4\frac{\zeta(2)^2}{N^4}\frac{\rho}{\sqrt{\pi}} = 10\frac{\zeta(4)}{N^4}\frac{\rho}{\sqrt{\pi}} \tag{8.22}$$

The nonzero part is more difficult.

$$= \sum_{k=1}^{\infty}\left(c_0(k)f_{1,dr}(-k)\right) \tag{8.23}$$

$$= \sum_{k=1}^{\infty} c_0(k)c_0(-k)\frac{\rho J'_{|kN|}(\rho)}{J_{|kN|}(\rho)} \tag{8.24}$$

$$= \frac{\rho}{\sqrt{\pi}}\sum_{k=1}^{\infty} c_0(k)c_0(-k)\frac{\rho J'_{|kN|}(\rho)}{J_{|kN|}(\rho)} \tag{8.25}$$

The Bessel J function contains an absolute value, and all values in c_0 are taken to even powers. This means the positive and negative ranges will be the same. Only the positive range is shown.

$$c_0(k)c_0(k)\frac{\rho J'_{|kN|}(\rho)}{J_{|kN|}(\rho)} = \frac{1}{k^3N^3} - \frac{10}{k^5N^5} - \frac{1}{2}\frac{\rho^2}{k^5N^5} + \frac{2}{3}\frac{\pi^2}{k^3N^5} + \frac{1}{2}\frac{\rho^2}{N^6k^6} \tag{8.26}$$

Each of these produces a ζ function when the sum is computed from $k = 1 \cdots \infty$.

$$\frac{1}{N^3}\sum_{k=1}^{\infty}\frac{1}{k^3} = \frac{\zeta(3)}{N^3} \tag{8.27}$$

$$\frac{-10}{N^5}\sum_{k=1}^{\infty}\frac{1}{k^5} = \frac{-10\zeta(5)}{N^5} \tag{8.28}$$

$$\frac{-\rho^2}{2N^5}\sum_{k=1}^{\infty}\frac{1}{k^5} = \frac{-\rho^2\zeta(5)}{2N^5} \tag{8.29}$$

$$\frac{2\pi^2}{3N^5}\sum_{k=1}^{\infty}\frac{1}{k^3} = \frac{2\pi^2\zeta(3)}{3N^5} \tag{8.30}$$

$$\frac{\rho^2}{2N^6}\sum_{k=1}^{\infty}\frac{1}{k^6} = \frac{\rho^2\zeta(6)}{2N^6} \tag{8.31}$$

These are combined to find the solution to the convolution.

$$\sum_{k=1}^{\infty}\cdots = \frac{\zeta(3)}{N^3} - \frac{10\zeta(5)}{N^5} - \frac{\rho^2\zeta(5)}{2N^5} + \frac{2\pi^2\zeta(3)}{3N^5} + \frac{1}{2}\frac{\rho^2\zeta(6)}{N^6} \tag{8.32}$$

The same thing happens for $k < 0$ giving a total of

$$\sum_{k\neq 0}\cdots = 2\left(\frac{\zeta(3)}{N^3} - \frac{10\zeta(5)}{N^5} - \frac{\rho^2\zeta(5)}{2N^5} + \frac{2\pi^2\zeta(3)}{3N^5} + \frac{1}{2}\frac{\rho^2\zeta(6)}{N^6}\right) \tag{8.33}$$

$$= \frac{2\zeta(3)}{N^3} - \frac{20\zeta(5)}{N^5} - \frac{\rho^2\zeta(5)}{N^5} + \frac{8\zeta(2)\zeta(3)}{N^5} + \frac{\rho^2\zeta(6)}{N^6} \tag{8.34}$$

In the λ_2 expression this sum is multiplied by $4\sqrt{\pi}\rho$. The contribution to λ_2 from this convolution is

$$T_2 = 4\sqrt{\pi}\mathrm{conv}(c_0, f_{1,dr})(0)\rho \tag{8.35}$$

$$= \rho^2\left(\frac{8\zeta(3)}{N^3} + 16\frac{\zeta(2)^2}{N^4} - \frac{80\zeta(5)}{N^5} + \frac{32\zeta(2)\zeta(3)}{N^5} - \frac{4\rho^2\zeta(5)}{N^5} + \frac{4\rho^2\zeta(6)}{N^6}\right) \tag{8.36}$$

Table 8.1: Confirmation of λ_2 by numerical approximation.

	Approx	Exact
$x_1\zeta(3)\rho^2/N^3$	7.999999999	8
$x_2\zeta(2)^2\rho^2/N^4$	24.0000000000	24
$x_3\zeta(4)\rho^2/N^4$	3.9999999999	4
$x_4\zeta(2)\zeta(3)\rho^2/N^5$	31.9999999979	32
$x_5\zeta(5)\rho^2/N^5$	-79.9999999999	-80
$x_6\zeta(5)\rho^4/N^5$	-3.9999999999	-4
$x_7\zeta(2)\zeta(4)\rho^2/N^6$	15.9999999999	16
$x_8\zeta(6)\rho^2/N^6$	-39.9999999999	-40
$x_9\zeta(6)\rho^4/N^6$	3.9999999999	4

8.3 Final Value for λ_2

All that remains to generate λ_2 is to combine the two values computed above.

$$\frac{\lambda_2}{\lambda} = \frac{8\zeta(3)}{N^3} + \frac{24\zeta(2)^2}{N^4} + \frac{4\zeta(4)}{N^4} - \frac{80\zeta(5)}{N^5} + \frac{32\zeta(2)\zeta(3)}{N^5} - \frac{4\rho^2\zeta(5)}{N^5} + \frac{16\zeta(2)\zeta(4)}{N^6} - \frac{40\zeta(6)}{N^6} + \frac{4\rho^2\zeta(6)}{N^6} \tag{8.37}$$

$$= \frac{8\zeta(3)}{N^3} + \frac{64\zeta(4)}{N^4} + \frac{32\zeta(2)\zeta(3)}{N^5} - \frac{(4\lambda+80)\zeta(5)}{N^5} + \frac{(4\lambda-12)\zeta(6)}{N^6} \tag{8.38}$$

The convolutions can also be approximated numerically. The numerical computation involves truncating the summations $\sum_{k=-\infty}^{\infty} \cdots = \sum_{k=-R}^{R}$. The exact results are compared to the approximate results in Table 8.1. The value of λ_3 will be dependent on u_2 and multiple derivatives of u_1. An automated method for dealing with Fourier series is now required. This library is described in Chapter 9.

9. Fourier Library for Maple

9.1 Introduction

The Fourier series library has two goals. The first is the symbolic manipulation of series. This is handled by the Fourier series object. When performing manipulations with this object, the target is a new Fourier series. The coefficients of this new series will be built from manipulations of the original coefficients, for example convolutions and derivatives.

The second and more difficult goal, is to find a closed from for the new coefficient function. No algorithm exists to determine if a closed form expression exists and find it in general. This means that the second goal will not always be possible, but a set of rules is introduced that deals with the type of series that appear in the Laplacian Eigenvalue problem on the N-sided polygon. Although not all series can be solved exactly, they can always be approximated numerically through truncation.

9.2 Fourier Series Manipulation

The foundation of the library is a Fourier Series object. This structure allows for the creation of a new series object and its use with the standard Maple interface for multiplication, addition, derivatives, etc. The object overloads the standard operations with the definitions provided below. During these calculations, the series will be kept in the object framework. Attempts to approximate or simplify the coefficients of the series are delayed until requested. This allows for efficient calculations by delaying the most time consuming parts until needed.

The series object is a tuple of two functions $\{f(k), g(k,\theta)\}$. They form the series

$$\sum_{k=-\infty}^{\infty} f(k)e^{g(k,\theta)} \tag{9.1}$$

The primary motivation for this library is handling multiplication. When two series are multiplied together a special convolution function is introduced. This function is completely inert until a value is requested and an evaluation method is chosen.

Given two series objects $A(\theta)$ and $B(\theta)$, the coefficient is a function with one input, the iteration index. For clarity, the exponent will be treated as a two input function, but in Maple it only has one input. The second input, θ, is evaluated using the `eval` command. The value θ is required to be symbolic in the return value of the function. This is enforced in the object constructor and mutators.

$$A(r,\theta) = \sum_{k=-\infty}^{\infty} f(k)e^{g(k,\theta)} \tag{9.2}$$

$$B(r,\theta) = \sum_{j=-\infty}^{\infty} x(j)e^{y(j,\theta)} \tag{9.3}$$

When these series are multiplied together, the object must confirm if the exponent functions are equivalent. If $g(k,\theta) = y(k,\theta)$ then the series can be combined.

$$AB = C \text{ if } g(k,\theta) = y(k,\theta) \tag{9.4}$$

$$C = \sum_{k=-\infty}^{\infty} h(k)e^{g(k,\theta)} \tag{9.5}$$

$$h(k) = \operatorname{conv}(f,x)(k) \tag{9.6}$$

The conv function is a symbolic representation of the sum

$$\operatorname{conv}(f,x)(k) = \sum_{j=-\infty}^{\infty} f(k-j)x(j) \tag{9.7}$$

The library will not attempt to evaluate this convolution until asked. It will store an infinite series of inert conv function calls.

Multiplication by scalar, s, is also supported.

$$sA = \sum_{k=-\infty}^{\infty} h(k)e^{g(k,\theta)} \tag{9.8}$$

$$h(k) = s\,h(k) \tag{9.9}$$

The addition of two series objects is dependent on the equivalence of their exponent functions.

$$A + B = C \text{ if } g(k,\theta) = y(k,\theta) \tag{9.10}$$

$$C = \sum_{k=-\infty}^{\infty} h(k) e^{g(k,\theta)} \tag{9.11}$$

$$h(k) = f(k) + x(k) \tag{9.12}$$

A piecewise function is used to add series to a scalar s.

$$A + s = \sum_{k=-\infty}^{\infty} h(k) e^{g(k,\theta)} \tag{9.13}$$

$$h(k) = \begin{cases} s + f(0) & k = 0 \\ f(k) & \text{otherwise} \end{cases} \tag{9.14}$$

Derivatives are also handled by a revision to the coefficient function. Taking the derivative with respect to variable v is implemented as

$$\frac{d}{dv} A = \sum_{k=-\infty}^{\infty} h(k) e^{g(k,\theta)} \tag{9.15}$$

$$h(k) = \frac{df(k)}{dv} + f(k) \frac{dg(k,\theta)}{dv} \tag{9.16}$$

One special case of integration is implemented.

$$\int_{-\pi}^{\pi} A d\theta = f(0) \tag{9.17}$$

This is supported for any equivalent range, for example $0 \cdots 2\pi$.

This functionality provides the algebraic manipulation needed for series calculations in our eigenvalue problem.

9.3 Numerical Evaluation

A method to approximate convolutions at specific points is given. First, the series is truncated. For example, to simplify a convolution of three functions we truncate two infinite series to have finite ranges.

$$a(n) = \operatorname{conv}(f, g, h)(n) \tag{9.18}$$

The series is truncated to run from $-t \cdots t$, which means calculating

$$\operatorname{trun}(a, n, t) = \sum_{k=-t}^{t} \sum_{j=-t}^{t} f(n-k)g(-k-j)h(j) \tag{9.19}$$

This is a double application of the formula from equation 9.7 with truncated ranges. Since these are truncated ranges, an exact solution would only be found if the series has a finite number of nonzero terms. If that were the case, selecting the correct t would complete the calculations. Since this is the exception rather than the rule, truncation is expected to produce only a numerical result.

To improve the results, Richardson Extrapolation is applied [94]. The approximate value is computed at multiple powers of two. These values are evaluated using Richardson Extrapolation to improve the numerical accuracy. The series are calculated from largest to smallest ranges. This allows the built in memoization features of Maple to quickly recompute repeated function calls.

A formula to cancel one order of error using Richardson with $t = 4, 8, 16$ is shown. This method is dependent on calculating at powers of two. It can be used recursively to eliminate additional error terms. Maple is used to determine the formula based on the number of powers of 2 available.

$$\operatorname{rich}(a, n, 16) = \frac{2^c \operatorname{trun}(a, n, 8) - \operatorname{trun}(a, n, 16)}{2^c - 1} \tag{9.20}$$

$$c = \frac{\operatorname{trun}(a, n, 16) - \operatorname{trun}(a, n, 8)}{\operatorname{trun}(a, n, 8) - \operatorname{trun}(a, n, 4)} \tag{9.21}$$

The value c is the rate of convergence. The larger $|c|$ is, the more accurate the calculations will be. A value $|c| < 1$ means a divergent series. The rate of convergence is also a numerical approximation.

The error in the truncated method is defined

$$\operatorname{trun}(a, n, t) = a(n) + Et^c + o\left(t^{c+1}\right) \tag{9.22}$$

One application of equation 9.20 will cancel the leading error Et^c, leaving an improved series.

$$\operatorname{rich}(a, n, t) = a(n) + o\left(t^{c+1}\right) \tag{9.23}$$

Although the goal is to find an exact solution, it is not always possible. Numerical approximations can give insight into the answer. These approximations are also crucial to testing the symbolic evaluation methods. They are also used to quickly determine the order in N of an expression. This is used to determine if expressions can be ignored because it does not contribute to the desired orders of the solution.

9.4 Symbolic Evaluation

The approach to simplifying convolutions is to split the problem up into smaller components. This reduces the expressions to a form that can be matched against a rule set. There are two main motivations for the approach to simplification, zero elements may be special and sums starting at 1 and running over positive integers are more likely to have simplification rules.

The first point is motivated simply by the fact that Fourier series created for our motivating problem tend to have special cases at zero. The implementation of scalar addition will introduce special zero conditions. The second point is motivated by the set of rules described in Section 9.4.1. In addition, Maple has more simplifications for positive ranges.

Convolutions are split into three ranges, zero, positive and negative. The original convolution formula is broken up into $k = 0$, $k < 0$, and $k > 0$.

$$\operatorname{conv}(f, g)(k) = \sum_{j=-\infty}^{\infty} f(k-j)g(j) \tag{9.24}$$

Zero Case

When k is equal to 0 then the convolution simplifies to

$$\operatorname{conv}(f, g)(0) = \sum_{j=-\infty}^{\infty} f(-j)g(j) \tag{9.25}$$

This is split into into three distinct summations.

$$z_1 = \sum_{j=1}^{\infty} f(j)g(-j) \tag{9.26}$$

$$z_2 = f(0)g(0) \tag{9.27}$$

$$z_3 = \sum_{j=1}^{\infty} f(-j)g(j) \tag{9.28}$$

Each of these summations is simplified using the strategy described in Section 9.4.1.

Positive Ranges

For positive ranges, assume that $k > 0$. The following sums can then be created

$$p_1 = \sum_{i=1}^{\infty} f(k+i)g(-i) \tag{9.29}$$

$$p_2 = f(k)g(0) \tag{9.30}$$

$$p_3 = \sum_{i=1}^{k} f(k-i)g(i) \tag{9.31}$$

$$p_4 = f(0)g(k) \tag{9.32}$$

$$p_5 = \sum_{i=1}^{\infty} f(-i)g(k+x) \tag{9.33}$$

There is one important note about how these sums are created. Function inputs with additions are not calculated directly. For example, $f(k+i)$ will first be evaluated as $f(m)$ and simplified using any assumptions on m. In this case, assuming that m is a positive integer. After this simplification is done the substitution $m = k+i$ is applied. Once the sums are created the rules from Section 9.4.1 are applied.

Negative Ranges

The approach behind negative k values mirrors that of the positive values. The ranges are different, but the motivation is the same. To handle the negative value of the number, sums are evaluated for $-|k|$ which is equivalent to k when $k < 0$.

$$n_1 = \sum_{i=1}^{\infty} f(i)g(i + |k|) \tag{9.34}$$

$$n_2 = f(0)g(-|k|) \tag{9.35}$$

$$n_3 = \sum_{i=1}^{|k|} f(-|k| + i)g(-i) \tag{9.36}$$

$$n_4 = f(-|k|)g(0) \tag{9.37}$$

$$n_5 = \sum_{i=1}^{\infty} f(-|k| - i)g(i) \tag{9.38}$$

The same process, used for positive ranges, of handling addition of function inputs using a two step process followed by simplification is performed.

Combining Ranges

Once each of these components has been simplified individually, they can be combined to determine a closed form for the convolution.

$$\operatorname{conv}(f, g)(k) = \begin{cases} p_1 + \cdots + p_5 & k > 0 \\ z_1 + z_2 + z_3 & k = 0 \\ n_1 + \cdots + n_5 & k < 0 \end{cases} \tag{9.39}$$

9.4.1 Simplification Rules

To simplify one algebraic expression over a range, the following steps are taken.

1. The expression is expanded into a sum of products normal form.
2. The normal form is then converted into a list of pairs. Each pair has an expression in terms of the summation variable as the first element and any constant multipliers as the second elements.
3. The pairs are then matched on known patterns.
4. After all matching has been completed, the pairs can be evaluated.
5. Finally, the sum can be created as a piecewise function.

9.4.2 Summation Patterns

The following patterns are matched to simplify series.

$$\sum_{k=x}^{y} \frac{1}{k^a} = Z(a, x, y) \tag{9.40}$$

$$\sum_{k=x}^{y} \psi(a, k) = P_{\text{sum}}(a, 0, x, y) \tag{9.41}$$

$$\sum_{k=x}^{y} \frac{\psi(a, k)}{k^b}|_{b>0} = P_{\text{sum}}(a, b, x, y) \tag{9.42}$$

$$\sum_{k=x}^{y} \frac{1}{(n-k)^a k^b} = P_{\text{nmx}}(a, b, n, x, y) \tag{9.43}$$

$$\sum_{k=x}^{y} \frac{1}{(n+k)^a k^b} = P_{\text{npx}}(a, b, n, x, y) \tag{9.44}$$

$$\sum_{k=x}^{y} \frac{1}{(k-n)^a k^b} = P_{\text{xmn}}(a, b, n, x, y) \tag{9.45}$$

$$\sum_{k=x}^{y} \psi_a(k)\psi_b(k) = P_{\text{sq0}}(a, b, x, y) \sum_{k=x}^{y} \frac{\psi_a(k)\psi_b(k)}{k^c}|_{c>0} = \quad P_{\text{sq1}}(a, b, c, x, y) \tag{9.46}$$

$$\sum_{k=x}^{y} \frac{\psi_a(k)}{k^b(n+k)^c} = P_{\text{sump}}(a, b, c, m) \tag{9.47}$$

$$\sum_{k=x}^{y} \frac{\psi_a(k)}{k^b(n-k)^c} = P_{\text{sumn}}(a, b, c, m) \tag{9.48}$$

$$\sum_{k+n\neq 0} \frac{\psi_a(k)}{(k+n)^b k^c} = P_{\text{sum2}}(a, b, c, n) \tag{9.49}$$

$$\sum_{k=x}^{y} \psi_a(k) P_{\text{sum2}}(b, c, d, k) k^f = P_{\text{d3}}(a, b, c, d, f, x, y) \tag{9.50}$$

$$\sum_{k=x}^{y} \frac{P_{\text{sumn}}(\cdots)}{k^a} = D_{xx}(\cdots, a) \tag{9.51}$$

$$\sum_{k=x}^{y} \frac{P_{\text{sump}}(\cdots)}{k^a} = D_{xx}(\cdots, a) \tag{9.52}$$

The functions P_{nmx}, P_{npx}, and P_{xmn} can be solved by Maple's sum and simplify commands. No additional code is needed for these. The subscript of D_{xx} tells the range and type of P_{sum} function.

The Z function has three outcomes. These are all only valid when $a > 1$. When $a = 1$, then $Z(1, 1, \infty) = \infty$. For all functions, any case not caught returns the undefined function.

$$Z(a,1,\infty) = \zeta(a) \tag{9.53}$$

$$Z(a,n+1,\infty) = (-1)^a \frac{\psi_{a-1}(n)}{(a-1)!} - \frac{1}{n^a} \tag{9.54}$$

$$Z(a,1,n+1) = (-1)^{a+1} \frac{\psi_{a-1}(n)}{(a-1)!} + \zeta(a) \tag{9.55}$$

The P_{sum} function uses a helper G.

$$G(a,b) = \sum_{x=1}^{\infty} \frac{H_{x,a}}{x^b} \tag{9.56}$$

$$G(a,b,c) = \sum_{x=1}^{\infty} \frac{H_{x,a} H_{x,b}}{x^c} \tag{9.57}$$

$$G(1,2) = 2\zeta(3) \tag{9.58}$$

$$G(1,m)|_{m>2} = \frac{1}{2}\left((m+2)\zeta(m+1) - \sum_{n=1}^{m-2} \zeta(m-n)\zeta(n+1)\right) \tag{9.59}$$

$$G(m,m)|_{m>1} = \frac{1}{2}\left(\zeta(m)^2 \zeta(2m)\right) \tag{9.60}$$

$$G(2,4) = \zeta(3)^2 - \frac{1}{3}\zeta(6) \tag{9.61}$$

$$G(4,2) = \frac{37}{12}\zeta(6) - \zeta(3)^2 \tag{9.62}$$

$$G(2,3) = 3\zeta(2)\zeta(3) - \frac{9}{2}\zeta(5) \tag{9.63}$$

$$G(3,2) = \frac{11}{2}\zeta(5) - 2\zeta(2)\zeta(3) \tag{9.64}$$

Also, note that

$$G(a,b) = \zeta(a)\zeta(b) + \zeta(a+b) - G(b,a) \tag{9.65}$$

The $P_{\text{sum}}(a,b,x,y)$ function only simplifies when $b \geq 0$, $x = 1$ and $y = \infty$.

$$P_{\text{sum}}(a,b,1,\infty)|_{b>1,a>0} = (-1)^{a+1}a!\,(\zeta(a+1)\zeta(b) + \zeta(a+b+1) - G(a+1,b)) \tag{9.66}$$

$$P_{\text{sum}}(0,b,1,\infty)|_{b>1} = -\gamma\zeta(b) + G(1,b) - \zeta(b+a+1) \tag{9.67}$$

$$P_{\text{sum}}(a,1,1,\infty) = (-1)^{a+1}a!G(m,a+1) \tag{9.68}$$

$$P_{\text{sum}}(a,0,1,\infty)|_{a>1} = (-1)^{a-1}a!\zeta(a) \tag{9.69}$$

There is only one solution known for $P_{\text{sq0}}(a,b,x,y)$. An approximate solution is also used, but only when numerical computations are done.

$$P_{\text{sq0}}(1,3,1,\infty) = 45\zeta(5) - 18\zeta(3)\zeta(2) \tag{9.70}$$

$$P_{\text{sq0}}(3,1,1,\infty) = P_{\text{sq0}}(1,3,1,\infty) \tag{9.71}$$

$$P_{\text{sq0}}(1,1,1,\infty) = 3.60617070947878285 \tag{9.72}$$

$$\tag{9.73}$$

The library also only has one known solution for $P_{\text{sq1}}(a,b,c,x,y)$.

$$P_{\text{sq1}}(0,3,1,1,\infty) = -18\gamma\zeta(5) + 6\gamma\zeta(3)\zeta(2) + \frac{105}{8}\zeta(6) - 9\zeta(3)^2 \tag{9.74}$$

$$P_{\text{sq1}}(3,0,1,1,\infty) = P_{\text{sq1}}(0,3,1,1,\infty) \tag{9.75}$$

$$P_{\text{sq1}}(0,0,3,1,\infty) = \zeta(3)\gamma^2 - \frac{1}{180}\gamma\pi^4 - \frac{3}{2}\zeta(5) + \frac{1}{6}\pi^2\zeta(3) \tag{9.76}$$

$$P_{\text{sq1}}(0,1,2,1,\infty) = -\frac{7}{360}\gamma\pi^4 + \frac{1}{12}\pi^2\zeta(3) - \frac{1}{72}\pi^4X(1) + \zeta(5) + \frac{1}{2}G_2(2,2,-1) \tag{9.77}$$

$$P_{\text{sq1}}(1,0,2,1,\infty) = P_{\text{sq1}}(0,1,2,1,\infty) \tag{9.78}$$

$$P_{\text{sq1}}(1,1,1,1,\infty) = \frac{1}{36}\pi^4X(1) - \frac{1}{3}\pi^2(\frac{1}{6}\pi^2X(1) - \zeta(3)) - \frac{2}{3}\pi^2\zeta(3) + G_2(2,2,-1) + 10\zeta(5) \tag{9.79}$$

$$\tag{9.80}$$

Some of the expression are not know exactly. They are evaluated using the below values if a numerical answer is requested, but remain symbolic for exact calculations.

$$P_{\text{sq1}}(0,0,2,1,\infty) = 2.1367505174652068044479905 8 \tag{9.81}$$

$$P_{\text{sq1}}(0,1,3,1,\infty) = -.8894998046438163845241 \tag{9.82}$$

$$P_{\text{sq1}}(1,1,3,1,\infty) = 2.836377920750922562 3 \tag{9.83}$$

$$P_{\text{sq1}}(0,0,4,1,\infty) = 0.3726076126807983213837 5 \tag{9.84}$$

$$P_{\text{sq1}}(1,2,1,1,\infty) = -4.115466336142813225664 2 \tag{9.85}$$

$$P_{\text{sq1}}(0,2,2,1,\infty) = 1.315515926331496906856 2 \tag{9.86}$$

$$P_{\text{sq1}}(1,1,2,1,\infty) = 2.836377920750922562384 \tag{9.87}$$

After all the replacements has been made, rebuilding the expression and simplifying it algebraically are handled by Maple's built-in tools.

9.5 Order of Evaluation

The order of multiplication can lead to divergent subexpressions. This is an additional motivation to delay evaluation of convolutions until all convolutions have been collected. A series can be examined to make predictions about the best way to order convolutions. The below example comes from evaluation of the eigenvalue problem. Only truncated functions are shown to highlight the problem terms.

$$a(k) = \begin{cases} 0 & k = 0 \\ \frac{i(-1)^k \rho}{\sqrt{\pi}} + O\left(\frac{1}{N}\right) & \text{otherwise} \end{cases} \tag{9.88}$$

$$b(k) = \begin{cases} 0 & k = 0 \\ \frac{i(-1)^k \rho}{Nk\sqrt{\pi}} + O\left(\frac{1}{N^2}\right) & \text{otherwise} \end{cases} \tag{9.89}$$

$$c(k) = \begin{cases} \frac{-2\zeta(2)}{N^2} + O\left(\frac{1}{N^3}\right) & k = 0 \\ \frac{(-1)^k}{N^2 k^2} + O\left(\frac{1}{N^3}\right) & \text{otherwise} \end{cases} \tag{9.90}$$

Our goal is to calculate the convolution

$$\mathrm{conv}\,(a,b,c,c)\,(0) \tag{9.91}$$

Since convolutions represent multiplications, the order in which the functions are multiplied will not change the final answer. It can, however, affect the intermediate calculations. Solving the convolution of a and b first leads to an infinity.

$$\sum_{k=-\infty}^{\infty} a(n-k)b(k) = \sum_{k\neq 0} \frac{(-1)^{n+1}\rho^2}{Nk\pi} \tag{9.92}$$

$$= \frac{(-1)^{n+1}\rho^2}{N\pi}\zeta(1) = \infty \tag{9.93}$$

At this point, symbolic and numerical results can have a problem. This problem is avoided by reordering the inputs

$$\mathrm{conv}\,(a,c,c,b)\,(0) = \frac{-24\rho^2\zeta(2)\zeta(3)}{\pi N^5} + \frac{60\rho^2\zeta(5)}{\pi N^5} \tag{9.94}$$

Determining the convolutions (a,c) and (b,c) avoids these problems. This is done by sorting the functions in the convolution. Although every ordering is mathematically equivalent, some produce intermediate infinities. The expressions are ordered to avoid this.

9.6 Example: Poisson's Equation Revisited

The example from Chapter 4.7 gave the first two terms in the target series. The third term required manipulation of infinite series. With the introduction of the Fourier library described in this chapter, the next term can be found.

The second energy variation, E_2 is given by the CMS expression.

$$E_2 = -\frac{1}{2} - 2CN^i\nabla_1 u_1 - \dot{\nabla} CN^i\nabla_i u$$
$$+ CZ^i_\alpha \nabla^\alpha C\nabla_i u - C^2 N^i N^j \nabla_i \nabla_j u \tag{9.95}$$

New symbols are needed. All but u_1 can easily be created.

```
C1:=ddt(surface_velocity()):
C1:=restrictTime(C1):
NI:=surface_normal(1):
BAb:=curvature_tensor(2,-2):
```

The three terms that do not depend on u_1 can be generated and integrated using the same methods as in Chapter 4.7. First, the terms are created.

```
Term2:=contract(CMSScalar(1/2)*C1*ddZi(u)*ddZI(u)
          ,[1,2]):
Term3:=contract(exponent(C0,2)*NI*ddZi(ddZi(u))
          *ddZI(u),[1,2,3,4]):
Term4:=contract(CMSScalar(-1/2)*exponent(C0,2)*BAb
          *ddZi(ddZI(u)),[1,2,3,4]):
```

Then they can be added and integrated.

```
partial:=Term2+Term3+Term4:
partial:=getCompts(restrictTime(partial))[1]:
partial:=eval(partial,psi=theta/N):
partial:=convert(expand(series(partial,N=infinity)),'polynom'):
partial:=expand(int(partial,theta=-Pi..Pi));
```

This gives a result of

$$\frac{1}{5}\frac{\pi^5}{N^4} \tag{9.96}$$

For the final term in E_2, two series representations are introduced.

$$C = \sum_{k=-\infty}^{\infty} c_0 e^{ikN\theta} \tag{9.97}$$

$$u_1 = \sum_{k=-\infty}^{\infty} f_1(k,r)^{ikN\theta} \tag{9.98}$$

To define these the Fourier library must be included.

```
libname :=
          "./CMSTensors",
          "./ExactConvolutionSolver",
          "./FourierCoefficientManip",
          libname;
with(CMSTensors);
with(ExactConvolutionSolver);
with(FourierCoefficientManip);
```

The library also needs to be initialized for derivatives and integration.

```
initialize_fourier([r],[theta,psi]);
```

These are defined using the library as

```
c0:=w->eval(piecewise(
          k=0,
          -2*Zeta(2)/N^2
          ,
          (-1)^k/(N^2*k^2)
),k=w);
f1:=w->eval(piecewise(
          k=0,
          c0(0)*(r*J(1,r*rho)-J(0,r*rho))/(2*J(1,1))
          ,
          c0(k)*J(k*N,r*rho)/(J(k*N,rho)*2)
          ),k=w);

C0_compts:=FourierSum(c0,k->I*k*N*psi,c[0]):
C0:=CMSObject([],Array([C0_compts]),true):
u1_compts:= FourierSum(f1,k->I*k*N*theta,f[1]):
u1:=CMSObject([],Array([u1_compts])):
```

The definition of u_1 is determined from conditions detailed in [10]. The method for computing these partial derivatives is explained, for different conditions, in Chapter 7.

The library can be used in conjunction with the CMS library to compute the last integrand.

```
Term1:=contract(C0*ddZi(u1)*ddZI(u),[1,2]):
```

The integral is the coefficient at 0 times 2π.

```
Term1:=CMSScalar(2*Pi)*Term1:
Term1:=getCompts(restrictTime(Term1))[1]:
```

This can be approximated numerically.

```
EVAL_METHOD:=EVAL_COEFFICIENT_NUMERICALLY:
Term1_approx:=sort(eval(getCoeffAt(Term1,0),rho=1),N);
```

This gives a series that starts with the term

$$\frac{3.774886343}{N^3} + O\left(\frac{1}{N^4}\right) \tag{9.99}$$

It can also be solved exactly.

```
EVAL_METHOD:=EVAL_COEFFICIENT_ALGEBRAICALLY:
Term1_exact:=sort(eval(getCoeffAt(Term1,0),rho=1),N);
```

This gives a series that starts with

$$\frac{\pi\zeta(3)}{N^3} + O\left(\frac{1}{N^4}\right) \tag{9.100}$$

This also produces terms of higher order, but these will be affected by higher order variations. Only the N^{-3} component is completely known at this point. The current series is computed as

```
energy[2]:=Term1_exact+partial:
E[N]=sum(1/i!*energy[i],i=0..2);
```

This gives the same solution presented in [10].

$$E_N = -\frac{1}{16}\pi + \frac{1}{12}\frac{\pi^3}{N^2} + \frac{1}{2}\frac{\pi\zeta(3)}{N^3} + O\left(\frac{1}{N^4}\right) \tag{9.101}$$

10. Level Set Surface

10.1 Introduction

In this chapter, we provide additional evidence to support the correctness of our calculations. We introduce another deformation that uses the same expressions but has a predictable answer. These calculations are an exact match for the other eigenvalue problems up until the convolutions are evaluated. The values of the c_k coefficients are computed differently. This leads to each variation being equal to zero.

Two deforming manifolds have been defined so far for the same CMS expressions, an ellipse and a polygon. When deforming the unit circle into the regular N-sided polygon, the CMS expressions are evaluated two ways, numerically and symbolically. If these results match, it shows the expressions were evaluated correctly. Independent numerical calculations for specific values of N are also used to give support for the correctness of the expressions [58].

Numerical methods are more promising for the deformation of the unit circle into the ellipse with semi-axis $A = 1$ and $B = 1 + \epsilon$. Using entirely independent numerical calculations, this series can be approximated to extremely high accuracy for different values of ϵ.

Both these problems use the same CMS expressions. In both cases, confirmation of the calculations is done through numerical means. In this Chapter, a third deformation is introduced where the deformation itself is non-trivial, but the answer is predicable and trivial. This deformation will lead to all eigenvalues except λ_0 being 0.

10.2 Computation of Surface Velocity

The surface velocity C_0 for the level set is

$$C_0 = \left(\nabla_i F \nabla^i F\right)^{-1/2} F_t \tag{10.1}$$

The value of F is chosen to ensure that all eigenvalues after λ_0 will be 0.

$$\rho = \sqrt{\lambda} \tag{10.2}$$

$$F = J_0(\rho r) + \alpha t J_1(\rho r)\cos\theta \tag{10.3}$$

$$F|_{t=0} = J_0(\rho r) \tag{10.4}$$

$$F_t = \alpha J_1(\rho r)\cos\theta \tag{10.5}$$

The value of C_0 is computed at $t = 0$.

$$C_0 = \left(\nabla_i F \nabla^i F\right)^{-1/2} F_t \tag{10.6}$$

$$= \left(\begin{bmatrix} \frac{dF}{dr} \\ \frac{dF}{d\theta} \end{bmatrix}_i \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{r^2} \end{bmatrix}^{ij} \begin{bmatrix} \frac{dF}{dr} \\ \frac{dF}{d\theta} \end{bmatrix}_j \right)^{-1/2} F_t \tag{10.7}$$

$$= \left(\begin{bmatrix} \frac{dF}{dr}^2 & \frac{dF}{dr}\frac{dF}{d\theta} \\ \frac{1}{r^2}\frac{dF}{d\theta}\frac{dF}{dr} & \frac{1}{r^2}\frac{dF}{d\theta}^2 \end{bmatrix}_i^i \right)^{-1/2} F_t \tag{10.8}$$

$$= \left(\frac{dF}{dr}^2 + \frac{1}{r^2}\frac{dF}{d\theta}^2 \right)^{-1/2} F_t \tag{10.9}$$

$$= \left(\frac{dF}{dr}^2 \right)^{-1/2} F_t \tag{10.10}$$

$$= \left((-J_1(\rho r)\rho)^2 \right)^{-1/2} F_t \tag{10.11}$$

$$= \left(J_1(\rho r)^2 \rho^2 \right)^{-1/2} F_t \tag{10.12}$$

$$= \frac{F_t}{J_1(\rho r)\rho} \tag{10.13}$$

$$= \frac{\alpha J_1(\rho r)\cos\theta}{J_1(\rho r)\rho} \tag{10.14}$$

$$= \frac{\alpha \cos\theta}{\rho} \tag{10.15}$$

Next, C_0 is converted to a Fourier series.

$$C_0 = \sum_{k=-\infty}^{\infty} c_0 e^{ik\theta} \tag{10.16}$$

An integral is taken to determine the Fourier coefficients.

$$c_0(k) = \frac{1}{2\pi}\int_{\pi}^{\pi} \frac{-\alpha\cos\theta}{\rho} e^{-ik\theta} \tag{10.17}$$

$$c_0(1) = \frac{1}{2}\frac{\alpha}{\rho} \tag{10.18}$$

$$c_0(-1) = \frac{1}{2}\frac{\alpha}{\rho} \tag{10.19}$$

$$c_0(|k| \neq 1) = 0 \tag{10.20}$$

10.3 Derivatives of the Surface Velocity

The invariant time derivatives of the surface velocity must be computed using the CMS.

To compute $C_1 = \dot{\nabla} C_0$, the rules of the CMS are applied. This is automated through the use of the CMSTRS.

$$\dot{\nabla} C_0 = \dot{\nabla}\left(\left(\nabla_i F \nabla^i F\right)^{-1/2} F_t\right) \tag{10.21}$$

$$C_1 = \dot{\nabla} F_t \left(\nabla_i F \nabla^i F\right)^{-1/2} + F_t \dot{\nabla}\left(\nabla_i F \nabla^i F\right)^{-1/2} \tag{10.22}$$

$$\begin{aligned} &= \left(\nabla_i F \nabla^i F\right)^{-1/2}\left(F_{tt} + CN^j \nabla_j F_t\right) \\ &\quad - \frac{1}{2} F_t \left(\nabla_j F \nabla^j F\right)^{-3/2} \dot{\nabla}\left(\nabla_i F \nabla^i F\right) \end{aligned} \tag{10.23}$$

$$\begin{aligned} &= F_{tt}\left(\nabla_i F \nabla^i F\right)^{-1/2} + CN^j \nabla_j F_t \left(\nabla_i F \nabla^i F\right)^{-1/2} \\ &\quad - \frac{1}{2} F_t \left(\nabla_k F \nabla^k F\right)^{-3/2}\left(2\nabla_i F_t \nabla^i F + 2CN^j \nabla_j \nabla_i F \nabla^i F\right) \end{aligned} \tag{10.24}$$

$$\begin{aligned} &= F_{tt}\left(\nabla_i F \nabla^i F\right)^{-1/2} + CN^j \nabla_j F_t \left(\nabla_i F \nabla^i F\right)^{-1/2} \\ &\quad - F_t \nabla_i F_t \nabla^i F \left(\nabla_k F \nabla^k F\right)^{-3/2} - F_t CN^j \nabla_j \nabla_i F \nabla^i F \left(\nabla_k F \nabla^k F\right)^{-3/2} \end{aligned} \tag{10.25}$$

These expressions can be evaluated automatically using the CMSTensor library. The calculations are performed by hand here for C_1, but the difficulty increases significantly with C_2. Each term in the sum can be evaluated independently. First, the subexpressions are simplified.

$$\left(\nabla_i F \nabla^i F\right)^{-1/2} = \frac{1}{J_1(\rho r)\rho} \tag{10.26}$$

$$\left(\nabla_i F \nabla^i F\right)^{-3/2} = \frac{1}{J_1(\rho r)^3 \rho^3} \tag{10.27}$$

$$F_{tt} = \frac{d}{dt}\left(\alpha J_1(\rho r)\cos\theta\right) = 0 \tag{10.28}$$

$$\nabla_i F_t \nabla^i F = \frac{dF_t}{dr}\frac{dF}{dr} + \frac{1}{r^2}\frac{dF_t}{d\theta}\frac{dF}{d\theta} \tag{10.29}$$

$$= \frac{dF_t}{dr}\frac{dF}{dr} + 0 \tag{10.30}$$

$$= J_1(\rho)^2 \rho\alpha\cos\theta \tag{10.31}$$

The terms of the sum are now computed. C_1 is a surface object, so the expressions are evaluated at $r = 1$.

$$T_1 = F_{tt}\left(\nabla_i F \nabla^i F\right)^{-1/2} \tag{10.32}$$

$$= 0 \tag{10.33}$$

$$T_2 = C N^j \nabla_j F_t \left(\nabla_i F \nabla^i F\right)^{-1/2} \tag{10.34}$$

$$= \left(\frac{\alpha\cos\theta}{\rho}\right)\left(\frac{1}{J_1(\rho)\rho}\right) N^j \nabla_j F_t \tag{10.35}$$

$$= \frac{\alpha\cos\theta}{J_1(\rho)\rho^2}\begin{bmatrix} 1 \\ 0 \end{bmatrix}^j \begin{bmatrix} \frac{dF_t}{dr} \\ \frac{dF_t}{d\theta} \end{bmatrix}_j \tag{10.36}$$

$$= \frac{\alpha\cos\theta}{J_1(\rho)\rho^2}\begin{bmatrix} \frac{dF_t}{dr} & 0 \\ \frac{dF_t}{d\theta} & 0 \end{bmatrix}^j_j \tag{10.37}$$

$$= \frac{\alpha\cos\theta}{J_1(\rho)\rho^2}\frac{dF_t}{dr} \tag{10.38}$$

$$= \frac{\alpha\cos\theta}{J_1(\rho)\rho^2}\left(\alpha\rho\cos\theta J_0(\rho) - \frac{\alpha\cos\theta J_1(\rho)}{1}\right) \tag{10.39}$$

$$= \frac{\alpha^2\cos^2\theta}{\rho}\frac{J_0(\rho)}{J_1(\rho)} - \frac{\alpha^2\cos^2\theta}{\rho^2} \tag{10.40}$$

$$= -\frac{\alpha^2\cos^2\theta}{\rho^2} \tag{10.41}$$

$$T_3 = - F_t \nabla_i F_t \nabla^i F \left(\nabla_k F \nabla^k F\right)^{-3/2} \quad (10.42)$$

$$= - (\alpha J_1(\rho) \cos\theta)) \left(J_1(\rho)^2 \rho \alpha \cos\theta\right) \left(\frac{1}{J_1(\rho)^3 \rho^3}\right) \quad (10.43)$$

$$= - \frac{\alpha^2 \cos^2\theta}{\rho^2} \quad (10.44)$$

$$T_4 = - F_t C N^j \nabla_j \nabla_i F \nabla^i F \left(\nabla_k F \nabla^k F\right)^{-3/2} \quad (10.45)$$

$$= - \left(\frac{1}{J_1(\rho)^3 \rho^3}\right) (\alpha J_1(\rho) \cos\theta) C N^j \nabla_j \nabla_i F \nabla^i F \quad (10.46)$$

$$= - \frac{\alpha \cos\theta}{J_1(\rho)^2 \rho^3} \left(\frac{\alpha \cos\theta}{\rho}\right) N^j \nabla_j \nabla_i F \nabla^i F \quad (10.47)$$

$$= - \frac{\alpha^2 \cos^2\theta}{J_1(\rho)^2 \rho^4} N^j \nabla_j \nabla_i F \nabla^i F \quad (10.48)$$

$$= - \frac{\alpha^2 \cos^2\theta}{J_1(\rho)^2 \rho^4} \left(\frac{d^2F}{dr^2} \frac{dF}{dr}\right) \quad (10.49)$$

$$= - \frac{\alpha^2 \cos^2\theta}{J_1(\rho)^2 \rho^4} \left(-\rho^2 J_1(\rho)^2\right) \quad (10.50)$$

$$= - \frac{\alpha^2 \cos^2\theta}{\rho^2} \quad (10.51)$$

These combined to give the expression for C_1.

$$C_1 = 0 - \frac{\alpha^2 \cos^2\theta}{\rho^2} - \frac{\alpha^2 \cos^2\theta}{\rho^2} + \frac{\alpha^2 \cos^2\theta}{\rho^2} \quad (10.52)$$

$$= \frac{-\alpha^2 \cos^2\theta}{\rho^2} \quad (10.53)$$

The integral used to determine the Fourier coefficients is only nonzero at three points.

$$c_1(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{-\alpha^2 \cos^2\theta}{\rho^2} e^{-ik\theta} \quad (10.54)$$

$$c_1(2) = - \frac{1}{4} \frac{\alpha^2}{\rho^2} \quad (10.55)$$

$$c_1(0) = - \frac{1}{2} \frac{\alpha^2}{\rho^2} \quad (10.56)$$

$$c_1(-2) = - \frac{1}{4} \frac{\alpha^2}{\rho^2} \quad (10.57)$$

$$c_1(\text{otherwise}) = 0 \quad (10.58)$$

10.4 Computation of λ

The first lambda value is trivially computed using the general solution to λ_1.

$$\lambda_1 = -2\rho^2 c_0 = -2\rho^2 0 = 0 \tag{10.59}$$

The second lambda value requires expanding convolutions.

$$\lambda_2 = 2\rho^2 \text{conv}(c_0, c_0)(0) + 4\sqrt{\pi}\rho \text{conv}(c_0, f_{1,dr})(0) - 2\rho^2 c_1(0) \tag{10.60}$$

These convolutions each have a finite number of terms and can be computed directly.

$$-2\rho^2 c_1(0) = -2\rho^2 \frac{-1}{2}\frac{\alpha^2}{\rho^2} = \alpha^2 \tag{10.61}$$

$$2\rho^2 \text{conv}(c_0, c_0)(0) = 2\rho^2 \sum_{k=-\infty}^{\infty} c_0(k) c_0(-k) \tag{10.62}$$

$$= 4\rho^2 (c_0(1) c_0(-1)) \tag{10.63}$$

$$= 4\rho^2 \left(\frac{1}{2}\frac{\alpha}{\rho}\right)\left(\frac{1}{2}\frac{\alpha}{\rho}\right) \tag{10.64}$$

$$= \alpha^2 \tag{10.65}$$

The last term requires the definition for $f_{1,dr}$ be evaluated on this deformation.

$$u_1 = \sum_{k=-\infty}^{\infty} f_1(k) e^{ik\theta} \tag{10.66}$$

$$f_1 = \begin{cases} \frac{1}{2}\frac{\lambda_1 (J_0(\rho r) - rJ(1,\rho r)\rho)}{\rho^2 \sqrt{\pi} J_1(\rho)} & k = 0 \\ \frac{c_0(k)\rho}{\sqrt{\pi}} \frac{J_{|k|}(\rho r)}{J_{|k|}(\rho)} & k \neq 0 \end{cases} \tag{10.67}$$

$$f_{1,dr} = \begin{cases} \frac{c_0(0)\rho}{\sqrt{\pi}} & k = 0 \\ \frac{c_0(k)\rho}{\sqrt{\pi}} \frac{\rho J'_{|k|}(\rho)}{J_{|k|}(\rho)} & k \neq 0 \end{cases} \tag{10.68}$$

Since c_0 is only nonzero at two points, this function simplifies greatly.

$$f_{1,dr} = \begin{cases} \frac{c_0(1)\rho}{\sqrt{\pi}} \frac{\rho J'_{|1|}(\rho)}{J_{|1|}(\rho)} & k = 1 \\ \frac{c_0(1)\rho}{\sqrt{\pi}} \frac{\rho J'_{|-1|}(\rho)}{J_{|-1|}(\rho)} & k = -1 \\ 0 & |k| \neq 1 \end{cases} \tag{10.69}$$

$$= \begin{cases} \frac{-c_0(1)\rho}{\sqrt{\pi}} & k = 1 \\ \frac{-c_0(1)\rho}{\sqrt{\pi}} & k = -1 \\ 0 & |k| \neq 1 \end{cases} \tag{10.70}$$

$$= \begin{cases} \frac{-\alpha}{2\sqrt{\pi}} & k = 1 \\ \frac{-\alpha}{2\sqrt{\pi}} & k = -1 \\ 0 & |k| \neq 1 \end{cases} \tag{10.71}$$

The last term is computed by expanding the convolutions.

$$4\sqrt{\pi}\rho\text{conv}(c_0, f_{1,dr})(0) = 4\sqrt{\pi}\rho \sum_{k=-\infty}^{\infty} c_0(k) f_{1,dr}(-k) \tag{10.72}$$

$$= 4\sqrt{\pi}\rho \left(c_0(1) f_{1,dr}(-1) + c_0(-1) f_{1,dr}(1)\right) \tag{10.73}$$

$$= 4\sqrt{\pi}\rho \left(\left(\frac{1}{2}\frac{\alpha}{\rho}\right)\left(-\frac{\alpha}{2\sqrt{\pi}}\right) + \left(\frac{1}{2}\frac{\alpha}{\rho}\right)\left(-\frac{\alpha}{2\sqrt{\pi}}\right) \right) \tag{10.74}$$

$$= 4\sqrt{\pi}\rho \left(-\frac{\alpha^2}{4\rho\sqrt{\pi}} - \frac{\alpha^2}{4\rho\sqrt{\pi}} \right) \tag{10.75}$$

$$= 4\sqrt{\pi}\rho \left(-\frac{\alpha^2}{2\rho\sqrt{\pi}} \right) \tag{10.76}$$

$$= -2\alpha^2 \tag{10.77}$$

Adding these values together shows that they cancel out.

$$\lambda_2 = \alpha^2 + \alpha^2 - 2\alpha^2 = 0 \tag{10.78}$$

The first and second λ values are zeroed. By design all higher lambda values will also be zero.

$$\lambda_0 = \rho^2 \tag{10.79}$$

$$\lambda_1 = 0 \tag{10.80}$$

$$\lambda_2 = 0 \tag{10.81}$$

$$\lambda_3 = 0 \tag{10.82}$$

$$\lambda_4 = 0 \tag{10.83}$$

This deformation gives an additional justification for the accuracy of the eigenvalue computations. On this boundary, the solution is known. The evaluation of this problem adds additional support that the CMS expressions and the convolution expressions found for the Laplacian eigenvalues are correct. This provides an additional method of testing for all our libraries.

11. Results and Analysis

11.1 Introduction

This chapter examines the results of our experiments with the SCMS. The results of all the problems described in earlier chapters are given. These results are analyzed to show that the SCMS solved each problem.

The final series found for each problem is also given. Expression swell will be analyzed for each problem. In all cases, the final answers are simple, but intermediate calculations are very large. The contour length problem from Chapter 5 is examined first. This problem has a well known answer that can be easily compared to.

Next, the level set problem is examined. This problem was shown in Chapter 10. This is the first of three Laplace eigenvalue problems on Dirichlet boundary conditions. This problem has a trivial solution and adds additional support to the correctness of our CMS expressions.

The two deformations described in Chapter 6 are examined last. First, the deformation of the unit circle to the ellipse with semi-axis 1 and $1+\epsilon$. This series can be numerically approximated to high orders for verification. The series presented here has the most terms found to date.

The most difficult and interesting problem will be reviewed last, the deformation from the unit circle to the N-sided regular polygon. In this series, we present two previously unknown terms and confirm them using our own library for numerical aproximations. Full tables for all terms evaluated in the N-sided regular polygon deformation can be found in Appendix A.

All computations in this section were completed in OS X Yosemite (10.10.5) with a dual-core 2.9 GHz Intel Core i7 and 8GB 1600 MHz DDR3 ram. The Java VM is Java(TM) SE Runtime Environment (build 1.8.0_60-b27) and the compiler is javac version 1.8.0_60. The version of Maple used in 17.00, Maple Build ID 813473.

11.2 Contour Length

The contour length problem was originally described in Chapter 5. In this problem, the unit circle was deformed into a ellipse with semi-axis $A = 1+\epsilon$ and $B = 1$. A series expansion of the contour length of the surface was then given in terms of ϵ. This series has a know answer which can be computed using alternative means.

The contour length is the integral of 1 over the surface. The surface is depended on time t. It is the unit circle at $t = 0$ and the ellipse at $t = 1$.

$$L(t) = \int 1dS \tag{11.1}$$

In Maple the below command can expand the series for any number of terms.

```
restart;
assume(epsilon >0);
max_order:=9;
L:=int(
        sqrt( (1+epsilon)^2*cos(theta)^2+sin(theta)^2)
        ,theta=0..2*Pi
        );
expand(series(L/Pi, epsilon=0,max_order));
```

The output of this command is the following series for the contour length $L(t)$ at time $t = 1$. Each term has a factor of π which is divided out for clarity.

$$\frac{L(1)}{\pi} = 2 + \epsilon + \frac{\epsilon^2}{8} - \frac{\epsilon^3}{16} + \frac{17\epsilon^4}{512} - \frac{19\epsilon^5}{1024} + \frac{89\epsilon^6}{8192} - \frac{109\epsilon^7}{16384} + O\left(\epsilon^8\right) \tag{11.2}$$

The values computed using the SCMS give the same series. The variations are computed as follows.

$$L_0 = L(0) = 2\pi \tag{11.3}$$

$$L_1 = \dot{\nabla} L\Big|_{t=0} = \pi\epsilon \tag{11.4}$$

$$L_2 = \dot{\nabla}^2 L\Big|_{t=0} = \frac{\pi\epsilon^2}{4} \tag{11.5}$$

$$L_3 = \dot{\nabla}^3 L\Big|_{t=0} = -\frac{3\pi\epsilon^3}{8} \tag{11.6}$$

$$L_4 = \dot{\nabla}^4 L\Big|_{t=0} = \frac{51\pi\epsilon^4}{64} \tag{11.7}$$

$$L_5 = \dot{\nabla}^5 L\Big|_{t=0} = -\frac{285\pi\epsilon^5}{128} \tag{11.8}$$

$$L_6 = \dot{\nabla}^6 L\Big|_{t=0} = \frac{4005\pi\epsilon^6}{512} \tag{11.9}$$

$$L_7 = \dot{\nabla}^7 L\Big|_{t=0} = -\frac{34335\pi\epsilon^7}{1024} \tag{11.10}$$

Creating the series expansion from these expressions gives Equation 11.2.

$$\frac{L(1)}{\pi} = \sum_{k=0}^{7} \frac{L_k}{k!\pi} = (\text{Eq } 11.2) \tag{11.11}$$

Automation of these solutions begins with the CMSTRS library. Application of the CMSTRS library is detailed in Table 11.1. The expression size shows the number of CMS products that are added together to make the final expression. This number is growing roughly exponentially, L_k has approximately 3^{k-1} terms. The number of rules applied is also growing roughly exponentially. The CMSTRS library can apply over 100,000 rules in under 20 seconds. Combining equivalent terms to simplify the expression does not contribute significantly to the computations. Only on the last two orders L_6 and L_7 do these computations overtake reductions. In the last order computed, 19,265 terms are combined taking a total of 6 minutes.

The expressions are next output to Maple for evaluation in a polar coordinate system. The main computational complexity in these expressions is the surface velocity C. Computation of $\dot{\nabla}^5 C$ takes 2 minutes, 4 seconds, and 719 milliseconds in Maple. The next order, $\dot{\nabla}^6 C$, takes 47 minutes, 19 seconds, and 301 milliseconds. This is by far the most demanding part of the computations. The time to evaluate the expressions in Maple once the C values have been computed is given in Table 11.2.

Table 11.1: This table shows the results of generating CMS expressions for the contour length problem using the CMSTRS library.

Order	Expr. Size	Rules	Time to Reduce	Combined	Comb. Time
L_1	1	8	11ms	0	0ms
L_2	2	36	11ms	1	4ms
L_3	8	118	36ms	3	3ms
L_4	27	946	204ms	75	36ms
L_5	84	$4,765$	738ms	627	372ms
L_6	264	$23,418$	3s 499ms	$3,354$	11s 522ms
L_7	827	$105,473$	18 sec 851ms	$19,265$	6m 29s

Table 11.2: This table shows time taken to evaluate the CMS expressions for the contour length on the circle to ellipse deformation in Maple.

Order	Time to Evaluate
L_0	19ms
L_1	63ms
L_2	308ms
L_3	213ms
L_4	544ms
L_5	1s 957ms
L_6	5s 302ms
L_7	18s 129ms

Computation of $\dot{\nabla}^k C$ is a purely algebraic problem. The value of C is derived in terms of t from the coordinate deformation. The time to compute $\dot{\nabla}^5 C$ is entirely devoted to allowing Maple to determine the derivative with respect to t. Computing this value for higher orders is not relevant to testing the SCMS. This time complexity is in Maple's underlying libraries. This is the motivation for stopping calculations at L_7. Computing L_8 would require $\dot{\nabla}^7 C$.

The contour length experiment uses the majority of rules in the CMSTRS libraries. Rules related to spatial integration and the eigenfunction u are not applied in this problem. Since the exact solution to this problem can be found using other automated means, confirming the series presented against the exact series shows that the SCMS accurately replicates the calculations of the CMS.

11.3 Level Set Problem

The level set problem described in Chapter 10 uses all the rules in the CMSTRS and also has a known answer. This problem uses the same expressions as our other eigenvalue problems. Since CMS expressions are can be evaluated in any coordinate system, the same expressions can be evaluated three different ways for three different problems. This deformation ends in the same state it started at $t = 1$. This means all the variations will be equal to 0.

Computation of the surface velocity presents a problem here as well. The value of C is itself a CMS expression. The SCMS must first be used to evaluate these values before they can be plugged into the eigenvalue formulas.

Table 11.3 shows time and size complexity of the CMS expressions. In this problem, combining equivalent terms becomes the bottleneck. If like terms are not combined, then C_4 will have 11,506 terms. This will just move the bottleneck to the CMSTensor library.

Table 11.3: This table shows the growth in the expressions for the surface velocity and its derivatives for the level set problem.

Order	Expr. Size	Rules	Time to Reduce	Combined	Comb. Time
C_1	4	206	145ms	2	12ms
C_2	20	$1,985$	567ms	23	50ms
C_3	112	$17,582$	2s 197ms	195	2s 834ms
C_4	497	$152,083$	18s 305ms	$2,766$	1h 9m 46s

Once the C_4 values are computed, the remaining expressions exactly match those of the ellipse and polygon problem. The convolutions for the Laplacian Eigenvalues under Dirichlet boundary conditions can be evaluated with the new c_k Fourier coefficients. This surface deformation has a finite number of terms in each convolution. They can be solved exactly by expanding the convolutions far enough that all non-zero terms are included. The results are all zeros as expected. This adds more justification that our expressions have be computed correctly.

$$\lambda(t=1) = \lambda_0 + \lambda_1 + \frac{\lambda_2}{2!} + \frac{\lambda_3}{3!} + \frac{\lambda_4}{4!} \tag{11.12}$$

$$= \rho^2 + 0 + 0 + 0 + 0 \tag{11.13}$$

$$= \rho^2 \tag{11.14}$$

11.4 Ellipse Eigenvalues

In the ellipse version of the problem, the surface velocity is computed using the same means as the contour length problems. There are three different sets of CMS expressions that must be derived to solve the problem. These are all also used for the level set problem and the regular polygon problem.

To create eigenfunctions, the surface condition and normalization condition must be calculated. These are needed to one order lower then the target. We have computed up to λ_6, meaning only u_5 is needed. The value of u_1 is the starting point and requires no rule applications. The derivation of surface conditions only contributes a small amount of complexity in the CMS. The growth is show in Table 11.4. The normalization condition is also needed and is shown in Table 11.5.

Table 11.4: This table shows the growth in the expressions for the surface condition of u_k using the CMSTRS library.

Order	Expr. Size	Rules	Time to Reduce	Combined	Comb. Time
u_2	4	82	19ms	1	0ms
u_3	15	540	61ms	7	4ms
u_4	48	3,521	333ms	70	91 ms
u_5	147	17,634	1s 718ms	437	1s 332ms

Table 11.5: This table shows the growth in the expressions for the normalization condition of u_k using the CMSTRS library.

Order	Expr. Size	Rules	Time to Reduce	Combined	Comb. Time
u_2	11	42	11ms	1	1ms
u_3	40	648	101ms	32	4ms
u_4	132	4,149	640ms	370	50ms
u_5	439	21,268	3s 717ms	2,518	1s 58ms

The values of λ_k need to be computed to order $k = 6$. The growth in these expressions is shown in Table 11.6. Combining equivalent terms to save computational complexity in the Maple evaluation is the first bottleneck encountered here. Maple's integration command could not fully simplify the Bessel expressions that appear in the sixth normalization condition without using hypergeometric functions. Maple considers these expression simple, but it is possible that integrals exist only it terms of other Bessel functions. For a general series to present itself here, all integrals of Bessel functions must be expressions of Bessel Functions. Hypergeometric functions do not lead to a simple answer, although it is an equivalent answer. We do not compute the next order because of this unexpected computational problem not related to the CMS itself.

Table 11.6: This table shows the growth in the expressions for λ_k using the CMSTRS library.

Order	Expr. Size	Rules	Time to Reduce	Combined	Comb. Time
λ_2	4	155	43ms	2	3ms
λ_3	17	973	154ms	17	13ms
λ_4	57	6,595	679ms	143	124ms
λ_5	205	33,004	3s 320ms	661	3s 832ms
λ_6	726	179,921	20s 867ms	4,129	4m 40s 901ms

When evaluating in Maple, there are two steps to the process. First, the expressions are converted into convolutions. Additional simplification takes place during the creation of these convolution expressions, further decreasing the number of terms that need to be evaluated. The comparison between the size of the CMS expressions and the convolution expressions is shown in three tables. The λ_k expressions are in Table 11.7. The u_k functions are in two tables, the surface condition in Table 11.8 and the normalization condition in Table 11.9.

Table 11.7: The number of convolutions generated from the λ_k's CMS expression.

Order	CMS Expr.	Conv. Expr
λ_1	1	1
λ_2	4	3
λ_3	17	9
λ_4	57	27
λ_5	205	82
λ_6	726	253

Evaluation of these convolution is straightforward. Every Fourier coefficient function has a finite number of non-zero entries. The convolutions can be expanded far enough to include all non-zero entries. Maple can then simply the expressions using its built in routines. The first seven values are computed below both symbolically and numerically when ρ is the first zero of J_0.

Table 11.8: The number of convolutions generated from the u_k surface condition's CMS expression.

Order	CMS Expr.	Conv. Expr
u_1	1	1
u_2	4	3
u_3	15	9
u_4	48	23
u_5	147	62

Table 11.9: The number of convolutions generated from the u_k normalization condition's CMS expression.

Order	CMS Expr.	Conv. Expr
u_1	1	1
u_2	4	2
u_3	15	5
u_4	48	12
u_5	147	36

$$\lambda_0 = \rho^2 \tag{11.15}$$

$$\approx 5.783185962946785 \tag{11.16}$$

$$\lambda_1 = -\rho^2 \tag{11.17}$$

$$\approx -5.783185962946785 \tag{11.18}$$

$$\lambda_2 = \frac{3}{2}\rho^2 + \frac{1}{4}\rho^4 \tag{11.19}$$

$$\approx 17.036088914926359 \tag{11.20}$$

$$\lambda_3 = -3\rho^2 - \frac{3}{2}\rho^4 \tag{11.21}$$

$$\approx -67.517417711877445 \tag{11.22}$$

$$\lambda_4 = \frac{15}{2}\rho^2 + \frac{15}{2}\rho^4 + \frac{87}{128}\rho^6 - \frac{21}{256}\rho^8 \tag{11.23}$$

$$\approx 333.919528952098846 \tag{11.24}$$

$$\lambda_5 = -\frac{45}{2}\rho^2 - \frac{75}{2}\rho^4 - \frac{1305}{128}\rho^6 + \frac{315}{256}\rho^8 \tag{11.25}$$

$$\approx -1979.913206464417536 \tag{11.26}$$

$$\lambda_6 = \frac{315}{4}\rho^2 + \frac{1575}{8}\rho^4 + \frac{27405}{256}\rho^6 - \frac{11155}{1536}\rho^8 - \frac{2665}{1536}\rho^{10} + \frac{145}{1024}\rho^{12} \tag{11.27}$$

$$\approx 13695.804996358888013 \tag{11.28}$$

The values of λ_0 to λ_3 are well known [59]. The values of λ_4 to λ_6 were computed to high numerical precision by Pavel Grinfeld [38]. Our results match these calculations giving further support that the SCMS system works correctly and that these exact symbolic expressions are correct.

11.5 N-sided Regular Polygon

At this point, we have tested all the problems where outside methods can easily be used to confirm our results. The final problem requires its own confirmation method. Recently, numerical progress has been made by Robert Jones which will also be used to verify our results [58].

One of the most amazing features of the CMS again provides huge advantages in this problem. CMS expressions can be evaluated in any coordinate system. All three of our eigenvalue problems start with the same surface, the unit circle. Their unique deformations are encapsulated in the surface velocity for each surface. Since all three start with the same coordinate system, a majority of the computations overlap. From the previous two problems, it has been shown that the CMS expressions generated are correct. Additionally, it has been shown that the evaluation of these CMS expressions in terms of abstract Fourier coefficients is also correct. All that remains to solve this version of the problem is to plug in the correct values for the Fourier coefficients of the surface velocity and its derivatives $c_0, c_1, \cdots, c_4$.

The values needed are given below expanded up to N^{-6}. The value of c_5 is of order N^{-8} and would appear in λ_6 but it is not needed for our series

$$\dot{\nabla}^m = \sum_{k=-\infty}^{\infty} c_m(k) e^{ikN\theta} \tag{11.29}$$

$$c_0(k) = \begin{cases} -2\,\frac{\zeta(2)}{N^2} - 3\,\frac{\zeta(6)}{N^6} + O\left(\frac{1}{N^8}\right) & k = 0 \\ \\ \frac{(-1)^k}{k^2 N^2} + 2\,\frac{(-1)^k \zeta(2)}{k^2 N^4} - 5\,\frac{(-1)^k}{k^4 N^4} + 12\,\frac{(-1)^k \zeta(4)}{k^2 N^6} & \\ -46\,\frac{(-1)^k \zeta(2)}{k^4 N^6} + 61\,\frac{(-1)^k}{k^6 N^6} + O\left(\frac{1}{N^8}\right) & k \neq 0 \end{cases} \tag{11.30}$$

$$c_1(k) = \begin{cases} 0 & k = 0 \\ \\ 0 & k \neq 0 \end{cases} \tag{11.31}$$

$$c_2(k) = \begin{cases} 12\,\frac{\zeta(4)}{N^4} - 24\,\frac{\zeta(6)}{N^6} + O\left(\frac{1}{N^8}\right) & k = 0 \\ \\ -12\,\frac{(-1)^k\zeta(2)}{k^2N^4} + 24\,\frac{(-1)^k}{k^4N^4} - 180\,\frac{(-1)^k\zeta(4)}{k^2N^6} & \\ +1044\,\frac{(-1)^k\zeta(2)}{k^4N^6} - 1500\,\frac{(-1)^k}{k^6N^6} + O\left(\frac{1}{N^8}\right) & k \neq 0 \end{cases} \tag{11.32}$$

$$c_3(k) = \begin{cases} 180\,\frac{\zeta(6)}{N^6} + O\left(\frac{1}{N^8}\right) & k = 0 \\ -1080\,\frac{(-1)^k\zeta(2)}{k^4N^6} + 1800\,\frac{(-1)^k}{k^6N^6} + O\left(\frac{1}{N^8}\right) & k \neq 0 \end{cases} \tag{11.33}$$

$$c_4(k) = \begin{cases} -324\,\frac{\zeta(6)}{N^6} + O\left(\frac{1}{N^8}\right) & k = 0 \\ 2160\,\frac{(-1)^k\zeta(4)}{k^2N^6} + 11880\,\frac{(-1)^k}{k^6N^6} - 8856\,\frac{(-1)^k\zeta(2)}{k^4N^6} + O\left(\frac{1}{N^8}\right) & k \neq 0 \end{cases} \tag{11.34}$$

$$c_5 = \begin{cases} O\left(\frac{1}{N^8}\right) & k = 0 \\ \\ O\left(\frac{1}{N^8}\right) & k \neq 0 \end{cases} \tag{11.35}$$

To show how the computations are tested, one convolution from λ_3 is examined. The expression for λ_3 contains 9 different convolution expressions. Tables like the ones below are given for all convolutions in Chapter A.3.

The term evaluated here is the fourth convolution and first non-trivial convolution that appears in λ_3. The closed form for this expression is determined using the Fourier library from Chapter 9. This library is also used to give numerical estimates for the convolutions using truncation.

The exact solution is

$$T_4 = \left(4\rho^4 - 4\rho^2\right)\text{conv}(c_0, c_0, c_0)(0) \tag{11.36}$$

$$= \frac{216\zeta(6)\rho^2}{N^6} - \frac{216\zeta(6)\rho^4}{N^6} \tag{11.37}$$

Table 11.10 shows the numerical results compared to the exact solution. Since this term is part of λ_3 when the Taylor series is formed, this value will be divided by 3!. This is accounted for in the tables. This convolution contributes to both the ρ^2 and ρ^4 component of the final series. After division by 3! the result should be $36\zeta(6)$. Taking the best numerical result in the table and dividing out the $\zeta(6)$ returns $35.9999999\cdots$ as expected.

The approx range column gives the truncation range. When this is set to 4 then the series is computed from -4 to 4, a total of 9 terms. This convolution contains three functions, the convolution of the first two is found exactly and tested. It is then combined with the third function. Table 11.10 only shows this final convolution of two functions.

The result column contains the numerical approximations. The relative and absolute error between this numerical result and the exact result is given. The convergence towards the exact answer is also shown in Figure 11.1.

The convergence rate is used to determine how fast the numerical computations are approaching some fixed value. There is not enough information to estimate a convergence rate until the third approximation. This approximates the error as $O\left(r^{x}\right)$ where r is the range used and x is the convergence rate. The final column shows the time it took to approximate each summation.

The table also shows the results of applying Richardson extrapolation, described in Chapter 9. Calculations are done with 64 digits of accuracy. Richardson extrapolation is used to take the list of numerical results and eliminate error terms. In this case, Richardson Extrapolation works extremely well. The extrapolated result is accurate to 60 digits. Table 11.11 repeats these calculations on the ρ^4 term, but the results are the same.

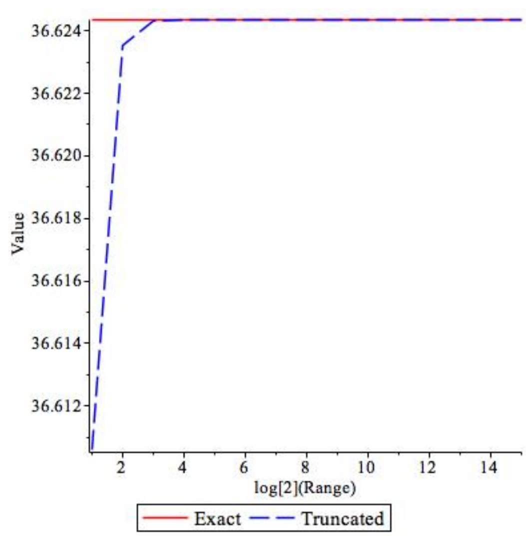

Figure 11.1: Convergence of numerical truncation to the exact answer for the fourth convolution of λ_3.

Table 11.10: Order ρ^2/N^6 contribution from convolution 4 of λ_3

$T_4/3! = \frac{v\rho^2}{N^6}, v = (216\,\zeta(6))/3!$					
$v = 36.6243502314401690297226454724731390044654296411827282263267118\,9$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	36.610605735565	$1.37445e-02$	$3.75283e-04$	n/a	0.003 sec.
4	36.623532797464	$8.17434e-04$	$2.23194e-05$	n/a	0.002 sec.
8	36.624314781685	$3.54498e-05$	$9.67929e-07$	−4.047	0.004 sec.
16	36.624348929133	$1.30231e-06$	$3.55585e-08$	−4.517	0.008 sec.
32	36.624350187365	$4.40747e-08$	$1.20343e-09$	−4.762	0.014 sec.
64	36.624350230007	$1.43282e-09$	$3.91220e-11$	−4.883	0.030 sec.
128	36.624350231395	$4.56637e-11$	$1.24681e-12$	−4.942	0.064 sec.
256	36.624350231439	$1.44104e-12$	$3.93464e-14$	−4.971	0.143 sec.
512	36.624350231440	$4.52531e-14$	$1.23560e-15$	−4.986	0.310 sec.
1024	36.624350231440	$1.41762e-15$	$3.87070e-17$	−4.993	0.726 sec.
2048	36.624350231440	$4.43547e-17$	$1.21107e-18$	−4.996	1.922 sec.
4096	36.624350231440	$1.38693e-18$	$3.78691e-20$	−4.998	6.105 sec.
8192	36.624350231440	$4.33549e-20$	$1.18377e-21$	−4.999	25.695 sec.
16384	36.624350231440	$1.35505e-21$	$3.69985e-23$	−5.000	92.262 sec.
32768	36.624350231440	$4.23484e-23$	$1.15629e-24$	−5.000	420.616 sec.
Richardson	36.624350231440	$8.38800e-59$	$2.29028e-60$	n/a	n/a
Error: $O\left(\text{range}^{-5.000}\right)$					

Table 11.11: Order ρ^4/N^6 contribution from convolution 4 of λ_3

$T_4/3! = \frac{v\rho^4}{N^6}, v = (-216\,\zeta(6))/3!$					
$v = -36.6243502314401690297226454724731390044654296411827282263267118\,9$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−36.610605735565	$1.37445e-02$	$3.75283e-04$	n/a	0.003 sec.
4	−36.623532797464	$8.17434e-04$	$2.23194e-05$	n/a	0.002 sec.
8	−36.624314781685	$3.54498e-05$	$9.67929e-07$	−4.047	0.004 sec.
16	−36.624348929133	$1.30231e-06$	$3.55585e-08$	−4.517	0.008 sec.
32	−36.624350187365	$4.40747e-08$	$1.20343e-09$	−4.762	0.014 sec.
64	−36.624350230007	$1.43282e-09$	$3.91220e-11$	−4.883	0.030 sec.
128	−36.624350231395	$4.56637e-11$	$1.24681e-12$	−4.942	0.064 sec.
256	−36.624350231439	$1.44104e-12$	$3.93464e-14$	−4.971	0.143 sec.
512	−36.624350231440	$4.52531e-14$	$1.23560e-15$	−4.986	0.310 sec.
1024	−36.624350231440	$1.41762e-15$	$3.87070e-17$	−4.993	0.726 sec.
2048	−36.624350231440	$4.43547e-17$	$1.21107e-18$	−4.996	1.922 sec.
4096	−36.624350231440	$1.38693e-18$	$3.78691e-20$	−4.998	6.105 sec.
8192	−36.624350231440	$4.33549e-20$	$1.18377e-21$	−4.999	25.695 sec.
16384	−36.624350231440	$1.35505e-21$	$3.69985e-23$	−5.000	92.262 sec.
32768	−36.624350231440	$4.23484e-23$	$1.15629e-24$	−5.000	420.616 sec.
Richardson	−36.624350231440	$8.38800e-59$	$2.29028e-60$	n/a	n/a
Error: $O\left(\text{range}^{-5.000}\right)$					

Table 11.12 shows a term from λ_3 that has a slower convergence rate. The convolution expression and exact answer are

$$T_7 = -4\pi \text{conv}(c_0, f_{1,dr}, f_{1,dr})(0) \tag{11.38}$$
$$= \frac{16\rho^2\zeta(4)}{N^4} + \frac{32\rho^2\zeta(2)\zeta(3)}{N^5} + \frac{8\rho^4\zeta(3)^2}{N^6} + \frac{224}{3}\frac{\zeta(6)\rho^2}{N^6} - \frac{88}{3}\frac{\rho^4\zeta(6)}{N^6} \tag{11.39}$$

Table 11.12 only looks at the ρ^2/N^5 contribution from this convolution. The convergence rate here is only -2 instead of -5. A larger exponent means a faster convergence. In this case, Richardson only gets 48 digits correct instead of 60 digits. Figure 11.2 shows that this is still clearly converging towards the exact answer.

Table 11.12: Order ρ^2/N^5 contribution from convolution 7 of λ_3

$T_7/3! = \frac{v\rho^2}{N^5}, v = (32\,\zeta(2)\,\zeta(3))\,/3!$					
$v = 10.545623201585579297051122354587336384438141155520715779711052 14$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	10.207613801337	$3.38009e-01$	$3.20521e-02$	n/a	0.017 sec.
4	10.438615344676	$1.07008e-01$	$1.01471e-02$	n/a	0.023 sec.
8	10.515371067399	$3.02521e-02$	$2.86869e-03$	-1.590	0.036 sec.
16	10.537574585531	$8.04862e-03$	$7.63219e-04$	-1.789	0.070 sec.
32	10.543547247410	$2.07595e-03$	$1.96855e-04$	-1.894	0.152 sec.
64	10.545096042486	$5.27159e-04$	$4.99884e-05$	-1.947	0.301 sec.
128	10.545490378244	$1.32823e-04$	$1.25951e-05$	-1.974	0.689 sec.
256	10.545589865789	$3.33358e-05$	$3.16110e-06$	-1.987	2.214 sec.
512	10.545614851343	$8.35024e-06$	$7.91821e-07$	-1.993	2.971 sec.
1024	10.545621111985	$2.08960e-06$	$1.98149e-07$	-1.997	8.499 sec.
2048	10.545622678930	$5.22655e-07$	$4.95613e-08$	-1.998	18.389 sec.
4096	10.545623070890	$1.30696e-07$	$1.23934e-08$	-1.999	37.881 sec.
8192	10.545623168908	$3.26779e-08$	$3.09872e-09$	-2.000	96.229 sec.
16384	10.545623193416	$8.16998e-09$	$7.74727e-10$	-2.000	279.892 sec.
32768	10.545623199543	$2.04256e-09$	$1.93688e-10$	-2.000	1005.533 sec.
Richardson	10.545623201586	$4.51200e-47$	$4.27856e-48$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Tables for every convolution are given in the appendix, Chapter A. The final values computed for each λ are below. The details of each convolution and its answer are also given in the appendix. For some convolutions in λ_5, only numerical approximations were possible. We have a hypothesis for some properties of this series based on our results.

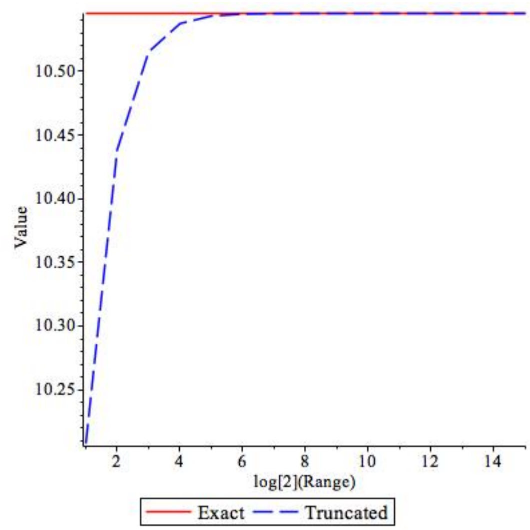

Figure 11.2: Convergence of numerical truncation to the exact answer for the seventh convolution of λ_3.

1. Each term in the series has the form $\frac{a_1 \rho^{a_3}}{N^{a_2}} \prod(z(b_i))$.
2. The inputs to the ζ function are related to the order of N^{a_2} by $\sum(b_i) = a_2$. They may not include $\zeta(1)$ because it is ∞.
3. The coefficient a_1 is always a small integer.
4. The order of ρ^{a_3} is related to the order of N^{a_2}. At $a_2 = 3$ there is just ρ^2. At $a_2 = 5$, we have ρ^2 and ρ^4. The next order, $a_2 = 7$ has three ρ components for ρ^2, ρ^4, and ρ^6. The powers of ρ take the even values from 2 to $a_2 - 1$.
5. When the series is scaled to a polygon of area π, all $\zeta(b_i)$ where b_i is even cancel.

Given the numerical results, only one answer was possible that would match all our computations and all these hypothesis. Specifically, the small number of terms that could only be approximated converged towards the $\zeta(6)$ value needed to meet the 5th point. If $\zeta(6)$ cancels out when the surface area is scaled to size π, then the value of λ_5 we compute must be correct. All our evidence supports this hypothesis and the final value we have computed.

$$\lambda_0 = \rho^2 \tag{11.40}$$

$$\lambda_1 = \frac{4\zeta(2)\rho^2}{N^2} + \frac{6\zeta(6)\rho^2}{N^6} + O\left(\frac{1}{N^7}\right) \tag{11.41}$$

$$\begin{aligned}\lambda_2 =& \frac{8\zeta(3)\rho^2}{N^3} + \frac{64\zeta(4)\rho^2}{N^4} + \frac{\left(32\zeta(2)\zeta(3) - 80\zeta(5) - 4\zeta(5)\rho^2\right)\rho^2}{N^5} \\ &+ \frac{\left(-12\zeta(6) + 4\zeta(6)\rho^2\right)\rho^2}{N^6} + O\left(\frac{1}{N^7}\right)\end{aligned} \tag{11.42}$$

$$\begin{aligned}\lambda_3 =& -\frac{24\zeta(4)\rho^2}{N^4} + \frac{\left(96\zeta(2)\zeta(3) + 144\zeta(5)\right)\rho^2}{N^5} \\ &+ \frac{948\zeta(6)\rho^2 + \left(24\zeta(3)^2 - 12\zeta(6)\right)\rho^4}{N^6} + O\left(\frac{1}{N^7}\right)\end{aligned} \tag{11.43}$$

$$\begin{aligned}\lambda_4 =& \frac{\left(672\zeta(5) - 384\zeta(2)\zeta(3)\right)\rho^2}{N^5} \\ &+ \frac{\left(192\zeta(3)^2 - 1152\zeta(6)\right)\rho^2}{N^6} + O\left(\frac{1}{N^7}\right)\end{aligned} \tag{11.44}$$

$$\lambda_5 = \frac{1680\zeta(6)\rho^2}{N^6} + O\left(\frac{1}{N^7}\right) \tag{11.45}$$

This forms the Taylor series

$$\begin{aligned}\frac{\lambda_N}{\lambda} =& 1 + \frac{4\zeta(2)}{N^2} + \frac{4\zeta(3)}{N^3} + \frac{28\zeta(4)}{N^4} \\ &+ \frac{12\zeta(5) + 16\zeta(2)\zeta(3) - 2\lambda\zeta(5)}{N^5} \\ &+ \frac{8\zeta(3)^2 + 124\zeta(6) + 4\zeta(3)^2\lambda}{N^6} + O\left(\frac{1}{N^7}\right)\end{aligned} \tag{11.46}$$

Robert Jones of HBE Labs recently used numerical methods to approximate the answer to 100 digits on the regular polygons for fixed N [58].

The series in Equation 11.46 is for a polygon with surface area 2π area. The numerical results in [58] are for a polygon with surface area π. This is only a difference in the scale of the shape, Equation 11.46 can be scaled to have area π.

The scaling factor is

$$\frac{N\sin\left(\frac{\pi}{N}\right)\cos\left(\frac{\pi}{N}\right)}{\pi} = 1 - \frac{4\zeta(2)}{N^2} + \frac{12\zeta(4)}{N^4} - \frac{12\zeta(6)}{N^6} + O\left(N^{-8}\right) \tag{11.47}$$

The scaled version of Equation 11.46 has the very interesting effect of canceling all even $\zeta(a)$ terms. Leaving only the Equation 11.48.

$$\frac{L_{N,\pi}}{\lambda} = 1 + \frac{4\zeta(3)}{N^3} - \frac{2\lambda\zeta(5)}{N^5} + \frac{12\zeta(5)}{N^5} + \frac{4\lambda\zeta(3)^2}{N^6} + \frac{8\zeta(3)^2}{N^6} + O\left(\frac{1}{N^7}\right) \tag{11.48}$$

The rescaled series is compared to the numerical results in Table 11.13. This gives an outside confirmation of our series.

Table 11.13: Comparison of Equation 11.48 to numerical approximations [58].

Comparison to Numerical Approximations			
N	Exact Result	Approx. Result	Difference
4	6.28372254763131031000	6.28318530717958647693	5.3724045172e-04
8	5.83856803931689371583	5.83849143359244285052	7.6605724451e-05
16	5.78999273707821970544	5.78999189999020853435	8.3708801117e-07
32	5.78403488119607130392	5.78403487370244431833	7.4936269856e-09
64	5.78329204396220764314	5.78329204389978502746	6.2422615677e-11
128	5.78319922243260221656	5.78319922243209895699	5.0325957804e-13
256	5.78318762036894686933	5.78318762036894287572	3.993610474e-15

The series for the Laplace eigenvalues on the regular N-sided polygon under Dirichlet boundary conditions have been verified in multiple ways. The series up to $\frac{1}{N^4}$ was computed by Pavel Grinfeld [47]. His calculations were checked using an early version of the CMSTRS library and an error was found. The full SCMS system confirmed these results up to $\frac{1}{N^4}$. Application of our system has also provided more terms in the series, up to $\frac{1}{N^6}$.

These results are confirmed using numerical approximations. Each convolution is computed exactly and numerically and the results are compared. For a small number of series, only numerical approximations were available. These series appear in Chapter A.5. An alternative numerical method was developed independently [58]. Our series is evaluated at specific values of N and compared against these results. All of these methods agree, showing that the expressions presented here are correct.

12. Conclusions

In this thesis, we have shown that a symbolic implementation of the CMS can provide advantages over the current state of the art methods. The CMS is an analytical framework for dealing for deforming manifolds. The CMS itself was presented in Chapter 2. The CMS can be applied to a wide range of problems.

The applications of the CMS are vast, some examples were given in Chapter 2.5. One of the most prominent applications of the CMS is in the study of fluid films. An example from this space is the deformation of a soap bubble or film. Imagine a soap bubble blowing in the wind. As time passes, the shape of the bubble is constantly deformed. The bubble itself has a surface tension, which is also changing as the bubble deforms. This real world event is what the CMS is designed to model. The expressions that appear in the soap bubble problem are far more complex then those that appear in our problems.

The CMS is an extension of tensor calculus. Symbolic computation has a long history of advancing research by automating tensor calculus. Chapter 2.6 examined the previous work done in this field. Many CASs have been developed, but the focus has been on General Relativity and stationary surfaces. The introduction of deforming manifolds significantly broadens the scope of problems that can be examined, but also increases the complexity of calculations.

We have designed and implemented SCMS, a symbolic computation system for the CMS. The first part of the SCMS system is a Java library for working in the CMS. We have built this library based on a TRS we developed. The CMSTRS library was first proposed in 2011 and the first full version was presented in 2013 [10, 11]. This library implements the high level symbolic framework of the CMS.

The CMSTRS library is not just for algebraic simplification using rewrite rules. It also performs equivalence testing. Chapter 3.3.3 looked at the types of equivalence that could be used to combine terms. This was itself a difficult task. The CMSTRS library also allowed expressions to be exported to a Maple program for evaluation.

One of the most important features of the CMS is that any expression derived using it can be evaluated on any coordinate system. Our second library, the CMSTensor library for Maple was described in Chapter 4. Evaluation of a CMS expression for a particular realization of a problem requires working in a coordinate system. The CMSTensor library allows CMS expressions to be evaluated on a coordinate system. This library was first presented in 2013 and a more comprehensive version is presented in this thesis [11].

By combining these two libraries together, problems in the CMS can be automated from start to finish. Chapter 5 showed how these libraries can be used in practice. In this chapter, the perimeter of the unit circle was stretched into an ellipse. A Taylor series for the contour length was given in terms of how far the semi-axis was stretched. This series can be easily computed outside the CMS and was used to verify the correctness of our system.

The SCMS was then applied to a series of non-trivial problems. These problems all came from boundary variations of the Laplacian eigenvalues under Dirichlet boundary condition. Chapter 6 explained this basic problem. The eigenvalues are know on the unit circle. There are many other shapes where the eigenvalues are not known. The CMS can be used to deform the unit circle into these shapes. An alternative to finding the solution directly on the new shape is to change a shape with a known solution into a shape with an unknown solution. This method creates a Taylor series for the eigenvalues on the new surface.

Three surface deformations were examined. The deformation from the unit circle in Chapter 10 does not change the final shape. All variations in this deformation are equal to zero. This deformation was used to further confirm the accuracy of our expressions.

The second deformation was from the unit circle into an ellipse. These eigenvalues could be computed numerically to high orders. Our system replicated the known exact expressions and found new terms in the series. The series presented here has more terms then previously known. The series is confirmed using numerical approximations.

Since expressions in the CMS can be evaluated in any coordinate system, both these eigenvalue problems use the same CMS expressions. At this point, we were confident that our system worked correctly.

The most important boundary variation problem we approached was the eigenvalues on the regular N-sided polygon. At the start of this project, the exact series was only know for a few specific values of N and numerically approximated on others [46]. Our system helped with the first exact series presented in 2012 [47].

The series is found in terms of $\frac{1}{N}$. The series given in [47] went up to $\frac{1}{N^4}$. The series presented in this thesis adds another two terms, taking the series up to $\frac{1}{N^6}$. Computation of these final two terms was beyond the scope of all previous attempts.

Evaluation of these new terms required the creation and manipulation of Fourier series. Chapter

7 and Chapter 8 showed why these series appear when solving the problem. Chapter 9 showed how we implemented an additional Maple library to automate this process. The new library could be combined with the SCMS to seamlessly compute the desired eigenvalues.

Chapter 11 showed the results of applying our system to these problems. In each case, our system computed values beyond the scope of existing systems. The SCMS can deal with expression growth that was previously intractable. This allows advances in research beyond previous limitations.

Analysis of the series found for the Laplace-Dirichlet eigenvalues on the regular polygon have lead to some hypothesis about the series.

1. Each term in the series has the form $\frac{a_1 \rho^{a_3}}{N^{a_2}} \prod(z(b_i))$.
2. The inputs to the ζ function are related to the order of N^{a_2} by $\sum(b_i) = a_2$. They may not include $\zeta(1)$ because it is ∞.
3. The coefficient a_1 is always a small integer.
4. The order of ρ^{a_3} is related to the order of N^{a_2}. At $a_2 = 3$ there is just ρ^2. At $a_2 = 5$, we have ρ^2 and ρ^4. The next order, $a_2 = 7$ has three ρ components for ρ^2, ρ^4, and ρ^6. The powers of ρ take the even values from 2 to $a_2 - 1$.
5. When the series is scaled to a polygon of area π, all $\zeta(b_i)$ where b_i is even cancel.

The fourth point in this list is not obvious from the series shown in the results section. Chapter A.6 shows λ_1 and λ_2 computed to N^{-11}. We have verified this pattern appears in these two terms up to order N^{-20}.

We can also propose a hypothesis about the evolution of the λ orders. From order analysis of the CMS rules, it can be seen that λ_a will contribute to order N^{a+1} and higher. Finding the N^{-7} term would require λ_6.

The main computational complexity that we have run into is related to the eigenfunction. Generating partial derivatives of the eigenfunction as described in Chapter 7 creates algebraically complex functions. Even in λ_5, convolutions involving surface velocity and u_1 can be easily simplified. Expressions with u_2 become more difficult and the expression containing u_4 pushed the limits of our design.

One area for future work to extend this series is to revise the generation and simplification of u_k values. The requirements presented in Chapter 7 can be met in multiple ways. The null space of the Helmholtz operators gives a significant amount of freedom in construction. If equivalent but algebraically simpler solutions can be found, then it will be easy to extend the series. It is unknown if this can be done, because computation of these series currently rely on the Helmholtz null space. A new simpliciation approach would be required, if not an entirely new approach to computing the values.

A second area for future work is in numerical approximations. We have used Richardson extrapolation exclusively to extend numerical results. For some series, this was clearly not the optimal acceleration method. Some experimentation has shown that other acceleration methods will produce more digits of accuracy from less data. Continued optimization of the numerical methods used to approximate convolutions can lead to more accurate results. The current results are of practical use, but we believe a pattern may appear in this series. Finding this pattern will require more accuracy.

In conclusion, we have shown that the CMS is in desperate need of more automated computation methods. Rapid expression swell has caused many problems, which seem to be perfect fits for the CMS, to become intractable. These problems have either been abandoned or left to alternative methods. Our symbolic implementation of the CMS gives a new foundation to fill this gap.

We have shown that the CMS can be implemented as both an evaluation method and a symbolic framework. We have used this system to solve a set of non-trivial problems. Our results have shown that automation can extend the reach of the CMS.

Our most important contribution to an open research problem is the series expansion for the Laplace eigenvalues on the N-sided regular polygon under Dirichlet boundary conditions. This expansion goes further then all previous attempts. In addition, it gives new insight into potential patterns that arise in this series.

We have successfully designed, implemented, and verified a symbolic computation system, the SCMS. We have released this software as a open source package. We have applied this system to open problems and shown that it can advance these problems beyond previous limitations. These results show that automated computation within the CMS is possible and that it is necessary to advance research in a wide range of fields.

13. Bibliography

[1] Tela the tensor language. http://www.ava.fmi.fi/prog/tela.html.

[2] The first eigenvalue of the laplacian-dirichlet. In *Extremum Problems for Eigenvalues of Elliptic Operators*, Frontiers in Mathematics, pages 45–60. Birkhuser Basel, 2006.

[3] O. Alexandrov. Surface normal illustration. https://commons.wikimedia.org/wiki/File:Surface_normal_illustration.png, 2007.

[4] P. Amore. Spectroscopy of drums and quantum billiards: perturbative and non-perturbative results. *J. Math. Phys.*, 51:052105, 2010.

[5] L. B. Buchberger and K. R. Loos. Algebraic simplification. *Computing, Suppl.*, 4:11–43, 1982.

[6] L. Bachmair and D. A. Plaisted. Termination orderings for associative-commutative rewrite systems. *J. Symbolic Computation*, 1:329–349, 1985.

[7] A. Balaz, I. Vidanovic, D. Stojiljkovic, D. Vudragovic, A. Belic, and A. Bogojevic. SPEEDUP Code for Calculation of Transition Amplitudes via the Effective Action Approach. *ArXiv e-prints*, May 2011.

[8] A. Benjamin, R. Benjamin, and P. Grinfeld. Minimal surface with a cavity of given perimeter. *J. Geom. Symmetry Phys.*, 28:59–66, 2012.

[9] M. Berg van den. On rayleigh's formula for the first dirichlet eigenvalue of a radial perturbation of a ball. *Journal of Geometric Analysis*, 23(3), 2013.

[10] M. Boady, P. Grinfeld, and J. Johnson. Boundary variation of poisson's equation: a model problem for symbolic calculus of moving surfaces. *Int. J. Math. Comp. Sci.*, 6(2), 2011.

[11] M. Boady, P. Grinfeld, and J. Johnson. A term rewriting system for the calculus of moving surfaces. In M. B. Monagan, G. Cooperman, and M. Giesbrecht, editors, *ISSAC*, pages 69–76. ACM, 2013.

[12] S. Bonanos. Riemannian geometry and tensor calculus @ mathematica. http://www.inp.demokritos.gr/%7esbonano/RGTC/.

[13] I. Booth, M. P. Heller, and M. Spaliński. Black brane entropy and hydrodynamics. *Phys. Rev. D*, 83:061901, Mar 2011.

[14] Booyabazooka. Graph isomorphism a. `https://commons.wikimedia.org/wiki/File:Graph_isomorphism_a.svg`, 2006.

[15] Booyabazooka. Graph isomorphism b. `https://commons.wikimedia.org/wiki/File:Graph_isomorphism_b.svg`, 2006.

[16] C. Boyer. *History of Analytic Geometry.* Scripta Mathematica, New York, 1956.

[17] L. Brewin. Riemann normal coordinate expansions using cadabra. *Classical and Quantum Gravity*, 26(17):175017, 2009.

[18] L. Brewin. A brief introduction to cadabra: A tool for tensor computations in general relativity. *Computer Physics Communications*, 181(3):489 – 498, 2010.

[19] A. J. Christopherson, K. A. Malik, D. R. Matravers, and K. Nakamura. Comparing two different formulations of metric cosmological perturbation theory. *Classical and Quantum Gravity*, 28(22):225024, 2011.

[20] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. The maude 2.0 system. *Proc. Rewriting Techniques and Applications*, pages 76–87, June 2003.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, Cambridge, Massachusetts, third edition, 2009.

[22] C. J. Costa. Complete minimal surfaces in $\mathbf{r}^3$ of genus one and four planar embedded ends. *Proceedings of the American Mathematical Society*, 119(4):1279–1287, 1993.

[23] R. Courant and D. Hilbert. *Methods of mathematical physics. Partial differential equations*, volume 2. Interscience, 1965.

[24] C. Deffayet, S. Deser, and G. Esposito-Farèse. Arbitrary p-form galileons. *Phys. Rev. D*, 82:061501, Sep 2010.

[25] C. Deffayet, G. Esposito-Farèse, and A. Vikman. Covariant galileon. *Phys. Rev. D*, 79:084003, Apr 2009.

[26] N. Dershowitz and D. A. Plaisted. *Handbook of Automated Reasoning*, volume 1, chapter 9. Elsevire, `http://rewriting.loria.fr/documents/rewriting.pdf`, 2001.

[27] R. Dodier. Maxima 5.22.1 manual: 29. atensor. `http://maxima.sourceforge.net/docs/manual/en/maxima_29.html#SEC131`.

[28] M. Durkee and H. S. Reall. Perturbations of higher-dimensional spacetimes. *Classical and Quantum Gravity*, 28(3):035011, 2011.

[29] A. Einstein. *The Berlin Years: Writings, 1914-1917*, volume 6 of *The Collected Papers of Albert Einstein*. Princeton University, New Jersey, 1997.

[30] Euclid. *The thirteen books of Euclid's Elements*. Cambridge, England, second edition, 1926.

[31] L. Euler. Du mouvement dun corps solide quelconque lorsquil tourne autour dun axe mobile. *Histoire de l'Acadmie royale des sciences et des belles lettres de Berlin*, pages 176–227, 1751.

[32] L. C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 2 edition, 2010.

[33] F. Franchetti, A. Sandryhaila, and J. Johnson. High assurance SPIRAL. In *SPIE*. Proceedings of SPIE 2014, 2014.

[34] J. v. z. Gathen and J. Gerhard, editors. *Modern Computer Algebra*. Cambridge, New York, third edition, 2013.

[35] O. Giraud and K. Thas. Hearing shapes of drums: Mathematical and physical aspects of isospectrality. *Rev. Mod. Phys.*, 82:2213–2255, Aug 2010.

[36] D. S. Grebenkov and B. T. Nguyen. Geometrical structure of laplacian eigenvalues. *SIAM Review*, 55(4):601–667, 2013.

[37] M. Grinfeld. *Thermodynamic Methods in the Theory of Heterogeneous Systems*. Longman, New York, 1991.

[38] P. Grinfeld. Email correspondence. Private Communications.

[39] P. Grinfeld. Exact nonlinear equations for fluid films and proper adaptations of conservation theorems from classical hydrodynamics. *J. Geom. Symm. Phys.*, 16:1–21, 2009.

[40] P. Grinfeld. Hadamard's formula inside and out. *J. Opt. Theory and Appl.*, 146(3):654–690, 2009.

[41] P. Grinfeld. Hamiltonian dynamic equations for fluid films. *Stud. Appl. Math.*, 125:223–264, 2010.

[42] P. Grinfeld. A variable thickness model for fluid films under large displacements. *Phys. Rev. Lett.*, 105:137802, 2010.

[43] P. Grinfeld. Viscous equations of fluid film dynamics. *Computers, Materials and Continua*, 2010.

[44] P. Grinfeld. A better calculus of moving surfaces. *J. Geom. Symm. Phys.*, 26:61–69, 2012.

[45] P. Grinfeld. *Introduction to Tensor Analysis and the Calculus of Moving Surfaces*. Springer, 2013.

[46] P. Grinfeld and G. Strang. Laplace eigenvalues on polygons. *Computers and Mathematics with Applications*, 48:1121–1133, 2004.

[47] P. Grinfeld and G. Strang. Laplace eigenvalues on regular polygons: A series in $1/N$. *Journal of Mathematical Analysis and Applications*, 385(1):135 – 149, 2012.

[48] P. Grinfeld and J. Wisdom. Total gravitational energy of a slightly ellipsoidal trilayer planet. *Quart. Appl. Math.*, 64(2):271–281, 2006.

[49] P. Grinfeld and J. Wisdom. A way to compute the gravitational potential for near-spherical geometries. *Quart. Appl. Math.*, 64(2):229–252, 2006.

[50] P. Guidotti and J. V. Lambers. Eigenvalue characterization and computation for the laplacian on general 2-d domains. *Numerical Functional Analysis and Optimization*, 29:507–531, 2008.

[51] Y. V. Gusev. Heat kernel expansion in the covariant perturbation theory. *Nuclear Physics B*, 807(3):566 – 590, 2009.

[52] J. Hadamard. *Leons sur la Propagation des Ondes et les quations de l'Hydrodynamique*. Hermann, Paris, 1903.

[53] E. Harrell. A short history of operator theory, 2009.

[54] T. Hawkins. Cauchy and the spectral theory of matrices. *Historia Mathematica*, 2(1):1 – 29, 1975.

[55] H. Howards, M. Hutchings, and F. Morgan. The isoperimetric problem on surfaces. *The American Mathematical Monthly*, 106(5):pp. 430–439, 1999.

[56] G. Huet and D. Lankford. On the uniform halting problem for term rewriting systems. *Rapport Laboria 283*, 1978.

[57] G. Huet and D. C. Oppen. Equations and rewrite rules – a survey. Technical report, Stanford University, Jan 1980.

[58] R. Jones. Computing ultra-precise eigenvalues of the Laplacian within polygons. *ArXiv e-prints*, Feb. 2016.

[59] D. Joseph. Parameter and domain dependence of eigenvalues of elliptic partial differential equations. *Archive for Rational Mechanics and Analysis*, 24(5):325–351, 1967.

[60] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. In *Proceedings of the 11th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, POPL '84, pages 83–92, New York, NY, USA, 1984. ACM.

[61] J.-P. Jouannaud and M. Munoz. Termination of a set of rules modulo a set of equations. In *Proceedings of the 7th International Conference on Automated Deduction*, pages 175–193, London, UK, UK, 1984. Springer-Verlag.

[62] M. Kac. Can one hear the shape of a drum? *The American Mathematical Monthly*, 73(4):pp. 1–23, 1966.

[63] J. Katz and G. I. Livshits. Superpotentials from variational derivatives rather than lagrangians in relativistic theories of gravity. *Classical and Quantum Gravity*, 25(17):175024, 2008.

[64] M. Khabou, L. Hermi, and M. Rhouma. Shape recognition using eigenvalues of the dirichlet laplacian. *Pattern Recognition*, 40(1):141–153, 2007.

[65] J. W. Klop. Term rewriting systems from church-rosser to knuth-bendix and beyond. In *Proceedings of the seventeenth international colloquium on Automata, languages and programming*, pages 350–369, 1990.

[66] D. Knuth and P. Bendix. Simple word problems in universal algebras. *Computational Problems in Abstract Algebra*, pages 263–297, 1970.

[67] B. G. Korenev. *Bessel Functions and their Applications*, volume 8 of *Analytical methods and Special Functions: An International Series of Monographs in Mathematics*. Taylor and Francis, 2002.

[68] V. Kozlov. Domain dependence of eigenvalues of elliptic type operators. *Mathematische Annalen*, 357(4), 2012.

[69] W. Landry. Implementing a high performance tensor library. *Sci. Program.*, 11:273–290, December 2003.

[70] J. M. Lee. Ricci: A mathematica package for doing tensor calculations in diferential goemetry. `http://www.math.washington.edu/%7elee/Ricci/`.

[71] T. Levi-Civita. *The Absolute Differential Calculus (Calculus of Tensors)*. Dover Publications, 1977.

[72] C.-H. L. Lin, M.-H. Lo, and Y.-C. Tsai. Shape Energy of Fluid Membranes —Analytic Expressions for the Fourth-Order Variation of the Bending Energy—. *Progress of Theoretical Physics*, 109:591–618, Apr. 2003.

[73] J. Lodder. Curvature in the calculus curriculum. *The American Mathematical Monthly*, 110(7):593–605, Aug.-Sep 2003.

[74] S. Mann, L. Nathan, and T. DeRose. A coordinate free geometry adt. Technical report, University of Waterloo, 1997.

[75] Maplesoft. Maple online help: Overview of the tensor package. `http://www.maplesoft.com/support/help/Maple/view.aspx?path=tensor`.

[76] MapleSoft. tensor(deprecated)/permute_indices - help. `http://www.maplesoft.com/support/help/maple/view.aspx?path=tensor(deprecated)%2Fpermute_indices`. Accessed: 2015-07-28.

[77] Maplesoft. *Maple Programming Guide*. Maplesoft, a division of Waterloo Maple Inc., Toronto, version 9.0 edition, 2011-2014.

[78] Maplesoft. *Maple User Manual*. Maplesoft, Waterloo, ON Canada, 2016.

[79] J. M. Maran-Garca. xperm: fast index canonicalization for tensor computer algebra. *Computer Physics Communications*, 179(8):597 – 603, 2008.

[80] J. M. Martin-Garcia. xtensor: Fast abstract tensor computer algebra. `http://www.xact.es/xTensor/index.html`.

[81] MathTensor Inc. Mathtensor - tensor analysis for mathematica. `http://smc.vnet.net/MathTensor.html`.

[82] A. McConnell. *Applications of Tensor Analysis*. Dover Publications, New York, 1957.

[83] T. Mlek and V. Pravda. Kerrschild spacetimes with an (a)ds background. *Classical and Quantum Gravity*, 28(12):125011, 2011.

[84] NASA. Internal structure of mercury. `https://commons.wikimedia.org/wiki/File:Internal_Structure_of_Mercury.jpg`, 2015.

[85] R. Osserman and A. Weinstein, editors. *Geometry of the Laplace Operator*, volume XXXVI of *Proceedings of Symposia in Pure Mathematics*. American Mathematical Society, Providence, Rhode Island, 1980.

[86] L. Parker and S. M. Christensen. *MathTensor: A System for Doing Tensor Analysis by Computer*. Addison-Wesley Professional, 1994.

[87] K. Peeters. Cadabra: a field-theory motivated symbolic computer algebra system. *Computer Physics Communications*, 176:550–558, Apr. 2007.

[88] K. Peeters. Introducing Cadabra: a symbolic computer algebra system for field theory problems. *ArXiv High Energy Physics - Theory e-prints*, Jan. 2007.

[89] K. Peeters. *Cadabra: reference guide and tutorial*. `http://cadabra.phi-sci.com/cadabra.pdf`, June 2008.

[90] M. H. Protter. Can one hear the shape of a drum? revisited. *SIAM Review*, 29(2):185–197, 1987.

[91] E. Racine, A. Buonanno, and L. Kidder. Recoil velocity at second post-newtonian order for spinning black hole binaries. *Phys. Rev. D*, 80:044010, Aug 2009.

[92] B. J. W. S. Rayleigh. *The theory of sound*, volume 2. Macmillan, London, 1894.

[93] M. M. G. Ricci and T. Levi-Civita. Méthodes de calcul différentiel absolu et leurs applications. *Mathematische Annalen*, 54(1):125–201, 1900.

[94] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357, 1911.

[95] T. Rowland and E. W. Weisstein. Linear transforms. `http://mathworld.wolfram.com/LinearTransformation.html`.

[96] A. Sandberg. Costa's minimal surface. `https://commons.wikimedia.org/w/index.php?curid=21538299`, 2012.

[97] U. Schning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37(3):312 – 323, 1988.

[98] H. Soleng. The mathematica packages cartan and mathtensor for tensor analysis. *Relativity and Scientific Computing*, pages 210–230, 1996.

[99] H. H. Soleng. Cartan: a mathematical package for tensor analysis. `http://www.adinfinitum.no/cartan/`.

[100] W.-H. Steedb and Y. Hardy. *Quantum Mechanics Using Computer Algebra*. World Scientific Publishing, Singapore, second edition, 2010.

[101] W. Stein et al. *Sage Mathematics Software (Version x.y.z)*. The Sage Development Team, YYYY. `http://www.sagemath.org`.

[102] C. F. Steinwachs and A. Y. Kamenshchik. One-loop divergences for gravity nonminimally coupled to a multiplet of scalar fields: Calculation in the jordan frame. i. the main results. *Physical Review D*, 84(2):024026, July 2011.

[103] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge, Massachusetts, 4th edition edition, 2009.

[104] J. Synge and A. Schild. *Tensor Calculus*. Dover Publications, Inc., 1949.

[105] T. Thomas. Extended compatibility conditions for the study of surface discontinuity in continuum mechanics. *J. Math. Mech.*, 6:311–322, 1957.

[106] T. Thomas. *Concepts from Tensor Analysis And Differential Geometry*. Academic Press, New York, 1965.

[107] L. N. Trefethen and M. Embree. *Spectra and Psuedospectra Analysis The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, Princeton, 2005.

[108] C. Truesdell and R. Toupin. *The Classical Field Theories*. Springer, Berlin, 1960.

[109] E. Tsantilis, R. A. Puntigam, and F. W. Hehl. A quadratic curvature Lagrangian of Pawłowski and Rączka: a finger exercise with MathTensor. *ArXiv General Relativity and Quantum Cosmology e-prints*, Jan. 1996.

[110] D. N. Vulcanov and G. Ciobanu. Maple procedures in teaching the canonical formalism of general relativity. *ArXiv Physics e-prints*, Oct. 2000.

[111] L. L. Williams. Field equations and lagrangian for the kaluza metric evaluated with tensor algebra software. *Journal of Gravity*, 2015.

[112] I. Wolfram Research. *Mathematica*. Wolfram Research, Inc., Champaign, Illinois, version 9.0 edition, 2012.

[113] S. Zuluaga and F. Fonseca. Can't one really hear the shape of a drum? *Acoustical Physics*, 57(4):465–472, 2011.

[114] E. lvarez, A. F. Faedo, and J. Lpez-Villarejo. Ultraviolet behavior of transverse gravity. *Journal of High Energy Physics*, 2008(10):023, 2008.

Appendix A. Full Numerical Results

A.1 Introduction

This appendix contains the full numerical results for every convolution that is used to determine the series for λ_N. Each variation of λ starting with λ_2 has its own section. The values of λ_0 and λ_1 are derived in Chapter 6. These results are analyzed in Chapter 11

The CMS expressions for each λ_k value are written as a summation of convolutions of basic functions. Each convolution is seperated into a set of T_j values for each λ_k.

The exact values for each convolution are presented. A table with the numerical approximations is given for each convolution. Some convolutions require multiple tables. Each constant A for orders x and y in $\frac{A\rho^x}{N^y}$ is given its own table. Although these are all computed from the same convolution, they are seperated algebraically in the computation. Each may also have a different convergence rate.

The convergence rate of a function determines how fast it approaches the exact answer. For each convolution, there is an exact answer F. This is approximated by a function $f(r)$. The convergence rate approximates the error introducted by the approximation.

$$|F - f(r)| = O\left(r^c\right) \tag{A.1}$$

This formula states that the error is a polynomial function in r where the highest order is c.

For each numerical approximation, the absolute and exact error is calculated. The exact error is $|F - f(r)|$ and the relative error is $\frac{|F-f(r)|}{|F|}$. The approximated convergence rate and time to compute are also given for each range r. The convergence rate cannot be approximated until three values have been computed. A relative error rate of $1e-8$ or smaller is considered strong evidence for the correctness of the exact formula. Richardson Extrapolation is applied to the entire table to produce an improved numerical estimate.

Each convolution is truncated at range r

$$\text{conv}(f,g)(0) = \sum_{j=-r}^{j=r} f(j)g(-j) \tag{A.2}$$

This means that at $r = 32$ there are actually 65 multiplications needed.

When the Taylor series is computed for λ_N the factors used are

$$\lambda_n = \sum_{k=0}^{5} \frac{\lambda_k}{k!} \tag{A.3}$$

The $k!$ is factored out in the numerical evaluations. This helps keep the size of the values down. For example, each term in λ_5 is divided by 120.

The value of each λ_k is given divided by $k!$ and ρ^2. These factors always appear. Removing them improves the clarity of the expressions.

The surface velocity's Fourier coefficient and its derivatives are given by c_m.

$$C = \sum_{k=-\infty}^{\infty} c_0(k)e^{ikN\theta} \tag{A.4}$$

$$\dot{\nabla}^x C = \sum_{k=-\infty}^{\infty} c_x(k)e^{ikN\theta} \tag{A.5}$$

The coefficients of the eigenfunctions $u_z = \frac{\partial^z u}{\partial t^z}|_{t=0,r=1}$ are given by f_z. If a derivative is taken before evaluation it is given in the subscript. For example, $f_{1,drd\theta}$ is the coefficient of u_1 with a derivative taken with respect to r and θ before evaluating at $r = 1$.

Additional commentary is given for λ_2. For λ_3 through λ_5 only minimal commentary is given. Each λ follows the pattern established in λ_2. Expressions that contribute to order $O\left(N^{-7}\right)$ are not shown in tables.

A.2 λ_2 Detailed Results

The second variation of lambda, λ_2, is determined by the following expressions.

$$\lambda_2 = T_1 + T_2 + T_3 \tag{A.6}$$

$$T_1 = 2\rho^2 \text{conv}(c_0, c_0)(0) \tag{A.7}$$

$$T_2 = 4\rho\sqrt{\pi}\text{conv}(c_0, f_{1,dr})(0) \tag{A.8}$$

$$T_3 = -2\rho^2 c_1(0) \tag{A.9}$$

Equation A.9 contains no convolutions and is known exactly.

$$T_3 = -2\rho^2 c_1(0) \tag{A.10}$$
$$= -2\rho^2 0 \tag{A.11}$$
$$= 0 \tag{A.12}$$

A.2.1 First Convolution of λ_2

$$T_1 = 2\rho^2 \text{conv}(c_0, c_0)(0) \tag{A.13}$$
$$= \frac{24\rho^2\zeta(4)}{N^4} - \frac{12\rho^2\zeta(6)}{N^6} + O\left(\frac{1}{N^7}\right) \tag{A.14}$$

Table A.1 and Table A.2 show that numerical approximation by truncation has an error of $O\left(R^{-3}\right)$ where R is the truncated range $\sum_{k=-R}^{R} c_0(-k)c_0(k)$. These numerical results confirm the exact solutions computed by the system.

A.2.2 Second Convolution of λ_2

$$T_2 = 4\rho\sqrt{\pi}\text{conv}(c_0, f_{1,dr})(0) \tag{A.15}$$
$$= \frac{8\zeta(3)\rho^2}{N^3} + \frac{40\zeta(4)\rho^2}{N^4} - \frac{80\zeta(5)\rho^2}{N^5} + \frac{32\zeta(2)\zeta(3)\rho^2}{N^5} - \frac{4\rho^4\zeta(5)}{N^5} + \frac{4\rho^4\zeta(6)}{N^6} + O\left(\frac{1}{N^7}\right) \tag{A.16}$$

Each combinations of orders for N and ρ is given a table. Tables A.3 to Table A.7 show the convergence and error for each term. These numerical results give clear evidence that the exact calculations are correct.

A.2.3 Complete λ_2

Combining these results give a final answer.

$$\lambda_2 = T_1 + T + 2 + T_3 \tag{A.17}$$

$$= \frac{8\zeta(3)\rho^2}{N^3} + \frac{64\zeta(4)\rho^2}{N^4} - \frac{80\zeta(5)\rho^2}{N^5} + \frac{32\zeta(2)\zeta(3)\rho^2}{N^5} - \frac{4\zeta(5)\rho^4}{N^5} - \frac{12\zeta(6)\rho^2}{N^6} + \frac{4\zeta(6)\rho^4}{N^6} + O\left(\frac{1}{N^7}\right) \tag{A.18}$$

In the Taylor series, these will be divided by 2!. All terms contain a ρ^2. These are both divided out in the next set of equations.

$$\frac{T_1}{2!\rho^2} = \frac{12\zeta(4)}{N^4} - \frac{6\zeta(6)}{N^6} \tag{A.19}$$

$$\frac{T_2}{2!\rho^2} = \frac{4\zeta(3)}{N^3} + \frac{20\zeta(4)}{N^4} - \frac{40\zeta(5)}{N^5} + \frac{16\zeta(2)\zeta(3)}{N^5} - \frac{2\rho^2\zeta(5)}{N^5} + \frac{3\rho^2\zeta(6)}{N^6} \tag{A.20}$$

$$\frac{T_3}{2!\rho^2} = 0 \tag{A.21}$$

$$\frac{\lambda_2}{2!\rho^2} = \frac{4\zeta(3)}{N^3} + \frac{32\zeta(4)}{N^4} - \frac{40\zeta(5)}{N^5} + \frac{16\zeta(2)\zeta(3)}{N^5} - \frac{2\zeta(5)\rho^2}{N^5} - \frac{6\zeta(6)}{N^6} + \frac{2\zeta(6)\rho^2}{N^6} \tag{A.22}$$

Table A.1: Order ρ^2/N^4 contribution from convolution 1 of λ_2

$T_1/2! = \frac{v\rho^2}{N^4}, v = (24\,\zeta\,(4))\,/2!$					
$v = 12.987878804533658298192044358494014833297011423024722892195714 58$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	12.948232337111	$3.96465e-02$	$3.05257e-03$	n/a	0.004 sec.
4	12.980736195136	$7.14261e-03$	$5.49944e-04$	n/a	0.004 sec.
8	12.986800672518	$1.07813e-03$	$8.30106e-05$	−2.422	0.007 sec.
16	12.987730668359	$1.48136e-04$	$1.14057e-05$	−2.705	0.010 sec.
32	12.987859393297	$1.94112e-05$	$1.49457e-06$	−2.853	0.024 sec.
64	12.987876320386	$2.48415e-06$	$1.91267e-07$	−2.927	0.041 sec.
128	12.987878490348	$3.14186e-07$	$2.41907e-08$	−2.964	0.102 sec.
256	12.987878765029	$3.95042e-08$	$3.04162e-09$	−2.982	0.178 sec.
512	12.987878799581	$4.95252e-09$	$3.81319e-10$	−2.991	0.410 sec.
1024	12.987878803914	$6.19973e-10$	$4.77347e-11$	−2.995	0.933 sec.
2048	12.987878804456	$7.75534e-11$	$5.97121e-12$	−2.998	2.149 sec.
4096	12.987878804524	$9.69772e-12$	$7.46675e-13$	−2.999	5.899 sec.
8192	12.987878804532	$1.21244e-12$	$9.33515e-14$	−2.999	19.186 sec.
16384	12.987878804534	$1.51569e-13$	$1.16700e-14$	−3.000	77.627 sec.
32768	12.987878804534	$1.89469e-14$	$1.45882e-15$	−3.000	393.877 sec.
Richardson	12.987878804534	$3.18845e-47$	$2.45494e-48$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.2: Order ρ^2/N^6 contribution from convolution 1 of λ_2

$T_1/2! = \frac{v\rho^2}{N^6}, v = (-12\,\zeta\,(6))\,/2!$					
$v = -6.1040583719066948382871075787455231674109049401971213710544519 80$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−6.330560431790	$2.26502e-01$	$3.71068e-02$	n/a	0.004 sec.
4	−6.149011273064	$4.49529e-02$	$7.36443e-03$	n/a	0.004 sec.
8	−6.111063571840	$7.00520e-03$	$1.14763e-03$	−2.258	0.007 sec.
16	−6.105029813102	$9.71441e-04$	$1.59147e-04$	−2.653	0.010 sec.
32	−6.104185982535	$1.27611e-04$	$2.09059e-05$	−2.838	0.024 sec.
64	−6.104074713361	$1.63415e-05$	$2.67715e-06$	−2.923	0.041 sec.
128	−6.104060439051	$2.06714e-06$	$3.38651e-07$	−2.963	0.102 sec.
256	−6.104058631830	$2.59924e-07$	$4.25821e-08$	−2.982	0.178 sec.
512	−6.104058404493	$3.25862e-08$	$5.33844e-09$	−2.991	0.410 sec.
1024	−6.104058375986	$4.07925e-09$	$6.68286e-10$	−2.995	0.933 sec.
2048	−6.104058372417	$5.10281e-10$	$8.35970e-11$	−2.998	2.149 sec.
4096	−6.104058371971	$6.38085e-11$	$1.04534e-11$	−2.999	5.899 sec.
8192	−6.104058371915	$7.97752e-12$	$1.30692e-12$	−2.999	19.186 sec.
16384	−6.104058371908	$9.97281e-13$	$1.63380e-13$	−3.000	77.627 sec.
32768	−6.104058371907	$1.24666e-13$	$2.04234e-14$	−3.000	393.877 sec.
Richardson	−6.104058371907	$2.08787e-39$	$3.42046e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.3: Order ρ^2/N^3 contribution from convolution 2 of λ_2

$T_2/2! = \frac{v\rho^2}{N^3}, v = (8\,\zeta(3))/2!$					
$v = 4.808227612638377141598952646045799963059945169361995527169086220$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	4.500000000000	$3.08228e-01$	$6.41042e-02$	n/a	0.007 sec.
4	4.710648148148	$9.75795e-02$	$2.02943e-02$	n/a	0.008 sec.
8	4.780640974247	$2.75866e-02$	$5.73738e-03$	-1.590	0.015 sec.
16	4.800888154890	$7.33946e-03$	$1.52644e-03$	-1.789	0.026 sec.
32	4.806334569430	$1.89304e-03$	$3.93709e-04$	-1.894	0.054 sec.
64	4.807746901183	$4.80711e-04$	$9.99769e-05$	-1.947	0.106 sec.
128	4.808106492275	$1.21120e-04$	$2.51902e-05$	-1.974	0.228 sec.
256	4.808197214037	$3.03986e-05$	$6.32221e-06$	-1.987	0.417 sec.
512	4.808219998130	$7.61451e-06$	$1.58364e-06$	-1.993	0.848 sec.
1024	4.808225707151	$1.90549e-06$	$3.96297e-07$	-1.997	1.953 sec.
2048	4.808227136034	$4.76604e-07$	$9.91227e-08$	-1.998	3.964 sec.
4096	4.808227493458	$1.19180e-07$	$2.47867e-08$	-1.999	11.556 sec.
8192	4.808227582840	$2.97987e-08$	$6.19744e-09$	-2.000	37.241 sec.
16384	4.808227605188	$7.45013e-09$	$1.54945e-09$	-2.000	134.642 sec.
32768	4.808227610776	$1.86259e-09$	$3.87375e-10$	-2.000	540.517 sec.
Richardson	4.808227612638	$4.11445e-47$	$8.55711e-48$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.4: Order ρ^2/N^4 contribution from convolution 2 of λ_2

$T_2/2! = \frac{v\rho^2}{N^4}, v = (40\,\zeta(4))/2!$					
$v = 21.646464674222763830320073930823358055495019038374538153659524 30$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.007 sec.
4	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.008 sec.
8	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.015 sec.
16	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.026 sec.
32	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.054 sec.
64	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.106 sec.
128	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.228 sec.
256	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.417 sec.
512	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	0.848 sec.
1024	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	1.953 sec.
2048	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	3.964 sec.
4096	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	11.556 sec.
8192	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	37.241 sec.
16384	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	134.642 sec.
32768	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	540.517 sec.
Richardson	21.646464674223	$0.00000e+00$	$0.00000e+00$	n/a	n/a
No Error Detected					

Table A.5: Order ρ^2/N^5 contribution from convolution 2 of λ_2

$T_2/2! = \frac{v\rho^2}{N^5}, v = (-80\,\zeta(5) + 32\,\zeta(2)\,\zeta(3))/2!$					
$v = -9.8402406009780591621012523945193575689688133135143651398345 5069$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-11.641186796732	$1.80095e+00$	$1.83019e-01$	n/a	0.007 sec.
4	-10.458849090200	$6.18608e-01$	$6.28652e-02$	n/a	0.008 sec.
8	-10.019859450639	$1.79619e-01$	$1.82535e-02$	-1.429	0.015 sec.
16	-9.888397792188	$4.81572e-02$	$4.89390e-03$	-1.740	0.026 sec.
32	-9.852687369821	$1.24468e-02$	$1.26488e-03$	-1.880	0.054 sec.
64	-9.843402977912	$3.16238e-03$	$3.21372e-04$	-1.943	0.106 sec.
128	-9.841037504351	$7.96903e-04$	$8.09841e-05$	-1.973	0.228 sec.
256	-9.840440613450	$2.00012e-04$	$2.03260e-05$	-1.987	0.417 sec.
512	-9.840290702287	$5.01013e-05$	$5.09147e-06$	-1.993	0.848 sec.
1024	-9.840253138570	$1.25376e-05$	$1.27411e-06$	-1.997	1.953 sec.
2048	-9.840243736909	$3.13593e-06$	$3.18684e-07$	-1.998	3.964 sec.
4096	-9.840241385152	$7.84174e-07$	$7.96905e-08$	-1.999	11.556 sec.
8192	-9.840240797046	$1.96067e-07$	$1.99251e-08$	-2.000	37.241 sec.
16384	-9.840240649998	$4.90199e-08$	$4.98157e-09$	-2.000	134.642 sec.
32768	-9.840240613233	$1.22553e-08$	$1.24543e-09$	-2.000	540.517 sec.
Richardson	-9.840240600978	$2.58726e-33$	$2.62926e-34$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.6: Order ρ^4/N^5 contribution from convolution 2 of λ_2

$T_2/2! = \frac{v\rho^4}{N^5}, v = (-4\,\zeta\,(5))\,/2!$					
$v = -2.073855510286739852662730972914068336114161839003825623948385356$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−2.062500000000	$1.13555e-02$	$5.47556e-03$	n/a	0.007 sec.
4	−2.072683577675	$1.17193e-03$	$5.65098e-04$	n/a	0.008 sec.
8	−2.073760812514	$9.46978e-05$	$4.56627e-05$	−3.241	0.015 sec.
16	−2.073848785031	$6.72526e-06$	$3.24288e-06$	−3.614	0.026 sec.
32	−2.073855062476	$4.47810e-07$	$2.15931e-07$	−3.809	0.054 sec.
64	−2.073855481404	$2.88831e-08$	$1.39273e-08$	−3.905	0.106 sec.
128	−2.073855508453	$1.83373e-09$	$8.84213e-10$	−3.953	0.228 sec.
256	−2.073855510171	$1.15509e-10$	$5.56976e-11$	−3.977	0.417 sec.
512	−2.073855510279	$7.24758e-12$	$3.49474e-12$	−3.988	0.848 sec.
1024	−2.073855510286	$4.53860e-13$	$2.18848e-13$	−3.994	1.953 sec.
2048	−2.073855510287	$2.83940e-14$	$1.36914e-14$	−3.997	3.964 sec.
4096	−2.073855510287	$1.77549e-15$	$8.56130e-16$	−3.999	11.556 sec.
8192	−2.073855510287	$1.10995e-16$	$5.35212e-17$	−3.999	37.241 sec.
16384	−2.073855510287	$6.93805e-18$	$3.34548e-18$	−4.000	134.642 sec.
32768	−2.073855510287	$4.33654e-19$	$2.09105e-19$	−4.000	540.517 sec.
Richardson	−2.073855510287	$2.79654e-51$	$1.34848e-51$	n/a	n/a
Error: $O\left(\text{range}^{-4.000}\right)$					

Table A.7: Order ρ^4/N^6 contribution from convolution 2 of λ_2

$T_2/2! = \frac{v\rho^4}{N^6}, v = (4\,\zeta\,(6))\,/2!$					
$v = 2.034686123968898279429035859581841055803634980065707123684817326$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	2.031250000000	$3.43612e-03$	$1.68877e-03$	n/a	0.007 sec.
4	2.034481765475	$2.04358e-04$	$1.00437e-04$	n/a	0.008 sec.
8	2.034677261530	$8.86244e-06$	$4.35568e-06$	−4.047	0.015 sec.
16	2.034685798392	$3.25577e-07$	$1.60013e-07$	−4.517	0.026 sec.
32	2.034686112950	$1.10187e-08$	$5.41542e-09$	−4.762	0.054 sec.
64	2.034686123611	$3.58204e-10$	$1.76049e-10$	−4.883	0.106 sec.
128	2.034686123957	$1.14159e-11$	$5.61066e-12$	−4.942	0.228 sec.
256	2.034686123969	$3.60259e-13$	$1.77059e-13$	−4.971	0.417 sec.
512	2.034686123969	$1.13133e-14$	$5.56021e-15$	−4.986	0.848 sec.
1024	2.034686123969	$3.54405e-16$	$1.74182e-16$	−4.993	1.953 sec.
2048	2.034686123969	$1.10887e-17$	$5.44983e-18$	−4.996	3.964 sec.
4096	2.034686123969	$3.46733e-19$	$1.70411e-19$	−4.998	11.556 sec.
8192	2.034686123969	$1.08387e-20$	$5.32697e-21$	−4.999	37.241 sec.
16384	2.034686123969	$3.38761e-22$	$1.66493e-22$	−5.000	134.642 sec.
32768	2.034686123969	$1.05871e-23$	$5.20331e-24$	−5.000	540.517 sec.
Richardson	2.034686123969	$2.09590e-59$	$1.03009e-59$	n/a	n/a
Error: $O\left(\text{range}^{-5.000}\right)$					

A.3 λ_3 Detailed Results

The third variation of lambda, λ_3, is determined by nine convolutions.

$$\lambda_3 = \sum_{k=1}^{9} T_k \tag{A.23}$$

$$T_1 = 6\rho^2 \mathrm{conv}(c_0, c_1)(0) = 0 \tag{A.24}$$

$$T_2 = 8\rho\sqrt{\pi}\mathrm{conv}(c_1, f_{1,dr})(0) = 0 \tag{A.25}$$

$$T_3 = -2\rho^2 c_2(0) \tag{A.26}$$

$$T_4 = \left(4\rho^4 - 4\rho^2\right) \mathrm{conv}(c_0, c_0, c_0)(0) \tag{A.27}$$

$$T_5 = 2\rho^2 \mathrm{conv}(c_0, c_0, c_{0,d\theta^2})(0) \tag{A.28}$$

$$T_6 = 8\rho\sqrt{\pi}\mathrm{conv}(c_0, c_0, f_{1,dr^2})(0) \tag{A.29}$$

$$T_7 = -4\pi \mathrm{conv}(c_0, f_{1,dr}, f_{1,dr})(0) \tag{A.30}$$

$$T_8 = -4\pi \mathrm{conv}(c_0, f_{1,d\theta}, f_{1,d\theta})(0) \tag{A.31}$$

$$T_9 = 4\rho\sqrt{\pi}\mathrm{conv}(c_0, f_{2,dr})(0) \tag{A.32}$$

Expressions containing $c_1 = 0$ are trivially zero. Term T_3 contains no convolutions and is also known.

$$T_3 = -2\rho^2 c_2(0) \tag{A.33}$$

$$= \frac{24\rho^2\zeta(4)}{N^4} + \frac{48\rho^2\zeta(6)}{N^6} \tag{A.34}$$

The nontrivial convolutions will be given in the next section.

A.3.1 Convolutions of Interest in λ_3

$$T_4 = \left(4\rho^4 - 4\rho^2\right)\text{conv}(c_0, c_0, c_0)(0) \tag{A.35}$$

$$= \frac{216\zeta(6)\rho^2}{N^6} - \frac{216\zeta(6)\rho^4}{N^6} \tag{A.36}$$

$$T_5 = 2\rho^2\text{conv}(c_0, c_0, c_{0,d\theta^2})(0) \tag{A.37}$$

$$= \frac{24\rho^2\zeta(4)}{N^4} - \frac{48\rho^2\zeta(6)}{N^6} \tag{A.38}$$

$$T_6 = 8\rho\sqrt{\pi}\text{conv}(c_0, c_0, f_{1,dr^2})(0) \tag{A.39}$$

$$= -\frac{96\zeta(4)\rho^2}{N^4} + \frac{96\zeta(5)\rho^2}{N^5} + \frac{528\zeta(6)\rho^2}{N^6} + \frac{432\zeta(6)\rho^4}{N^6} \tag{A.40}$$

$$T_7 = -4\pi\text{conv}(c_0, f_{1,dr}, f_{1,dr})(0) \tag{A.41}$$

$$= \frac{16\rho^2\zeta(4)}{N^4} + \frac{32\rho^2\zeta(2)\zeta(3)}{N^5} + \frac{8\rho^4\zeta(3)^2}{N^6} + \frac{224}{3}\frac{\zeta(6)\rho^2}{N^6} - \frac{88}{3}\frac{\rho^4\zeta(6)}{N^6} \tag{A.42}$$

$$T_8 = -4\pi\text{conv}(c_0, f_{1,d\theta}, f_{1,d\theta})(0) \tag{A.43}$$

$$= \frac{24\rho^2\zeta(4)}{N^4} - \frac{48\rho^2\zeta(6)}{N^6} \tag{A.44}$$

$$T_9 = 4\rho\sqrt{\pi}\text{conv}(c_0, f_{2,dr})(0) \tag{A.45}$$

$$= \frac{32\zeta(4)\rho^2}{N^4} + \frac{48\rho^2\zeta(5)}{N^5} + \frac{64\zeta(2)\zeta(3)\rho^2}{N^5} + \frac{532}{3}\frac{\zeta(6)\rho^2}{N^6} + \frac{16\rho^4\zeta(3)^2}{N^6} - \frac{596}{3}\frac{\zeta(6)\rho^4}{N^6} \tag{A.46}$$

The values divided by $3!\rho^2$ are given below.

$$\frac{T_3}{3!\rho^2} = -\frac{4\zeta(4)}{N^4} + \frac{8\zeta(6)}{N^6} \tag{A.47}$$

$$\frac{T_4}{3!\rho^2} = \frac{36\zeta(6)(1-\rho^2)}{N^6} \tag{A.48}$$

$$\frac{T_5}{3!\rho^2} = \frac{4\zeta(4)}{N^4} - \frac{8\zeta(6)}{N^6} \tag{A.49}$$

$$\frac{T_6}{3!\rho^2} = -\frac{16\zeta(4)}{N^4} + \frac{16\zeta(5)}{N^5} + \frac{88\zeta(6)}{N^6} + \frac{72\zeta(6)\rho^2}{N^6} \tag{A.50}$$

$$\frac{T_7}{3!\rho^2} = \frac{8\zeta(4)}{3N^4} + \frac{16\zeta(2)\zeta(3)}{3N^5} + \frac{112\zeta(6)}{9N^6} + \frac{4\zeta(3)^2\rho^2}{3N^6} - \frac{44\zeta(6)\rho^2}{9N^6} \tag{A.51}$$

$$\frac{T_8}{3!\rho^2} = \frac{4\zeta(4)}{N^4} - \frac{8\zeta(6)}{N^6} \tag{A.52}$$

$$\frac{T_9}{3!\rho^2} = \frac{16\zeta(4)}{3N^4} + \frac{8\zeta(5)}{N^5} + \frac{32\zeta(2)\zeta(3)}{3N^5} + \frac{266\zeta(6)}{9N^6} + \frac{8\zeta(3)^2\rho^2}{3N^6} - \frac{298\zeta(6)\rho^2}{9N^6} \tag{A.53}$$

A.3.2 Complete λ_3

$$\begin{aligned}\lambda_3 = &-\frac{24\zeta(2)\rho^2}{N^4} + \frac{144\zeta(5)\rho^2}{N^5} + \frac{96\zeta(2)\zeta(3)\rho^2}{N^5} \\ &+ \frac{948\zeta(6)\rho^2}{N^6} - \frac{12\rho^4\zeta(6)}{N^6} + \frac{24\rho^4\zeta(3)^2}{N^6}\end{aligned} \tag{A.54}$$

$$\frac{\lambda_3}{3!\rho^2} = -\frac{4\zeta(4)}{N^4} + \frac{24\zeta(5)}{N^5} + \frac{16\zeta(2)\zeta(3)}{N^5} + \frac{158\zeta(6)}{N^6} - \frac{2\zeta(6)\rho^2}{N^6} + \frac{4\zeta(3)^2\rho^2}{N^6} \tag{A.55}$$

Table A.8: Order ρ^2/N^6 contribution from convolution 4 of λ_3

$T_4/3! = \frac{v\rho^2}{N^6}, v = (216\,\zeta\,(6))\,/3!$					
$v = 36.624350231440169029722645472473139004465429641182728226326 71189$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	36.610605735565	$1.37445e-02$	$3.75283e-04$	n/a	0.003 sec.
4	36.623532797464	$8.17434e-04$	$2.23194e-05$	n/a	0.002 sec.
8	36.624314781685	$3.54498e-05$	$9.67929e-07$	-4.047	0.004 sec.
16	36.624348929133	$1.30231e-06$	$3.55585e-08$	-4.517	0.008 sec.
32	36.624350187365	$4.40747e-08$	$1.20343e-09$	-4.762	0.014 sec.
64	36.624350230007	$1.43282e-09$	$3.91220e-11$	-4.883	0.030 sec.
128	36.624350231395	$4.56637e-11$	$1.24681e-12$	-4.942	0.064 sec.
256	36.624350231439	$1.44104e-12$	$3.93464e-14$	-4.971	0.143 sec.
512	36.624350231440	$4.52531e-14$	$1.23560e-15$	-4.986	0.310 sec.
1024	36.624350231440	$1.41762e-15$	$3.87070e-17$	-4.993	0.726 sec.
2048	36.624350231440	$4.43547e-17$	$1.21107e-18$	-4.996	1.922 sec.
4096	36.624350231440	$1.38693e-18$	$3.78691e-20$	-4.998	6.105 sec.
8192	36.624350231440	$4.33549e-20$	$1.18377e-21$	-4.999	25.695 sec.
16384	36.624350231440	$1.35505e-21$	$3.69985e-23$	-5.000	92.262 sec.
32768	36.624350231440	$4.23484e-23$	$1.15629e-24$	-5.000	420.616 sec.
Richardson	36.624350231440	$8.38800e-59$	$2.29028e-60$	n/a	n/a
Error: $O\left(\text{range}^{-5.000}\right)$					

Table A.9: Order ρ^4/N^6 contribution from convolution 4 of λ_3

$T_4/3! = \frac{v\rho^4}{N^6}, v = (-216\,\zeta(6))/3!$					
$v = -36.624350231440169029722645472473139004465429641182728226326711\,89$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-36.610605735565	$1.37445e-02$	$3.75283e-04$	n/a	0.003 sec.
4	-36.623532797464	$8.17434e-04$	$2.23194e-05$	n/a	0.002 sec.
8	-36.624314781685	$3.54498e-05$	$9.67929e-07$	-4.047	0.004 sec.
16	-36.624348929133	$1.30231e-06$	$3.55585e-08$	-4.517	0.008 sec.
32	-36.624350187365	$4.40747e-08$	$1.20343e-09$	-4.762	0.014 sec.
64	-36.624350230007	$1.43282e-09$	$3.91220e-11$	-4.883	0.030 sec.
128	-36.624350231395	$4.56637e-11$	$1.24681e-12$	-4.942	0.064 sec.
256	-36.624350231439	$1.44104e-12$	$3.93464e-14$	-4.971	0.143 sec.
512	-36.624350231440	$4.52531e-14$	$1.23560e-15$	-4.986	0.310 sec.
1024	-36.624350231440	$1.41762e-15$	$3.87070e-17$	-4.993	0.726 sec.
2048	-36.624350231440	$4.43547e-17$	$1.21107e-18$	-4.996	1.922 sec.
4096	-36.624350231440	$1.38693e-18$	$3.78691e-20$	-4.998	6.105 sec.
8192	-36.624350231440	$4.33549e-20$	$1.18377e-21$	-4.999	25.695 sec.
16384	-36.624350231440	$1.35505e-21$	$3.69985e-23$	-5.000	92.262 sec.
32768	-36.624350231440	$4.23484e-23$	$1.15629e-24$	-5.000	420.616 sec.
Richardson	-36.624350231440	$8.38800e-59$	$2.29028e-60$	n/a	n/a
Error: $O\left(\text{range}^{-5.000}\right)$					

Table A.10: Order ρ^2/N^4 contribution from convolution 5 of λ_3

$T_5/3! = \frac{v\rho^2}{N^4}, v = (24\,\zeta(4))/3!$					
$v = 4.329292934844552766064014786164671611099003807674907630731904860$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	4.250000000000	$7.92929e-02$	$1.83154e-02$	n/a	0.002 sec.
4	4.315007716049	$1.42852e-02$	$3.29967e-03$	n/a	0.002 sec.
8	4.327136670814	$2.15626e-03$	$4.98064e-04$	-2.422	0.003 sec.
16	4.328996662494	$2.96272e-04$	$6.84344e-05$	-2.705	0.006 sec.
32	4.329254112372	$3.88225e-05$	$8.96739e-06$	-2.853	0.013 sec.
64	4.329287966549	$4.96830e-06$	$1.14760e-06$	-2.927	0.032 sec.
128	4.329292306473	$6.28371e-07$	$1.45144e-07$	-2.964	0.064 sec.
256	4.329292855836	$7.90084e-08$	$1.82497e-08$	-2.982	0.141 sec.
512	4.329292924940	$9.90504e-09$	$2.28791e-09$	-2.991	0.313 sec.
1024	4.329292933605	$1.23995e-09$	$2.86408e-10$	-2.995	0.674 sec.
2048	4.329292934689	$1.55107e-10$	$3.58273e-11$	-2.998	1.893 sec.
4096	4.329292934825	$1.93954e-11$	$4.48005e-12$	-2.999	5.251 sec.
8192	4.329292934842	$2.42488e-12$	$5.60109e-13$	-2.999	20.531 sec.
16384	4.329292934844	$3.03137e-13$	$7.00200e-14$	-3.000	93.700 sec.
32768	4.329292934845	$3.78939e-14$	$8.75290e-15$	-3.000	399.164 sec.
Richardson	4.329292934845	$6.37690e-47$	$1.47297e-47$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.11: Order ρ^2/N^6 contribution from convolution 5 of λ_3

$T_5/3! = \frac{v\rho^2}{N^6}, v = (-48\,\zeta(6))/3!$					
$v = -8.1387444958755931177161434383273642232145399202628284947392693 07$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−10.019483454321	$1.88074e+00$	$2.31085e-01$	n/a	0.002 sec.
4	−8.502454875017	$3.63710e-01$	$4.46888e-02$	n/a	0.002 sec.
8	−8.194963344120	$5.62188e-02$	$6.90756e-03$	−2.303	0.003 sec.
16	−8.146522536974	$7.77804e-03$	$9.55681e-04$	−2.666	0.006 sec.
32	−8.139765601278	$1.02111e-03$	$1.25462e-04$	−2.842	0.013 sec.
64	−8.138875234676	$1.30739e-04$	$1.60638e-05$	−2.924	0.032 sec.
128	−8.138761033255	$1.65374e-05$	$2.03193e-06$	−2.963	0.064 sec.
256	−8.138746575272	$2.07940e-06$	$2.55494e-07$	−2.982	0.141 sec.
512	−8.138744756565	$2.60690e-07$	$3.20307e-08$	−2.991	0.313 sec.
1024	−8.138744528510	$3.26340e-08$	$4.00971e-09$	−2.995	0.674 sec.
2048	−8.138744499958	$4.08225e-09$	$5.01582e-10$	−2.998	1.893 sec.
4096	−8.138744496386	$5.10468e-10$	$6.27207e-11$	−2.999	5.251 sec.
8192	−8.138744495939	$6.38202e-11$	$7.84152e-12$	−2.999	20.531 sec.
16384	−8.138744495884	$7.97825e-12$	$9.80280e-13$	−3.000	93.700 sec.
32768	−8.138744495877	$9.97327e-13$	$1.22541e-13$	−3.000	399.164 sec.
Richardson	−8.138744495876	$1.35609e-38$	$1.66622e-39$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.12: Order ρ^2/N^4 contribution from convolution 6 of λ_3

$T_6/3! = \frac{v\rho^2}{N^4}, v = (-96\,\zeta(4))/3!$					
$v = -17.317171739378211064256059144658686444396015230699630522927619 44$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−17.000000000000	$3.17172e-01$	$1.83154e-02$	n/a	0.009 sec.
4	−17.260030864198	$5.71409e-02$	$3.29967e-03$	n/a	0.015 sec.
8	−17.308546683256	$8.62506e-03$	$4.98064e-04$	−2.422	0.023 sec.
16	−17.315986649977	$1.18509e-03$	$6.84344e-05$	−2.705	0.040 sec.
32	−17.317016449488	$1.55290e-04$	$8.96739e-06$	−2.853	0.079 sec.
64	−17.317151866197	$1.98732e-05$	$1.14760e-06$	−2.927	0.166 sec.
128	−17.317169225894	$2.51348e-06$	$1.45144e-07$	−2.964	0.325 sec.
256	−17.317171423345	$3.16034e-07$	$1.82497e-08$	−2.982	0.845 sec.
512	−17.317171699758	$3.96202e-08$	$2.28791e-09$	−2.991	1.369 sec.
1024	−17.317171734418	$4.95978e-09$	$2.86408e-10$	−2.995	3.015 sec.
2048	−17.317171738758	$6.20427e-10$	$3.58273e-11$	−2.998	6.620 sec.
4096	−17.317171739301	$7.75818e-11$	$4.48005e-12$	−2.999	16.881 sec.
8192	−17.317171739369	$9.69950e-12$	$5.60109e-13$	−2.999	48.551 sec.
16384	−17.317171739377	$1.21255e-12$	$7.00200e-14$	−3.000	153.422 sec.
32768	−17.317171739378	$1.51576e-13$	$8.75290e-15$	−3.000	622.884 sec.
Richardson	−17.317171739378	$2.55076e-46$	$1.47297e-47$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.13: Order ρ^2/N^5 contribution from convolution 6 of λ_3

$T_6/3! = \frac{v\rho^2}{N^5}, v = (96\,\zeta(5))/3!$					
$v = 16.590844082293918821301847783312546688913294712030604991587082\,85$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	16.500000000000	$9.08441e-02$	$5.47556e-03$	n/a	0.009 sec.
4	16.581468621399	$9.37546e-03$	$5.65098e-04$	n/a	0.015 sec.
8	16.590086500110	$7.57582e-04$	$4.56627e-05$	-3.241	0.023 sec.
16	16.590790280247	$5.38020e-05$	$3.24288e-06$	-3.614	0.040 sec.
32	16.590840499811	$3.58248e-06$	$2.15931e-07$	-3.809	0.079 sec.
64	16.590843851229	$2.31065e-07$	$1.39273e-08$	-3.905	0.166 sec.
128	16.590844067624	$1.46698e-08$	$8.84213e-10$	-3.953	0.325 sec.
256	16.590844081370	$9.24070e-10$	$5.56976e-11$	-3.977	0.845 sec.
512	16.590844082236	$5.79807e-11$	$3.49474e-12$	-3.988	1.369 sec.
1024	16.590844082290	$3.63088e-12$	$2.18848e-13$	-3.994	3.015 sec.
2048	16.590844082294	$2.27152e-13$	$1.36914e-14$	-3.997	6.620 sec.
4096	16.590844082294	$1.42039e-14$	$8.56130e-16$	-3.999	16.881 sec.
8192	16.590844082294	$8.87962e-16$	$5.35212e-17$	-3.999	48.551 sec.
16384	16.590844082294	$5.55044e-17$	$3.34548e-18$	-4.000	153.422 sec.
32768	16.590844082294	$3.46924e-18$	$2.09105e-19$	-4.000	622.884 sec.
Richardson	16.590844082294	$2.23723e-50$	$1.34848e-51$	n/a	n/a
Error: $O\left(\text{range}^{-4.000}\right)$					

Table A.14: Order ρ^2/N^6 contribution from convolution 6 of λ_3

$T_6/3! = \frac{v\rho^2}{N^6}, v = (528\,\zeta(6))/3!$					
$v = 89.526189454631524294877577821601006455359939122891113442131962\,38$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	97.049145288412	$7.52296e+00$	$8.40308e-02$	n/a	0.009 sec.
4	90.981030971195	$1.45484e+00$	$1.62505e-02$	n/a	0.015 sec.
8	89.751064847611	$2.24875e-01$	$2.51184e-03$	-2.303	0.023 sec.
16	89.557301619023	$3.11122e-02$	$3.47520e-04$	-2.666	0.040 sec.
32	89.530273876241	$4.08442e-03$	$4.56226e-05$	-2.842	0.079 sec.
64	89.526712409831	$5.22955e-04$	$5.84137e-06$	-2.924	0.166 sec.
128	89.526255604150	$6.61495e-05$	$7.38885e-07$	-2.963	0.325 sec.
256	89.526197772217	$8.31759e-06$	$9.29067e-08$	-2.982	0.845 sec.
512	89.526190497390	$1.04276e-06$	$1.16475e-08$	-2.991	1.369 sec.
1024	89.526189585168	$1.30536e-07$	$1.45808e-09$	-2.995	3.015 sec.
2048	89.526189470961	$1.63290e-08$	$1.82393e-10$	-2.998	6.620 sec.
4096	89.526189456673	$2.04187e-09$	$2.28075e-11$	-2.999	16.881 sec.
8192	89.526189454887	$2.55281e-10$	$2.85146e-12$	-2.999	48.551 sec.
16384	89.526189454663	$3.19130e-11$	$3.56466e-13$	-3.000	153.422 sec.
32768	89.526189454636	$3.98931e-12$	$4.45602e-14$	-3.000	622.884 sec.
Richardson	89.526189454632	$5.42436e-38$	$6.05897e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.15: Order ρ^4/N^6 contribution from convolution 6 of λ_3

$T_6/3! = \frac{v\rho^4}{N^6}, v = (432\,\zeta\,(6))\,/3!$					
$v = 73.248700462880338059445290944946278008930859282365456452653423 76$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	73.221211471129	$2.74890e-02$	$3.75283e-04$	n/a	0.009 sec.
4	73.247065594929	$1.63487e-03$	$2.23194e-05$	n/a	0.015 sec.
8	73.248629563369	$7.08995e-05$	$9.67929e-07$	-4.047	0.023 sec.
16	73.248697858265	$2.60462e-06$	$3.55585e-08$	-4.517	0.040 sec.
32	73.248700374731	$8.81495e-08$	$1.20343e-09$	-4.762	0.079 sec.
64	73.248700460015	$2.86564e-09$	$3.91220e-11$	-4.883	0.166 sec.
128	73.248700462789	$9.13275e-11$	$1.24681e-12$	-4.942	0.325 sec.
256	73.248700462877	$2.88207e-12$	$3.93464e-14$	-4.971	0.845 sec.
512	73.248700462880	$9.05062e-14$	$1.23560e-15$	-4.986	1.369 sec.
1024	73.248700462880	$2.83524e-15$	$3.87070e-17$	-4.993	3.015 sec.
2048	73.248700462880	$8.87095e-17$	$1.21107e-18$	-4.996	6.620 sec.
4096	73.248700462880	$2.77386e-18$	$3.78691e-20$	-4.998	16.881 sec.
8192	73.248700462880	$8.67097e-20$	$1.18377e-21$	-4.999	48.551 sec.
16384	73.248700462880	$2.71009e-21$	$3.69985e-23$	-5.000	153.422 sec.
32768	73.248700462880	$8.46968e-23$	$1.15629e-24$	-5.000	622.884 sec.
Richardson	73.248700462880	$1.67010e-58$	$2.28004e-60$	n/a	n/a
Error: $O\left(\text{range}^{-5.000}\right)$					

Table A.16: Order ρ^2/N^4 contribution from convolution 7 of λ_3

$T_7/3! = \frac{v\rho^2}{N^4}, v = (16\,\zeta\,(4))\,/3!$					
$v = 2.8861952898963685107093431907764477407326692051166050871546032 40$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	3.066446889494	$1.80252e-01$	$6.24530e-02$	n/a	0.017 sec.
4	2.984561941018	$9.83667e-02$	$3.40818e-02$	n/a	0.023 sec.
8	2.926275815681	$4.00805e-02$	$1.38870e-02$	-0.490	0.036 sec.
16	2.900192604426	$1.39973e-02$	$4.84975e-03$	-1.160	0.070 sec.
32	2.890673322250	$4.47803e-03$	$1.55153e-03$	-1.454	0.152 sec.
64	2.887553694988	$1.35841e-03$	$4.70656e-04$	-1.609	0.301 sec.
128	2.886593417022	$3.98127e-04$	$1.37942e-04$	-1.700	0.689 sec.
256	2.886309245054	$1.13955e-04$	$3.94828e-05$	-1.757	2.214 sec.
512	2.886227351372	$3.20615e-05$	$1.11086e-05$	-1.795	2.971 sec.
1024	2.886204193411	$8.90352e-06$	$3.08486e-06$	-1.822	8.499 sec.
2048	2.886197737074	$2.44718e-06$	$8.47891e-07$	-1.843	18.389 sec.
4096	2.886195956910	$6.67014e-07$	$2.31105e-07$	-1.859	37.881 sec.
8192	2.886195470440	$1.80543e-07$	$6.25541e-08$	-1.872	96.229 sec.
16384	2.886195338478	$4.85812e-08$	$1.68323e-08$	-1.882	279.892 sec.
32768	2.886195302903	$1.30064e-08$	$4.50641e-09$	-1.891	1005.533 sec.
Richardson	2.886195289896	$7.87774e-18$	$2.72945e-18$	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.17: Order ρ^2/N^5 contribution from convolution 7 of λ_3

$T_7/3! = \frac{v\rho^2}{N^5}, v = (32\,\zeta(2)\,\zeta(3))/3!$					
$v = 10.545623201585579297051122354587336384438141155520715779711052 14$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	10.207613801337	$3.38009e-01$	$3.20521e-02$	n/a	0.017 sec.
4	10.438615344676	$1.07008e-01$	$1.01471e-02$	n/a	0.023 sec.
8	10.515371067399	$3.02521e-02$	$2.86869e-03$	-1.590	0.036 sec.
16	10.537574585531	$8.04862e-03$	$7.63219e-04$	-1.789	0.070 sec.
32	10.543547247410	$2.07595e-03$	$1.96855e-04$	-1.894	0.152 sec.
64	10.545096042486	$5.27159e-04$	$4.99884e-05$	-1.947	0.301 sec.
128	10.545490378244	$1.32823e-04$	$1.25951e-05$	-1.974	0.689 sec.
256	10.545589865789	$3.33358e-05$	$3.16110e-06$	-1.987	2.214 sec.
512	10.545614851343	$8.35024e-06$	$7.91821e-07$	-1.993	2.971 sec.
1024	10.545621111985	$2.08960e-06$	$1.98149e-07$	-1.997	8.499 sec.
2048	10.545622678930	$5.22655e-07$	$4.95613e-08$	-1.998	18.389 sec.
4096	10.545623070890	$1.30696e-07$	$1.23934e-08$	-1.999	37.881 sec.
8192	10.545623168908	$3.26779e-08$	$3.09872e-09$	-2.000	96.229 sec.
16384	10.545623193416	$8.16998e-09$	$7.74727e-10$	-2.000	279.892 sec.
32768	10.545623199543	$2.04256e-09$	$1.93688e-10$	-2.000	1005.533 sec.
Richardson	10.545623201586	$4.51200e-47$	$4.27856e-48$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.18: Order ρ^2/N^6 contribution from convolution 7 of λ_3

$T_7/3! = \frac{v\rho^2}{N^6}, v = \left(\frac{224}{3}\,\zeta(6)\right)/3!$					
$v = 12.660269215806478183114000904064788791667062098186622102927752 26$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	12.482436706676	$1.77833e-01$	$1.40465e-02$	n/a	0.017 sec.
4	13.105805773806	$4.45537e-01$	$3.51917e-02$	n/a	0.023 sec.
8	12.926771792767	$2.66503e-01$	$2.10503e-02$	-1.800	0.036 sec.
16	12.766751046340	$1.06482e-01$	$8.41071e-03$	-0.162	0.070 sec.
32	12.696615026449	$3.63458e-02$	$2.87086e-03$	-1.190	0.152 sec.
64	12.671718692883	$1.14495e-02$	$9.04363e-04$	-1.494	0.301 sec.
128	12.663709515500	$3.44030e-03$	$2.71740e-04$	-1.636	0.689 sec.
256	12.661271650513	$1.00243e-03$	$7.91796e-05$	-1.716	2.214 sec.
512	12.660555083819	$2.85868e-04$	$2.25799e-05$	-1.766	2.971 sec.
1024	12.660349448061	$8.02323e-05$	$6.33733e-06$	-1.801	8.499 sec.
2048	12.660291457942	$2.22421e-05$	$1.75685e-06$	-1.826	18.389 sec.
4096	12.660275321322	$6.10552e-06$	$4.82258e-07$	-1.845	37.881 sec.
8192	12.660270878285	$1.66248e-06$	$1.31315e-07$	-1.861	96.229 sec.
16384	12.660269665431	$4.49624e-07$	$3.55146e-08$	-1.873	279.892 sec.
32768	12.660269336711	$1.20904e-07$	$9.54990e-09$	-1.883	1005.533 sec.
Richardson	12.660269215808	$1.75428e-12$	$1.38566e-13$	n/a	n/a
Error: $O\left(\text{range}^{-1.883}\right)$					

Table A.19: Order ρ^4/N^6 contribution from convolution 7 of λ_3

$T_7/3! = \frac{v\rho^4}{N^6}, v = \left(8\,(\zeta(3))^2 - \frac{88}{3}\zeta(6)\right)/3!$					
$v = -3.0470894606791279267193964405685165890527938546976547368320856 96$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-3.201311167997	$1.54222e-01$	$5.06128e-02$	n/a	0.017 sec.
4	-3.091472748182	$4.43833e-02$	$1.45658e-02$	n/a	0.023 sec.
8	-3.058746313678	$1.16569e-02$	$3.82557e-03$	-1.747	0.036 sec.
16	-3.050085677383	$2.99622e-03$	$9.83304e-04$	-1.918	0.070 sec.
32	-3.047852498395	$7.63038e-04$	$2.50415e-04$	-1.955	0.152 sec.
64	-3.047282420149	$1.92959e-04$	$6.33258e-05$	-1.970	0.301 sec.
128	-3.047138017188	$4.85565e-05$	$1.59354e-05$	-1.981	0.689 sec.
256	-3.047101642771	$1.21821e-05$	$3.99794e-06$	-1.989	2.214 sec.
512	-3.047092511830	$3.05115e-06$	$1.00133e-06$	-1.994	2.971 sec.
1024	-3.047090224189	$7.63510e-07$	$2.50570e-07$	-1.997	8.499 sec.
2048	-3.047089651648	$1.90969e-07$	$6.26726e-08$	-1.998	18.389 sec.
4096	-3.047089508433	$4.77538e-08$	$1.56719e-08$	-1.999	37.881 sec.
8192	-3.047089472619	$1.19399e-08$	$3.91846e-09$	-2.000	96.229 sec.
16384	-3.047089463664	$2.98516e-09$	$9.79675e-10$	-2.000	279.892 sec.
32768	-3.047089461425	$7.46312e-10$	$2.44926e-10$	-2.000	1005.533 sec.
Richardson	-3.047089460679	$3.02411e-21$	$9.92460e-22$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.20: Order ρ^2/N^4 contribution from convolution 8 of λ_3

$T_8/3! = \frac{v\rho^2}{N^4}, v = (24\,\zeta(4))/3!$					
$v = 4.3292929348445527660640147861646716110990038076749076307319048 60$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	4.250000000000	$7.92929e-02$	$1.83154e-02$	n/a	0.004 sec.
4	4.315007716049	$1.42852e-02$	$3.29967e-03$	n/a	0.006 sec.
8	4.327136670814	$2.15626e-03$	$4.98064e-04$	-2.422	0.012 sec.
16	4.328996662494	$2.96272e-04$	$6.84344e-05$	-2.705	0.025 sec.
32	4.329254112372	$3.88225e-05$	$8.96739e-06$	-2.853	0.049 sec.
64	4.329287966549	$4.96830e-06$	$1.14760e-06$	-2.927	0.099 sec.
128	4.329292306473	$6.28371e-07$	$1.45144e-07$	-2.964	0.205 sec.
256	4.329292855836	$7.90084e-08$	$1.82497e-08$	-2.982	0.443 sec.
512	4.329292924940	$9.90504e-09$	$2.28791e-09$	-2.991	0.884 sec.
1024	4.329292933605	$1.23995e-09$	$2.86408e-10$	-2.995	3.358 sec.
2048	4.329292934689	$1.55107e-10$	$3.58273e-11$	-2.998	4.546 sec.
4096	4.329292934825	$1.93954e-11$	$4.48005e-12$	-2.999	15.526 sec.
8192	4.329292934842	$2.42488e-12$	$5.60109e-13$	-2.999	52.412 sec.
16384	4.329292934844	$3.03137e-13$	$7.00200e-14$	-3.000	210.710 sec.
32768	4.329292934845	$3.78939e-14$	$8.75290e-15$	-3.000	899.548 sec.
Richardson	4.329292934845	$6.37690e-47$	$1.47297e-47$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.21: Order ρ^2/N^6 contribution from convolution 8 of λ_3

$T_8/3! = \frac{v\rho^2}{N^6}, v = (-48\,\zeta(6))/3!$					
$v = -8.138744495875593117716143438327364223214539920262828494739269307$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-10.019483454321	$1.88074e+00$	$2.31085e-01$	n/a	0.004 sec.
4	-8.502454875017	$3.63710e-01$	$4.46888e-02$	n/a	0.006 sec.
8	-8.194963344120	$5.62188e-02$	$6.90756e-03$	-2.303	0.012 sec.
16	-8.146522536974	$7.77804e-03$	$9.55681e-04$	-2.666	0.025 sec.
32	-8.139765601278	$1.02111e-03$	$1.25462e-04$	-2.842	0.049 sec.
64	-8.138875234676	$1.30739e-04$	$1.60638e-05$	-2.924	0.099 sec.
128	-8.138761033255	$1.65374e-05$	$2.03193e-06$	-2.963	0.205 sec.
256	-8.138746575272	$2.07940e-06$	$2.55494e-07$	-2.982	0.443 sec.
512	-8.138744756565	$2.60690e-07$	$3.20307e-08$	-2.991	0.884 sec.
1024	-8.138744528510	$3.26340e-08$	$4.00971e-09$	-2.995	3.358 sec.
2048	-8.138744499958	$4.08225e-09$	$5.01582e-10$	-2.998	4.546 sec.
4096	-8.138744496386	$5.10468e-10$	$6.27207e-11$	-2.999	15.526 sec.
8192	-8.138744495939	$6.38202e-11$	$7.84152e-12$	-2.999	52.412 sec.
16384	-8.138744495884	$7.97825e-12$	$9.80280e-13$	-3.000	210.710 sec.
32768	-8.138744495877	$9.97327e-13$	$1.22541e-13$	-3.000	899.548 sec.
Richardson	-8.138744495876	$1.35609e-38$	$1.66622e-39$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.22: Order ρ^2/N^4 contribution from convolution 9 of λ_3

$T_9/3! = \frac{v\rho^2}{N^4}, v = (32\,\zeta(4))/3!$					
$v = 5.772390579792737021418686381552895481465338410233210174309206480$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	6.132893778988	$3.60503e-01$	$6.24530e-02$	n/a	0.021 sec.
4	5.969123882037	$1.96733e-01$	$3.40818e-02$	n/a	0.028 sec.
8	5.852551631362	$8.01611e-02$	$1.38870e-02$	-0.490	0.055 sec.
16	5.800385208851	$2.79946e-02$	$4.84975e-03$	-1.160	0.098 sec.
32	5.781346644500	$8.95606e-03$	$1.55153e-03$	-1.454	0.203 sec.
64	5.775107389977	$2.71681e-03$	$4.70656e-04$	-1.609	0.439 sec.
128	5.773186834044	$7.96254e-04$	$1.37942e-04$	-1.700	1.007 sec.
256	5.772618490109	$2.27910e-04$	$3.94828e-05$	-1.757	3.282 sec.
512	5.772454702745	$6.41230e-05$	$1.11086e-05$	-1.795	4.295 sec.
1024	5.772408386823	$1.78070e-05$	$3.08486e-06$	-1.822	10.684 sec.
2048	5.772395474149	$4.89436e-06$	$8.47891e-07$	-1.843	23.074 sec.
4096	5.772391913820	$1.33403e-06$	$2.31105e-07$	-1.859	49.618 sec.
8192	5.772390940879	$3.61087e-07$	$6.25541e-08$	-1.872	122.442 sec.
16384	5.772390676955	$9.71624e-08$	$1.68323e-08$	-1.882	367.899 sec.
32768	5.772390605805	$2.60127e-08$	$4.50641e-09$	-1.891	1250.308 sec.
Richardson	5.772390579793	$1.57555e-17$	$2.72945e-18$	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.23: Order ρ^2/N^5 contribution from convolution 9 of λ_3

$T_9/3! = \frac{v\rho^2}{N^5}, v = (48\,\zeta(5) + 64\,\zeta(2)\,\zeta(3))\,/3!$					
$v = 29.386668444318118004753168600830946113332929667056734055215645 71$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	28.665227602675	$7.21441e-01$	$2.45499e-02$	n/a	0.021 sec.
4	29.167965000051	$2.18703e-01$	$7.44227e-03$	n/a	0.028 sec.
8	29.325785384853	$6.08831e-02$	$2.07179e-03$	-1.672	0.055 sec.
16	29.370544311186	$1.61241e-02$	$5.48689e-04$	-1.818	0.098 sec.
32	29.382514744726	$4.15370e-03$	$1.41346e-04$	-1.903	0.203 sec.
64	29.385614010587	$1.05443e-03$	$3.58814e-05$	-1.949	0.439 sec.
128	29.386402790301	$2.65654e-04$	$9.03995e-06$	-1.974	1.007 sec.
256	29.386601772262	$6.66721e-05$	$2.26879e-06$	-1.987	3.282 sec.
512	29.386651743804	$1.67005e-05$	$5.68302e-07$	-1.993	4.295 sec.
1024	29.386664265116	$4.17920e-06$	$1.42214e-07$	-1.997	10.684 sec.
2048	29.386667399008	$1.04531e-06$	$3.55709e-08$	-1.998	23.074 sec.
4096	29.386668182927	$2.61391e-07$	$8.89490e-09$	-1.999	49.618 sec.
8192	29.386668378962	$6.53558e-08$	$2.22400e-09$	-2.000	122.442 sec.
16384	29.386668427978	$1.63400e-08$	$5.56033e-10$	-2.000	367.899 sec.
32768	29.386668440233	$4.08511e-09$	$1.39012e-10$	-2.000	1250.308 sec.
Richardson	29.386668444318	$3.93947e-40$	$1.34056e-41$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.24: Order ρ^2/N^6 contribution from convolution 9 of λ_3

$T_9/3! = \frac{v\rho^2}{N^6}, v = \left(\frac{532}{3}\,\zeta(6)\right)/3!$					
$v = 30.068139387540385684895752147153873380209272483193227494453411 60$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	29.712474369280	$3.55665e-01$	$1.18286e-02$	n/a	0.021 sec.
4	30.959212503539	$8.91073e-01$	$2.96351e-02$	n/a	0.028 sec.
8	30.601144541462	$5.33005e-01$	$1.77266e-02$	-1.800	0.055 sec.
16	30.281103048608	$2.12964e-01$	$7.08270e-03$	-0.162	0.098 sec.
32	30.140831008826	$7.26916e-02$	$2.41756e-03$	-1.190	0.203 sec.
64	30.091038341693	$2.28990e-02$	$7.61569e-04$	-1.494	0.439 sec.
128	30.075019986927	$6.88060e-03$	$2.28834e-04$	-1.636	1.007 sec.
256	30.070144256954	$2.00487e-03$	$6.66775e-05$	-1.716	3.282 sec.
512	30.068711123565	$5.71736e-04$	$1.90147e-05$	-1.766	4.295 sec.
1024	30.068299852049	$1.60465e-04$	$5.33670e-06$	-1.801	10.684 sec.
2048	30.068183871811	$4.44843e-05$	$1.47945e-06$	-1.826	23.074 sec.
4096	30.068151598571	$1.22110e-05$	$4.06112e-07$	-1.845	49.618 sec.
8192	30.068142712497	$3.32496e-06$	$1.10581e-07$	-1.861	122.442 sec.
16384	30.068140286789	$8.99248e-07$	$2.99070e-08$	-1.873	367.899 sec.
32768	30.068139629349	$2.41809e-07$	$8.04202e-09$	-1.883	1250.308 sec.
Richardson	30.068139387544	$3.50857e-12$	$1.16687e-13$	n/a	n/a
Error: $O\left(\text{range}^{-1.883}\right)$					

Table A.25: Order ρ^4/N^6 contribution from convolution 9 of λ_3

$T_9/3! = \frac{v\rho^4}{N^6}, v = \left(16\,(\zeta(3))^2 - \frac{596}{3}\,\zeta(6)\right)/3!$					
$v = -29.83218370099540244677754457625851216248132914349522591665370687$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-30.140627115632	$3.08443e-01$	$1.03393e-02$	n/a	0.021 sec.
4	-29.920950276000	$8.87666e-02$	$2.97553e-03$	n/a	0.028 sec.
8	-29.855497406992	$2.33137e-02$	$7.81495e-04$	-1.747	0.055 sec.
16	-29.838176134403	$5.99243e-03$	$2.00871e-04$	-1.918	0.098 sec.
32	-29.833709776428	$1.52608e-03$	$5.11553e-05$	-1.955	0.203 sec.
64	-29.832569619935	$3.85919e-04$	$1.29363e-05$	-1.970	0.439 sec.
128	-29.832280814012	$9.71130e-05$	$3.25531e-06$	-1.981	1.007 sec.
256	-29.832208065179	$2.43642e-05$	$8.16708e-07$	-1.989	3.282 sec.
512	-29.832189803297	$6.10230e-06$	$2.04554e-07$	-1.994	4.295 sec.
1024	-29.832185228015	$1.52702e-06$	$5.11870e-08$	-1.997	10.684 sec.
2048	-29.832184082934	$3.81938e-07$	$1.28029e-08$	-1.998	23.074 sec.
4096	-29.832183796503	$9.55077e-08$	$3.20150e-09$	-1.999	49.618 sec.
8192	-29.832183724875	$2.38798e-08$	$8.00472e-10$	-2.000	122.442 sec.
16384	-29.832183706966	$5.97032e-09$	$2.00130e-10$	-2.000	367.899 sec.
32768	-29.832183702488	$1.49262e-09$	$5.00340e-11$	-2.000	1250.308 sec.
Richardson	-29.832183700995	$6.04823e-21$	$2.02742e-22$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

A.4 λ_4 Detailed Results

The third variation, λ_4, is determined by 27 expressions. The list of convolutions starts with the simplest expressions.

$$\lambda_4 = \sum_{k=1}^{27} T_k \tag{A.56}$$

$$T_1 = -2\rho^2 c_3(0) \tag{A.57}$$

$$T_2 = (24\rho^4 - 24\rho^2)\mathrm{conv}\,(c_0, c_0, c_1)\,(0) = 0 \tag{A.58}$$

$$T_3 = 12\rho\sqrt{\pi}\mathrm{conv}\,(c_1 f_{2,dr})\,(0) = 0 \tag{A.59}$$

$$T_4 = 36\rho\sqrt{\pi}\mathrm{conv}\left(c_0, c_1, f_{1,dr^2}\right)(0) = 0 \tag{A.60}$$

$$T_5 = 6\rho^2\mathrm{conv}\,(c_1, c_1)\,(0) = 0 \tag{A.61}$$

$$T_6 = -12\pi\mathrm{conv}\,(c_1, f_{1,dr}, f_{1,dr})\,(0) = 0 \tag{A.62}$$

$$T_7 = -12\pi\mathrm{conv}\,(c_1, f_{1,d\theta}, f_{1,d\theta})\,(0) = 0 \tag{A.63}$$

$$T_8 = -2\rho^2\mathrm{conv}\,(R_{AbC}, c_0, c_0, c_{0,d\theta})\,(0) = 0 \tag{A.64}$$

$$T_9 = 2\rho^2\mathrm{conv}\left(c_0, c_0, c_{1,d\theta^2}\right)(0) = 0 \tag{A.65}$$

$$T_{10} = 10\rho^2\mathrm{conv}\left(c_0, c_{0,d\theta^2}, c_1\right)(0) = 0 \tag{A.66}$$

$$T_{11} = (-8\rho^4 + 12\rho^2)\mathrm{conv}\,(c_0, c_0, c_0, c_0)\,(0) \tag{A.67}$$

$$T_{12} = -14\rho^2\mathrm{conv}\left(c_0, c_0, c_0, c_{0,d\theta^2}\right)(0) \tag{A.68}$$

$$T_{13} = -6\rho^2\mathrm{conv}\,(c_0, c_0, c_{0,d\theta}, c_{0,d\theta})\,(0) \tag{A.69}$$

$$T_{14} = 8\rho^2\mathrm{conv}\,(c_0, c_2)\,(0) \tag{A.70}$$

The next set of terms depends on f_1 and its derivatives.

$$T_{15} = 12\rho\sqrt{\pi}\text{conv}\left(c_2, f_{1,dr}\right)(0) \tag{A.71}$$

$$T_{16} = -12\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_{0,d\theta}, f_{1,drd\theta}\right)(0) \tag{A.72}$$

$$T_{17} = -12\rho^2\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, f_{1,dr}\right)(0) \tag{A.73}$$

$$T_{18} = -12\rho\sqrt{\pi}\text{convconv}\left(c_0, c_0, c_{0,d\theta^2}, f_{1,dr}\right)(0) \tag{A.74}$$

$$T_{19} = 12\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, f_{1,dr^3}\right)(0) \tag{A.75}$$

$$T_{20} = -12\pi\text{conv}\left(c_0, c_0, f_{1,dr}, f_{1,dr}\right)(0) \tag{A.76}$$

$$T_{21} = 12\pi\text{conv}\left(c_0, c_0, f_{1,d\theta}, f_{1,d\theta}\right)(0) \tag{A.77}$$

$$T_{22} = -24\pi\text{conv}\left(c_0, c_0, f_{1,dr}, f_{1,dr^2}\right)(0) \tag{A.78}$$

$$T_{23} = -24\pi\text{conv}\left(c_0, c_0, f_{1,d\theta}, f_{1,drd\theta}\right)(0) \tag{A.79}$$

$$\tag{A.80}$$

The most difficult convolutions to work with contain higher partial derivatives of u.

$$T_{24} = 12\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, f_{2,dr^2}\right)(0) \tag{A.81}$$

$$T_{25} = -12\pi\text{conv}\left(c_0, f_{1,d\theta}, f_{2,d\theta}\right)(0) \tag{A.82}$$

$$T_{26} = -12\pi\text{conv}\left(c_0, f_{1,dr}f_{2,dr}\right)(0) \tag{A.83}$$

$$T_{27} = 4\rho\sqrt{\pi}\text{conv}\left(c_0, f_{3,dr}\right)(0) \tag{A.84}$$

Expressions containing $c_1 = 0$ or R_{AbC} are trivially zero. This eliminates T_2 throught T_{10}. Term T_1 contains no convolutions and is also known.

$$T_1 = -2\rho^2 c_2(0) \tag{A.85}$$

$$= -\frac{360\rho^2\zeta(6)}{N^6} \tag{A.86}$$

A.4.1 Convolutions of Interest in λ_4

$$T_{11} = -\frac{1920\zeta(8)\rho^4}{N^8} + \frac{2880\zeta(8)\rho^2}{N^8} \tag{A.87}$$
$$T_{12} = \frac{756\rho^2\zeta(6)}{N^6} - \frac{1680\rho^2\zeta(8)}{N^8} \tag{A.88}$$
$$T_{13} = -\frac{108\rho^2\zeta(6)}{N^6} + \frac{240\rho^2\zeta(8)}{N^8} \tag{A.89}$$
$$T_{14} = -\frac{288\rho^2\zeta(6)}{N^6} + \frac{960\rho^2\zeta(8)}{N^8} \tag{A.90}$$
$$T_{15} = -\frac{288\rho^2\zeta(2)\zeta(3)}{N^5} + \frac{576\rho^2\zeta(5)}{N^5} - \frac{504\zeta(6)\rho^2}{N^6} \tag{A.91}$$
$$T_{16} = -\frac{720\rho^2\zeta(5)}{N^5} + \frac{288\rho^2\zeta(2)\zeta(3)}{N^5} \tag{A.92}$$
$$T_{17} = \frac{864\rho^4\zeta(2)\zeta(5)}{N^7} - \frac{2160\rho^4\zeta(7)}{N^7} \tag{A.93}$$
$$T_{18} = \frac{144\rho^2\zeta(5)}{N^5} + \frac{504\rho^2\zeta(6)}{N^6} \tag{A.94}$$
$$T_{19} = \frac{2160\rho^2\zeta(5)}{N^5} - \frac{864\rho^2\zeta(2)\zeta(3)}{N^5} - \frac{1944\rho^2\zeta(6)}{N^6} \tag{A.95}$$
$$T_{20} = \frac{216\rho^2\zeta(6)}{N^6} - \frac{288\rho^2\zeta(3)^2}{N^6} \tag{A.96}$$
$$T_{21} = \frac{216\rho^2\zeta(6)}{N^6} - \frac{480\rho^2\zeta(8)}{N^8} \tag{A.97}$$
$$T_{22} = -\frac{288\rho^2\zeta(5)}{N^5} - \frac{1440\rho^2\zeta(6)}{N^6} + \frac{576\rho^2\zeta(3)^2}{N^6} \tag{A.98}$$
$$T_{23} = \frac{576\rho^2\zeta(2)\zeta(3)}{N^5} - \frac{1440\rho^2\zeta(5)}{N^5} \tag{A.99}$$
$$T_{24} = \frac{576\rho^2\zeta(2)\zeta(3)}{N^5} - \frac{1728\rho^2\zeta(5)}{N^5} - \frac{816\rho^2\zeta(6)}{N^6} - \frac{288\rho^2\zeta(3)^2}{N^6} \tag{A.100}$$
$$T_{25} = -\frac{288\rho^2\zeta(2)\zeta(3)}{N^5} + \frac{864\rho^2\zeta(5)}{N^5} + \frac{936\rho^2\zeta(6)}{N^6} \tag{A.101}$$
$$T_{26} = -\frac{192\rho^2\zeta(2)\zeta(3)}{N^5} + \frac{624\rho^2\zeta(5)}{N^5} - \frac{48\rho^2\zeta(3)^2}{N^6} + \frac{1200\zeta(6)\rho^2}{N^6} \tag{A.102}$$
$$T_{27} = -\frac{192\rho^2\zeta(2)\zeta(3)}{N^5} + \frac{480\rho^2\zeta(5)}{N^5} + \frac{240\rho^2\zeta(3)^2}{N^6} + \frac{480\rho^2\zeta(6)}{N^6} \tag{A.103}$$

The values divided by $3!\rho^2$ are given below.

$$\frac{T_1}{4!\rho^2} = -\frac{15\zeta(6)}{N^6} \tag{A.104}$$

$$\frac{T_{12}}{4!\rho^2} = \frac{63}{2}\frac{\zeta(6)}{N^6} \tag{A.105}$$

$$\frac{T_{13}}{4!\rho^2} = -\frac{9}{2}\frac{\zeta(6)}{N^6} \tag{A.106}$$

$$\frac{T_{14}}{4!\rho^2} = -\frac{12\zeta(6)}{N^6} \tag{A.107}$$

$$\frac{T_{15}}{4!\rho^2} = -\frac{12\zeta(2)\zeta(3)}{N^5} + \frac{24\zeta(5)}{N^5} - \frac{21\zeta(6)}{N^6} \tag{A.108}$$

$$\frac{T_{16}}{4!\rho^2} = -\frac{30\zeta(5)}{N^5} + \frac{12\zeta(2)\zeta(3)}{N^5} \tag{A.109}$$

$$\frac{T_{18}}{4!\rho^2} = \frac{6\zeta(5)}{N^5} + \frac{21\zeta(6)}{N^6} \tag{A.110}$$

$$\frac{T_{19}}{4!\rho^2} = \frac{90\zeta(5)}{N^5} - \frac{36\zeta(2)\zeta(3)}{N^5} - \frac{81\zeta(6)}{N^6} \tag{A.111}$$

$$\frac{T_{20}}{4!\rho^2} = \frac{9\zeta(6)}{N^6} - \frac{12\zeta(3)^2}{N^6} \tag{A.112}$$

$$\frac{T_{21}}{4!\rho^2} = \frac{9\zeta(6)}{N^6} \tag{A.113}$$

$$\frac{T_{22}}{4!\rho^2} = -\frac{12\zeta(5)}{N^5} - \frac{60\zeta(6)}{N^6} + \frac{24\zeta(3)^2}{N^6} \tag{A.114}$$

$$\frac{T_{23}}{4!\rho^2} = \frac{24\zeta(2)\zeta(3)}{N^5} - \frac{60\zeta(5)}{N^5} \tag{A.115}$$

$$\frac{T_{24}}{4!\rho^2} = \frac{24\zeta(2)\zeta(3)}{N^5} - \frac{72\zeta(5)}{N^5} - \frac{34\zeta(6)}{N^6} - \frac{12\zeta(3)^2}{N^6} \tag{A.116}$$

$$\frac{T_{25}}{4!\rho^2} = -\frac{12\zeta(2)\zeta(3)}{N^5} + \frac{36\zeta(5)}{N^5} + \frac{39\zeta(6)}{N^6} \tag{A.117}$$

$$\frac{T_{26}}{4!\rho^2} = -\frac{8\zeta(2)\zeta(3)}{N^5} + \frac{26\zeta(5)}{N^5} - \frac{2\zeta(3)^2}{N^6} + \frac{50\zeta(6)}{N^6} \tag{A.118}$$

$$\frac{T_{27}}{4!\rho^2} = -\frac{8\zeta(2)\zeta(3)}{N^5} + \frac{20\zeta(5)}{N^5} + \frac{10\zeta(3)^2}{N^6} + \frac{20\zeta(6)}{N^6} \tag{A.119}$$

A.4.2 Complete λ_4

$$\lambda_4 = -\frac{384\zeta(2)\zeta(3)\rho^2}{N^5} + \frac{672\zeta(5)\rho^2}{N^5} + \frac{192\zeta(3)^2\rho^2}{N^6} - \frac{1152\zeta(6)\rho^2}{N^6} \tag{A.120}$$

$$\frac{\lambda_4}{4!\rho^2} = -\frac{16\zeta(2)\zeta(3)}{N^5} + \frac{28\zeta(5)}{N^5} + \frac{8\zeta(3)^2}{N^6} - \frac{48\zeta(6)}{N^6} \tag{A.121}$$

Table A.26: Order ρ^2/N^6 contribution from convolution 12 of λ_4

$T_{12}/4! = \frac{v\rho^2}{N^6}, v = (756\,\zeta(6))/4!$					
$v = 32.046306452510147901007314788413996628907250936034887198035872 89$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	33.235442266898	$1.18914e+00$	$3.71068e-02$	n/a	0.003 sec.
4	32.282309183589	$2.36003e-01$	$7.36443e-03$	n/a	0.002 sec.
8	32.083083752160	$3.67773e-02$	$1.14763e-03$	-2.258	0.004 sec.
16	32.051406518784	$5.10007e-03$	$1.59147e-04$	-2.653	0.009 sec.
32	32.046976408310	$6.69956e-04$	$2.09059e-05$	-2.838	0.015 sec.
64	32.046392245146	$8.57926e-05$	$2.67715e-06$	-2.923	0.031 sec.
128	32.046317305016	$1.08525e-05$	$3.38651e-07$	-2.963	0.066 sec.
256	32.046307817109	$1.36460e-06$	$4.25821e-08$	-2.982	0.145 sec.
512	32.046306623588	$1.71077e-07$	$5.33844e-09$	-2.991	0.307 sec.
1024	32.046306473926	$2.14161e-08$	$6.68286e-10$	-2.995	0.718 sec.
2048	32.046306455189	$2.67897e-09$	$8.35970e-11$	-2.998	2.100 sec.
4096	32.046306452845	$3.34994e-10$	$1.04534e-11$	-2.999	7.915 sec.
8192	32.046306452552	$4.18820e-11$	$1.30692e-12$	-2.999	23.870 sec.
16384	32.046306452515	$5.23573e-12$	$1.63380e-13$	-3.000	106.785 sec.
32768	32.046306452511	$6.54496e-13$	$2.04234e-14$	-3.000	469.569 sec.
Richardson	32.046306452510	$1.09613e-38$	$3.42046e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.27: Order ρ^2/N^6 contribution from convolution 13 of λ_4

$T_{13}/4! = \frac{v\rho^2}{N^6}, v = (-108\,\zeta(6))/4!$					
$v = -4.5780437789300211287153306840591423755581787051478410282908389 85$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-4.747920323843	$1.69877e-01$	$3.71068e-02$	n/a	0.003 sec.
4	-4.611758454798	$3.37147e-02$	$7.36443e-03$	n/a	0.002 sec.
8	-4.583297678880	$5.25390e-03$	$1.14763e-03$	-2.258	0.002 sec.
16	-4.578772359826	$7.28581e-04$	$1.59147e-04$	-2.653	0.006 sec.
32	-4.578139486901	$9.57080e-05$	$2.09059e-05$	-2.838	0.010 sec.
64	-4.578056035021	$1.22561e-05$	$2.67715e-06$	-2.923	0.020 sec.
128	-4.578045329288	$1.55036e-06$	$3.38651e-07$	-2.963	0.042 sec.
256	-4.578043973873	$1.94943e-07$	$4.25821e-08$	-2.982	0.093 sec.
512	-4.578043803370	$2.44396e-08$	$5.33844e-09$	-2.991	0.194 sec.
1024	-4.578043781989	$3.05944e-09$	$6.68286e-10$	-2.995	0.378 sec.
2048	-4.578043779313	$3.82711e-10$	$8.35970e-11$	-2.998	0.798 sec.
4096	-4.578043778978	$4.78564e-11$	$1.04534e-11$	-2.999	1.809 sec.
8192	-4.578043778936	$5.98314e-12$	$1.30692e-12$	-2.999	4.580 sec.
16384	-4.578043778931	$7.47961e-13$	$1.63380e-13$	-3.000	6.490 sec.
32768	-4.578043778930	$9.34994e-14$	$2.04234e-14$	-3.000	13.078 sec.
Richardson	-4.578043778930	$1.56590e-39$	$3.42046e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.28: Order ρ^2/N^6 contribution from convolution 14 of λ_4

$T_{14}/4! = \frac{v\rho^2}{N^6}, v = (-288\,\zeta(6))/4!$					
$v = -12.20811674381338967657421515749104633482180988039424274210890396$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-11.974742435992	$2.33374e-01$	$1.91163e-02$	n/a	0.005 sec.
4	-12.162755125668	$4.53616e-02$	$3.71569e-03$	n/a	0.005 sec.
8	-12.201093819002	$7.02292e-03$	$5.75267e-04$	-2.294	0.008 sec.
16	-12.207144651465	$9.72092e-04$	$7.96267e-05$	-2.664	0.012 sec.
32	-12.207989111147	$1.27633e-04$	$1.04547e-05$	-2.841	0.025 sec.
64	-12.208100401642	$1.63422e-05$	$1.33863e-06$	-2.924	0.058 sec.
128	-12.208114676647	$2.06717e-06$	$1.69327e-07$	-2.963	0.103 sec.
256	-12.208116483889	$2.59924e-07$	$2.12911e-08$	-2.982	0.189 sec.
512	-12.208116711227	$3.25862e-08$	$2.66922e-09$	-2.991	0.413 sec.
1024	-12.208116739734	$4.07925e-09$	$3.34143e-10$	-2.995	0.989 sec.
2048	-12.208116743303	$5.10281e-10$	$4.17985e-11$	-2.998	2.265 sec.
4096	-12.208116743750	$6.38085e-11$	$5.22672e-12$	-2.999	6.886 sec.
8192	-12.208116743805	$7.97752e-12$	$6.53460e-13$	-2.999	33.431 sec.
16384	-12.208116743812	$9.97281e-13$	$8.16900e-14$	-3.000	109.439 sec.
32768	-12.208116743813	$1.24666e-13$	$1.02117e-14$	-3.000	444.328 sec.
Richardson	-12.208116743813	$1.77366e-39$	$1.45286e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.29: Order ρ^2/N^5 contribution from convolution 15 of λ_4

$T_{15}/4! = \frac{v\rho^2}{N^5}, v = (-288\,\zeta(2)\,\zeta(3) + 576\,\zeta(5))/4!$					
$v = 1.15861391987332481358774637714731316838412446812429698303075695$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	2.543390097549	$1.38478e+00$	$1.19520e+00$	n/a	0.002 sec.
4	1.626086084626	$4.67472e-01$	$4.03475e-01$	n/a	0.001 sec.
8	1.293612150438	$1.34998e-01$	$1.16517e-01$	-1.464	0.003 sec.
16	1.194751989049	$3.61381e-02$	$3.11908e-02$	-1.750	0.004 sec.
32	1.167950339937	$9.33642e-03$	$8.05827e-03$	-1.883	0.007 sec.
64	1.160985789223	$2.37187e-03$	$2.04716e-03$	-1.944	0.019 sec.
128	1.159211602904	$5.97683e-04$	$5.15860e-04$	-1.973	0.032 sec.
256	1.158763929574	$1.50010e-04$	$1.29473e-04$	-1.987	0.066 sec.
512	1.158651495877	$3.75760e-05$	$3.24319e-05$	-1.993	0.155 sec.
1024	1.158623323069	$9.40320e-06$	$8.11590e-06$	-1.997	0.278 sec.
2048	1.158616271821	$2.35195e-06$	$2.02997e-06$	-1.998	0.596 sec.
4096	1.158614508004	$5.88131e-07$	$5.07616e-07$	-1.999	1.218 sec.
8192	1.158614066924	$1.47051e-07$	$1.26919e-07$	-2.000	2.495 sec.
16384	1.158613956638	$3.67649e-08$	$3.17318e-08$	-2.000	4.846 sec.
32768	1.158613929065	$9.19150e-09$	$7.93319e-09$	-2.000	9.693 sec.
Richardson	1.158613919873	$2.10269e-23$	$1.81483e-23$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.30: Order ρ^2/N^6 contribution from convolution 15 of λ_4

$T_{15}/4! = \frac{v\rho^2}{N^6}, v = (-504\,\zeta(6))/4!$					
$v = -21.364204301673431934004876525609331085938167290689924798690 58193$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.002 sec.
4	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.001 sec.
8	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.003 sec.
16	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.004 sec.
32	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.007 sec.
64	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.019 sec.
128	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.032 sec.
256	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.066 sec.
512	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.155 sec.
1024	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.278 sec.
2048	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.596 sec.
4096	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	1.218 sec.
8192	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	2.495 sec.
16384	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	4.846 sec.
32768	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	9.693 sec.
Richardson	−21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	n/a
No Error Detected					

Table A.31: Order ρ^2/N^5 contribution from convolution 16 of λ_4

$T_{16}/4! = \frac{v\rho^2}{N^5}, v = (-720\,\zeta(5) + 288\,\zeta(2)\,\zeta(3))/4!$					
$v = -7.3801804507335443715759392958895181767266099851357738548759 1302$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−8.730890097549	$1.35071e+00$	$1.83019e-01$	n/a	0.003 sec.
4	−7.844136817650	$4.63956e-01$	$6.28652e-02$	n/a	0.002 sec.
8	−7.514894587979	$1.34714e-01$	$1.82535e-02$	−1.429	0.003 sec.
16	−7.416298344141	$3.61179e-02$	$4.89390e-03$	−1.740	0.006 sec.
32	−7.389515527366	$9.33508e-03$	$1.26488e-03$	−1.880	0.011 sec.
64	−7.382552233434	$2.37178e-03$	$3.21372e-04$	−1.943	0.026 sec.
128	−7.380778128263	$5.97678e-04$	$8.09841e-05$	−1.973	0.045 sec.
256	−7.380330460087	$1.50009e-04$	$2.03260e-05$	−1.987	0.097 sec.
512	−7.380218026715	$3.75760e-05$	$5.09147e-06$	−1.993	0.219 sec.
1024	−7.380189853928	$9.40319e-06$	$1.27411e-06$	−1.997	0.445 sec.
2048	−7.380182802681	$2.35195e-06$	$3.18684e-07$	−1.998	0.952 sec.
4096	−7.380181038864	$5.88131e-07$	$7.96905e-08$	−1.999	1.773 sec.
8192	−7.380180597784	$1.47051e-07$	$1.99251e-08$	−2.000	3.517 sec.
16384	−7.380180487498	$3.67649e-08$	$4.98157e-09$	−2.000	7.094 sec.
32768	−7.380180459925	$9.19150e-09$	$1.24543e-09$	−2.000	14.271 sec.
Richardson	−7.380180450734	$1.94044e-33$	$2.62926e-34$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.32: Order ρ^2/N^5 contribution from convolution 18 of λ_4

$T_{18}/4! = \frac{v\rho^2}{N^5}, v = (144\,\zeta(5))/4!$					
$v = 6.2215665308602195579881929187422050083424855170114768718451560 68$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	6.187500000000	$3.40665e-02$	$5.47556e-03$	n/a	0.002 sec.
4	6.218050733025	$3.51580e-03$	$5.65098e-04$	n/a	0.001 sec.
8	6.221282437541	$2.84093e-04$	$4.56627e-05$	-3.241	0.003 sec.
16	6.221546355093	$2.01758e-05$	$3.24288e-06$	-3.614	0.005 sec.
32	6.221565187429	$1.34343e-06$	$2.15931e-07$	-3.809	0.010 sec.
64	6.221566444211	$8.66494e-08$	$1.39273e-08$	-3.905	0.020 sec.
128	6.221566525359	$5.50119e-09$	$8.84213e-10$	-3.953	0.040 sec.
256	6.221566530514	$3.46526e-10$	$5.56976e-11$	-3.977	0.085 sec.
512	6.221566530838	$2.17427e-11$	$3.49474e-12$	-3.988	0.223 sec.
1024	6.221566530859	$1.36158e-12$	$2.18848e-13$	-3.994	0.339 sec.
2048	6.221566530860	$8.51819e-14$	$1.36914e-14$	-3.997	0.700 sec.
4096	6.221566530860	$5.32647e-15$	$8.56130e-16$	-3.999	1.537 sec.
8192	6.221566530860	$3.32986e-16$	$5.35212e-17$	-3.999	3.163 sec.
16384	6.221566530860	$2.08141e-17$	$3.34548e-18$	-4.000	6.098 sec.
32768	6.221566530860	$1.30096e-18$	$2.09105e-19$	-4.000	13.442 sec.
Richardson	6.221566530860	$8.38963e-51$	$1.34848e-51$	n/a	n/a
Error: $O\left(\text{range}^{-4.000}\right)$					

Table A.33: Order ρ^2/N^6 contribution from convolution 18 of λ_4

$T_{18}/4! = \frac{v\rho^2}{N^6}, v = (504\,\zeta(6))/4!$					
$v = 21.364204301673431934004876525609331085938167290689924798690581 93$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.002 sec.
4	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.001 sec.
8	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.003 sec.
16	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.005 sec.
32	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.010 sec.
64	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.020 sec.
128	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.040 sec.
256	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.085 sec.
512	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.223 sec.
1024	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.339 sec.
2048	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	0.700 sec.
4096	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	1.537 sec.
8192	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	3.163 sec.
16384	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	6.098 sec.
32768	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	13.442 sec.
Richardson	21.364204301673	$1.00000e-62$	$4.68073e-64$	n/a	n/a
No Error Detected					

Table A.34: Order ρ^2/N^5 contribution from convolution 19 of λ_4

$T_{19}/4! = \frac{v\rho^2}{N^5}, v = (2160\,\zeta(5) - 864\,\zeta(2)\,\zeta(3))\,/4!$					
$v = 22.14054135220063311472781788766855453017982995540732156462773906$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	26.192670292647	$4.05213e+00$	$1.83019e-01$	n/a	0.003 sec.
4	23.532410452951	$1.39187e+00$	$6.28652e-02$	n/a	0.002 sec.
8	22.544683763938	$4.04142e-01$	$1.82535e-02$	-1.429	0.003 sec.
16	22.248895032424	$1.08354e-01$	$4.89390e-03$	-1.740	0.006 sec.
32	22.168546582097	$2.80052e-02$	$1.26488e-03$	-1.880	0.013 sec.
64	22.147656700302	$7.11535e-03$	$3.21372e-04$	-1.943	0.031 sec.
128	22.142334384790	$1.79303e-03$	$8.09841e-05$	-1.973	0.058 sec.
256	22.140991380262	$4.50028e-04$	$2.03260e-05$	-1.987	0.129 sec.
512	22.140654080146	$1.12728e-04$	$5.09147e-06$	-1.993	0.292 sec.
1024	22.140569561783	$2.82096e-05$	$1.27411e-06$	-1.997	0.681 sec.
2048	22.140548408044	$7.05584e-06$	$3.18684e-07$	-1.998	1.791 sec.
4096	22.140543116593	$1.76439e-06$	$7.96905e-08$	-1.999	6.303 sec.
8192	22.140541793352	$4.41152e-07$	$1.99251e-08$	-2.000	29.894 sec.
16384	22.140541462495	$1.10295e-07$	$4.98157e-09$	-2.000	86.969 sec.
32768	22.140541379775	$2.75745e-08$	$1.24543e-09$	-2.000	421.251 sec.
Richardson	22.140541352201	$5.82133e-33$	$2.62926e-34$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.35: Order ρ^2/N^6 contribution from convolution 19 of λ_4

$T_{19}/4! = \frac{v\rho^2}{N^6}, v = (-1944\,\zeta(6))/4!$					
$v = -82.404788020740380316875952313064562760047216692661138509235101 73$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-85.462565829166	$3.05778e+00$	$3.71068e-02$	n/a	0.003 sec.
4	-83.011652186370	$6.06864e-01$	$7.36443e-03$	n/a	0.002 sec.
8	-82.499358219841	$9.45702e-02$	$1.14763e-03$	-2.258	0.003 sec.
16	-82.417902476873	$1.31145e-02$	$1.59147e-04$	-2.653	0.006 sec.
32	-82.406510764226	$1.72274e-03$	$2.09059e-05$	-2.838	0.013 sec.
64	-82.405008630376	$2.20610e-04$	$2.67715e-06$	-2.923	0.031 sec.
128	-82.404815927183	$2.79064e-05$	$3.38651e-07$	-2.963	0.058 sec.
256	-82.404791529710	$3.50897e-06$	$4.25821e-08$	-2.982	0.129 sec.
512	-82.404788460654	$4.39913e-07$	$5.33844e-09$	-2.991	0.292 sec.
1024	-82.404788075810	$5.50699e-08$	$6.68286e-10$	-2.995	0.681 sec.
2048	-82.404788027629	$6.88879e-09$	$8.35970e-11$	-2.998	1.791 sec.
4096	-82.404788021602	$8.61414e-10$	$1.04534e-11$	-2.999	6.303 sec.
8192	-82.404788020848	$1.07697e-10$	$1.30692e-12$	-2.999	29.894 sec.
16384	-82.404788020754	$1.34633e-11$	$1.63380e-13$	-3.000	86.969 sec.
32768	-82.404788020742	$1.68299e-12$	$2.04234e-14$	-3.000	421.251 sec.
Richardson	-82.404788020740	$2.81862e-38$	$3.42046e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.36: Order ρ^2/N^6 contribution from convolution 20 of λ_4

$T_{20}/4! = \frac{v\rho^2}{N^6}, v = \left(216\,\zeta(6) - 288\,(\zeta(3))^2\right)/4!$					
$v = -8.1832020233435685495335595775655691750884674588709820329955319 59$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-8.556366698322	$3.73165e-01$	$4.56013e-02$	n/a	0.021 sec.
4	-8.256525473173	$7.33234e-02$	$8.96024e-03$	n/a	0.005 sec.
8	-8.194459067404	$1.12570e-02$	$1.37563e-03$	-2.272	0.012 sec.
16	-8.184735586904	$1.53356e-03$	$1.87404e-04$	-2.674	0.022 sec.
32	-8.183400221365	$1.98198e-04$	$2.42201e-05$	-2.864	0.058 sec.
64	-8.183227087535	$2.50642e-05$	$3.06288e-06$	-2.947	0.104 sec.
128	-8.183205166553	$3.14321e-06$	$3.84105e-07$	-2.982	0.402 sec.
256	-8.183202416381	$3.93037e-07$	$4.80297e-08$	-2.995	1.154 sec.
512	-8.183202072451	$4.91070e-08$	$6.00095e-09$	-2.999	2.465 sec.
1024	-8.183202029479	$6.13502e-09$	$7.49709e-10$	-3.001	5.160 sec.
2048	-8.183202024110	$7.66549e-10$	$9.36735e-11$	-3.001	9.132 sec.
4096	-8.183202023439	$9.57906e-11$	$1.17058e-11$	-3.001	17.731 sec.
8192	-8.183202023356	$1.19716e-11$	$1.46295e-12$	-3.000	34.980 sec.
16384	-8.183202023345	$1.49628e-12$	$1.82848e-13$	-3.000	72.249 sec.
32768	-8.183202023344	$1.87023e-13$	$2.28545e-14$	-3.000	141.755 sec.
Richardson	-8.183202023344	$6.42631e-15$	$7.85304e-16$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.37: Order ρ^2/N^6 contribution from convolution 21 of λ_4

$T_{21}/4! = \frac{v\rho^2}{N^6}, v = (216\,\zeta(6))/4!$					
$v = 9.1560875578600422574306613681182847511163574102956820565816779 71$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	9.495840647685	$3.39753e-01$	$3.71068e-02$	n/a	0.005 sec.
4	9.223516909597	$6.74294e-02$	$7.36443e-03$	n/a	0.002 sec.
8	9.166595357760	$1.05078e-02$	$1.14763e-03$	-2.258	0.004 sec.
16	9.157544719653	$1.45716e-03$	$1.59147e-04$	-2.653	0.008 sec.
32	9.156278973803	$1.91416e-04$	$2.09059e-05$	-2.838	0.016 sec.
64	9.156112070042	$2.45122e-05$	$2.67715e-06$	-2.923	0.033 sec.
128	9.156090658576	$3.10072e-06$	$3.38651e-07$	-2.963	0.070 sec.
256	9.156087947746	$3.89885e-07$	$4.25821e-08$	-2.982	0.174 sec.
512	9.156087606739	$4.88793e-08$	$5.33844e-09$	-2.991	0.297 sec.
1024	9.156087563979	$6.11888e-09$	$6.68286e-10$	-2.995	0.626 sec.
2048	9.156087558625	$7.65421e-10$	$8.35970e-11$	-2.998	1.261 sec.
4096	9.156087557956	$9.57127e-11$	$1.04534e-11$	-2.999	2.593 sec.
8192	9.156087557872	$1.19663e-11$	$1.30692e-12$	-2.999	5.050 sec.
16384	9.156087557862	$1.49592e-12$	$1.63380e-13$	-3.000	10.459 sec.
32768	9.156087557860	$1.86999e-13$	$2.04234e-14$	-3.000	20.602 sec.
Richardson	9.156087557860	$3.13180e-39$	$3.42046e-40$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.38: Order ρ^2/N^5 contribution from convolution 22 of λ_4

$T_{22}/4! = \frac{v\rho^2}{N^5}, v = (-288\,\zeta(5))/4!$					
$v = -12.443133061720439115976385837484410016684971034022953743690312 14$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−15.400343508389	$2.95721e+00$	$2.37658e-01$	n/a	0.019 sec.
4	−13.448157960176	$1.00502e+00$	$8.07694e-02$	n/a	0.006 sec.
8	−12.730546019612	$2.87413e-01$	$2.30981e-02$	−1.444	0.013 sec.
16	−12.518871875974	$7.57388e-02$	$6.08680e-03$	−1.761	0.025 sec.
32	−12.462409870009	$1.92768e-02$	$1.54919e-03$	−1.906	0.052 sec.
64	−12.447974018174	$4.84096e-03$	$3.89046e-04$	−1.968	0.114 sec.
128	−12.444343289588	$1.21023e-03$	$9.72607e-05$	−1.991	0.525 sec.
256	−12.443435275371	$3.02214e-04$	$2.42876e-05$	−1.999	1.292 sec.
512	−12.443208529820	$7.54681e-05$	$6.06504e-06$	−2.002	2.577 sec.
1024	−12.443151912826	$1.88511e-05$	$1.51498e-06$	−2.002	4.985 sec.
2048	−12.443137771854	$4.71013e-06$	$3.78533e-07$	−2.001	10.349 sec.
4096	−12.443134238842	$1.17712e-06$	$9.46001e-08$	−2.001	22.157 sec.
8192	−12.443133355939	$2.94219e-07$	$2.36451e-08$	−2.001	58.426 sec.
16384	−12.443133135266	$7.35458e-08$	$5.91055e-09$	−2.000	162.609 sec.
Richardson	−12.443133059765	$1.95579e-09$	$1.57178e-10$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.39: Order ρ^2/N^6 contribution from convolution 22 of λ_4

$T_{22}/4! = \frac{v\rho^2}{N^6}, v = \left(-1440\,\zeta(6) + 576\,(\zeta(3))^2\right)/4!$					
$v = -26.362004556659726768942633896087523821699399663637885531390099 94$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−24.833085307987	$1.52892e+00$	$5.79971e-02$	n/a	0.019 sec.
4	−26.074368198709	$2.87636e-01$	$1.09110e-02$	n/a	0.006 sec.
8	−26.318208995571	$4.37956e-02$	$1.66131e-03$	−2.348	0.013 sec.
16	−26.356013338647	$5.99122e-03$	$2.27267e-04$	−2.689	0.025 sec.
32	−26.361224998170	$7.79558e-04$	$2.95713e-05$	−2.859	0.052 sec.
64	−26.361905393167	$9.91635e-05$	$3.76161e-06$	−2.937	0.114 sec.
128	−26.361992068466	$1.24882e-05$	$4.73719e-07$	−2.973	0.525 sec.
256	−26.362002990804	$1.56586e-06$	$5.93982e-08$	−2.988	1.292 sec.
512	−26.362004360687	$1.95973e-07$	$7.43391e-09$	−2.995	2.577 sec.
1024	−26.362004532152	$2.45078e-08$	$9.29664e-10$	−2.998	4.985 sec.
2048	−26.362004553596	$3.06394e-09$	$1.16226e-10$	−2.999	10.349 sec.
4096	−26.362004556277	$3.83007e-10$	$1.45287e-11$	−3.000	22.157 sec.
8192	−26.362004556612	$4.78758e-11$	$1.81609e-12$	−3.000	58.426 sec.
16384	−26.362004556654	$5.98441e-12$	$2.27009e-13$	−3.000	162.609 sec.
Richardson	−26.362004556660	$7.52251e-16$	$2.85354e-17$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.40: Order ρ^2/N^5 contribution from convolution 23 of λ_4

$T_{23}/4! = \frac{v\rho^2}{N^5}, v = (576\,\zeta(2)\,\zeta(3) - 1440\,\zeta(5))\,/4!$					
$v = -14.76036090146708874315187859177903635345321997027154770975182604$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-17.461780195098	$2.70142e+00$	$1.83019e-01$	n/a	0.003 sec.
4	-15.688273635301	$9.27913e-01$	$6.28652e-02$	n/a	0.002 sec.
8	-15.029789175958	$2.69428e-01$	$1.82535e-02$	-1.429	0.004 sec.
16	-14.832596688283	$7.22358e-02$	$4.89390e-03$	-1.740	0.008 sec.
32	-14.779031054731	$1.86702e-02$	$1.26488e-03$	-1.880	0.015 sec.
64	-14.765104466868	$4.74357e-03$	$3.21372e-04$	-1.943	0.031 sec.
128	-14.761556256527	$1.19536e-03$	$8.09841e-05$	-1.973	0.065 sec.
256	-14.760660920175	$3.00019e-04$	$2.03260e-05$	-1.987	0.176 sec.
512	-14.760436053431	$7.51520e-05$	$5.09147e-06$	-1.993	0.281 sec.
1024	-14.760379707855	$1.88064e-05$	$1.27411e-06$	-1.997	0.596 sec.
2048	-14.760365605363	$4.70390e-06$	$3.18684e-07$	-1.998	1.207 sec.
4096	-14.760362077728	$1.17626e-06$	$7.96905e-08$	-1.999	2.489 sec.
8192	-14.760361195568	$2.94101e-07$	$1.99251e-08$	-2.000	4.909 sec.
16384	-14.760360974997	$7.35298e-08$	$4.98157e-09$	-2.000	10.693 sec.
32768	-14.760360919850	$1.83830e-08$	$1.24543e-09$	-2.000	20.367 sec.
Richardson	-14.760360901467	$3.88089e-33$	$2.62926e-34$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.41: Order ρ^2/N^5 contribution from convolution 24 of λ_4

$T_{24}/4! = \frac{v\rho^2}{N^5}, v = (576\,\zeta(2)\,\zeta(3) - 1728\,\zeta(5))\,/4!$					
$v = -27.20349396318752785912826442926344637013819100429450145344213818$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−27.538219804902	3.34726e − 01	1.23045e − 02	n/a	0.007 sec.
4	−27.318285006675	1.14791e − 01	4.21972e − 03	n/a	0.006 sec.
8	−27.229480814853	2.59869e − 02	9.55276e − 04	−1.308	0.015 sec.
16	−27.208278871196	4.78491e − 03	1.75893e − 04	−2.066	0.023 sec.
32	−27.204281373371	7.87410e − 04	2.89452e − 05	−2.407	0.039 sec.
64	−27.203615331730	1.21369e − 04	4.46151e − 06	−2.585	0.087 sec.
128	−27.203511922919	1.79597e − 05	6.60199e − 07	−2.687	0.306 sec.
256	−27.203496549710	2.58652e − 06	9.50805e − 08	−2.750	0.795 sec.
512	−27.203494328631	3.65444e − 07	1.34337e − 08	−2.791	1.489 sec.
1024	−27.203494014091	5.09033e − 08	1.87120e − 09	−2.820	3.141 sec.
2048	−27.203493970200	7.01259e − 09	2.57783e − 10	−2.841	7.542 sec.
4096	−27.203493964145	9.57555e − 10	3.51997e − 11	−2.858	18.243 sec.
8192	−27.203493963317	1.29801e − 10	4.77148e − 12	−2.871	53.565 sec.
16384	−27.203493963205	1.74873e − 11	6.42832e − 13	−2.882	176.472 sec.
32768	−27.203493963190	2.34361e − 12	8.61509e − 14	−2.891	690.478 sec.
Richardson	−27.203493963188	1.25361e − 20	4.60827e − 22	n/a	n/a
Error: $O\left(\text{range}^{-2.891}\right)$					

Table A.42: Order ρ^2/N^6 contribution from convolution 24 of λ_4

$T_{24}/4! = \frac{v\rho^2}{N^6}, v = \left(-816\,\zeta(6) - 288\,(\zeta(3))^2\right)/4!$					
$v = -51.92895368867488155725783055857515187486661953028368519221910449$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−51.000000000000	9.28954e − 01	1.78889e − 02	n/a	0.007 sec.
4	−51.766653806584	1.62300e − 01	3.12542e − 03	n/a	0.006 sec.
8	−51.905327083914	2.36266e − 02	4.54979e − 04	−2.467	0.015 sec.
16	−51.925813333853	3.14035e − 03	6.04741e − 05	−2.759	0.023 sec.
32	−51.928552614215	4.01074e − 04	7.72352e − 06	−2.903	0.039 sec.
64	−51.928903264980	5.04237e − 05	9.71013e − 07	−2.966	0.087 sec.
128	−51.928947383712	6.30496e − 06	1.21415e − 07	−2.991	0.306 sec.
256	−51.928952901439	7.87235e − 07	1.51599e − 08	−2.999	0.795 sec.
512	−51.928953590388	9.82866e − 08	1.89271e − 09	−3.002	1.489 sec.
1024	−51.928953676400	1.22746e − 08	2.36373e − 10	−3.002	3.141 sec.
2048	−51.928953687141	1.53338e − 09	2.95285e − 11	−3.001	7.542 sec.
4096	−51.928953688483	1.91599e − 10	3.68964e − 12	−3.001	18.243 sec.
8192	−51.928953688651	2.39443e − 11	4.61098e − 13	−3.001	53.565 sec.
16384	−51.928953688672	2.99264e − 12	5.76295e − 14	−3.000	176.472 sec.
32768	−51.928953688675	3.74051e − 13	7.20313e − 15	−3.000	690.478 sec.
Richardson	−51.928953688675	3.81171e − 14	7.34024e − 16	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.43: Order ρ^2/N^5 contribution from convolution 25 of λ_4

$T_{25}/4! = \frac{v\rho^2}{N^5}, v = (-288\,\zeta(2)\,\zeta(3) + 864\,\zeta(5))/4!$					
$v = 13.60174698159376392956413221463172318506909550214725072672106909$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	13.769109902451	$1.67363e-01$	$1.23045e-02$	n/a	0.008 sec.
4	13.659142503337	$5.73955e-02$	$4.21972e-03$	n/a	0.004 sec.
8	13.614740407427	$1.29934e-02$	$9.55276e-04$	-1.308	0.009 sec.
16	13.604139435598	$2.39245e-03$	$1.75893e-04$	-2.066	0.016 sec.
32	13.602140686686	$3.93705e-04$	$2.89452e-05$	-2.407	0.034 sec.
64	13.601807665865	$6.06843e-05$	$4.46151e-06$	-2.585	0.078 sec.
128	13.601755961460	$8.97987e-06$	$6.60199e-07$	-2.687	0.275 sec.
256	13.601748274855	$1.29326e-06$	$9.50805e-08$	-2.750	0.927 sec.
512	13.601747164316	$1.82722e-07$	$1.34337e-08$	-2.791	1.770 sec.
1024	13.601747007045	$2.54517e-08$	$1.87120e-09$	-2.820	4.319 sec.
2048	13.601746985100	$3.50630e-09$	$2.57783e-10$	-2.841	12.974 sec.
4096	13.601746982073	$4.78777e-10$	$3.51997e-11$	-2.858	44.837 sec.
8192	13.601746981659	$6.49004e-11$	$4.77148e-12$	-2.871	165.731 sec.
16384	13.601746981603	$8.74364e-12$	$6.42832e-13$	-2.882	692.501 sec.
Richardson	13.601746981594	$1.65815e-19$	$1.21907e-20$	n/a	n/a
Error: $O\left(\text{range}^{-2.882}\right)$					

Table A.44: Order ρ^2/N^6 contribution from convolution 25 of λ_4

$T_{25}/4! = \frac{v\rho^2}{N^6}, v = (936\,\zeta\,(6))\,/4!$					
$v = 39.676379417393516448866199261845900588170882111281288911853937 87$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	39.254159352315	$4.22220e-01$	$1.06416e-02$	n/a	0.008 sec.
4	39.604045461802	$7.23340e-02$	$1.82310e-03$	n/a	0.004 sec.
8	39.665658918960	$1.07205e-02$	$2.70199e-04$	-2.506	0.009 sec.
16	39.674914441755	$1.46498e-03$	$3.69231e-05$	-2.735	0.016 sec.
32	39.676187737002	$1.91680e-04$	$4.83110e-06$	-2.862	0.034 sec.
64	39.676354896615	$2.45208e-05$	$6.18020e-07$	-2.929	0.078 sec.
128	39.676376316404	$3.10099e-06$	$7.81571e-08$	-2.964	0.275 sec.
256	39.676379027499	$3.89894e-07$	$9.82686e-09$	-2.982	0.927 sec.
512	39.676379368514	$4.88795e-08$	$1.23196e-09$	-2.991	1.770 sec.
1024	39.676379411275	$6.11889e-09$	$1.54220e-10$	-2.995	4.319 sec.
2048	39.676379416628	$7.65421e-10$	$1.92916e-11$	-2.998	12.974 sec.
4096	39.676379417298	$9.57127e-11$	$2.41233e-12$	-2.999	44.837 sec.
8192	39.676379417382	$1.19663e-11$	$3.01597e-13$	-2.999	165.731 sec.
16384	39.676379417392	$1.49592e-12$	$3.77031e-14$	-3.000	692.501 sec.
Richardson	39.676379417394	$1.30805e-36$	$3.29680e-38$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.45: Order ρ^2/N^5 contribution from convolution 26 of λ_4

$T_{26}/4! = \frac{v\rho^2}{N^5}, v = (-192\,\zeta(2)\,\zeta(3) + 624\,\zeta(5))/4!$					
$v = 11.141686831349249139038819116001883792826892173768659441762431 42$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	10.488581607521	$6.53105e-01$	$5.86182e-02$	n/a	0.015 sec.
4	10.642054307191	$4.99633e-01$	$4.48435e-02$	n/a	0.010 sec.
8	10.865135559246	$2.76551e-01$	$2.48213e-02$	0.540	0.010 sec.
16	11.017792576679	$1.23894e-01$	$1.11199e-02$	-0.547	0.026 sec.
32	11.093059451792	$4.86274e-02$	$4.36445e-03$	-1.020	0.042 sec.
64	11.124171070700	$1.75158e-02$	$1.57209e-03$	-1.275	0.105 sec.
128	11.135737098003	$5.94973e-03$	$5.34007e-04$	-1.428	0.283 sec.
256	11.139749280411	$1.93755e-03$	$1.73901e-04$	-1.527	0.675 sec.
512	11.141075532809	$6.11299e-04$	$5.48659e-05$	-1.597	1.419 sec.
1024	11.141498667980	$1.88163e-04$	$1.68882e-05$	-1.648	3.190 sec.
2048	11.141630049117	$5.67822e-05$	$5.09638e-06$	-1.687	6.681 sec.
4096	11.141669972971	$1.68584e-05$	$1.51309e-06$	-1.718	15.765 sec.
8192	11.141681894037	$4.93731e-06$	$4.43139e-07$	-1.744	41.602 sec.
16384	11.141685402064	$1.42929e-06$	$1.28283e-07$	-1.765	152.240 sec.
Richardson	11.141686831389	$3.95959e-11$	$3.55385e-12$	n/a	n/a
Error: $O\left(\text{range}^{-1.765}\right)$					

Table A.46: Order ρ^2/N^6 contribution from convolution 26 of λ_4

$T_{26}/4! = \frac{v\rho^2}{N^6}, v = \left(-48\,(\zeta(3))^2 + 1200\,\zeta(6)\right)/4!$					
$v = 47.977271502355188517898526331932050740723403690114900743857564 85$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	49.798540262307	$1.82127e+00$	$3.79611e-02$	n/a	0.015 sec.
4	48.956927420645	$9.79656e-01$	$2.04192e-02$	n/a	0.010 sec.
8	48.373943239983	$3.96672e-01$	$8.26791e-03$	-0.530	0.010 sec.
16	48.115524661209	$1.38253e-01$	$2.88164e-03$	-1.174	0.026 sec.
32	48.021476767021	$4.42053e-02$	$9.21379e-04$	-1.458	0.042 sec.
64	47.990679114298	$1.34076e-02$	$2.79458e-04$	-1.611	0.105 sec.
128	47.981200911079	$3.92941e-03$	$8.19015e-05$	-1.700	0.283 sec.
256	47.978396198409	$1.12470e-03$	$2.34423e-05$	-1.757	0.675 sec.
512	47.977587936703	$3.16434e-04$	$6.59551e-06$	-1.795	1.419 sec.
1024	47.977359376544	$8.78742e-05$	$1.83158e-06$	-1.822	3.190 sec.
2048	47.977295655035	$2.41527e-05$	$5.03419e-07$	-1.843	6.681 sec.
4096	47.977278085515	$6.58316e-06$	$1.37214e-07$	-1.859	15.765 sec.
8192	47.977273284246	$1.78189e-06$	$3.71403e-08$	-1.872	41.602 sec.
16384	47.977271981833	$4.79477e-07$	$9.99384e-09$	-1.882	152.240 sec.
Richardson	47.977271502355	$1.90022e-15$	$3.96068e-17$	n/a	n/a
Error: $O\left(\text{range}^{-1.882}\right)$					

Table A.47: Order ρ^2/N^5 contribution from convolution 27 of λ_4

$T_{27}/4! = \frac{v\rho^2}{N^5}, v = \left(-192\,\zeta\left(2\right)\zeta\left(3\right) + 480\,\zeta\left(5\right)\right)/4!$					
$v = 4.920120300489029581050626197259678784484406656757182569917275 35$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	6.483303611663	$1.56318e+00$	$3.17712e-01$	n/a	0.023 sec.
4	4.873805980320	$4.63143e-02$	$9.41325e-03$	n/a	0.022 sec.
8	4.629176920551	$2.90943e-01$	$5.91334e-02$	-2.718	0.052 sec.
16	4.734138733374	$1.85982e-01$	$3.78002e-02$	-1.221	0.126 sec.
32	4.835090289306	$8.50300e-02$	$1.72821e-02$	-0.056	0.360 sec.
64	4.886808129022	$3.33122e-02$	$6.77060e-03$	-0.965	1.043 sec.
128	4.908191537590	$1.19288e-02$	$2.42449e-03$	-1.274	3.172 sec.
256	4.916092939031	$4.02736e-03$	$8.18549e-04$	-1.436	14.975 sec.
512	4.918816000929	$1.30430e-03$	$2.65095e-04$	-1.537	155.602 sec.
1024	4.919710802577	$4.09498e-04$	$8.32292e-05$	-1.606	425.495 sec.
Richardson	4.920144146884	$2.38464e-05$	$4.84671e-06$	n/a	n/a
Error: $O\left(\text{range}^{-1.606}\right)$					

Table A.48: Order ρ^2/N^6 contribution from convolution 27 of λ_4

$T_{27}/4! = \frac{v\rho^2}{N^6}, v = \left(240\,(\zeta(3))^2 + 480\,\zeta(6)\right)/4!$					
$v = 34.796269224025325133427209383888288829873703858295957978162514 87$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	37.034171346438	$2.23790e+00$	$6.43144e-02$	n/a	0.023 sec.
4	35.628350173444	$8.32081e-01$	$2.39129e-02$	n/a	0.022 sec.
8	35.075973898852	$2.79705e-01$	$8.03835e-03$	-1.348	0.052 sec.
16	34.884915068834	$8.86458e-02$	$2.54757e-03$	-1.532	0.126 sec.
32	34.823148562911	$2.68793e-02$	$7.72478e-04$	-1.629	0.360 sec.
64	34.804155850289	$7.88663e-03$	$2.26651e-04$	-1.701	1.043 sec.
128	34.798528409366	$2.25919e-03$	$6.49261e-05$	-1.755	3.172 sec.
256	34.796905057479	$6.35833e-04$	$1.82730e-05$	-1.794	14.975 sec.
512	34.796445797196	$1.76573e-04$	$5.07449e-06$	-1.822	155.602 sec.
1024	34.796317748314	$4.85243e-05$	$1.39453e-06$	-1.843	425.495 sec.
Richardson	34.796269209172	$1.48535e-08$	$4.26870e-10$	n/a	n/a
Error: $O\left(\text{range}^{-1.843}\right)$					

A.5 λ_5 Detailed Results

Due to the number of convolutions that appear in λ_5, they are organized into groups.

A.5.1 Trivial Zeros

The following convolutions are trivially zero because they contain $c_1 = 0$.

$$T_1 = -96\rho^3\sqrt{\pi}\text{conv}\left(c_0, c_0, c_1, f_{1,dr}\right)(0) \tag{A.122}$$

$$T_2 = -80\rho\sqrt{\pi}\text{conv}\left(c_0, c_{0,d\theta^2}, c_1, f_{1,dr}\right)(0) \tag{A.123}$$

$$T_3 = -16\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_{1,d\theta^2}, f_{1,dr}\right)(0) \tag{A.124}$$

$$T_4 = +96\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_1, f_{1,dr^3}\right)(0) \tag{A.125}$$

$$T_5 = +72\rho\sqrt{\pi}\text{conv}\left(c_0, c_1, f_{2,dr^2}\right)(0) \tag{A.126}$$

$$T_6 = -16\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_{1,d\theta} f_{1,drd\theta}\right)(0) \tag{A.127}$$

$$T_7 = -80\rho\sqrt{\pi}\text{conv}\left(c_0, c_{0,d\theta}, c_1, f_{1,drd\theta}\right)(0) \tag{A.128}$$

$$T_8 = +16\rho\sqrt{\pi}\text{conv}\left(c_1, f_{3,dr}\right)(0) \tag{A.129}$$

$$T_9 = +48\rho\sqrt{\pi}\text{conv}\left(c_1, c_1, f_{1,dr^2}\right)(0) \tag{A.130}$$

$$T_{10} = -144\pi\text{conv}\left(c_0, c_1, f_{1,dr}, f_{1,dr^2}\right)(0) \tag{A.131}$$

$$T_{11} = -144\pi\text{conv}\left(c_0, c_1, f_{1,d\theta}, f_{1,drd\theta}\right)(0) \tag{A.132}$$

$$T_{12} = +72\pi\text{conv}\left(c_0, c_1, f_{1,d\theta}, f_{1,d\theta}\right)(0) \tag{A.133}$$

$$T_{13} = +14\rho^2\text{conv}\left(c_0, c_1, c_{1,d\theta^2}\right)(0) \tag{A.134}$$

$$T_{14} = -42\rho^2\text{conv}\left(c_0, c_{0,d\theta}, c_{0,d\theta} c_1\right)(0) \tag{A.135}$$

$$T_{15} = -20\rho^2\text{conv}\left(c_0, c_0, c_0, c_{1,d\theta^2}\right)(0) \tag{A.136}$$

$$T_{16} = -120\rho^2\text{conv}\left(c_0, c_0, c_{0,d\theta^2}, c_1\right)(0) \tag{A.137}$$

$$T_{17} = -4\rho^2\text{conv}\left(r_{AbC}, c_0, c_0, c_{1,d\theta}\right)(0) \tag{A.138}$$

$$T_{18} = +16\rho^2\text{conv}\left(c_{0,d\theta^2}, c_1, c_1\right)(0) \tag{A.139}$$

$$T_{19} = +20\rho^2\text{conv}\left(c_1, c_2\right)(0) \tag{A.140}$$

$$T_{20} = -18\rho^2\text{conv}\left(c_0, c_0, c_{0,d\theta}, c_{1,d\theta}\right)(0) \tag{A.141}$$

$$T_{21} = -\,14\rho^2 \mathrm{conv}\left(r_{AbC}, c_0, c_{0,d\theta}, c_1\right)(0) \tag{A.142}$$

$$T_{22} = -\,72\pi \mathrm{conv}\left(c_0, c_1, f_{1,dr}, f_{1,dr}\right)(0) \tag{A.143}$$

$$T_{23} = -\,48\pi \mathrm{conv}\left(c_1, f_{1,dr}, f_{2,dr}\right)(0) \tag{A.144}$$

$$T_{24} = -\,48\pi \mathrm{conv}\left(c_1, f_{1,d\theta}, f_{2,d\theta}\right)(0) \tag{A.145}$$

$$T_{25} = \left(-80\rho^4 + 120\rho^2\right) \mathrm{conv}\left(c_0, c_0, c_0, c_1\right)(0) \tag{A.146}$$

$$T_{26} = \left(60\rho^4 - 60\rho^2\right) \mathrm{conv}\left(c_0, c_1, c_1\right)(0) \tag{A.147}$$

The following terms are zero because of the communtator R.

$$T_{27} = +\,16\rho\sqrt{\pi} \mathrm{conv}\left(r_{AbC}, c_0, c_0, c_{0,d\theta}, f_{1,dr}\right)(0) \tag{A.148}$$

$$T_{28} = +\,2\rho^2 \mathrm{conv}\left(r_{ABC}, c_0, c_0, c_0, c_{0,d\theta}\right)(0) \tag{A.149}$$

$$T_{29} = +\,16\rho^2 \mathrm{conv}\left(r_{AbC}, c_0, c_0, c_0, c_{0,d\theta}\right)(0) \tag{A.150}$$

A.5.2 Convolutions Involving Only Velocity

The following convolutions only have coefficients related to the surface velocity.

$$T_{30} = +\,2\rho^2 \mathrm{conv}\left(c_0, c_0, c_{0,d\theta}, c_{0,d\theta}, c_{0,d\theta^2}\right)(0) \tag{A.151}$$

$$= \frac{36\rho^2\zeta(6)}{N^6} \tag{A.152}$$

$$T_{31} = +\,2\rho^2 \mathrm{conv}\left(c_0, c_0, c_0, c_{0,d\theta}, c_{0,d\theta^3}\right)(0) \tag{A.153}$$

$$= \frac{-600\zeta(8)\rho^2}{N^8} = O\left(\frac{1}{N^7}\right) \tag{A.154}$$

$$T_{32} = +\,2\rho^2 \mathrm{conv}\left(c_0, c_0, c_0, c_{0,d\theta^2}, c_{0,d\theta^2}\right)(0) \tag{A.155}$$

$$= -\,\frac{108\rho^2\zeta(6)}{N^6} \tag{A.156}$$

$$T_{33} = \left(-16\rho^4 + 80\rho^2\right) \text{conv}\left(c_0, c_0, c_0, c_{0,d\theta}, c_{0,d\theta}\right)(0) \tag{A.157}$$

$$= \frac{960\zeta(8)\rho^4}{N^8} - \frac{4800\zeta(8)\rho^2}{N^8} = O\left(\frac{1}{N^7}\right) \tag{A.158}$$

$$T_{34} = \left(-16\rho^6 + 28\rho^4 - 48\rho^2\right) \text{conv}\left(c_0, c_0, c_0, c_0, c_0\right)(0) \tag{A.159}$$

$$= \frac{17280\zeta(10)\rho^6}{N^{10}} - \frac{30240\zeta(10)\rho^4}{N^{10}} + \frac{51840\zeta(10)\rho^2}{N^{10}} = O\left(\frac{1}{N^7}\right) \tag{A.160}$$

$$T_{35} = \left(-24\rho^4 + 92\rho^2\right) \text{conv}\left(c_0, c_0, c_0, c_0, c_{0,d\theta^2}\right)(0) \tag{A.161}$$

$$= -\frac{5760\zeta(8)\rho^4}{N^8} + \frac{22080\zeta(8)\rho^2}{N^8} = O\left(\frac{1}{N^7}\right) \tag{A.162}$$

$$T_{36} = +\,18\rho^2 \text{conv}\left(c_0, c_{0,d\theta^2}, c_2\right)(0) \tag{A.163}$$

$$= -\frac{648\rho^2\zeta(6)}{N^6} \tag{A.164}$$

$$T_{37} = +\,2\rho^2 \text{conv}\left(c_0, c_0, c_{2,d\theta^2}\right)(0) \tag{A.165}$$

$$= \frac{72\rho^2\zeta(6)}{N^6} \tag{A.166}$$

$$T_{38} = \left(40\rho^4 - 40\rho^2\right) \text{conv}\left(c_0, c_0, c_2\right)(0) \tag{A.167}$$

$$= \frac{4800\zeta(8)\rho^2(\rho^2 - 1)}{N^8} = O\left(\frac{1}{N^7}\right) \tag{A.168}$$

$$T_{39} = +\,10\rho^2 \text{conv}\left(c_0, c_3\right)(0) \tag{A.169}$$

$$= -\frac{6000\rho^2\zeta(8)}{N^8} = O\left(\frac{1}{N^7}\right) \tag{A.170}$$

$$T_{40} = -\,2\rho^2 c_4(0) \tag{A.171}$$

$$= \frac{648\rho^2\zeta(6)}{N^6} \tag{A.172}$$

A.5.3 Convolutions Involving the First Eigenfunction

$$T_{41} = 16\rho\sqrt{\pi}\text{conv}\left(c_3, f_{1,dr}\right) \quad \text{(A.173)}$$

$$= \frac{57600\rho^2\zeta(7)}{N^7} - \frac{34560\zeta(2)\zeta(5)\rho^2}{N^7} = O\left(\frac{1}{N^7}\right) \quad \text{(A.174)}$$

$$T_{42} = 64\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_{0,d\theta}, f_{1,drd\theta}\right)(0) \quad \text{(A.175)}$$

$$= -\frac{80640\rho^2\zeta(7)}{N^7} + \frac{46080\zeta(2)\zeta(5)\rho^2}{N^7} = O\left(\frac{1}{N^7}\right) \quad \text{(A.176)}$$

$$T_{43} = 32\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_{0,d\theta}, c_{0,d\theta}, f_{1,dr}\right)(0) \quad \text{(A.177)}$$

$$= \frac{11520\rho^2\zeta(7)}{N^7} - \frac{6912\rho^2\zeta(2)\zeta(5)}{N^7} = O\left(\frac{1}{N^7}\right) \quad \text{(A.178)}$$

$$T_{44} = -48\rho^3\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_0, f_{1,dr^2}\right)(0) \quad \text{(A.179)}$$

$$= \frac{11520\rho^4\zeta(8)}{N^8} = O\left(\frac{1}{N^7}\right) \quad \text{(A.180)}$$

$$T_{45} = -16\rho^3\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_0, f_{1,dr}\right)(0) \quad \text{(A.181)}$$

$$= \frac{80640\rho^4\zeta(9)}{N^9} - \frac{46080\rho^4\zeta(2)\zeta(7)}{N^9} = O\left(\frac{1}{N^7}\right) \quad \text{(A.182)}$$

$$T_{46} = -48\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_{0,d\theta^2}, f_{1,dr^2}\right)(0) \quad \text{(A.183)}$$

$$= -\frac{2592\zeta(6)\rho^2}{N^6} \quad \text{(A.184)}$$

$$T_{47} = 16\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_0, f_{1,dr^4}\right)(0) \quad \text{(A.185)}$$

$$= \frac{483840\rho^2\zeta(7)}{N^7} - \frac{276480\rho^2\zeta(2)\zeta(5)}{N^7} = O\left(\frac{1}{N^7}\right) \quad \text{(A.186)}$$

$$T_{48} = -16\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_{0,d\theta}, c_{0,d\theta}, f_{1,dr^2}\right)(0) \quad \text{(A.187)}$$

$$= \frac{288\zeta(6)\rho^2}{N^6} \quad \text{(A.188)}$$

$$T_{49} = -48\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_{0,d\theta}, f_{1,dr^2d\theta}\right)(0) \quad \text{(A.189)}$$

$$= \frac{34560\rho^2\zeta(2)\zeta(5)}{N^7} - \frac{60480\rho^2\zeta(7)}{N^7} = O\left(\frac{1}{N^7}\right) \quad \text{(A.190)}$$

$$T_{50} = 64\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, c_0, c_{0,d\theta}, f_{1,dr}\right)(0) \tag{A.191}$$

$$= O\left(\frac{1}{N^7}\right) \tag{A.192}$$

$$T_{51} = 64\rho\sqrt{\pi}\text{conv}\left(c_0, c_2, f_{1,dr^2}\right)(0) \tag{A.193}$$

$$= \frac{2304\zeta(6)\rho^2}{N^6} \tag{A.194}$$

$$T_{52} = 48\pi\text{conv}\left(c_0, c_0, c_{0,d\theta}, f_{1,d\theta}, f_{1,d\theta^2}\right)(0) \tag{A.195}$$

$$= \frac{864\zeta(6)\rho^2}{N^6} \tag{A.196}$$

$$T_{53} = -48\pi\text{conv}\left(c_0, c_0, c_0, f_{1,dr}, f_{1,dr^3}\right)(0) \tag{A.197}$$

$$= \frac{8640\rho^2\zeta(3)^2}{N^6} - \frac{6912\rho^2\zeta(2)\zeta(3)}{N^6} - \frac{15120\rho^2\zeta(6)}{N^6} + \frac{17280\rho^2\zeta(5)}{N^6} \tag{A.198}$$

$$T_{54} = -48\pi\text{conv}\left(c_0, c_0, c_0, f_{1,d\theta}, f_{1,dr^2d\theta}\right)(0) \tag{A.199}$$

$$= \frac{34560\rho^2\zeta(2)\zeta(5)}{N^7} - \frac{60480\rho^2\zeta(7)}{N^7} = O\left(\frac{1}{N^7}\right) \tag{A.200}$$

$$T_{55} = 48\pi\text{conv}\left(c_0, c_0, c_0, f_{1,d\theta}, f_{1,drd\theta}\right)(0) \tag{A.201}$$

$$= -\frac{60480\rho^2\zeta(7)}{N^7} + \frac{34560\rho^2\zeta(2)\zeta(5)}{N^7} = O\left(\frac{1}{N^7}\right) \tag{A.202}$$

$$T_{56} = 24\pi\text{conv}\left(c_0, c_0, c_{0,d\theta^2}, f_{1,dr}, f_{1,dr}\right)(0) \tag{A.203}$$

$$= \frac{576\rho^2\zeta(3)^2}{N^6} - \frac{432\rho^2\zeta(6)}{N^6} \tag{A.204}$$

$$T_{57} = 24\pi\text{conv}\left(c_0, c_0, c_{0,d\theta^2}, f_{1,d\theta}, f_{1,d\theta}\right)(0) \tag{A.205}$$

$$= \frac{432\rho^2\zeta(6)}{N^6} \tag{A.206}$$

$$T_{58} = -48\pi\text{conv}\left(c_0, c_0, c_0, f_{1,dr^2}, f_{1,dr^2}\right)(0) \tag{A.207}$$

$$= \frac{2592\rho^2\zeta(6)}{N^6} \tag{A.208}$$

$$T_{59} = -48\pi\text{conv}\left(c_0, c_0, c_0, f_{1,drd\theta}, f_{1,drd\theta}\right)(0) \tag{A.209}$$

$$= -\frac{6912\rho^2\zeta(2)\zeta(3)}{N^6} - \frac{2592\zeta(6)\rho^2}{N^6} + \frac{17280\rho^2\zeta(5)}{N^6} \tag{A.210}$$

$$T_{60} = -96\pi\text{conv}\left(c_0, c_0, c_0, f_{1,dr}, f_{1,dr^2}\right)(0) \tag{A.211}$$

$$= O\left(\frac{1}{N^7}\right) \tag{A.212}$$

$$T_{61} = 48\pi\text{conv}\left(c_0, c_0, c_0, f_{1,d\theta}, f_{1,drd\theta}\right)(0) \tag{A.213}$$

$$= \frac{34560\rho^2\zeta(2)\zeta(5)}{N^7} - \frac{60480\rho^2\zeta(7)}{N^7} = O\left(\frac{1}{N^7}\right) \tag{A.214}$$

$$T_{62} =- 48\pi\mathrm{conv}\left(c_0, c_0, c_0, f_{1,d\theta} f_{1,d\theta}\right)(0) \tag{A.215}$$
$$=\frac{2880\rho^2\zeta(8)}{N^8} = O\left(\frac{1}{N^7}\right) \tag{A.216}$$
$$T_{63} =- 24\pi\mathrm{conv}\left(c_2, f_{1,dr}, f_{1,dr}\right)(0) \tag{A.217}$$
$$=- \frac{3744\rho^2\zeta(6)}{N^6} + \frac{2304\rho^2\zeta(3)^2}{N^6} \tag{A.218}$$
$$T_{64} =- 24\pi\mathrm{conv}\left(c_2, f_{1,d\theta}, f_{1,d\theta}\right)(0) \tag{A.219}$$
$$=- \frac{1296\rho^2\zeta(6)}{N^6} \tag{A.220}$$
$$T_{65} =48\pi\mathrm{conv}\left(c_0, c_0, c_{0,d\theta}, f_{1,dr}, f_{1,drd\theta}\right)(0) \tag{A.221}$$
$$=- \frac{2880\rho^2\zeta(3)^2}{N^6} + \frac{4176\rho^2\zeta(6)}{N^6} \tag{A.222}$$

A.5.4 Convolutions Involving the Second Eigenfunction

$$T_{66} =- 24\rho^3\sqrt{\pi}\mathrm{conv}\left(c_0, c_0, c_0, f_{2,dr}\right)(0) \tag{A.223}$$
$$=O\left(\frac{1}{N^7}\right) \tag{A.224}$$
$$T_{67} =- 24\rho\sqrt{\pi}\mathrm{conv}\left(c_0, c_0, c_{0,d\theta^2}, f_{2,dr}\right)(0) \tag{A.225}$$
$$=- \frac{576\rho^2\zeta(3)^2}{N^6} + \frac{2112\zeta(6)\rho^2}{N^6} \tag{A.226}$$
$$T_{68} =24\rho\sqrt{\pi}\mathrm{conv}\left(c_2, f_{2,dr}\right)(0) \tag{A.227}$$
$$=\frac{4416\rho^2\zeta(6)}{N^6} - \frac{2304\rho^2\zeta(3)^2}{N^6} \tag{A.228}$$
$$T_{69} =24\rho\sqrt{\pi}\mathrm{conv}\left(c_0, c_0, c_0, f_{2,dr^3}\right)(0) \tag{A.229}$$
$$=\frac{19584\zeta(6)\rho^2}{N^6} - \frac{8640\rho^2\zeta(3)^2}{N^6} \tag{A.230}$$
$$T_{70} =- 24\rho\sqrt{\pi}\mathrm{conv}\left(c_0, c_0, c_{0,d\theta}, f_{2,drd\theta}\right)(0) \tag{A.231}$$
$$=- \frac{6528\zeta(6)\rho^2}{N^6} + \frac{2880\rho^2\zeta(3)^2}{N^6} \tag{A.232}$$

$$T_{71} =-48\pi\text{conv}\left(c_0, c_0, f_{1,dr}, f_{2,dr}\right)(0) \tag{A.233}$$
$$=O\left(\frac{1}{N^7}\right) \tag{A.234}$$

$$T_{72} =-12\pi\text{conv}\left(c_0, f_{2,dr}, f_{2,dr}\right)(0) \tag{A.235}$$
$$=\frac{1919.8217678467\rho^2}{N^6} \tag{A.236}$$

$$T_{73} =-12\pi\text{conv}\left(c_0, f_{2,d\theta}, f_{2,d\theta}\right)(0) \tag{A.237}$$
$$=\frac{1708.980680\rho^2}{N^6} \tag{A.238}$$

$$T_{74} =-48\pi\text{conv}\left(c_0, c_0, f_{1,dr^2}, f_{2,dr}\right)(0) \tag{A.239}$$
$$=\frac{1152\zeta(3)^2\rho^2}{N^6}-\frac{4224\zeta(6)\rho^2}{N^6} \tag{A.240}$$

$$T_{75} =-48\pi\text{conv}\left(c_0, c_0, f_{1,drd\theta}, f_{2,d\theta}\right)(0) \tag{A.241}$$
$$=\frac{13536\zeta(6)\rho^2}{N^6}+\frac{13824\zeta(2)\zeta(3)\rho^2}{N^6}-\frac{34560\zeta(5)\rho^2}{N^6}-\frac{5760\zeta(3)^2\rho^2}{N^6} \tag{A.242}$$

$$T_{76} =48\pi\text{conv}\left(c_0, c_0, f_{1,d\theta}, f_{2,d\theta}\right)(0) \tag{A.243}$$
$$=O\left(\frac{1}{N^7}\right) \tag{A.244}$$

$$T_{77} =-48\pi\text{conv}\left(c_0, c_0, f_{1,dr}, f_{2,dr^2}\right)(0) \tag{A.245}$$
$$=-\frac{4625.2459797\rho^2}{N^6} \tag{A.246}$$

$$T_{78} =-48\pi\text{conv}\left(c_0, c_0, f_{1,d\theta}, f_{2,drd\theta}\right)(0) \tag{A.247}$$
$$=-\frac{13056\rho^2\zeta(6)}{N^6}+\frac{5760\rho^2\zeta(3)^2}{N^6} \tag{A.248}$$

A.5.5 Convolutions Involving the Third Eigenfunction

$$T_{79} =16\rho\sqrt{\pi}\text{conv}\left(c_0, c_0, f_{3,dr^2}\right)(0) \tag{A.249}$$
$$=\frac{-4076.319\rho^2}{N^6} \tag{A.250}$$

$$T_{80} =-16\pi\text{conv}\left(c_0, f_{1,dr}, f_{3,dr}\right)(0) \tag{A.251}$$
$$=\frac{2065\,\zeta\,(6)\,\rho^2}{N^6} \tag{A.252}$$

$$T_{81} =-16\pi\text{conv}\left(c_0, f_{1,d\theta}, f_{3,d\theta}\right)(0) \tag{A.253}$$
$$=\frac{2038.159818\rho^2}{N^6} \tag{A.254}$$

A.5.6 Convolution Involving the Fourth Eigenfunction

$$T_{82} = 4\rho\sqrt{\pi}\text{conv}\left(c_0, f_{4,dr}\right)(0) \tag{A.255}$$

$$= \frac{240.4192524\rho^2}{N^6} \tag{A.256}$$

A.5.7 Complete λ_5

This number is not determined exactly, but looking that the remaining numerical results and using a hypothesis about the final series gives an answer. The hypothesis used is given in Chapter 11.5.

$$\lambda_5 = \frac{1680\zeta(6)\rho^2}{N^6} \tag{A.257}$$

$$= \frac{1709.136346\rho^2}{N^6} \tag{A.258}$$

$$\frac{\lambda_5}{5!\rho^2} = \frac{14\zeta(6)}{N^6} \tag{A.259}$$

A.5.8 Numerical Results

Some of the results in this section could not be approximated only at the topmost convolution. Tables that have smaller maximum ranges required multiple nested convolutions.

Table A.49: Order ρ^6/N^2 contribution from convolution 30 of λ_5

$T_{30}/5! = \frac{v\rho^2}{N^6}, v = (36\,\zeta\,(6))\,/5!$					
$v = 0.3052029185953347419143553789372761583705452470098560685527225991$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	0.198122805694	$1.07080e-01$	$3.50849e-01$	n/a	0.004 sec.
4	0.284667575333	$2.05353e-02$	$6.72842e-02$	n/a	0.002 sec.
8	0.302037284967	$3.16563e-03$	$1.03722e-02$	-2.317	0.004 sec.
16	0.304765281692	$4.37637e-04$	$1.43392e-03$	-2.671	0.006 sec.
32	0.305145477284	$5.74413e-05$	$1.88207e-04$	-2.843	0.013 sec.
64	0.305195564404	$7.35419e-06$	$2.40961e-05$	-2.924	0.040 sec.
128	0.305201988363	$9.30232e-07$	$3.04791e-06$	-2.963	0.064 sec.
256	0.305202801629	$1.16966e-07$	$3.83241e-07$	-2.982	0.118 sec.
512	0.305202903932	$1.46638e-08$	$4.80460e-08$	-2.991	0.290 sec.
1024	0.305202916760	$1.83566e-09$	$6.01457e-09$	-2.995	0.641 sec.
2048	0.305202918366	$2.29626e-10$	$7.52373e-10$	-2.998	1.572 sec.
4096	0.305202918567	$2.87138e-11$	$9.40810e-11$	-2.999	4.597 sec.
8192	0.305202918592	$3.58988e-12$	$1.17623e-11$	-2.999	15.237 sec.
16384	0.305202918595	$4.48777e-13$	$1.47042e-12$	-3.000	74.366 sec.
32768	0.305202918595	$5.60996e-14$	$1.83811e-13$	-3.000	256.377 sec.
Richardson	0.305202918595	$2.93496e-18$	$9.61644e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.50: Order ρ^6/N^2 contribution from convolution 31 of λ_5

$T_{31}/5! = \frac{v\rho^2}{N^6}, v = (0)\,/5!$					
$v = 0.0$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	0.355215647685	$3.55216e-01$	Inf	n/a	0.005 sec.
4	0.068348964959	$6.83490e-02$	Inf	n/a	0.007 sec.
8	0.010547680875	$1.05477e-02$	Inf	-2.311	0.006 sec.
16	0.001458626889	$1.45863e-03$	Inf	-2.669	0.011 sec.
32	0.000191465527	$1.91466e-04$	Inf	-2.843	0.022 sec.
64	0.000024513794	$2.45138e-05$	Inf	-2.924	0.047 sec.
128	0.000003100767	$3.10077e-06$	Inf	-2.963	0.094 sec.
256	0.000000389887	$3.89887e-07$	Inf	-2.982	0.224 sec.
512	0.000000048879	$4.88793e-08$	Inf	-2.991	0.479 sec.
1024	0.000000006119	$6.11888e-09$	Inf	-2.995	1.097 sec.
2048	0.000000000765	$7.65421e-10$	Inf	-2.998	2.874 sec.
4096	0.000000000096	$9.57127e-11$	Inf	-2.999	7.900 sec.
8192	0.000000000012	$1.19663e-11$	Inf	-2.999	28.429 sec.
16384	0.000000000001	$1.49592e-12$	Inf	-3.000	141.946 sec.
32768	0.000000000000	$1.86999e-13$	Inf	-3.000	626.996 sec.
Richardson	0.000000000000	$9.78323e-18$	Inf	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.51: Order ρ^6/N^2 contribution from convolution 32 of λ_5

$T_{32}/5! = \frac{v\rho^2}{N^6}, v = (-108\,\zeta(6)\,(6))\,/5!$					
$v = -0.9156087557860042257430661368118284751116357410295682056581677971$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-0.949584064769	$3.39753e-02$	$3.71068e-02$	n/a	0.003 sec.
4	-0.922351690960	$6.74294e-03$	$7.36443e-03$	n/a	0.002 sec.
8	-0.916659535776	$1.05078e-03$	$1.14763e-03$	-2.258	0.004 sec.
16	-0.915754471965	$1.45716e-04$	$1.59147e-04$	-2.653	0.009 sec.
32	-0.915627897380	$1.91416e-05$	$2.09059e-05$	-2.838	0.017 sec.
64	-0.915611207004	$2.45122e-06$	$2.67715e-06$	-2.923	0.033 sec.
128	-0.915609065858	$3.10072e-07$	$3.38651e-07$	-2.963	0.075 sec.
256	-0.915608794775	$3.89885e-08$	$4.25821e-08$	-2.982	0.168 sec.
512	-0.915608760674	$4.88793e-09$	$5.33844e-09$	-2.991	0.348 sec.
1024	-0.915608756398	$6.11888e-10$	$6.68286e-10$	-2.995	0.757 sec.
2048	-0.915608755863	$7.65421e-11$	$8.35970e-11$	-2.998	1.879 sec.
4096	-0.915608755796	$9.57127e-12$	$1.04534e-11$	-2.999	5.328 sec.
8192	-0.915608755787	$1.19663e-12$	$1.30692e-12$	-2.999	19.694 sec.
16384	-0.915608755786	$1.49592e-13$	$1.63380e-13$	-3.000	119.563 sec.
32768	-0.915608755786	$1.86999e-14$	$2.04234e-14$	-3.000	331.078 sec.
Richardson	-0.915608755786	$9.78340e-19$	$1.06851e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.52: Order ρ^6/N^2 contribution from convolution 36 of λ_5

$T_{36}/5! = \frac{v\rho^2}{N^6}, v = (-648\,\zeta\,(6))\,/5!$					
$v = -5.4936525347160253544583968208709708506698144461774092339490067782$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-5.388634096196	$1.05018e-01$	$1.91163e-02$	n/a	0.004 sec.
4	-5.473239806550	$2.04127e-02$	$3.71569e-03$	n/a	0.004 sec.
8	-5.490492218551	$3.16032e-03$	$5.75267e-04$	-2.294	0.010 sec.
16	-5.493215093159	$4.37442e-04$	$7.96267e-05$	-2.664	0.014 sec.
32	-5.493595100016	$5.74347e-05$	$1.04547e-05$	-2.841	0.032 sec.
64	-5.493645180739	$7.35398e-06$	$1.33863e-06$	-2.924	0.064 sec.
128	-5.493651604491	$9.30225e-07$	$1.69327e-07$	-2.963	0.133 sec.
256	-5.493652417750	$1.16966e-07$	$2.12911e-08$	-2.982	0.249 sec.
512	-5.493652520052	$1.46638e-08$	$2.66922e-09$	-2.991	0.557 sec.
1024	-5.493652532880	$1.83566e-09$	$3.34143e-10$	-2.995	1.210 sec.
2048	-5.493652534486	$2.29626e-10$	$4.17985e-11$	-2.998	2.775 sec.
4096	-5.493652534687	$2.87138e-11$	$5.22672e-12$	-2.999	7.522 sec.
8192	-5.493652534712	$3.58988e-12$	$6.53460e-13$	-2.999	21.566 sec.
16384	-5.493652534716	$4.48777e-13$	$8.16900e-14$	-3.000	89.189 sec.
32768	-5.493652534716	$5.60996e-14$	$1.02117e-14$	-3.000	406.313 sec.
Richardson	-5.493652534716	$2.93499e-18$	$5.34250e-19$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.53: Order ρ^6/N^2 contribution from convolution 37 of λ_5

$T_{37}/5! = \frac{v\rho^2}{N^6}, v = (72\,\zeta(6))/5!$					
$v = 0.6104058371906694838287107578745523167410904940197121371054451980$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	0.680418129537	$7.00123e-02$	$1.14698e-01$	n/a	0.003 sec.
4	0.624014322634	$1.36085e-02$	$2.22942e-02$	n/a	0.002 sec.
8	0.612512714634	$2.10688e-03$	$3.45160e-03$	-2.294	0.004 sec.
16	0.610697464895	$2.91628e-04$	$4.77760e-04$	-2.664	0.007 sec.
32	0.610444126990	$3.82898e-05$	$6.27284e-05$	-2.841	0.011 sec.
64	0.610410739842	$4.90265e-06$	$8.03179e-06$	-2.924	0.027 sec.
128	0.610406457341	$6.20150e-07$	$1.01596e-06$	-2.963	0.046 sec.
256	0.610405915168	$7.79773e-08$	$1.27747e-07$	-2.982	0.104 sec.
512	0.610405846967	$9.77586e-09$	$1.60153e-08$	-2.991	0.245 sec.
1024	0.610405838414	$1.22378e-09$	$2.00486e-09$	-2.995	0.555 sec.
2048	0.610405837344	$1.53084e-10$	$2.50791e-10$	-2.998	1.465 sec.
4096	0.610405837210	$1.91425e-11$	$3.13603e-11$	-2.999	4.450 sec.
8192	0.610405837193	$2.39326e-12$	$3.92076e-12$	-2.999	17.862 sec.
16384	0.610405837191	$2.99184e-13$	$4.90140e-13$	-3.000	75.223 sec.
32768	0.610405837191	$3.73998e-14$	$6.12703e-14$	-3.000	318.219 sec.
Richardson	0.610405837191	$1.95666e-18$	$3.20550e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.54: Order ρ^6/N^2 contribution from convolution 46 of λ_5

$T_{46}/5! = \frac{v\rho^2}{N^6}, v = (-2592\,\zeta(6))/5!$					
$v = -21.974610138864101417833587283483883402679257784709636935796027 13$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-22.790017554444	$8.15407e-01$	$3.71068e-02$	n/a	0.007 sec.
4	-22.136440583032	$1.61830e-01$	$7.36443e-03$	n/a	0.010 sec.
8	-21.999828858624	$2.52187e-02$	$1.14763e-03$	-2.258	0.018 sec.
16	-21.978107327166	$3.49719e-03$	$1.59147e-04$	-2.653	0.029 sec.
32	-21.975069537127	$4.59398e-04$	$2.09059e-05$	-2.838	0.062 sec.
64	-21.974668968100	$5.88292e-05$	$2.67715e-06$	-2.923	0.127 sec.
128	-21.974617580582	$7.44172e-06$	$3.38651e-07$	-2.963	0.241 sec.
256	-21.974611074589	$9.35725e-07$	$4.25821e-08$	-2.982	0.499 sec.
512	-21.974610256174	$1.17310e-07$	$5.33844e-09$	-2.991	1.064 sec.
1024	-21.974610153549	$1.46853e-08$	$6.68286e-10$	-2.995	2.296 sec.
2048	-21.974610140701	$1.83701e-09$	$8.35970e-11$	-2.998	4.912 sec.
4096	-21.974610139094	$2.29710e-10$	$1.04534e-11$	-2.999	12.483 sec.
8192	-21.974610138893	$2.87191e-11$	$1.30692e-12$	-2.999	36.028 sec.
16384	-21.974610138868	$3.59021e-12$	$1.63380e-13$	-3.000	163.035 sec.
32768	-21.974610138865	$4.48797e-13$	$2.04234e-14$	-3.000	555.420 sec.
Richardson	-21.974610138864	$2.34801e-17$	$1.06851e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.55: Order ρ^6/N^2 contribution from convolution 47 of λ_5

$T_{47}/5! = \frac{v\rho^2}{N^6}, v = (0)/5!$					
$v = 0.0$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-11.366900725924	$1.13669e+01$	Inf	n/a	0.020 sec.
4	-2.187166878700	$2.18717e+00$	Inf	n/a	0.028 sec.
8	-0.337525788002	$3.37526e-01$	Inf	-2.311	0.048 sec.
16	-0.046676060433	$4.66761e-02$	Inf	-2.669	0.121 sec.
32	-0.006126896863	$6.12690e-03$	Inf	-2.843	0.196 sec.
64	-0.000784441397	$7.84441e-04$	Inf	-2.924	0.432 sec.
128	-0.000099224551	$9.92246e-05$	Inf	-2.963	1.102 sec.
256	-0.000012476387	$1.24764e-05$	Inf	-2.982	1.922 sec.
512	-0.000001564138	$1.56414e-06$	Inf	-2.991	3.612 sec.
1024	-0.000000195804	$1.95804e-07$	Inf	-2.995	7.669 sec.
2048	-0.000000024493	$2.44935e-08$	Inf	-2.998	16.289 sec.
4096	-0.000000003063	$3.06281e-09$	Inf	-2.999	37.832 sec.
8192	-0.000000000383	$3.82921e-10$	Inf	-2.999	92.810 sec.
16384	-0.000000000048	$4.78695e-11$	Inf	-3.000	293.204 sec.
32768	-0.000000000006	$5.98396e-12$	Inf	-3.000	1001.152 sec.
Richardson	-0.000000000000	$3.13063e-16$	Inf	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.56: Order ρ^6/N^2 contribution from convolution 48 of λ_5

$T_{48}/5! = \frac{v\rho^2}{N^6}, v = (288\,\zeta(6))/5!$					
$v = 2.441623348762677935314843031498209266964361976078848548421780792$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	1.584982445556	$8.56641e-01$	$3.50849e-01$	n/a	0.008 sec.
4	2.277340602667	$1.64283e-01$	$6.72842e-02$	n/a	0.007 sec.
8	2.416298279736	$2.53251e-02$	$1.03722e-02$	-2.317	0.014 sec.
16	2.438122253538	$3.50110e-03$	$1.43392e-03$	-2.671	0.027 sec.
32	2.441163818276	$4.59530e-04$	$1.88207e-04$	-2.843	0.065 sec.
64	2.441564515228	$5.88335e-05$	$2.40961e-05$	-2.924	0.119 sec.
128	2.441615906908	$7.44186e-06$	$3.04791e-06$	-2.963	0.228 sec.
256	2.441622413033	$9.35729e-07$	$3.83241e-07$	-2.982	0.447 sec.
512	2.441623231452	$1.17310e-07$	$4.80460e-08$	-2.991	0.977 sec.
1024	2.441623334077	$1.46853e-08$	$6.01457e-09$	-2.995	1.985 sec.
2048	2.441623346926	$1.83701e-09$	$7.52373e-10$	-2.998	4.419 sec.
4096	2.441623348533	$2.29710e-10$	$9.40810e-11$	-2.999	11.499 sec.
8192	2.441623348734	$2.87191e-11$	$1.17623e-11$	-2.999	34.070 sec.
16384	2.441623348759	$3.59021e-12$	$1.47042e-12$	-3.000	120.347 sec.
32768	2.441623348762	$4.48797e-13$	$1.83811e-13$	-3.000	491.473 sec.
Richardson	2.441623348763	$2.34797e-17$	$9.61644e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.57: Order ρ^6/N^2 contribution from convolution 51 of λ_5

$T_{51}/5! = \frac{v\rho^2}{N^6}, v = (2304\,\zeta(6))/5!$					
$v = 19.53298679010142348251874425198567413571489580863078838737424634$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	12.679859564446	$6.85313e+00$	$3.50849e-01$	n/a	0.007 sec.
4	18.218724821340	$1.31426e+00$	$6.72842e-02$	n/a	0.009 sec.
8	19.330386237887	$2.02601e-01$	$1.03722e-02$	-2.317	0.019 sec.
16	19.504978028303	$2.80088e-02$	$1.43392e-03$	-2.671	0.042 sec.
32	19.529310546204	$3.67624e-03$	$1.88207e-04$	-2.843	0.062 sec.
64	19.532516121825	$4.70668e-04$	$2.40961e-05$	-2.924	0.139 sec.
128	19.532927255261	$5.95348e-05$	$3.04791e-06$	-2.963	0.280 sec.
256	19.532979304266	$7.48584e-06$	$3.83241e-07$	-2.982	0.650 sec.
512	19.532985851619	$9.38483e-07$	$4.80460e-08$	-2.991	1.288 sec.
1024	19.532986672619	$1.17483e-07$	$6.01457e-09$	-2.995	2.397 sec.
2048	19.532986775405	$1.46961e-08$	$7.52373e-10$	-2.998	5.627 sec.
4096	19.532986788264	$1.83768e-09$	$9.40810e-11$	-2.999	13.288 sec.
8192	19.532986789872	$2.29753e-10$	$1.17623e-11$	-2.999	38.629 sec.
16384	19.532986790073	$2.87217e-11$	$1.47042e-12$	-3.000	136.818 sec.
32768	19.532986790098	$3.59038e-12$	$1.83811e-13$	-3.000	584.589 sec.
Richardson	19.532986790101	$1.87838e-16$	$9.61644e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.58: Order ρ^6/N^2 contribution from convolution 52 of λ_5

$T_{52}/5! = \frac{v\rho^2}{N^6}, v = (864\,\zeta(6))/5!$					
$v = 7.324870046288033805944529094494627800893085928236545645265342376$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	4.754947336667	$2.56992e+00$	$3.50849e-01$	n/a	0.005 sec.
4	6.832021808002	$4.92848e-01$	$6.72842e-02$	n/a	0.003 sec.
8	7.248894839208	$7.59752e-02$	$1.03722e-02$	-2.317	0.005 sec.
16	7.314366760614	$1.05033e-02$	$1.43392e-03$	-2.671	0.010 sec.
32	7.323491454827	$1.37859e-03$	$1.88207e-04$	-2.843	0.020 sec.
64	7.324693545684	$1.76501e-04$	$2.40961e-05$	-2.924	0.038 sec.
128	7.324847720723	$2.23256e-05$	$3.04791e-06$	-2.963	0.107 sec.
256	7.324867239100	$2.80719e-06$	$3.83241e-07$	-2.982	0.185 sec.
512	7.324869694357	$3.51931e-07$	$4.80460e-08$	-2.991	0.397 sec.
1024	7.324870002232	$4.40560e-08$	$6.01457e-09$	-2.995	0.886 sec.
2048	7.324870040777	$5.51103e-09$	$7.52373e-10$	-2.998	2.021 sec.
4096	7.324870045599	$6.89131e-10$	$9.40810e-11$	-2.999	5.489 sec.
8192	7.324870046202	$8.61572e-11$	$1.17623e-11$	-2.999	17.697 sec.
16384	7.324870046277	$1.07706e-11$	$1.47042e-12$	-3.000	77.522 sec.
32768	7.324870046287	$1.34639e-12$	$1.83811e-13$	-3.000	290.302 sec.
Richardson	7.324870046288	$7.04391e-17$	$9.61644e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.59: Order ρ^6/N^2 contribution from convolution 53 of λ_5

$T_{53}/5! = \frac{v\rho^2}{N^6}, v = \left(8640\ (\zeta(3))^2 - 6912\,\zeta(2)\,\zeta(3) - 15120\,\zeta(6) + 17280\,\zeta(5)\right)/5!$					
$v = 11.275377840702086221320575140716824289887673399512150248724150 5$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	15.301784668868	$4.02641e+00$	$3.57097e-01$	n/a	0.025 sec.
4	10.577996657418	$6.97381e-01$	$6.18499e-02$	n/a	0.009 sec.
8	10.530398794140	$7.44979e-01$	$6.60713e-02$	-6.633	0.021 sec.
16	10.941289725770	$3.34088e-01$	$2.96299e-02$	3.110	0.044 sec.
32	11.156014168103	$1.19364e-01$	$1.05862e-02$	-0.936	0.081 sec.
64	11.236902421650	$3.84754e-02$	$3.41234e-03$	-1.408	0.231 sec.
128	11.263660684583	$1.17172e-02$	$1.03918e-03$	-1.596	0.862 sec.
256	11.271933708552	$3.44413e-03$	$3.05456e-04$	-1.693	1.959 sec.
512	11.274389571611	$9.88269e-04$	$8.76484e-05$	-1.752	4.101 sec.
1024	11.275099180109	$2.78661e-04$	$2.47141e-05$	-1.791	8.286 sec.
2048	11.275300308051	$7.75327e-05$	$6.87628e-06$	-1.819	14.910 sec.
4096	11.275356494998	$2.13457e-05$	$1.89313e-06$	-1.840	28.204 sec.
8192	11.275372014242	$5.82646e-06$	$5.16742e-07$	-1.856	52.668 sec.
16384	11.275376261661	$1.57904e-06$	$1.40043e-07$	-1.869	111.265 sec.
32768	11.275377415347	$4.25355e-07$	$3.77243e-08$	-1.880	243.915 sec.
Richardson	11.275377799909	$4.07936e-08$	$3.61793e-09$	n/a	n/a
Error: $O\left(\text{range}^{-1.880}\right)$					

Table A.60: Order ρ^6/N^2 contribution from convolution 56 of λ_5

$T_{56}/5! = \frac{v\rho^2}{N^6}, v = \left(576\ (\zeta(3))^2 - 432\,\zeta(6)\right)/5!$					
$v = 3.2732808093374274198134238310262276700353869835483928131982127 84$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	3.422546679329	$1.49266e-01$	$4.56013e-02$	n/a	0.029 sec.
4	3.302610189269	$2.93294e-02$	$8.96024e-03$	n/a	0.017 sec.
8	3.277783626962	$4.50282e-03$	$1.37563e-03$	-2.272	0.040 sec.
16	3.273894234762	$6.13425e-04$	$1.87404e-04$	-2.674	0.062 sec.
32	3.273360088546	$7.92792e-05$	$2.42201e-05$	-2.864	0.142 sec.
64	3.273290835014	$1.00257e-05$	$3.06288e-06$	-2.947	0.291 sec.
128	3.273282066621	$1.25728e-06$	$3.84105e-07$	-2.982	0.793 sec.
256	3.273280966552	$1.57215e-07$	$4.80297e-08$	-2.995	1.667 sec.
512	3.273280828980	$1.96428e-08$	$6.00095e-09$	-2.999	3.472 sec.
1024	3.273280811791	$2.45401e-09$	$7.49709e-10$	-3.001	6.878 sec.
2048	3.273280809644	$3.06620e-10$	$9.36735e-11$	-3.001	14.629 sec.
4096	3.273280809376	$3.83162e-11$	$1.17058e-11$	-3.001	33.454 sec.
8192	3.273280809342	$4.78864e-12$	$1.46295e-12$	-3.000	78.456 sec.
16384	3.273280809338	$5.98514e-13$	$1.82848e-13$	-3.000	220.941 sec.
32768	3.273280809338	$7.48093e-14$	$2.28545e-14$	-3.000	881.211 sec.
Richardson	3.273280809337	$5.60693e-18$	$1.71294e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.61: Order ρ^6/N^2 contribution from convolution 57 of λ_5

$T_{57}/5! = \frac{v\rho^2}{N^6}, v = (432\,\zeta(6))/5!$					
$v = 3.662435023144016902972264547247313900446542964118272822632671189$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	3.798336259074	$1.35901e-01$	$3.71068e-02$	n/a	0.005 sec.
4	3.689406763839	$2.69717e-02$	$7.36443e-03$	n/a	0.006 sec.
8	3.666638143104	$4.20312e-03$	$1.14763e-03$	-2.258	0.011 sec.
16	3.663017887861	$5.82865e-04$	$1.59147e-04$	-2.653	0.021 sec.
32	3.662511589521	$7.65664e-05$	$2.09059e-05$	-2.838	0.043 sec.
64	3.662444828017	$9.80487e-06$	$2.67715e-06$	-2.923	0.084 sec.
128	3.662436263430	$1.24029e-06$	$3.38651e-07$	-2.963	0.181 sec.
256	3.662435179098	$1.55954e-07$	$4.25821e-08$	-2.982	0.395 sec.
512	3.662435042696	$1.95517e-08$	$5.33844e-09$	-2.991	0.786 sec.
1024	3.662435025592	$2.44755e-09$	$6.68286e-10$	-2.995	1.845 sec.
2048	3.662435023450	$3.06168e-10$	$8.35970e-11$	-2.998	4.481 sec.
4096	3.662435023182	$3.82851e-11$	$1.04534e-11$	-2.999	11.229 sec.
8192	3.662435023149	$4.78651e-12$	$1.30692e-12$	-2.999	38.797 sec.
16384	3.662435023145	$5.98369e-13$	$1.63380e-13$	-3.000	178.178 sec.
32768	3.662435023144	$7.47995e-14$	$2.04234e-14$	-3.000	781.390 sec.
Richardson	3.662435023144	$3.91336e-18$	$1.06851e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.62: Order ρ^6/N^2 contribution from convolution 58 of λ_5

$T_{58}/5! = \frac{v\rho^2}{N^6}, v = (2592\,\zeta(6))/5!$					
$v = 21.974610138864101417833587283483883402679257784709636935796027 13$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	22.790017554444	$8.15407e-01$	$3.71068e-02$	n/a	0.029 sec.
4	22.136440583032	$1.61830e-01$	$7.36443e-03$	n/a	0.019 sec.
8	21.999828858624	$2.52187e-02$	$1.14763e-03$	-2.258	0.036 sec.
16	21.978107327166	$3.49719e-03$	$1.59147e-04$	-2.653	0.063 sec.
32	21.975069537127	$4.59398e-04$	$2.09059e-05$	-2.838	0.153 sec.
64	21.974668968100	$5.88292e-05$	$2.67715e-06$	-2.923	0.297 sec.
128	21.974617580582	$7.44172e-06$	$3.38651e-07$	-2.963	0.788 sec.
256	21.974611074589	$9.35725e-07$	$4.25821e-08$	-2.982	1.730 sec.
512	21.974610256174	$1.17310e-07$	$5.33844e-09$	-2.991	3.591 sec.
1024	21.974610153549	$1.46853e-08$	$6.68286e-10$	-2.995	7.481 sec.
2048	21.974610140701	$1.83701e-09$	$8.35970e-11$	-2.998	14.755 sec.
4096	21.974610139094	$2.29710e-10$	$1.04534e-11$	-2.999	35.104 sec.
8192	21.974610138893	$2.87191e-11$	$1.30692e-12$	-2.999	81.651 sec.
16384	21.974610138868	$3.59021e-12$	$1.63380e-13$	-3.000	234.566 sec.
32768	21.974610138865	$4.48797e-13$	$2.04234e-14$	-3.000	885.615 sec.
Richardson	21.974610138864	$2.34801e-17$	$1.06851e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.63: Order ρ^6/N^2 contribution from convolution 59 of λ_5

$T_{59}/5! = \frac{v\rho^2}{N^6}, v = (-6912\,\zeta(2)\,\zeta(3) - 2592\,\zeta(6) + 17280\,\zeta(5))\,/5!$					
$v = 13.4502560246569115657309213367858038456084701439420775676083554$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	15.856625251470	$2.40637e+00$	$1.78909e-01$	n/a	0.021 sec.
4	14.220772588345	$7.70517e-01$	$5.72864e-02$	n/a	0.012 sec.
8	13.668165647513	$2.17910e-01$	$1.62012e-02$	-1.566	0.022 sec.
16	13.508214699049	$5.79587e-02$	$4.30911e-03$	-1.789	0.047 sec.
32	13.465203505400	$1.49475e-02$	$1.11132e-03$	-1.895	0.123 sec.
64	13.454051610156	$3.79559e-03$	$2.82194e-04$	-1.947	0.205 sec.
128	13.451212355262	$9.56331e-04$	$7.11013e-05$	-1.974	0.861 sec.
256	13.450496042556	$2.40018e-04$	$1.78449e-05$	-1.987	2.089 sec.
512	13.450316146412	$6.01218e-05$	$4.46993e-06$	-1.993	4.233 sec.
1024	13.450271069779	$1.50451e-05$	$1.11858e-06$	-1.997	9.086 sec.
2048	13.450259787774	$3.76312e-06$	$2.79780e-07$	-1.998	21.254 sec.
4096	13.450256965666	$9.41009e-07$	$6.99622e-08$	-1.999	72.065 sec.
8192	13.450256259938	$2.35281e-07$	$1.74927e-08$	-2.000	198.401 sec.
16384	13.450256083481	$5.88238e-08$	$4.37344e-09$	-2.000	676.692 sec.
32768	13.450256039363	$1.47064e-08$	$1.09339e-09$	-2.000	2681.493 sec.
Richardson	13.450256024658	$5.98394e-13$	$4.44894e-14$	n/a	n/a
Error: $O\left(\text{range}^{-2.000}\right)$					

Table A.64: Order ρ^6/N^2 contribution from convolution 63 of λ_5

$T_{63}/5! = \frac{v\rho^2}{N^6}, v = \left(-3744\,\zeta(6) + 2304\,(\zeta(3))^2\right)/5!$					
$v = -3.99824020398903586795020589638255418860898589835836858615961441$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	-4.468583912440	$4.70344e-01$	$1.17638e-01$	n/a	0.032 sec.
4	-4.463426643963	$4.65186e-01$	$1.16348e-01$	n/a	0.022 sec.
8	-4.217576293700	$2.19336e-01$	$5.48582e-02$	5.575	0.050 sec.
16	-4.078675276541	$8.04351e-02$	$2.01176e-02$	-0.824	0.081 sec.
32	-4.024440931851	$2.62007e-02$	$6.55306e-03$	-1.357	0.184 sec.
64	-4.006244253807	$8.00405e-03$	$2.00189e-03$	-1.576	0.397 sec.
128	-4.000592789193	$2.35259e-03$	$5.88405e-04$	-1.687	0.961 sec.
256	-3.998914390528	$6.74187e-04$	$1.68621e-04$	-1.752	2.157 sec.
512	-3.998429985868	$1.89782e-04$	$4.74664e-05$	-1.793	4.360 sec.
1024	-3.998292918676	$5.27147e-05$	$1.31845e-05$	-1.821	8.948 sec.
2048	-3.998254694370	$1.44904e-05$	$3.62419e-06$	-1.842	18.522 sec.
4096	-3.998244153732	$3.94974e-06$	$9.87870e-07$	-1.859	42.377 sec.
8192	-3.998241273104	$1.06912e-06$	$2.67396e-07$	-1.872	104.026 sec.
16384	-3.998240491673	$2.87684e-07$	$7.19527e-08$	-1.882	292.722 sec.
32768	-3.998240281009	$7.70203e-08$	$1.92636e-08$	-1.891	1192.477 sec.
Richardson	-3.998240210788	$6.79908e-09$	$1.70052e-09$	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.65: Order ρ^6/N^2 contribution from convolution 64 of λ_5

$T_{64}/5! = \frac{v\rho^2}{N^6}, v = (-1296\,\zeta(6))/5!$					
$v = -10.987305069432050708916793641741941701339628892354818467898013 57$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−9.974146186482	$1.01316e+00$	$9.22118e-02$	n/a	0.006 sec.
4	−10.794824431679	$1.92481e-01$	$1.75185e-02$	n/a	0.005 sec.
8	−10.957723705812	$2.95814e-02$	$2.69232e-03$	−2.333	0.010 sec.
16	−10.983219156029	$4.08591e-03$	$3.71876e-04$	−2.676	0.020 sec.
32	−10.986768906456	$5.36163e-04$	$4.87984e-05$	−2.844	0.042 sec.
64	−10.987236428876	$6.86406e-05$	$6.24726e-06$	−2.925	0.110 sec.
128	−10.987296387222	$8.68221e-06$	$7.90204e-07$	−2.963	0.167 sec.
256	−10.987303977746	$1.09169e-06$	$9.93588e-08$	−2.982	0.433 sec.
512	−10.987304932570	$1.36862e-07$	$1.24564e-08$	−2.991	0.817 sec.
1024	−10.987305052299	$1.71329e-08$	$1.55933e-09$	−2.995	1.914 sec.
2048	−10.987305067289	$2.14318e-09$	$1.95060e-10$	−2.998	4.996 sec.
4096	−10.987305069164	$2.67996e-10$	$2.43914e-11$	−2.999	10.162 sec.
8192	−10.987305069399	$3.35056e-11$	$3.04948e-12$	−2.999	33.412 sec.
16384	−10.987305069428	$4.18858e-12$	$3.81220e-13$	−3.000	155.095 sec.
32768	−10.987305069432	$5.23597e-13$	$4.76547e-14$	−3.000	689.000 sec.
Richardson	−10.987305069432	$2.73929e-17$	$2.49314e-18$	n/a	n/a
Error: $O\left(\text{range}^{-3.000}\right)$					

Table A.66: Order ρ^6/N^2 contribution from convolution 65 of λ_5

$T_{65}/5! = \frac{v\rho^2}{N^6}, v = \left(-2880\,(\zeta(3))^2 + 4176\,\zeta(6)\right)/5!$					
$v = 0.72495939465160844813678206535632651857359891480997577296140163$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	0.184946860867	$5.40013e-01$	$7.44887e-01$	n/a	0.020 sec.
4	1.214258643642	$4.89299e-01$	$6.74933e-01$	n/a	0.007 sec.
8	1.045922284458	$3.20963e-01$	$4.42732e-01$	−2.612	0.020 sec.
16	0.855641657760	$1.30682e-01$	$1.80261e-01$	0.177	0.034 sec.
32	0.769729779099	$4.47704e-02$	$6.17557e-02$	−1.147	0.064 sec.
64	0.739049729502	$1.40903e-02$	$1.94360e-02$	−1.486	0.179 sec.
128	0.729183890226	$4.22450e-03$	$5.82722e-03$	−1.637	0.626 sec.
256	0.726187444668	$1.22805e-03$	$1.69396e-03$	−1.719	1.507 sec.
512	0.725308858267	$3.49464e-04$	$4.82046e-04$	−1.770	2.988 sec.
1024	0.725057296557	$9.79019e-05$	$1.35045e-04$	−1.804	5.818 sec.
2048	0.724986493241	$2.70986e-05$	$3.73795e-05$	−1.829	11.348 sec.
4096	0.724966823556	$7.42890e-06$	$1.02473e-05$	−1.848	22.074 sec.
8192	0.724961415232	$2.02058e-06$	$2.78716e-06$	−1.863	43.482 sec.
16384	0.724959940606	$5.45955e-07$	$7.53083e-07$	−1.875	90.180 sec.
32768	0.724959541339	$1.46687e-07$	$2.02339e-07$	−1.885	179.394 sec.
Richardson	0.724959408250	$1.35981e-08$	$1.87570e-08$	n/a	n/a
Error: $O\left(\text{range}^{-1.885}\right)$					

Table A.67: Order ρ^6/N^2 contribution from convolution 67 of λ_5

$T_{67}/5! = \frac{v\rho^2}{N^6}, v = \left(-576\ (\zeta\,(3))^2 + 2112\,\zeta\,(6)\right)/5!$					
$v = 10.969522058444860536189827186046659720590057876911557052595508 50$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	11.003327481852	$3.38054e-02$	$3.08176e-03$	n/a	0.028 sec.
4	10.976574846808	$7.05279e-03$	$6.42944e-04$	n/a	0.025 sec.
8	10.970396301603	$8.74243e-04$	$7.96975e-05$	-2.114	0.050 sec.
16	10.969606219863	$8.41614e-05$	$7.67230e-06$	-2.967	0.110 sec.
32	10.969529143916	$7.08547e-06$	$6.45923e-07$	-3.358	0.177 sec.
64	10.969522611300	$5.52855e-07$	$5.03992e-08$	-3.561	0.428 sec.
128	10.969522099638	$4.11933e-08$	$3.75525e-09$	-3.674	1.045 sec.
256	10.969522061424	$2.97890e-09$	$2.71562e-10$	-3.743	2.312 sec.
512	10.969522058656	$2.11017e-10$	$1.92367e-11$	-3.787	4.615 sec.
1024	10.969522058460	$1.47242e-11$	$1.34228e-12$	-3.818	10.283 sec.
2048	10.969522058446	$1.01562e-12$	$9.25861e-14$	-3.840	24.473 sec.
4096	10.969522058445	$6.94146e-14$	$6.32795e-15$	-3.857	45.449 sec.
8192	10.969522058445	$4.70876e-15$	$4.29258e-16$	-3.870	105.166 sec.
16384	10.969522058445	$3.17417e-16$	$2.89362e-17$	-3.881	292.875 sec.
32768	10.969522058445	$2.12826e-17$	$1.94015e-18$	-3.890	1033.326 sec.
Richardson	10.969522058445	$1.54029e-18$	$1.40415e-19$	n/a	n/a
Error: $O\left(\text{range}^{-3.890}\right)$					

Table A.68: Order ρ^6/N^2 contribution from convolution 68 of λ_5

$T_{68}/5! = \frac{v\rho^2}{N^6}, v = \left(4416\,\zeta(6)\,(6) - 2304\ (\zeta\,(3))^2\right)/5!$					
$v = 9.695361351101951050351506303211709144859163842542348532477102 92$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	7.695768667898	$1.99959e+00$	$2.06242e-01$	n/a	0.031 sec.
4	8.558564585153	$1.13680e+00$	$1.17252e-01$	n/a	0.027 sec.
8	9.224163603317	$4.71198e-01$	$4.86003e-02$	-0.374	0.045 sec.
16	9.529920448278	$1.65441e-01$	$1.70639e-02$	-1.122	0.113 sec.
32	9.642354003593	$5.30073e-02$	$5.46729e-03$	-1.443	0.183 sec.
64	9.679275257472	$1.60861e-02$	$1.65915e-03$	-1.607	0.412 sec.
128	9.690646287196	$4.71506e-03$	$4.86322e-04$	-1.699	1.045 sec.
256	9.694011732221	$1.34962e-03$	$1.39203e-04$	-1.756	2.299 sec.
512	9.694981631045	$3.79720e-04$	$3.91651e-05$	-1.795	4.637 sec.
1024	9.695255902156	$1.05449e-04$	$1.08762e-05$	-1.822	9.418 sec.
2048	9.695332367892	$2.89832e-05$	$2.98939e-06$	-1.843	19.472 sec.
4096	9.695353451310	$7.89979e-06$	$8.14801e-07$	-1.859	41.865 sec.
8192	9.695359212833	$2.13827e-06$	$2.20546e-07$	-1.872	100.420 sec.
16384	9.695360775729	$5.75373e-07$	$5.93452e-08$	-1.882	298.913 sec.
32768	9.695361197061	$1.54041e-07$	$1.58881e-08$	-1.891	1058.892 sec.
Richardson	9.695361337505	$1.35974e-08$	$1.40246e-09$	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.69: Order ρ^6/N^2 contribution from convolution 69 of λ_5

$T_{69}/5! = \frac{v\rho^2}{N^6}, v = \left(19584\,\zeta(6) - 8640\,(\zeta(3))^2\right)/5!$					
$v = 61.99465022864043475962400046777510659634766515836171675521783 43$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	56.097288449251	$5.89736e+00$	$9.51269e-02$	n/a	0.058 sec.
4	58.605418295884	$3.38923e+00$	$5.46697e-02$	n/a	0.081 sec.
8	60.583679714760	$1.41097e+00$	$2.27596e-02$	−0.342	0.188 sec.
16	61.498580004423	$4.96070e-01$	$8.00182e-03$	−1.113	0.356 sec.
32	61.835649442526	$1.59001e-01$	$2.56475e-03$	−1.441	0.705 sec.
64	61.946393606315	$4.82566e-02$	$7.78400e-04$	−1.606	1.424 sec.
128	61.980505160503	$1.41451e-02$	$2.28166e-04$	−1.699	3.080 sec.
256	61.990601380935	$4.04885e-03$	$6.53096e-05$	−1.756	6.281 sec.
512	61.993511069104	$1.13916e-03$	$1.83751e-05$	−1.795	12.479 sec.
1024	61.994333881846	$3.16347e-04$	$5.10281e-06$	−1.822	27.159 sec.
2048	61.994563279013	$8.69496e-05$	$1.40253e-06$	−1.843	55.556 sec.
4096	61.994626529265	$2.36994e-05$	$3.82281e-07$	−1.859	112.126 sec.
8192	61.994643813834	$6.41481e-06$	$1.03474e-07$	−1.872	263.502 sec.
16384	61.994648502522	$1.72612e-06$	$2.78430e-08$	−1.882	670.459 sec.
32768	61.994649766517	$4.62124e-07$	$7.45425e-09$	−1.891	2116.209 sec.
Richardson	61.994650187848	$4.07921e-08$	$6.57994e-10$	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.70: Order ρ^6/N^2 contribution from convolution 70 of λ_5

$T_{70}/5! = \frac{v\rho^2}{N^6}, v = \left(-6528\,\zeta(6) + 2880\,(\zeta(3))^2\right)/5!$					
$v = -20.66488340954681158654133348925836886544922171945390558507261143$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−18.699096149750	$1.96579e+00$	$9.51269e-02$	n/a	0.027 sec.
4	−19.535139431961	$1.12974e+00$	$5.46697e-02$	n/a	0.030 sec.
8	−20.194559904920	$4.70324e-01$	$2.27596e-02$	−0.342	0.060 sec.
16	−20.499526668141	$1.65357e-01$	$8.00182e-03$	−1.113	0.141 sec.
32	−20.611883147509	$5.30003e-02$	$2.56475e-03$	−1.441	0.252 sec.
64	−20.648797868772	$1.60855e-02$	$7.78400e-04$	−1.606	0.538 sec.
128	−20.660168386834	$4.71502e-03$	$2.28166e-04$	−1.699	1.306 sec.
256	−20.663533793645	$1.34962e-03$	$6.53096e-05$	−1.756	2.858 sec.
512	−20.664503689701	$3.79720e-04$	$1.83751e-05$	−1.795	8.152 sec.
1024	−20.664777960615	$1.05449e-04$	$5.10281e-06$	−1.822	12.261 sec.
2048	−20.664854426338	$2.89832e-05$	$1.40253e-06$	−1.843	25.070 sec.
4096	−20.664875509755	$7.89979e-06$	$3.82281e-07$	−1.859	58.707 sec.
8192	−20.664881271278	$2.13827e-06$	$1.03474e-07$	−1.872	133.628 sec.
16384	−20.664882834174	$5.75373e-07$	$2.78430e-08$	−1.882	367.183 sec.
32768	−20.664883255506	$1.54041e-07$	$7.45425e-09$	−1.891	1403.477 sec.
Richardson	−20.664883395949	$1.35974e-08$	$6.57994e-10$	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.71: Order ρ^2/N^6 contribution from convolution 72 of λ_5

$T_{72}/5! = \frac{v\rho^2}{N^6}, v = (1708.9)\,/5!$					
$v = 14.24083333$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	13.678465164908	$5.62368e-01$	$3.94898e-02$	n/a	2.017 sec.
8	14.263397472244	$2.25641e-02$	$1.58447e-03$	n/a	4.123 sec.
16	14.410544881832	$1.69712e-01$	$1.19172e-02$	-1.991	8.963 sec.
32	14.372601509840	$1.31768e-01$	$9.25284e-03$	-1.955	19.173 sec.
64	14.315144889039	$7.43116e-02$	$5.21820e-03$	0.599	41.382 sec.
128	14.279098561857	$3.82652e-02$	$2.68701e-03$	-0.673	98.126 sec.
256	14.261677036172	$2.08437e-02$	$1.46366e-03$	-1.049	220.044 sec.
512	14.254330401915	$1.34971e-02$	$9.47772e-04$	-1.246	546.546 sec.
Richardson	14.241505696376	$6.72363e-04$	$4.72137e-05$	n/a	n/a
Error: $O\left(\text{range}^{-1.246}\right)$					

Table A.72: Order ρ^2/N^6 contribution from convolution 73 of λ_5

$T_{73}/5! = \frac{v\rho^2}{N^6}, v = (1919.8217678467)\,/5!$					
$v = 15.9985147320558$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	15.449544134647	$5.48971e-01$	$3.43138e-02$	n/a	0.816 sec.
8	15.751538329719	$2.46976e-01$	$1.54375e-02$	n/a	1.661 sec.
16	15.897387257881	$1.01127e-01$	$6.32105e-03$	-1.050	4.099 sec.
32	15.960391647095	$3.81231e-02$	$2.38291e-03$	-1.211	9.621 sec.
64	15.985059805826	$1.34549e-02$	$8.41011e-04$	-1.353	28.648 sec.
128	15.994019632050	$4.49510e-03$	$2.80970e-04$	-1.461	92.939 sec.
256	15.997097118442	$1.41761e-03$	$8.86091e-05$	-1.542	423.810 sec.
512	15.998110561338	$4.04171e-04$	$2.52630e-05$	-1.602	2442.930 sec.
Richardson	15.998585778738	$7.10467e-05$	$4.44083e-06$	n/a	n/a
Error: $O\left(\text{range}^{-1.602}\right)$					

Table A.73: Order ρ^2/N^6 contribution from convolution 74 of λ_5

$T_{74}/5! = \frac{v\rho^2}{N^6}, v = \left(1152\,(\zeta(3))^2 - 4224\,\zeta(6)\right)/5!$					
$v = -21.93904411688972107237965437209331944118011575382311410519101700$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−22.006654963704	6.76108e − 02	3.08176e − 03	n/a	0.006 sec.
4	−21.953149693616	1.41056e − 02	6.42944e − 04	n/a	0.004 sec.
8	−21.940792603206	1.74849e − 03	7.96975e − 05	−2.114	0.007 sec.
16	−21.939212439726	1.68323e − 04	7.67230e − 06	−2.967	0.013 sec.
32	−21.939058287832	1.41709e − 05	6.45923e − 07	−3.358	0.027 sec.
64	−21.939045222600	1.10571e − 06	5.03992e − 08	−3.561	0.057 sec.
128	−21.939044199276	8.23866e − 08	3.75525e − 09	−3.674	0.232 sec.
256	−21.939044122848	5.95780e − 09	2.71562e − 10	−3.743	0.465 sec.
512	−21.939044117312	4.22034e − 10	1.92367e − 11	−3.787	1.116 sec.
1024	−21.939044116919	2.94484e − 11	1.34228e − 12	−3.818	2.466 sec.
2048	−21.939044116892	2.03125e − 12	9.25861e − 14	−3.840	4.700 sec.
4096	−21.939044116890	1.38829e − 13	6.32795e − 15	−3.857	9.341 sec.
8192	−21.939044116890	9.41752e − 15	4.29258e − 16	−3.870	18.733 sec.
16384	−21.939044116890	6.34833e − 16	2.89362e − 17	−3.881	42.047 sec.
32768	−21.939044116890	4.25651e − 17	1.94015e − 18	−3.890	79.015 sec.
Richardson	−21.939044116890	4.35113e − 26	1.98328e − 27	n/a	n/a
Error: $O\left(\text{range}^{-3.890}\right)$					

Table A.74: Order ρ^2/N^6 contribution from convolution 75 of λ_5

$T_{75}/5! = \frac{v\rho^2}{N^6}, v = \left(13536\,\zeta(6) + 13824\,\zeta(2)\,\zeta(3) - 34560\,\zeta(5) - 5760\,(\zeta(3))^2\right)/5!$					
$v = -25.45059326001060623518827854285895465406974245826420358929390743$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−25.641208529205	1.90615e − 01	7.48962e − 03	n/a	0.006 sec.
4	−25.513365122634	6.27719e − 02	2.46642e − 03	n/a	0.003 sec.
8	−25.464558619743	1.39654e − 02	5.48724e − 04	−1.389	0.007 sec.
16	−25.453150530902	2.55727e − 03	1.00480e − 04	−2.097	0.019 sec.
32	−25.451013440932	4.20181e − 04	1.65097e − 05	−2.416	0.033 sec.
64	−25.450657998926	6.47389e − 05	2.54371e − 06	−2.588	0.072 sec.
128	−25.450602838872	9.57886e − 06	3.76371e − 07	−2.688	0.290 sec.
256	−25.450594639502	1.37949e − 06	5.42027e − 08	−2.750	0.760 sec.
512	−25.450593454914	1.94904e − 07	7.65812e − 09	−2.791	1.720 sec.
1024	−25.450593287159	2.71484e − 08	1.06671e − 09	−2.820	3.360 sec.
2048	−25.450593263751	3.74005e − 09	1.46953e − 10	−2.841	6.512 sec.
4096	−25.450593260521	5.10696e − 10	2.00662e − 11	−2.858	13.268 sec.
8192	−25.450593260080	6.92271e − 11	2.72006e − 12	−2.871	25.750 sec.
16384	−25.450593260020	9.32655e − 12	3.66457e − 13	−2.882	53.523 sec.
32768	−25.450593260012	1.24992e − 12	4.91118e − 14	−2.891	105.205 sec.
Richardson	−25.450593260011	1.03266e − 20	4.05752e − 22	n/a	n/a
Error: $O\left(\text{range}^{-2.891}\right)$					

Table A.75: Order ρ^2/N^6 contribution from convolution 77 of λ_5

$T_{77}/5! = \frac{v\rho^2}{N^6}, v = (-4625.2459797399881985791341932018)/5!$					
$v = -38.543716497833234988159451610015$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−34.267176613578	4.27654e + 00	1.10953e − 01	n/a	0.018 sec.
4	−36.106661088883	2.43706e + 00	6.32283e − 02	n/a	0.006 sec.
8	−37.543204037088	1.00051e + 00	2.59579e − 02	−0.357	0.015 sec.
16	−38.197616713233	3.46100e − 01	8.97941e − 03	−1.134	0.028 sec.
32	−38.434394481224	1.09322e − 01	2.83631e − 03	−1.467	0.056 sec.
64	−38.510905751881	3.28107e − 02	8.51261e − 04	−1.630	0.154 sec.
128	−38.534172429867	9.54407e − 03	2.47617e − 04	−1.717	0.551 sec.
256	−38.540998019817	2.71848e − 03	7.05297e − 05	−1.769	1.195 sec.
512	−38.542953936791	7.62561e − 04	1.97843e − 05	−1.803	2.562 sec.
1024	−38.543505108638	2.11389e − 04	5.48440e − 06	−1.827	5.233 sec.
2048	−38.543658455570	5.80423e − 05	1.50588e − 06	−1.846	11.958 sec.
4096	−38.543700686425	1.58114e − 05	4.10220e − 07	−1.860	21.551 sec.
8192	−38.543712219141	4.27869e − 06	1.11009e − 07	−1.873	42.333 sec.
16384	−38.543715346369	1.15146e − 06	2.98742e − 08	−1.883	80.331 sec.
32768	−38.543716189242	3.08591e − 07	8.00626e − 09	−1.891	158.802 sec.
Richardson	−38.543716517185	1.93518e − 08	5.02074e − 10	n/a	n/a
Error: $O\left(\text{range}^{-1.891}\right)$					

Table A.76: Order ρ^6/N^2 contribution from convolution 78 of λ_5

$T_{78}/5! = \frac{v\rho^2}{N^6}, v = \left(-13056\,\zeta(6) + 5760\,(\zeta(3))^2\right)/5!$					
$v = -41.329766819093623173082666978516737730898443438907811170145222 77$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	−37.398192299501	3.93157e + 00	9.51269e − 02	n/a	0.026 sec.
4	−39.070278863923	2.25949e + 00	5.46697e − 02	n/a	0.026 sec.
8	−40.389119809840	9.40647e − 01	2.27596e − 02	−0.342	0.059 sec.
16	−40.999053336282	3.30713e − 01	8.00182e − 03	−1.113	0.128 sec.
32	−41.223766295018	1.06001e − 01	2.56475e − 03	−1.441	0.245 sec.
64	−41.297595737543	3.21711e − 02	7.78400e − 04	−1.606	0.512 sec.
128	−41.320336773669	9.43005e − 03	2.28166e − 04	−1.699	1.378 sec.
256	−41.327067587290	2.69923e − 03	6.53096e − 05	−1.756	2.828 sec.
512	−41.329007379403	7.59440e − 04	1.83751e − 05	−1.795	5.855 sec.
1024	−41.329555921231	2.10898e − 04	5.10281e − 06	−1.822	12.845 sec.
2048	−41.329708852675	5.79664e − 05	1.40253e − 06	−1.843	29.555 sec.
4096	−41.329751019510	1.57996e − 05	3.82281e − 07	−1.859	80.991 sec.
8192	−41.329762542556	4.27654e − 06	1.03474e − 07	−1.872	255.302 sec.
16384	−41.329765668348	1.15075e − 06	2.78430e − 08	−1.882	913.390 sec.
Richardson	−41.329766710279	1.08815e − 07	2.63285e − 09	n/a	n/a
Error: $O\left(\text{range}^{-1.882}\right)$					

Table A.77: Order ρ^2/N^6 contribution from convolution 79 of λ_5

$T_{79}/5! = \frac{v\rho^2}{N^6}, v = (-4076.319)/5!$					
$v = -33.969325$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	−32.888292404058	$1.08103e+00$	$3.18238e-02$	n/a	1.109 sec.
8	−33.251967030482	$7.17358e-01$	$2.11178e-02$	n/a	2.106 sec.
16	−33.686326268748	$2.82999e-01$	$8.33101e-03$	0.256	4.379 sec.
32	−33.875343403167	$9.39816e-02$	$2.76666e-03$	−1.200	9.228 sec.
64	−33.940429373732	$2.88956e-02$	$8.50639e-04$	−1.538	18.511 sec.
128	−33.960796647701	$8.52835e-03$	$2.51060e-04$	−1.676	37.543 sec.
256	−33.966872500990	$2.45250e-03$	$7.21975e-05$	−1.745	81.189 sec.
512	−33.968634762391	$6.90238e-04$	$2.03194e-05$	−1.786	209.694 sec.
Richardson	−33.969330296440	$5.29644e-06$	$1.55918e-07$	n/a	n/a
Error: $O\left(\text{range}^{-1.786}\right)$					

Table A.78: Order ρ^2/N^6 contribution from convolution 80 of λ_5

$T_{80}/5! = \frac{v\rho^2}{N^6}, v = (2065\,\zeta(6))/5!$					
$v = 17.50677852498239561258732937515209075097710930764868837670478241$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
2	15.241004034565	$2.26577e+00$	$1.29423e-01$	n/a	0.017 sec.
4	16.380355483006	$1.12642e+00$	$6.43421e-02$	n/a	0.015 sec.
8	17.556398642886	$4.96201e-02$	$2.83434e-03$	0.046	0.030 sec.
16	17.851483543801	$3.44705e-01$	$1.96898e-02$	−1.995	0.084 sec.
32	17.775673875765	$2.68895e-01$	$1.53595e-02$	−1.961	0.223 sec.
64	17.660767166779	$1.53989e-01$	$8.79594e-03$	0.600	0.666 sec.
128	17.588675024075	$8.18965e-02$	$4.67799e-03$	−0.673	2.481 sec.
256	17.553832010921	$4.70535e-02$	$2.68773e-03$	−1.049	11.780 sec.
512	17.539138745173	$3.23602e-02$	$1.84844e-03$	−1.246	86.972 sec.
1024	17.533453940184	$2.66754e-02$	$1.52372e-03$	−1.370	378.795 sec.
Richardson	17.507633340670	$8.54816e-04$	$4.88277e-05$	n/a	n/a
Error: $O\left(\text{range}^{-1.370}\right)$					

Table A.79: Order ρ^2/N^6 contribution from convolution 81 of λ_5

$T_{81}/5! = \frac{v\rho^2}{N^6}, v = (2038.159818)/5!$					
$v = 16.98466515$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	16.444146202028	$5.40519e-01$	$3.18239e-02$	n/a	0.341 sec.
8	16.625983515241	$3.58682e-01$	$2.11180e-02$	n/a	0.623 sec.
16	16.843163134374	$1.41502e-01$	$8.33116e-03$	0.256	1.215 sec.
32	16.937671701587	$4.69934e-02$	$2.76682e-03$	-1.200	2.576 sec.
64	16.970214686866	$1.44505e-02$	$8.50795e-04$	-1.538	4.974 sec.
128	16.980398323850	$4.26683e-03$	$2.51216e-04$	-1.676	10.522 sec.
256	16.983436250495	$1.22890e-03$	$7.23535e-05$	-1.745	25.510 sec.
512	16.984317381196	$3.47769e-04$	$2.04755e-05$	-1.786	74.386 sec.
Richardson	16.984665148220	$1.77953e-09$	$1.04773e-10$	n/a	n/a
Error: $O\left(\text{range}^{-1.786}\right)$					

Table A.80: Order ρ^2/N^6 contribution from convolution 82 of λ_5

$T_{82}/5! = \frac{v\rho^2}{N^6}, v = (240.42)/5!$					
$v = 2.0035$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	0.268236236742	$1.73526e+00$	$8.66116e-01$	n/a	0.000 sec.
8	1.808406665877	$1.95093e-01$	$9.73763e-02$	n/a	0.000 sec.
16	2.255171714306	$2.51672e-01$	$1.25616e-01$	-1.786	0.000 sec.
32	2.235317867984	$2.31818e-01$	$1.15706e-01$	-4.492	0.000 sec.
64	2.139102761494	$1.35603e-01$	$6.76829e-02$	2.277	0.000 sec.
128	2.071781601308	$6.82816e-02$	$3.40812e-02$	-0.515	0.000 sec.
256	2.038380249006	$3.48802e-02$	$1.74097e-02$	-1.011	0.000 sec.
512	2.024109313127	$2.06093e-02$	$1.02867e-02$	-1.227	0.000 sec.
1024	2.018545574235	$1.50456e-02$	$7.50965e-03$	-1.359	0.000 sec.
Richardson	2.003493769215	$6.23079e-06$	$3.10995e-06$	n/a	n/a
Error: $O\left(\text{range}^{-1.359}\right)$					

A.5.9 Lambda 5 Term 82 by Element

By order analysis it can be determined that only the surf_4 component of f_4 contributes to term T_{82} of λ_5. The surface can be broken up into multiple parts. The only components that need to be evaluated are

$$T_{82} = 4\rho\sqrt{\pi}\text{conv}\left(c_0, f_{4,dr}\right)(0) \tag{A.260}$$

$$=4\rho\sqrt{\pi}\sum_{k=-\infty}^{\infty} c_0(k)\text{surf}_4(-k)\frac{J'_{|kN|}(\rho)\rho}{J_{|kN|}(\rho)} \tag{A.261}$$

Each term in $\text{surf}_4(k)$ can be evaluated individually. The sub convolutions are given below. Each table shows the result of computing

$$4\rho\sqrt{\pi}\sum_{k=-\infty}^{\infty} c_0(k)Q_i(-k)\frac{J'_{|kN|}(\rho)\rho}{J_{|kN|}(\rho)} \tag{A.262}$$

$$Q_1|\text{surf}_4(k) = 4\text{conv}\left(c_0, c_0, c_{0,d\theta}, f_{1,drd\theta}\right)(k) \tag{A.263}$$

$$Q_2|\text{surf}_4(k) = 4\text{conv}\left(c_0, c_0, c_{0,d\theta}, f_{1,drd\theta}\right)(k) \tag{A.264}$$

$$Q_3|\text{surf}_4(k) = -4\text{conv}\left(c_0, c_0, c_0, f_{1,dr^3}\right)(k) \tag{A.265}$$

$$Q_4|\text{surf}_4(k) = -6\text{conv}\left(c_0, c_0, f_{2,dr^2}\right)(k) \tag{A.266}$$

$$Q_5|\text{surf}_4(k) = -4\text{conv}\left(c_0, f_{3,dr}\right)(k) \tag{A.267}$$

$$Q_6|\text{surf}_4(k) = 6\text{conv}\left(c_0, c_{0,d\theta}, f_{2,d\theta}\right)(k) \tag{A.268}$$

$$Q_7|\text{surf}_4(k) = 8\text{conv}\left(c_0, c_0, c_{0,d\theta}, f_{1,drd\theta}\right)(k) \tag{A.269}$$

$$Q_8|\text{surf}_4(k) = 4\text{conv}\left(c_0, c_{0,d\theta}, c_{0,d\theta}, f_{1,dr}\right)(k) \tag{A.270}$$

Table A.81: Order ρ^2/N^6 contribution from convolution Q_1 of T_{82} of λ_5

$T_{Q_1}/5! = \frac{v\rho^2}{N^6}, v = \left(-960\ (\zeta(3))^2 + 1392\,\zeta(6)\right)/5!$					
$v = 0.24165315$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	0.369124755200	$1.27472e-01$	$5.27498e-01$	n/a	0.436 sec.
8	0.277887683000	$3.62345e-02$	$1.49944e-01$	n/a	0.706 sec.
16	0.251306808600	$9.65366e-03$	$3.99484e-02$	-1.779	1.247 sec.
32	0.244143969200	$2.49082e-03$	$1.03074e-02$	-1.892	2.524 sec.
64	0.242285702700	$6.32553e-04$	$2.61761e-03$	-1.947	6.007 sec.
128	0.241812518300	$1.59368e-04$	$6.59492e-04$	-1.973	11.445 sec.
256	0.241693134400	$3.99844e-05$	$1.65462e-04$	-1.987	26.365 sec.
512	0.241663151800	$1.00018e-05$	$4.13891e-05$	-1.993	52.259 sec.
1024	0.241655639100	$2.48910e-06$	$1.03003e-05$	-1.997	82.802 sec.
Richardson	0.241653131300	$1.87000e-08$	$7.73836e-08$	n/a	n/a
Error: $O\left(\text{range}^{-1.997}\right)$					

Table A.82: Order ρ^2/N^6 contribution from convolution Q_2 of T_{82} of λ_5

$T_{Q_2}/5! = \frac{v\rho^2}{N^6}, v = \left(2880\ (\zeta(3))^2 - 2304\,\zeta(2)\,\zeta(3) - 5040\,\zeta(6) + 5760\,\zeta(5)\right)/5!$					
$v = 3.75845924$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	3.632883263000	$1.25576e-01$	$3.34116e-02$	n/a	0.487 sec.
8	3.722392167000	$3.60671e-02$	$9.59624e-03$	n/a	0.723 sec.
16	3.748817808000	$9.64143e-03$	$2.56526e-03$	-1.760	1.356 sec.
32	3.755969261000	$2.48998e-03$	$6.62500e-04$	-1.886	2.775 sec.
64	3.757826762000	$6.32478e-04$	$1.68281e-04$	-1.945	5.325 sec.
128	3.758299897000	$1.59343e-04$	$4.23958e-05$	-1.973	10.606 sec.
256	3.758419278000	$3.99620e-05$	$1.06325e-05$	-1.987	22.078 sec.
512	3.758449260000	$9.98000e-06$	$2.65534e-06$	-1.993	47.574 sec.
1024	3.758456772000	$2.46800e-06$	$6.56652e-07$	-1.997	105.261 sec.
Richardson	3.758459280000	$4.00000e-08$	$1.06427e-08$	n/a	n/a
Error: $O\left(\text{range}^{-1.997}\right)$					

Table A.83: Order ρ^2/N^6 contribution from convolution Q_3 of T_{82} of λ_5

$T_{Q_3}/5! = \frac{v\rho^2}{N^6}, v = (-2312.6229898281276944790484243\,44)/5!$					
$v = -19.27185825$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	-19.544160820000	$2.72303e-01$	$1.41295e-02$	n/a	2.289 sec.
8	-19.335929820000	$6.40716e-02$	$3.32462e-03$	n/a	6.294 sec.
16	-19.289446870000	$1.75886e-02$	$9.12658e-04$	-2.163	22.680 sec.
32	-19.276973710000	$5.11546e-03$	$2.65437e-04$	-1.898	92.272 sec.
64	-19.273320390000	$1.46214e-03$	$7.58692e-05$	-1.772	332.928 sec.
Richardson	-19.271389230000	$4.69020e-04$	$2.43370e-05$	n/a	n/a
Error: $O\left(\text{range}^{-1.772}\right)$					

Table A.84: Order ρ^2/N^6 contribution from convolution Q_4 of T_{82} of λ_5

$T_{Q_4}/5! = \frac{v\rho^2}{N^6}, v = (2065\,\zeta(6))/5!$					
$v = 17.50677854$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	16.380355480000	$1.12642e+00$	$6.43421e-02$	n/a	0.046 sec.
8	17.556398640000	$4.96201e-02$	$2.83434e-03$	n/a	0.102 sec.
16	17.851483540000	$3.44705e-01$	$1.96898e-02$	-1.995	0.080 sec.
32	17.775673880000	$2.68895e-01$	$1.53595e-02$	-1.961	0.175 sec.
64	17.660767170000	$1.53989e-01$	$8.79594e-03$	0.600	0.518 sec.
128	17.588675020000	$8.18965e-02$	$4.67799e-03$	-0.673	1.473 sec.
256	17.553832010000	$4.70535e-02$	$2.68773e-03$	-1.049	7.170 sec.
512	17.539138740000	$3.23602e-02$	$1.84844e-03$	-1.246	67.915 sec.
1024	17.533453940000	$2.66754e-02$	$1.52372e-03$	-1.370	251.376 sec.
Richardson	17.507633340000	$8.54800e-04$	$4.88268e-05$	n/a	n/a
Error: $O\left(\text{range}^{-1.370}\right)$					

Table A.85: Order ρ^2/N^6 contribution from convolution Q_5 of T_{82} of λ_5

$T_{Q_5}/5! = \frac{v\rho^2}{N^6}, v = (392.79369707470334264633169794 30)/5!$					
$v = 3.273280809$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	3.155210682000	$1.18070e-01$	$3.60709e-02$	n/a	0.805 sec.
8	3.249458923000	$2.38219e-02$	$7.27768e-03$	n/a	1.108 sec.
16	3.269072077000	$4.20873e-03$	$1.28578e-03$	-2.265	1.950 sec.
32	3.272657467000	$6.23342e-04$	$1.90433e-04$	-2.452	4.152 sec.
64	3.273216367000	$6.44420e-05$	$1.96873e-05$	-2.681	8.354 sec.
128	3.273282303000	$1.49400e-06$	$4.56423e-07$	-3.083	18.181 sec.
256	3.273284336000	$3.52700e-06$	$1.07751e-06$	-5.019	40.246 sec.
512	3.273282227000	$1.41800e-06$	$4.33205e-07$	0.053	112.137 sec.
1024	3.273281251000	$4.42000e-07$	$1.35033e-07$	-1.112	590.358 sec.
Richardson	3.273352974000	$7.21650e-05$	$2.20467e-05$	n/a	n/a
Error: $O\left(\text{range}^{-1.112}\right)$					

Table A.86: Order ρ^2/N^6 contribution from convolution Q_6 of T_{82} of λ_5

$T_{Q_6}/5! = \frac{v\rho^2}{N^6}, v = \left(-1920\,(\zeta(3))^2 + 2784\,\zeta(6)\right)/5!$					
$v = 0.48330629$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	0.738249510500	$2.54943e-01$	$5.27498e-01$	n/a	0.408 sec.
8	0.555775366000	$7.24691e-02$	$1.49944e-01$	n/a	0.660 sec.
16	0.502613617200	$1.93073e-02$	$3.99484e-02$	-1.779	1.169 sec.
32	0.488287938300	$4.98165e-03$	$1.03074e-02$	-1.892	2.374 sec.
64	0.484571405400	$1.26512e-03$	$2.61763e-03$	-1.947	4.787 sec.
128	0.483625036600	$3.18747e-04$	$6.59513e-04$	-1.973	9.258 sec.
256	0.483386268800	$7.99788e-05$	$1.65483e-04$	-1.987	22.321 sec.
512	0.483326303700	$2.00137e-05$	$4.14100e-05$	-1.993	47.486 sec.
1024	0.483311278200	$4.98820e-06$	$1.03210e-05$	-1.997	90.698 sec.
Richardson	0.483306263200	$2.68000e-08$	$5.54514e-08$	n/a	n/a
Error: $O\left(\text{range}^{-1.997}\right)$					

Table A.87: Order ρ^2/N^6 contribution from convolution Q_7 of T_{82} of λ_5

$T_{Q_7}/5! = \frac{v\rho^2}{N^6}, v = \left(768\ (\zeta(3))^2 - 1248\,\zeta(6)\right)/5!$					
$v = -1.332746753$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	-1.487808882000	$1.55062e-01$	$1.16348e-01$	n/a	0.583 sec.
8	-1.405858764000	$7.31120e-02$	$5.48581e-02$	n/a	0.833 sec.
16	-1.359558426000	$2.68117e-02$	$2.01176e-02$	-0.824	1.492 sec.
32	-1.341480311000	$8.73356e-03$	$6.55305e-03$	-1.357	3.114 sec.
64	-1.335414752000	$2.66800e-03$	$2.00188e-03$	-1.576	6.292 sec.
128	-1.333530930000	$7.84177e-04$	$5.88392e-04$	-1.687	12.575 sec.
256	-1.332971463000	$2.24710e-04$	$1.68607e-04$	-1.752	24.422 sec.
512	-1.332809995000	$6.32420e-05$	$4.74524e-05$	-1.793	56.691 sec.
1024	-1.332764306000	$1.75530e-05$	$1.31705e-05$	-1.821	99.592 sec.
Richardson	-1.332746732000	$2.10000e-08$	$1.57569e-08$	n/a	n/a
Error: $O\left(\text{range}^{-1.821}\right)$					

Table A.88: Order ρ^2/N^6 contribution from convolution Q_8 of T_{82} of λ_5

$T_{Q_8}/5! = \frac{v\rho^2}{N^6}, v = \left(1536\ (\zeta(3))^2 - 2496\,\zeta(6)\right)/5!$					
$v = -2.66549349$					
Approx Range	Result	Abs. Error	Rel. Error	Conv.	Time
4	-2.975617762000	$3.10124e-01$	$1.16348e-01$	n/a	0.508 sec.
8	-2.811717529000	$1.46224e-01$	$5.48581e-02$	n/a	0.768 sec.
16	-2.719116851000	$5.36234e-02$	$2.01176e-02$	-0.824	1.406 sec.
32	-2.682960621000	$1.74671e-02$	$6.55306e-03$	-1.357	2.941 sec.
64	-2.670829502000	$5.33601e-03$	$2.00189e-03$	-1.576	5.677 sec.
128	-2.667061859000	$1.56837e-03$	$5.88397e-04$	-1.687	10.865 sec.
256	-2.665942927000	$4.49437e-04$	$1.68613e-04$	-1.752	24.667 sec.
512	-2.665619991000	$1.26501e-04$	$4.74588e-05$	-1.793	47.677 sec.
1024	-2.665528612000	$3.51220e-05$	$1.31765e-05$	-1.821	99.352 sec.
Richardson	-2.665493462000	$2.80000e-08$	$1.05046e-08$	n/a	n/a
Error: $O\left(\text{range}^{-1.821}\right)$					

A.6 High Order N Values

The values of λ_1 and λ_2 can be computed exactly up to arbitrary orders of N. This is used to look for a pattern in the ρ components. It can be seen from these expansions that a new ρ order is added for every odd N order starting with 3.

$$\begin{aligned}
\frac{\lambda_1}{\rho^2} =& \frac{4\zeta(2)}{N^2} + \frac{6\zeta(6)}{N^6} + \frac{40}{2}\frac{\zeta(8)}{N^8} + \frac{36\zeta(10)}{N^{10}} \\
&+ \frac{70980}{691}\frac{\zeta(12)}{N^{12}} + \frac{107533}{35}\frac{\zeta(14)}{N^{14}} + \frac{3445560}{3617}\frac{\zeta(16)}{N^{16}} \\
&+ \frac{133290276}{43867}\frac{\zeta(18)}{N^{18}} + \frac{12118908560}{1222277}\frac{\zeta(20)}{N^{20}} + O\left(\frac{1}{N^{21}}\right)
\end{aligned} \tag{A.271}$$

$$\begin{aligned}
\frac{\lambda_2}{\rho^2} =& 8\,\frac{\zeta(3)}{N^3} + 64\,\frac{\zeta(4)}{N^4} + 32\,\frac{\zeta(2)\,\zeta(3)}{N^5} - 80\,\frac{\zeta(5)}{N^5} - 4\,\frac{\rho^2\zeta(5)}{N^5} \\
&- 12\,\frac{\zeta(6)}{N^6} + 4\,\frac{\rho^2\zeta(6)}{N^6} + 1176\,\frac{\zeta(7)}{N^7} - 896\,\frac{\zeta(2)\,\zeta(5)}{N^7} \\
&+ 272\,\frac{\zeta(4)\,\zeta(3)}{N^7} - 16\,\frac{\rho^2\zeta(2)\,\zeta(5)}{N^7} + 36\,\frac{\rho^2\zeta(7)}{N^7} - \frac{\rho^4\zeta(7)}{N^7} \\
&+ 120\,\frac{\zeta(8)}{N^8} - \frac{28}{3}\,\frac{\rho^2\zeta(8)}{N^8} + 4\,\frac{\rho^4\zeta(8)}{N^8} + 1488\,\frac{\zeta(6)\,\zeta(3)}{N^9} \\
&- 14240\,\frac{\zeta(4)\,\zeta(5)}{N^9} + 24864\,\frac{\zeta(2)\,\zeta(7)}{N^9} - 27040\,\frac{\zeta(9)}{N^9} \\
&+ 432\,\frac{\rho^2\zeta(2)\,\zeta(7)}{N^9} - 552\,\frac{\rho^2\zeta(9)}{N^9} - 136\,\frac{\rho^2\zeta(4)\,\zeta(5)}{N^9} \\
&- \frac{\rho^4\zeta(9)}{N^9} - 4\,\frac{\rho^4\zeta(2)\,\zeta(7)}{N^9} - 1/2\,\frac{\rho^6\zeta(9)}{N^9} + \frac{1216}{5}\,\frac{\zeta(10)}{N^{10}} \\
&- \frac{56}{5}\,\frac{\rho^2\zeta(10)}{N^{10}} + \frac{62}{5}\,\frac{\rho^4\zeta(10)}{N^{10}} + 4\,\frac{\rho^6\zeta(10)}{N^{10}} + 948904\,\frac{\zeta(11)}{N^{11}} \\
&+ \frac{22112}{3}\,\frac{\zeta(8)\,\zeta(3)}{N^{11}} + 641904\,\frac{\zeta(4)\,\zeta(7)}{N^{11}} - 123744\,\frac{\zeta(6)\,\zeta(5)}{N^{11}} \\
&- 927232\,\frac{\zeta(2)\,\zeta(9)}{N^{11}} + 6984\,\frac{\rho^2\zeta(4)\,\zeta(7)}{N^{11}} + 12968\,\frac{\rho^2\zeta(11)}{N^{11}} \\
&- 12000\,\frac{\rho^2\zeta(2)\,\zeta(9)}{N^{11}} - 744\,\frac{\rho^2\zeta(6)\,\zeta(5)}{N^{11}} - 94\,\frac{\rho^4\zeta(11)}{N^{11}} \\
&- 34\,\frac{\rho^4\zeta(4)\,\zeta(7)}{N^{11}} + 68\,\frac{\rho^4\zeta(2)\,\zeta(9)}{N^{11}} - 15\,\frac{\rho^6\zeta(11)}{N^{11}} \\
&- 2\,\frac{\rho^6\zeta(2)\,\zeta(9)}{N^{11}} - 1/4\,\frac{\rho^8\zeta(11)}{N^{11}} + O\left(\frac{1}{N^{12}}\right)
\end{aligned} \tag{A.272}$$

Vita

Mark Boady was born December 24, 1982 in the Roxborough area of Philadelphia, PA. He graduated with a Bachelor of Science degree from Drexel University in 2006. He received his Masters in 2012 and Doctorate in 2016.

He is currently a Professor of Computer Science at Drexel University.

His article "Boundary Variation of Poisson's Equation: A Model Problem For Symbolic Calculus of Moving Surfaces" was published in the International Journal of Applied Mathematics and Computer Science. The article "A Term Rewrite System for the Calculus of Moving Surfaces" was presented at the International Symposium on Symbolic and Algebraic Computation.

He lives in Perkasie with his wife, Petrina. They have two cats, two chinchillas, and a goldfish.