A Term Rewriting System for the Calculus of Moving Surfaces

Mark Boady
Department of Computer Science
Drexel University
3141 Chestnut St.
Philadelphia, PA 19104, USA
mwb33@drexel.edu

Pavel Grinfeld
Department of Mathematics
Drexel University
Korman Center
33rd and Market Streets
Philadelphia, PA 19104, USA
pg77@drexel.edu

Jeremy Johnson
Department of Computer Science
Drexel University
3141 Chestnut St.
Philadelphia, PA 19104, USA
jjohnson@cs.drexel.edu

ABSTRACT

The calculus of moving surfaces (CMS) is an analytic framework that extends the tensor calculus to deforming manifolds. We have applied the CMS to a number of boundary variation problems using a Term Rewrite System (TRS). The TRS is used to convert the initial CMS expression into a form that can be evaluated. The CMS produces expressions that are true for all coordinate spaces. This makes it very powerful but applications remain limited by a rapid growth in the size of expressions. We have extended results on existing problems to orders that had been previously intractable. In this paper, we describe our TRS and our method for evaluating CMS expressions on a specific coordinate system. Our work has already provided new insight into problems of current interest to researchers in the CMS.

Categories and Subject Descriptors

I.1 [**SYMBOLIC AND ALGEBRAIC MANIPULATION**]: Simplification of expressions; I.1.3 [**SYMBOLIC AND ALGEBRAIC MANIPULATION**]: Languages and Systems—*Special-purpose algebraic systems*

Keywords

Calculus of Moving Surfaces, Tensor Analysis, Term Rewrite Systems

1. INTRODUCTION

In recent papers, [3] and [4], we have examined boundary variation problems using the calculus of moving surfaces (CMS). The CMS provided valuable insight into these problems. The results provided in our previous publications, [3] and [4], were derived by using a custom Term Rewrite System (TRS) and evaluated in Maple [24]. The TRS was treated as a black box in those papers. Here we detail its implementation.

ISSAC'13, June 26–29, 2013, Boston, Massachusetts, USA.

In 2004, Grinfeld and Strang [10] posed the following boundary problem. *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?* More precisely, consider a regular polygon Ω_N with N sides inscribed into a unit circle Ω_∞. We view Ω_N as a boundary perturbation of Ω_∞ and ask the question of the relationship between the radial eigenvalues $\lambda_{\infty,n}$ on the unit circle and the corresponding eigenvalues $\lambda_{N,n}$ on the polygon. In [10], the idea of expressing $\lambda_{N,n}$ as a series in $1/N$ was put forth and in [11] the first several terms were computed using the calculus of moving surfaces.

Determining additional terms in this series was restricted by the rapid growth in complexity. A recipe for calculating terms was known, but the process is error-prone when done by hand and quickly becomes intractable. This problem is one of the motivations for our TRS. The TRS automates the derivation and assists the final evaluation. Our automated tools found an error in the fourth term of the hand calculation in the series expansion in [11] and was used to compute additional terms.

A model CMS problem was examined in [4]. We analyzed the evolution of Poisson's equation under arbitrary smooth deformations of the domain. Specifically, an approximate solution to $\Delta u = 1$ on a regular N-sided polygon was found. We also derived a partial series in $1/N$ for the Poisson energy E_N. This series was used to study the asymptotic behavior as $N \to \infty$.

In [3], we examined a classical question in boundary variations [17] that had been established correctly to second order. The seminal work [17] gives a third order expression. We obtained a partial series for the Laplace eigenvalues on an ellipse. The deforming surface was an expanding ellipse $\Omega(t)$ with semi-axes $A = 1$ and $B = 1 + t$. We determined the first seven Taylor terms of the series. The sixth term required calculating the sum of 11,024 terms and the seventh included 115,249 terms. This rapid increase in the number of terms is a consistent feature of high order perturbation problems. Many problems can be described by the CMS, but their calculation quickly becomes intractable.

Equations from the CMS were translated into directional rewrite rules. The direction of each rule was selected to provide results that can be evaluated, most importantly removal of the $\frac{\delta}{\delta t}$-derivative, the central differential operator in the CMS. Our rule set leads to normal forms that can be evaluated. The normal form expression is true for any coordinate system. The evaluation is performed by generating Maple code in a specific coordinate system. We demo how these rules are applied to an example problem in Section 7. Given a circle being stretched into an ellipse at rate c, determine an expression for the contour length of the shape in terms of c.

We begin with background information in Section 2. The formal language of the CMS and the signature of our TRS are given in Section 3. After listing our reduction rules in Section 4 and showing that they produce a unique normal form in Section 5, we describe the implementation in Section 6. Finally, we detail the solution to our example problem in Section 7 and describe our applications in Section 8.

2. BACKGROUND

The CMS is an extension of tensor calculus to support deforming manifolds. To our knowledge, existing algebraic packages do not address moving surfaces. A key feature of the tensor calculus is that its expressions can be evaluated in any coordinate system. A general purpose computer algebra system can be used if a problem is restricted to a specific coordinate system.

A TRS is a computational model for equational reasoning. The TRS has two components. A signature describes the language of the TRS. The signature contains all the function symbols that are used to generate terms. The TRS also contains a set of directional rewrite rules. Each rewrite rule follows the form $l \to r$. When a term is matched against the l pattern it is rewritten or reduced to the r pattern. Reduction continues until the term can no longer be matched to the left side of any rules. A term that does not match any rules is called a normal form. A normal form is the result of applying the TRS to an input term. The TRS provides an important method for algebraic simplification [1]. It can also automate mathematical formalism outside of general purpose computer algebra systems. A survey on the theory of TRSs can be found in [16].

Although no computer algebra systems have been developed for the CMS, software exists for tensor calculus. These packages originated in the theory of relativity and do not implement stationary manifolds, let alone the CMS. The two packages which closely resemble our TRS are MathTensor [26] and Cadabra [28]. In addition to rewriting, the symbolic manipulation of tensor indexes is also a challenging problem [25]. Other symbolic manipulation packages focusing on relativity have been developed.

These packages have proven quite successful in their fields, typically relativity. MathTensor has supported research in general relativity, such as [29] and [19]. It has also been successful in quantum field theory [13]. Cadabra is also used in general relativity [5, 23].

These packages could be modified to solve our example problem in Section 7, but the extensions needed for [3] and [4] would require fundamental changes to the basic data structures. This provided a motivation to begin with a new TRS instead of extending existing ones.

3. CALCULUS OF MOVING SURFACES

The signature of our TRS is derived from the formalism of the CMS. In this section, we describe the language of the CMS. We restrict our view to those objects and functions required for our model problems. The CMS has been described in great detail in earlier publications [12, 8]. The CMS, deeply rooted in tensor calculus [21], [27], [31], was originated by Jacques Hadamard. A historical review of the CMS can be found in [7].

In Section 7, we will use the CMS to examine a circle being stretched into a ellipse. At time zero, the unit circle has a contour length of 2π. We stretch the horizontal axis of the circle to $1 + ct$ to create an ellipse. Our goal is to create a Taylor series, in terms of c, for the contour length at time $t = 1$. For this simple problem, a series can be calculated without the CMS. The beauty of the CMS is its generality. The CMS expressions we derive are correct for the contour length of any deforming manifold.

The fundamental object of the CMS is the tensor. A tensor is a geometric field defining a linear and homogeneous transformation [30]. In the TRS, a tensor is a named value and has a set of properties. The tensor can be a spacial tensor, existing in the entire space, or a surface tensor, restricted to a manifold. The tensor also has an ordered list of named indexes. Each index can be a spacial or surface index. The index is either a contravariant or covariant index. This property describes how the tensor changes with respect to a change in coordinate systems [30].

We give two example tensors to clarify this description $X^{i}_{\cdot j}$ and Y_{α}. The symbol $X^{i}_{\cdot j}$ is a spacial tensor named X. It has two indexes. The first index is named i and is a contravariant space index. The second index, j, is a covariant space index. Lower case Roman letters are used for space indexes and lower case Greek letters for surface indexes. Contravariant indexes are upper indexes. Covariant indexes are lower indexes. The indexes are a single ordered list. A dot is used to show that j is after i in $X^{i}_{\cdot j}$. This will be done when the position of indexes is not clear. The second tensor, Y_{α}, is a surface tensor with one covariant surface index named α.

The indexes of the tensor are ordered. Returning to the previous examples, $X^{i}_{\cdot j}$ is a two dimensional system and Y_{α} is a one dimensional system. To give explicit values, a coordinate system must be selected. The number of elements in each dimension is determined by the dimensions of the space.

We now give the tensor objects needed to describe our TRS. The covariant metric tensor, Z_{ij}, describes the ambient coordinate space.

The surface is described by its own coordinate system. The shift tensor $Z^{i}_{\cdot\alpha}$ selects the surface part of a space tensor. Multiplication is the outer product. Contraction is a summation over a pair of indexes, shown by a repeated index. The covariant or contravariant property of an index can be changed by multiplication and contraction with the metric tensor. Raising or lowering an index is called index juggling. When contracting an index, the contracted name must appear exactly twice, once as a covariant index and once as a contravariant index. Both indexes must be of the same type, space or surface. This notation provides a shorthand for the summation $Z_{ki}Z^{k}_{\cdot\alpha} = \sum_{k=\ldots}\left(Z_{ki}Z^{k}_{\cdot\alpha}\right)$.

The surface metric tensors are defined by the shift tensor $S_{\alpha\beta} = Z^{i\alpha}Z_{i\beta}$ and it's inverse $S^{\alpha\beta}$.The surface normal is given by N^{i}. The curvature tensor is B^{α}_{β} and its trace B^{α}_{α} is mean curvature.

We can take derivatives with respect to the space and surface coordinate. The covariant surface derivative is a function of one input ∇_{α}. Contravariant and spacial derivatives

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
$\frac{\delta}{\delta t}$	$\delta/\delta t$-derivative
$\frac{\partial}{\partial t}$	Partial Time Derivative

Table 1: TRS Signature

all exist and are defined by the index of the ∇ symbol. The mechanics can be found in [30].

The CMS describes surfaces that deform over time. The surface velocity C is the rate of deformation of the surface in the normal direction. Tensors on the surface now changes as the surface changes. We can study how these fields change over time by using the $\frac{\delta}{\delta t}$-derivative. These functions are described in [7]. The mean curvature of an ellipse can be described with respect to time, and the derivative taken directly. The curvature of a red blood cell is more difficult to describe and evaluate [22].

We introduce the commutator tensor to facilitate switching the order of $\frac{\delta}{\delta t}$ and ∇_α [7]. It is a shorthand for the following:

$$R^{\gamma}_{.\alpha\beta} = \nabla^{\gamma}(CB_{\alpha\beta}) - \nabla^{\alpha}(CB^{\gamma}_{\beta}) - \nabla_{\beta}(CB^{\gamma}_{\alpha}) \tag{1}$$

The signature of our TRS is summarized in Table 1. Tensors that are only required for the evaluation of expressions are not included. Transformations given by index juggling are implemented but not explicitly listed. The integers and rationals are implemented but not listed in the table.

4. REWRITE RULES

The majority of our rewrite rules are related to derivative calculations. For the reduction rules, 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 $x_1, x_2, \cdots, x_n$.

The covariant and contravariant derivatives are defined by rules (2) to (5). These rules are true for any index of the derivative. Only the rules for ∇_α are shown. These rules are repeated for the other indexes of the derivative. The summation symbol is used to show a contraction summation in rule (5).

$$\nabla_\alpha(FG) \to G\nabla_\alpha(F) + F\nabla_\alpha(G) \tag{2}$$
$$\nabla_\alpha(x_1) \to 0 \tag{3}$$
$$\nabla_\alpha(F+G) \to \nabla_\alpha(F) + \nabla_\alpha(G) \tag{4}$$
$$\nabla_\alpha(\sum_i F^{\cdots i \cdots}_{\cdots i \cdots}) \to \sum_i \nabla_\alpha(F^{\cdots i \cdots}_{\cdots i \cdots}) \tag{5}$$

The $\frac{\delta}{\delta t}$-derivative is at the heart of the CMS. Calculating expressions using this derivative was the original motivation for our TRS. Although a mechanism for evaluating this derivative exists, it is challenging to evaluate. In [22], the Helfrich energy governing the shape of a red blood cell is examined. The fourth order derivative is essential to its analysis, but remains an open problem for a distorted torus. The application of the $\frac{\delta}{\delta t}$-derivative to the problem in [10] becomes intractable because some fields are described by infinite Fourier series. Our target normal form is an expression where $\frac{\delta}{\delta t}$-derivatives can be evaluated. The only appearance of the $\frac{\delta}{\delta t}$-derivative is when it is applied to the Surface Velocity or Commutator tensors.

First, the differentiation table for specific tensor objects is given.

$$\frac{\delta B^{\alpha}_{.\beta}}{\delta t} \to \nabla^{\alpha}\nabla_{\beta}C + CB^{\alpha}_{.\gamma}B^{\gamma}_{.\beta} \tag{6}$$
$$\frac{\delta B^{\alpha\beta}}{\delta t} \to \nabla^{\alpha}\nabla^{\beta}C + 3CB^{\alpha}_{.\gamma}B^{\gamma\beta} \tag{7}$$
$$\frac{\delta B_{\alpha\beta}}{\delta t} \to \nabla_{\alpha}\nabla_{\beta}C - CB_{\alpha\gamma}B^{\gamma}_{.\beta} \tag{8}$$
$$\frac{\delta N^i}{\delta t} \to -Z^i_{.\alpha}\nabla^{\alpha}C \tag{9}$$
$$\frac{\delta N_i}{\delta t} \to -Z_{i\alpha}\nabla^{\alpha}C \tag{10}$$
$$\frac{\delta Z^i_{.\alpha}}{\delta t} \to N^i\nabla_{\alpha}C - CZ^i_{.\beta}B^{\beta}_{.\alpha} \tag{11}$$
$$\frac{\delta Z_{i\alpha}}{\delta t} \to N_i\nabla_{\alpha}C - CZ_{i\beta}B^{\beta}_{.\alpha} \tag{12}$$
$$\frac{\delta Z^{i\alpha}}{\delta t} \to N^i\nabla^{\alpha}C - CZ^i_{.\beta}B^{\beta\alpha} \tag{13}$$
$$\frac{\delta Z_i^{.\alpha}}{\delta t} \to N_i\nabla^{\alpha}C - CZ_{i\beta}B^{\beta\alpha} \tag{14}$$
$$\frac{\delta C^{x_1}}{\delta t} \to x_1 C^{x_1-1}\frac{\delta C}{\delta t} \tag{15}$$
$$\frac{\delta x_1}{\delta t} \to 0 \tag{16}$$

The $\frac{\delta}{\delta t}$-derivative commutes with contraction and satisfies the sum and the product rules. The summation symbol is again used to show a contraction.

$$\frac{\delta \sum_i F^{\cdots i \cdots}_{\cdots i \cdots}}{\delta t} \to \sum_i \frac{\delta F^{\cdots i \cdots}_{\cdots i \cdots}}{\delta t} \tag{17}$$
$$\frac{\delta FG}{\delta t} \to G\frac{\delta F}{\delta t} + F\frac{\delta G}{\delta t} \tag{18}$$
$$\frac{\delta (F+G)}{\delta t} \to \frac{\delta F}{\delta t} + \frac{\delta G}{\delta t} \tag{19}$$

Reordering the $\frac{\delta}{\delta t}$-derivative 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.

$$\frac{\delta \nabla_\alpha A}{\delta t} \rightarrow \nabla_\alpha \frac{\delta A}{\delta t} \tag{20}$$

$$\frac{\delta \nabla_\alpha A^\beta}{\delta t} \rightarrow\rightarrow \nabla_\alpha \frac{\delta A^\beta}{\delta t} + R^{\beta}_{.\alpha\gamma} A^\gamma \tag{21}$$

$$\frac{\delta \nabla_\alpha A_\beta}{\delta t} \rightarrow\rightarrow \nabla_\alpha \frac{\delta A_\beta}{\delta t} - R^{\gamma}_{.\alpha\beta} A_\gamma \tag{22}$$

$$\frac{\delta \nabla^\alpha A}{\delta t} \rightarrow \nabla^\alpha \frac{\delta A}{\delta t} + 2CB^{\alpha\gamma}\nabla_\gamma A \tag{23}$$

$$\frac{\delta \nabla^\alpha A^\beta}{\delta t} \rightarrow \nabla^\alpha \frac{\delta A^\beta}{\delta t} + 2CB^{\alpha\gamma}\nabla_\gamma A^\beta + R^{\beta\alpha}_{..\gamma} A^\gamma \tag{24}$$

$$\frac{\delta \nabla^\alpha A_\beta}{\delta t} \rightarrow \nabla^\alpha \frac{\delta A_\beta}{\delta t} + 2CB^{\alpha\gamma}\nabla_\gamma A_\beta - R^{\gamma\alpha}_{..\beta} A_\gamma \tag{25}$$

For each index in the A tensor, a new $R^{:::}_{:::}A^{:::}_{:::}$ term is added. The form and contraction are defined by the index. The four possible terms are shown. The sign of the term is determined by the contracted index's status as covariant or contravariant.

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

$$\frac{\partial FG}{\partial t} \rightarrow F\frac{\partial G}{\partial t} + G\frac{\partial F}{\partial t} \tag{26}$$

$$\frac{\partial x_1}{\partial t} \rightarrow 0 \tag{27}$$

$$\frac{\partial (F+G)}{\partial t} \rightarrow \frac{\partial F}{\partial t} + \frac{\partial G}{\partial t} \tag{28}$$

$$\frac{\partial \sum_i F^{\cdots i \cdots}_{\cdots i \cdots}}{\partial t} \rightarrow \sum_i \frac{\partial F^{\cdots i \cdots}_{\cdots i \cdots}}{\partial t} \tag{29}$$

$$\frac{\partial \nabla_\alpha F}{\partial t} \rightarrow \nabla_\alpha \frac{\partial F}{\partial t} \tag{30}$$

Rule (30) is true for all index variations of the derivative.

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

$$A^{x_1}A^{x_2} \rightarrow A^{x_1+x_2} \tag{31}$$

$$A + 0 \rightarrow A \tag{32}$$

$$A(F+G) \rightarrow AF + AG \tag{33}$$

$$0\,A \rightarrow 0 \tag{34}$$

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 outside the TRS. We store the terms as an expression tree and implement tree matching algorithms to combine terms.

5. CONFLUENCE AND TERMINATION

For successful application of our TRS, it is crucial that expressions only contain symbols that can be evaluated. The TRS terminates when it reaches a normal form, a term that does not match any reduction rules. We oriented our rules so that if a normal form is reached it can be evaluated. Additionally, we required that the TRS provides a consistent output given a consistent input.

If a term matches more than one rule, the decision of which rule to apply could result in multiple normal forms. If a TRS is confluent, all paths that diverge will eventually converge. This property ensures that normal forms are unique. We will briefly justify that our system is confluent and terminating. A more comprehensive examination of these concepts is presented in [16].

Our TRS includes associative and commutative operators, but both these properties can be extended to TRSs modulo sets of equations as shown in [18] and [2]. This extensions means that normal forms will be unique in the equivalence class defined by the associative and commutative properties. Index juggling and renaming of contracted indexes follow the same logic as these properties, allowing them to be used as an equivalence class.

Critical pairs can be used to prove confluence in a TRS as shown in the classic work by Knuth & Bendix [20]. A terminating TRS is confluent if and only if no critical pair diverges [20, 15]. This can be extended to TRS modulo equational theories [18].

A critical pair (s,t), is a pair of terms created by a pattern that matches multiple rules. Using the tensor variables F and G we create a pattern that matches reduction rules (32) and (33). The pattern $F(G+0)$ is matched to both rules. A critical pair is created by following both possible reduction paths. First, $F(G+0) \rightarrow_{32} FG$. The second possible reduction is $F(G+0) \rightarrow_{33} FG + F0$. The critical pair is the set $(FG, FG+F0)$. The critical pair (s,t) converges if $s \twoheadrightarrow s'$, $t \twoheadrightarrow t'$, and $s' = t'$. The $\twoheadrightarrow$ notation is used to show a sequence of reductions. In this case, equals is defined specifically as syntactically equal with respect to associativity, commutativity, and index juggling. For our example critical pair, FG has no remaining reductions and $FG + F0 \rightarrow_{32} FG$. This shows that the critical pair converges to the term FG. It is easy to show that there exist no diverging critical pairs in our TRS, therefore it is confluent if it is terminating.

A TRS is terminating if there exists a well-founded ordering on the terms. Again referring to [20], a TRS is terminating if there exists a well-founded ordering $>$ such that for each reduction rule $l \rightarrow r$, $l > r$. An ordering is well-founded if it contains a minimum element. We define the ordering on our reduction rules by giving an ordering on the signature and referring to the process of Associative Path Ordering described in [2]. A termination ordering on the function symbols is given in equation (35). For readability, we have introduced symbols: x^y for exponents, $\sum$ for contraction, and * for multiplication. A constant is any function symbol that takes no inputs, therefore both 10 and B^α_β would be considered constants for this ordering.

$$\frac{\delta}{\delta t} > \frac{\partial}{\partial t} > \nabla > * > x^y > + > \sum > Constant \tag{35}$$

We now compare the left and right hand sides of our rules using this ordering. Rules that select elements, such as rule (32), or reduce to a constant, rule (16), are trivially ordered.

The remaining rules are ordered using a recursive examination of the function symbols. As an example, we examine rule (20). The outermost function symbol for this rule is decreasing by the ordering from equation (35), $\frac{\delta}{\delta t} > \nabla$. In addition, we must recursively compare the inputs to the function on the right hand side. The rule is not decreasing if a larger element has been created as a subterm of the right hand side. This means that the next test is between $\frac{\delta \nabla_\alpha A}{\delta t}$ and $\frac{\delta A}{\delta t}$. In this recursive test, the outermost function symbols are equal and $\nabla_\alpha A > A$. This proves that the entire reduction rule is decreasing with respect to our ordering.

In rule (21), the same process is used. The outermost level is decreasing, $\frac{\delta}{\delta t} > +$. The first input to the addition function is less then the left side by the same process we used for rule (20). It remains to be shown that $\frac{\delta \nabla_\alpha A^\beta}{\delta t} > R^\beta_{\alpha\gamma} A^\gamma$. A continuation of the recursive process shows that this ordering is correct and the reduction is decreasing. By extension, the reduction patterns given by rules (21) to (25) are decreasing.

The process of critical pair convergence and path ordering show that our TRS is both confluent and terminating. This means that any input will lead to a unique normal form. These properties ensure our TRS produces expressions that we can evaluate.

This is true for the CMS objects for which we have defined rules. Additional objects, such as ϵ_{ijk}, a Levi-Civita Symbol, have rules that do not lead to obvious termination orderings. We will attempt the Knuth-Bendix Algorithm [20] when extending our rule set for these objects.

6. IMPLEMENTATION

We implemented our TRS in the Java programming language. In addition to providing basic TRS functionality, we also implemented automatic code generation to Maple. This allowed for the automated testing of results without an intermediate program. The TRS implemented class objects for the signature described in Section 3. Expressions were then stored in a tree structure. The rules from Section 4 were implemented to match and transform the expression tree. The associative and commutative properties of functions were handled by allowing these objects children to have an unordered set of children and flattening repeated function applications. Automating the solution to a problem such as that in Section 7 required creating a driver program to generate terms, call the TRS for reduction to normal form, and then generate Maple code. The expression tree resulting from one iteration was used to create the starting term of the next iteration.

There were a number of reasons we decided to implement a custom TRS instead of building on an existing TRS such as Maude [6] or Mathematica [32]. The symbols of the CMS are inherently objects. This meant that implementing them as objects proved more efficient then using a flat functional notation. The language of object oriented programming closely matches the native language of the CMS. The ability to use inheritance also decreased the total number of rules that needed to be implemented. For example, rules (2) to (5) were easily implemented using an inheritance model on derivatives. The sequence of rules starting with rule (20), was also simple to implement using loops and objects. These problems could all be overcome in a standard TRS framework, but we felt that these gains were significant. Additionally, using a custom TRS allowed for two additional components to be built in the same framework. To minimize the number of terms that need to be evaluated, it is convenient to combine terms using associativity, commutativity, and index juggling. The most important of these equivalence classes is the ability to exchange the contravariant and covariant property of contracted indexes. While we could implemented a secondary module for compressing the normal form by combining terms, it was helpful to do this as part of the TRS. Since the result of the TRS was already a tree structure, matching subterms was reduced to a graph matching problem. A final motivation for creating our own TRS was code generation.

To evaluate expressions, they were translated to a computer algebra system. The object oriented nature of our TRS allowed this code generation to be encapsulated by the object classes. Maple code for the expression could then be generated using a simple tree walk. We created a custom Maple library for working with CMS expressions. Maple provides a library for tensor calculus, but it does not provide support for embedded manifolds, which required more general data structures and operations. Our libraries directly implemented the textbook definitions of the objects and functions from [30] and [7].

We initialized our Maple worksheet by defining the surface and spacial coordinate system. We provided function calls to create the primary objects which are generated from the coordinate system. Using function calls, we built up the remaining objects. For example, B^α_α is created by evaluating $A^{\alpha\beta} B_{\alpha\beta}$. The Maple code for this using our library is `BAb:=contract(prod(SAB,Bab),[2,3]):`. Our Maple code can generate all the objects in our signature. We can describe our expressions using these objects and functions. The generated code allows us to evaluate our CMS expressions for a defined coordinate system.

7. CONTOUR LENGTH

In this section, we show an example of calculations performed using our TRS. We have selected a simple contour length problem. This problem is based on [4] but can be easily calculated outside the CMS.

Consider the problem of evaluating the contour length of an ellipse with semiaxes $1 + c$ and 1. It is a simple problem for which the answer is known by other means. This makes it a good test problem. The CMS approaches this problem by considering a smooth evolution from the unit circle at time $t = 0$ to the ellipse at time $t = 1$. Such an evolution may be parameterized as follows:

$$x(t, \theta) = (1 + ct) \cos\theta \tag{36}$$

$$y(t, \theta) = \sin\theta \tag{37}$$

Let L(t) denote the contour length at time t:

$$L(t) = \int_{S(t)} 1dS \tag{38}$$

The contour length of the ellipse with semiaxes 1+c and 1 is given by L(1). It is estimated by the Taylor series

$$L(1) = L(0) + L'(0) + \frac{1}{2}L^2(0) + \frac{1}{6}L^3(0) + \cdots \tag{39}$$

where the expressions for the derivatives, $L^n(0)$, are derived by evaluating expressions generated by our TRS. We calculate repeated derivatives and evaluate them at $t = 0$. The first term in the series is $L(0) = \int_S 1dS = 2\pi$.

The CMS is used to determine the higher order derivatives of $L(t)$. The first derivative is

$$
\begin{aligned}
L'(t) &= \frac{\delta}{\delta t}\int_S 1dS \\
&= \int_S \left(\frac{\delta 1}{\delta t} - CB^\alpha_\alpha\right)dS \\
&= -\int_S CB^\alpha_\alpha dS
\end{aligned} \tag{40}
$$

The CMS provides a means for taking the $\frac{\delta}{\delta t}$ derivatives of an integral $\int_S dS$. In addition to the base derivative $\frac{\delta 1}{\delta t} = 0$, new terms are added to account for the changing surface.

The integral range will be the same for each order derivative. Our TRS calculates the series of integrands, which are then evaluated and integrated in Maple. To calculate the nth order derivative, we determine

$$M_n = \frac{\delta M_{n-1}}{\delta t} - CB^\alpha_\alpha M_{n-1} \tag{41}$$

$$L^n(0) = \int_S M_n|_{t=0}\, dS \tag{42}$$

Having already determined the first integrand $M_1 = -CB^\alpha_\alpha$, we apply this recursive definition for M_2.

$$M_2 = -\frac{\delta\left(CB^\alpha_\alpha\right)}{\delta t} + C^2 B^\alpha_\alpha B^\beta_\beta \tag{43}$$

Equation (43) is the first expression that requires rewriting by our TRS. This expression is invariant; it is valid for the contour length of any surface deformation. In our specific case, the value $\frac{\delta B^\alpha_\alpha}{\delta t}$ can be evaluated directly. For many surfaces, evaluating the derivative of mean curvature is extremely complex. Our TRS produces a normal form where all terms can be evaluated. This means the $\frac{\delta}{\delta t}$-derivatives will only be on the Surface Velocity and Commutator. These are defined by the change in surface and their $\frac{\delta}{\delta t}$-derivatives can be evaluated.

The TRS reduces this expression to a normal form. The total reduction takes 31 rewrites including structure changes like rule (17) and simple reductions like rule (32). We highlight some of key reductions below. Subscripts are attached to the arrow symbol referencing the rule list in Section 4.

$$
\begin{aligned}
&\frac{\delta\left(-CB^\alpha_\alpha\right)}{\delta t} + C^2 B^\alpha_\alpha B^\beta_\beta \rightarrow_{18} -B^\alpha_\alpha \frac{\delta C}{\delta t} - C\frac{\delta B^\alpha_\alpha}{\delta t} + C^2 B^\alpha_\alpha B^\beta_\beta \\
&\rightarrow_6 -B^\alpha_\alpha \frac{\delta C}{\delta t} - C(\nabla^\alpha\nabla_\alpha C + CB^\alpha_\gamma B^\gamma_\alpha) + C^2 B^\alpha_\alpha B^\beta_\beta \\
&\rightarrow_{33} -B^\alpha_\alpha \frac{\delta C}{\delta t} - C\nabla^\alpha\nabla_\alpha C - C^2 B^\alpha_\gamma B^\gamma_\alpha + C^2 B^\alpha_\alpha B^\beta_\beta
\end{aligned} \tag{44}
$$

The normal form for M_2 is given by equation (44). This expression is true for the contour length of any deforming manifold. We now generate code to evaluate the expression for our realization of the problem. Given a specific coordinate system, our Maple library calculates the value of the symbols. We precalculate the derivatives of C as $C0, C1, C2, \cdots$. Tensors are stored as multidimensional Maple Arrays. The Maple tensor library does not support rectangular tensors, so we created our own functions using the existing libraries. The Maple code used for evaluation is shown below.

```
  Temp0:=contract(prodlist(intTensor(-1/1), BAb, C1),
[1,2]):
  Temp0:=Temp0:-apply([θ,0]):
  Temp1:=contract(prodlist(intTensor(-1/1), C0, ddSA(
ddSa( C0 ))), [1,2]):
  Temp1:=Temp1:-apply([θ,0]):
  Temp2:=contract(prodlist(intTensor(-1/1), BAb, BAb,
TensorExp(C0, 2)), [1,4 , 2,3]):
  Temp2:=Temp2:-apply([θ,0]):
  Temp3:=contract(prodlist( BAb, BAb, TensorExp( C0,
2)), [3,4 , 1,2]):
  Temp3:=Temp3:-apply([θ,0]):
  solution:=lin_com(Temp0, Temp1, Temp2, Temp3);
```

Each term in the sum is calculated, evaluated at $t = 0$, and then the sum is evaluated. The result of this expression in our coordinate system, equation (48), is

$$M_2|_{t=0} = c^2(7\cos^2\theta - 5)\cos^2\theta \tag{45}$$

Taking the integral gives the second term in the series.

$$\int_0^{2\pi} (M_2|_{t=0})\, d\theta = \frac{1}{4}c^2\pi \tag{46}$$

The TRS repeats this process and determines the normal form of M_3 which requires 118 rewrites. In addition, we have combined terms to shorten the expression.

$$
\begin{aligned}
M_3 = &-C^3 B^\alpha_\alpha B^\beta_\beta B^\gamma_\gamma + 3C^3 B^\alpha_\beta B^\beta_\alpha B^\gamma_\gamma - 2C^3 B^\alpha_\beta B^\gamma_\alpha B^\beta_\gamma \\
&+ 3C^2 B^\alpha_\alpha \nabla^\beta\nabla_\beta C - 4C^2 B^{\alpha\beta}\nabla_\beta\nabla_\alpha C \\
&+ 3C\frac{\delta C}{\delta t} B^\alpha_\alpha B^\beta_\beta - 3C\frac{\delta C}{\delta t} B^\alpha_\beta B^\beta_\alpha - 2\frac{\delta C}{\delta t}\nabla^\alpha\nabla_\alpha C \\
&- \frac{\delta^2 C}{\delta^2 t} B^\alpha_\alpha - C\nabla^\alpha\nabla_\alpha \frac{\delta C}{\delta t} + CR^{\alpha\beta}_\beta \nabla_\alpha C
\end{aligned} \tag{47}
$$

Equation (47) 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.

Although these expressions are true for any surface, the values for each tensor can only be defined after selecting surface and space coordinate systems. In Maple, these are defined as global variables and the objects are calculated by library calls. We will now evaluate equation (40) to show the process. For the ambient space, we will use Cartesian coordinates $[x, y]$. For the surface, we will define a circle that is being stretched into an ellipse. The function $S(\theta, t)$ maps a point in the surface coordinates to the ambient coordinates. The constant c controls the rate of change over time.

$$S(\theta, t) = [(1 + ct)\cos\theta, \sin\theta] \tag{48}$$

First, we generate the tensors needed to determine B^α_α and C. We evaluate all tensors at $t = 0$. The shift tensor is defined by the surface, it is used to restrict a spacial field to the surface.

$$Z^i_\alpha = \begin{bmatrix} [-(1+ct)\sin\theta] \\ [\cos\theta] \end{bmatrix} \tag{49}$$

The surface normal, N_i, surface velocity, C, and surface metric $S^{\alpha\beta}$, are also derived from the surface.

$$N_i|_{t=0} = \begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix}$$
$$C|_{t=0} = c\cos^2\theta$$
$$S^{\alpha\beta}|_{t=0} = [[1]]$$

Having defined these symbols, we now evaluate the mean curvature B^α_α.

$$\nabla_\beta N_i|_{t=0} = \left[\begin{bmatrix}-\sin\theta\\ \cos\theta\end{bmatrix}\right]$$
$$B_{\alpha\beta}|_{t=0} = \left(-Z^i_\alpha \nabla_\beta N_i\right)|_{t=0} = \left[\left[-\sin^2\theta - \cos^2\theta\right]\right]$$
$$B^\alpha_\beta|_{t=0} = \left(S^{\alpha\gamma}B_{\gamma\beta}\right)|_{t=0} = \left[\left[-\sin^2\theta - \cos^2\theta\right]\right]$$
$$B^\alpha_\alpha|_{t=0} = -\sin^2\theta - \cos^2\theta = -1$$

Now that the values of C and B^α_α are known, we evaluate

$$M_1|_{t=0} = -CB^\alpha_\alpha = c\cos^2\theta \tag{50}$$

Next we determine the integral of equation (50) to find $L'(0)$.

$$L'(0) = \int_0^{2\pi} c\cos^2\theta d\theta = c\pi \tag{51}$$

The series for $L(1)$ can be calculated independently of the CMS to determine if this result is correct.

$$\begin{aligned} L(1) &= \int_0^{2\pi}\left(\sqrt{(1+c)^2\cos^2(\theta)+\sin^2(\theta)}\right)d\theta \\ &= \left(2 + c + \frac{1}{8}c^2 - \frac{1}{16}c^3 + \frac{17}{512}c^4 - \frac{19}{1024}c^5 + \cdots\right)\pi \end{aligned} \tag{52}$$

We calculate the next two terms in the series using equations (44) and (47).

$$\frac{1}{2!}L^2(0) = \frac{1}{2}\left(\frac{1}{4}c^2\pi\right) = \frac{1}{8}c^2\pi \tag{53}$$
$$\frac{1}{3!}L^3(0) = \frac{1}{6}\left(-\frac{3}{8}\pi c^3\right) = -\frac{1}{16}c^3\pi \tag{54}$$

Using our TRS and evaluating in Maple, we were able to confirm that the first 7 terms in the series are correct.

8. APPLICATIONS

We have applied our TRS to problems that are of current interest to researchers in the CMS. In [4], our implementation was used to analyze the evolution of Poisson's equation under arbitrary smooth deformations of the domain. We minimized the energy E

$$E = \int_\Omega \left(\frac{1}{2}\nabla_i u \nabla^i u + fu\right)d\Omega \tag{55}$$

on the domain Ω with the zero Dirichlet boundary condition

$$u|_S = 0 \tag{56}$$

Order	Products	TRS	Maple
2	5	85 millisec	1 second
3	25	149 millisec	1 second
4	152	746 millisec	6 seconds
5	1,138	7,169 millisec	2.6 minutes

Table 2: Number of terms and time to evaluate the Poisson series.

Order	Products	TRS	Maple
2	4	1 sec	1 sec
3	24	1 sec	1 sec
4	155	5 sec	4 min
5	1,221	42 sec	6 min
6	11,024	11 min	1.5 hours
7	115,249	3 hours	18 hours

Table 3: Number of terms and time to evaluate the $\lambda(c)$ series.

along the boundary S. Minimizing u is governed by the Poisson equation

$$\nabla_i u \nabla^i u = f. \tag{57}$$

We have calculated the first 5 derivatives and evaluated them at time $t = 0$, providing the terms in a series for the energy. The number of CMS products that needed to be summed and the time to calculate values are given in Table 2.[1] We give the initial terms in the series:

$$E_N = -\frac{\pi}{16}\left(1 - \frac{8\zeta(2)}{N^2} - \frac{8\zeta(3)}{N^3} + \cdots\right). \tag{58}$$

We have also used our system to calculate the Laplace-Dirichlet eigenvalues on an ellipse in [3]. This is a classical question in boundary variations [17] and has been established correctly to second order. We have used our system to determine the first 7 terms. We give the first four terms in the series in eccentricity e for the lowest eigenvalue $\lambda(e)$ below

$$\begin{aligned} \lambda(e) =& \lambda - \frac{\lambda}{2}e^2 + \frac{\lambda-6}{32}e^4\lambda + \frac{\lambda-6}{64}e^6\lambda \\ &- \frac{7\lambda^3 - 58\lambda^2 - 192\lambda + 1792}{32768}e^8\lambda + \cdots \end{aligned} \tag{59}$$

In this equation, λ is the corresponding eigenvalue on the unit circle. The number of products that needed to be summed and time to evaluate each expression is given in Table 3. The first 7 terms are presented in [3].

9. CONCLUSIONS

Although we have limited the objects defined in our initial implementation, the CMS has a wide range of uses. In high order stability analysis, the CMS can be used for shape optimization [14]. Biological models, such as blood cell membranes, can also be modeled [8]. The CMS introduces a great deal of analytical order into fluid film dynamics [9]. Having shown the success of a TRS for the CMS on boundary variation problems, we plan to expand the rule set to include additional rules and objects from the CMS. This will make proving confluence and termination more difficult problems.

[1] All timings run on Mac OS 10.7 3.06 GHz Intel 4GB ram with Maple 16 and Java 1.7.

The CMS has been hindered by a lack of automated computation and a rapid growth in expression length. We have proposed a TRS to solve boundary variation problems in the CMS. This system has already shown the power of applying a TRS to the CMS. It was used to support current research in the CMS, with results presented in [3] and [4].

Although we have only implemented a subset of the CMS, these results illustrate the power of applying a TRS to the problems of the CMS. We have found high order variations in problems that had previously been intractable. We view this TRS as a first step towards comprehensive implementation of the CMS.

10. REFERENCES

[1] L. B. Buchberger and K. R. Loos. Algebraic simplification. *Computing, Suppl.*, 4:11–43, 1982.

[2] L. Bachmair and D. A. Plaisted. Termination orderings for associative-commutative rewrite systems. *J. Symbolic Computation*, 1:329–349, 1985.

[3] M. Boady, P. Grinfeld, and J. Johnson. Laplace eigenvalues on the ellipse and the symbolic calculus of moving surfaces. *In preparation*.

[4] 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.

[5] 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.

[6] 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.

[7] P. Grinfeld. Hadamard's formula inside and out. *J. Opt. Theory and Appl.*, 146(3):654–690, 2009.

[8] P. Grinfeld. Hamiltonian dynamic equations for fluid films. *Stud. Appl. Math.*, 125:223–264, 2010.

[9] P. Grinfeld. A variable thickness model for fluid films under large displacements. *Phys. Rev. Lett.*, 105:137802, 2010.

[10] P. Grinfeld and G. Strang. Laplace eigenvalues on polygons. *Computers and Mathematics with Applications*, 48:1121–1133, 2004.

[11] 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.

[12] P. Grinfeld and J. Wisdom. A way to compute the gravitational potential for near-spherical geometries. *Quart. Appl. Math.*, 64(2):229–252, 2006.

[13] Y. V. Gusev. Heat kernel expansion in the covariant perturbation theory. *Nuclear Physics B*, 807(3):566 – 590, 2009.

[14] H. Howards, M. Hutchings, and F. Morgan. The isoperimetric problem on surfaces. *The American Mathematical Monthly*, 106(5):pp. 430–439, 1999.

[15] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. Assoc. Comp. Mach.*, 27:797–821, 1980.

[16] G. Huet and D. C. Oppen. Equations and rewrite rules – a survey. Technical report, Stanford University, Jan 1980.

[17] D. Joseph. Parameter and domain dependence of eigenvalues of elliptic partial differential equations. 24(5):325–361, 1967.

[18] 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.

[19] 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.

[20] D. Knuth and P. Bendix. Simple word problems in universal algebras. *Computational Problems in Abstract Algebra*, pages 263–297, 1970.

[21] T. Levi-Civita. *The Absolute Differential Calculus (Calculus of Tensors)*. Dover Publications, 1977.

[22] 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.

[23] T. Málek and V. Pravda. Kerra-schild spacetimes with an (a)ds background. *Classical and Quantum Gravity*, 28(12):125011, 2011.

[24] Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario, Canada. *Maple User Manual*, 2012.

[25] J. M. Maran-Garca. xperm: fast index canonicalization for tensor computer algebra. *Computer Physics Communications*, 179(8):597 – 603, 2008.

[26] MathTensor Inc. Mathtensor - tensor analysis for mathematica. http://smc.vnet.net/MathTensor.html.

[27] A. McConnell. *Applications of Tensor Analysis*. Dover Publications, New York, 1957.

[28] K. Peeters. *Cadabra: reference guide and tutorial*. http://cadabra.phi-sci.com/cadabra.pdf, June 2008.

[29] 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.

[30] J. Synge and A. Schild. *Tensor Calculus*. Dover Publications, Inc., 1949.

[31] T. Thomas. *Concepts from Tensor Analysis And Differential Geometry*. Academic Press, New York, 1965.

[32] Wolfram Research, Champaign, IL. *Wolfram Mathematica 9 Documentation Center*, 2012.