A Term Rewriting System for the Calculus of Moving Surfaces

Mark Boady, Pavel Grinfeld, Jeremy Johnson

Drexel University

June 27, 2013
Overview

- The Calculus of Moving Surfaces
- The kinds of problems it can solve
- Why it needs an automated approach
- The implementation of our Term Rewrite System
- Research Results
We have a surface and the we know properties of the surface.
We change the shape of the surface.
How does the surface changing affect the properties of the surface?
The Calculus of Moving Surfaces is an analytic framework for studying surfaces with evolving shapes.
We know the acoustic properties of a concert hall.
How do the acoustics change if we extend the concert hall?
Calculus of Moving Surfaces

- The Calculus of Moving Surfaces
  - A calculus for studying surfaces that change shape
  - Pioneered by Hadamard in 1903

- The invariant time derivative \( \dot{\nabla} \)
  - Allows us to study how properties change with respect to the surface changing shape over time

- The coordinate system on the surface is changing as time passes
  - The CMS extends Tensor Calculus
  - Tensor Calculus gives expressions that can be evaluated in any coordinate system
  - Automation has been very successful in the Tensor Calculus
We have a Scalar Field $F$

- Given a point on the surface it maps to a number
- At time $t$, the surface is $S_t$
- After $h$ time has passed the new surface is $S_{t+h}$

\[
\dot{\nabla} F = \lim_{h \to 0} \frac{F(P^*) - F(P)}{h}
\]
The Calculus of Moving Surfaces is a Symbolic Calculus

Analytical study of deforming surfaces

Objects represent properties of the surface

$B_{\alpha\beta}$, the curvature tensor

- Describes the curvature of the surface
- The trace is the mean curvature of the surface

The $\dot{\nabla}$ derivative is difficult to calculate

We have rules to take the $\dot{\nabla}$ derivative of each object

$$\dot{\nabla} B_{\beta}^\alpha = \nabla^\alpha \nabla_\beta C + CB_\gamma^\alpha B_\beta^\gamma$$

Rules transform expressions into equivalent expressions that can be evaluated

Evaluate the expression by choosing a coordinate system
Term Rewriting System

- Encode the properties of the CMS as rewrite rules
- Start with a symbolic expression
- Systematically apply rules to transform the expression
- Stop when no more rules can be applied (Normal Form)
- Normal Forms are Simplified Expressions
  - An expression that is easier to evaluate
  - All functions only need to be evaluated on the initial surface ($t = 0$)
The contour length of the unit circle is $2\pi$.

We stretch the circle into an ellipse with eccentricity $\epsilon$.

What is the formula for the new contour length in terms of $\epsilon$?
The contour length is $L(t) = \int_{S_t} 1dS_t$

What is the contour length, $L(t = 1)$, of an ellipse with semiaxes 1 and $1 + \epsilon$?

- Form a Taylor Series by taking derivatives of $L(t)$
- $L(1) = L(0) + L'(0) + \frac{1}{2} L''(0) + \frac{1}{6} L'''(0) + \cdots$
- More terms lead to a more accurate result

$L'(t) = \dot{\nabla} \left( \int_S 1dS \right) = \int_S \left( \dot{\nabla} 1 - CB_\alpha^\alpha \right) ds = -\int_S CB_\alpha^\alpha dS$

$L'(0) = \epsilon \pi$
\[ L''(t) = \dot{\nabla} \left( - \int_S C B^\alpha dS \right) \]
\[ L''(t) = \dot{\nabla} \left( - \int_S CB_\alpha dS \right) \]
\[ \rightarrow \int_S \left( -\dot{\nabla} \left( CB_\alpha \right) + C^2 B_\alpha B_\beta \right) dS \]
\[ L''(t) = \dot{\nabla} \left( - \int_S C B_\alpha^\alpha dS \right) \]
\[ \rightarrow \int_S \left( -\dot{\nabla} (C B_\alpha^\alpha) + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\[ \rightarrow \int_S \left( -B_\alpha^\alpha \dot{\nabla} C - C \dot{\nabla} B_\alpha^\alpha + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\[ L''(t) = \dot{\nabla} \left( - \int_S C B_\alpha^\alpha dS \right) \]

\[ \rightarrow \int_S \left( -\dot{\nabla} \left( C B_\alpha^\alpha \right) + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]

\[ \rightarrow \int_S \left( -B_\alpha^\alpha \dot{\nabla} C - C \dot{\nabla} B_\alpha^\alpha + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]

\[ \rightarrow \int_S \left( -B_\alpha^\alpha \dot{\nabla} C - C \left( \nabla^\alpha \nabla_\alpha C + C B_\gamma^\alpha B_\gamma^\gamma \right) + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\[ L''(t) = \dot{\nabla} \left( - \int_S CB_\alpha^\alpha dS \right) \]
\[ \rightarrow \int_S \left( - \dot{\nabla} (CB_\alpha^\alpha) + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\[ \rightarrow \int_S \left( - B_\alpha^\alpha \dot{\nabla} C - C \dot{\nabla} B_\alpha^\alpha + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\[ \rightarrow \int_S \left( - B_\alpha^\alpha \dot{\nabla} C - C \left( \nabla^\alpha \nabla_\alpha C + CB_\gamma^\alpha B_\alpha^\gamma \right) + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\[ \rightarrow \int_S \left( - B_\alpha^\alpha \dot{\nabla} C - C \nabla^\alpha \nabla_\alpha C + C^2 B_\gamma^\alpha B_\alpha^\gamma + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]
\( -B_{\alpha}^\alpha \nabla C \)

\[ \text{Temp0} := \text{contract} \left( \text{prodlist} \left( \begin{array}{c}
\text{intTensor}(-1),
BAb, C1
\end{array} \right) \right) , [1,2] : \]

\[ \text{Temp0} := \text{Temp0} - \text{apply}([\theta,0]) : \]

\( -C \nabla^\alpha \nabla_\alpha C \)

\[ \text{Temp1} := \text{contract} \left( \text{prodlist} \left( \begin{array}{c}
\text{intTensor}(-1),
C0, \text{ddSA}(\text{ddSa}(C0))
\end{array} \right) \right) , [1,2] : \]

\[ \text{Temp1} := \text{Temp1} - \text{apply}([\theta,0]) : \]
\[ C^2 B_\gamma^\alpha B_\alpha^\gamma \]

Temp2 := contract(prodlist(
intTensor(-1),
BAb, BAb, TensorExp(C0, 2))
, [1, 4, 2, 3]):
Temp2 := Temp2:-apply([\theta, 0]):
\[ C^2 B_\alpha^\alpha B_\beta^\beta \]
Temp3 := contract(prodlist(
BAb, BAb, TensorExp(C0, 2))
, [3, 4, 1, 2]):
Temp3 := Temp3:-apply([\theta, 0]):
\int_S \left(-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 \right) \, dS

total := lin_com(Temp0, Temp1, Temp2, Temp3):
solution := int(total, theta=0..2\pi);
Reach Normal Form for Expression

\[ L''(t) = \int_S \left( -B_\alpha^\alpha \dot{\nabla} C - C \nabla^\alpha \nabla_\alpha C + C^2 B_\gamma^\alpha B_\alpha^\gamma + C^2 B_\alpha^\alpha B_\beta^\beta \right) dS \]

Select Coordinate System and Export to Maple
Evaluate to get final answer

\[ L''(0) = \frac{1}{4} \epsilon^2 \pi \]

We use these terms to form a Taylor Series

\[ L(t = 1) = \left( 2 + \epsilon + \frac{1}{8} \epsilon^2 - \frac{1}{16} \epsilon^3 + \frac{17}{512} \epsilon^4 - \frac{19}{1024} \epsilon^5 + \cdots \right) \pi \]

\[ L(t = 1) = \int_0^{2\pi} \sqrt{(1 + \epsilon)^2 \cos^2 \theta + \sin^2 \theta} d\theta \]
What is the series in $\epsilon$ for the simple Laplace-Dirchlet eigenvalues on an ellipse with semi-axes 1 and $1 + \epsilon$?

Solve the system of equations for unknowns $u$ and $\lambda$

- $\Delta u + \lambda u = 0$
- $u|_S = 0$
- $\int_{\Omega} u^2 d\Omega = 1$

The Laplace-Dirchlet eigenvalues are given by the Taylor series

- $\lambda_{ellipse}(1) = \lambda(0) + \lambda'(0) + \frac{1}{2}\lambda''(0) + \frac{1}{6}\lambda'''(0) + \cdots$
- $\lambda(0) = \lambda_{circle}$

The accuracy of the answer depends on the number of derivatives
Properties of the Covariant Derivative

\[ \nabla_\alpha (FG) \rightarrow G \nabla_\alpha (F) + F \nabla_\alpha (G) \]

\[ \nabla_\alpha (x_1) \rightarrow 0 \]

\[ \nabla_\alpha (F + G) \rightarrow \nabla_\alpha (F) + \nabla_\alpha (G) \]

\[ \nabla_\alpha (\sum_i F \cdots i \cdots) \rightarrow \sum_i \nabla_\alpha (F \cdots i \cdots) \]

Properties of the Partial Derivative

\[ \frac{\partial FG}{\partial t} \rightarrow F \frac{\partial G}{\partial t} + G \frac{\partial F}{\partial t} \]

\[ \frac{\partial x_1}{\partial t} \rightarrow 0 \]

\[ \frac{\partial (F + G)}{\partial t} \rightarrow \frac{\partial F}{\partial t} + \frac{\partial G}{\partial t} \]

\[ \frac{\partial \sum_i F \cdots i \cdots}{\partial t} \rightarrow \sum_i \frac{\partial F \cdots i \cdots}{\partial t} \]

\[ \frac{\partial \nabla_\alpha F}{\partial t} \rightarrow \nabla_\alpha \frac{\partial F}{\partial t} \]
### Differential Table

<table>
<thead>
<tr>
<th>Term</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\dot{\nabla} Z^i_\alpha$</td>
<td>$N^i \nabla_\alpha C$</td>
</tr>
<tr>
<td>$\dot{\nabla} N^i$</td>
<td>$-Z^i_\alpha \nabla_\alpha C$</td>
</tr>
<tr>
<td>$\dot{\nabla} B^\alpha_\beta$</td>
<td>$\nabla^\alpha \nabla_\beta C + CB^\alpha_\gamma B^\gamma_\beta$</td>
</tr>
<tr>
<td>$\dot{\nabla} C^{x_1}$</td>
<td>$x_1 C^{x_1-1} \dot{\nabla} C$</td>
</tr>
<tr>
<td>$\dot{\nabla} x_1$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\dot{\nabla} F$</td>
<td>$\frac{\partial F}{\partial t} + CN^i \nabla_i F$</td>
</tr>
</tbody>
</table>
Properties of $\dot{\nabla}$

$\dot{\nabla} \left( \sum_i F_{i\cdots} \right) \rightarrow \sum_i \left( \dot{\nabla} F_{i\cdots} \right)$

$\dot{\nabla} (FG) \rightarrow G \dot{\nabla} (F) + F \dot{\nabla} (G)$

$\dot{\nabla} (F + G) \rightarrow \dot{\nabla} F + \dot{\nabla} G$

$\dot{\nabla}\nabla_\alpha A \rightarrow \nabla_\alpha \dot{\nabla} A + CB^\gamma_\alpha \nabla_\gamma A$

$\dot{\nabla}\nabla_\alpha A^\beta \rightarrow \nabla_\alpha \dot{\nabla} A^\beta + CB^\gamma_\alpha \nabla_\gamma A + \dot{R}^\beta_{\gamma\alpha} A^\gamma$

$\dot{\nabla}\nabla_\alpha A_\beta \rightarrow \dot{\nabla}_\alpha \dot{\nabla} A_\beta + CB^\gamma_\alpha \nabla_\gamma A_\beta - \dot{R}^\gamma_{\beta\alpha} A_\gamma$

Additional Rules

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

$A + 0 \rightarrow A$

$A(F + G) \rightarrow AF + AG$

$0 A \rightarrow 0$
These rules are sufficient to find arbitrary combinations of derivatives in boundary perturbation problems.

The rules are confluent.
- Order of application does not change final answer.

The rules are terminating.
- A normal form will be reached for any input.
Implemented a custom TRS in JAVA
- Inheritance for related objects and rules
- Expression Subtree Matching

Equivalence modulo
- Associativity
- Commutativity
- Index Juggling
- Index Renaming
- Example: $B^\alpha_\beta B^\beta_\gamma B^\gamma_\alpha B^\delta_\delta = B^\gamma_\gamma B^\delta\beta_\beta B^\beta_\alpha B^\delta_\delta$

Maple Library for Evaluation
- Built on existing Array manipulation tools

Code Generation
- Objects used by the TRS are also used for code generation
- Select a coordinate system
- Generate Maple code for evaluation in coordinate system
Laplace-Dirchlet Eigenvalues

\[ \lambda'(t) = - \int_S C \nabla_i u \nabla^i u \, dS \]

\[ \lambda'(0) = - \lambda \]

\[ \lambda''(t) = \dot{\nabla} \left( - \int_S C \nabla_i u \nabla^i u \, dS \right) \]

\[ = \int_S \left( C^2 B^\alpha \nabla^i u \nabla^i u \right) \, dS - \int_S \left( \dot{\nabla} C \nabla^i u \nabla^i u \right) \, dS \]

\[ - \int_S \left( 2 C \nabla_i \frac{\partial u}{\partial t} \nabla^i u \right) \, dS - \int_S \left( 2 C^2 N^i \nabla_j u \nabla^j \nabla^i u \right) \, dS \]

\[ \lambda''(0) = \frac{3}{2} \lambda + \frac{1}{4} \lambda^2 \]
\[ \chi''(t) = \int_S C^3 B_\alpha^\beta B_\beta^\alpha \nabla_i \nabla_i u \nabla_i u dS - \int_S 2C^3 N^j N^k \nabla_i \nabla^i \nabla_k \nabla_j u dS \]

+ \int_S 4C^3 B_\alpha^\alpha N^j \nabla_i \nabla_i ^{j} u \nabla_i u dS - \int_S 4 \nabla \nabla^i \frac{\partial u}{\partial t} \nabla_i u dS

- \int_S 2C^3 N^j N^k \nabla_i \nabla_i ^{k} u \nabla_k \nabla_i u dS - \int_S C^3 B_\beta^\beta B_\alpha^\alpha \nabla_i \nabla_i u \nabla_i u dS

+ \int_S 4C^2 B_\alpha^\alpha \nabla_i \frac{\partial u}{\partial t} \nabla_i u dS + \int_S C^2 \nabla_i \nabla_i u \nabla_i u \nabla_\alpha \nabla_\alpha C dS

- \int_S 4C^2 N^j \nabla_i \nabla_i ^{j} u \frac{\partial u}{\partial t} dS + \int_S 3CB_\alpha^\alpha \nabla \nabla^i \nabla_i u \nabla_i u dS

- \int_S 4C^2 N^j \nabla_i \nabla_i ^{j} u \frac{\partial u}{\partial t} \nabla_j \nabla_i u dS - \int_S 2C \nabla_i \nabla^2 \nabla_i u dS

- \int_S \nabla \nabla^2 \nabla_i u \nabla_i u dS + \int_S 2C^2 Z_\alpha^j \nabla_i \nabla_i ^{j} u \nabla_i u \nabla_\alpha C dS

- \int_S 2C \nabla \nabla_i \frac{\partial u}{\partial t} \nabla_i \frac{\partial u}{\partial t} dS - \int_S 6C \nabla \nabla C N^j \nabla_i \nabla_i ^{j} u \nabla_i u \nabla_j u dS
Taking repeated derivatives causes expression swell
Higher order derivatives provide a more accurate series
Number of terms grows exponentially
The time to evaluate also increases with the order of derivatives.
\[ \lambda'(0) = -\lambda \]
\[ \lambda''(0) = \frac{3}{2}\lambda + \frac{1}{4}\lambda^2 \]
\[ \lambda'''(0) = -3\lambda - \frac{3}{2}\lambda^2 \]
\[ \lambda^4(0) = \frac{15}{2}\lambda + \frac{15}{2}\lambda^2 + \frac{87}{128}\lambda^3 - \frac{21}{256}\lambda^4 \]
\[ \lambda^5(0) = -\frac{45}{2}\lambda - \frac{75}{2}\lambda^2 - \frac{1305}{128}\lambda^3 + \frac{315}{256}\lambda^4 \]
\[ \lambda^6(0) = \frac{315}{4}\lambda + \frac{1575}{8}\lambda^2 + \frac{27405}{256}\lambda^3 - \frac{11155}{1536}\lambda^4 - \frac{2665}{1536}\lambda^5 + \frac{145}{1024}\lambda^6 \]
\[ \lambda(\varepsilon) = \lambda + \left(-\frac{1}{2} \lambda \varepsilon^2 + \left(-\frac{3}{16} \lambda + \frac{1}{32} \lambda^2 \right) \varepsilon^4 + \left(-\frac{3}{32} \lambda + \frac{1}{64} \lambda^2 \right) \varepsilon^6 \right. \\
\left. + \left(-\frac{7}{128} \lambda + \frac{3}{512} \lambda^2 + \frac{29}{16384} \lambda^3 - \frac{7}{32768} \lambda^4 \right) \varepsilon^8 \\
+ \left(-\frac{9}{256} \lambda + \frac{1}{1024} \lambda^2 + \frac{87}{32768} \lambda^3 - \frac{21}{65536} \lambda^4 \right) \varepsilon^{10} + O(\varepsilon^{12}) \right] \\

- Expanded Series to higher order than previously calculated \\
- Found errors in previously published lower terms
Conclusions

- The Calculus of Moving Surfaces is an Analytic Framework for studying the deformation of surfaces.
- It can be used to study the properties of surfaces that change shape.
  - Boundary Variations
  - Shape Optimizations
  - Biological Models (Modeling Blood Cell Membranes)
  - Fluid Film Dynamics (Modeling Soap Films)
- We have implemented the first TRS for the CMS
  - Enables research to use the CMS
  - Easily extensible TRS
- We have used our system find a more accurate series for the Laplace-Dirchlet Eigenvalues on an ellipse.
Future Work

- What is the series in $\frac{1}{N}$ for the simple Laplace-Dirchlet eigenvalues on a regular polygon with N sides? (Grinfeld and Strang 2004)
  \[ \lambda_N = \lambda \left( 1 + \frac{4\zeta(2)}{N^2} + \frac{4\zeta(3)}{N^3} + \frac{28\zeta(4)}{N^4} + \cdots \right) \]

- Manipulation of Fourier Series
- Implement Additional Rules from the CMS
- Improve Efficiency and Decrease Expression Swell
- Eliminate Redundent Calculations