A Symbolic Computation System for the Calculus of Moving Surfaces

Mark Boady, Pavel Grinfeld, Jeremy Johnson

Drexel UNIVERSITY

Objective

No programs are currently available to symbolically solve problems in the Calculus of Moving Surfaces (CMS). Many research fields take advantage of the CMS and would benefit from a symbolic system. We are developing a system to fill this gap and help advance ongoing research in the CMS.

The goals of the system are:
- Simplify complex expressions.
- Take derivatives and integrals.
- Combine like terms.
- Support very long and complex symbolic expressions.
- Convert expressions for use in other packages (Mathematica, Maple).
- Support all the rules of the CMS.
- Allow users to create their own rules.

Motivation

The Calculus of Moving Surfaces is an extension of Tensor Calculus on stationary surfaces to moving surfaces.
The CMS provides analytic tools for finding solutions to a wide range of problems with moving surfaces such as
- Fluid Film Dynamics
- Boundary Variation Problems
- Shape Optimization Problems

A symbolic system will provide advantages over existing methods. As with any analytic framework, the complexity of calculations grows rapidly with the order of approximation. This means that hand calculations quickly become error prone or intractable. Automated symbolic computations will not make errors or become hindered by complex calculations. Symbolic Computation also offers advantages over numerical methods, particularly when the boundary perturbation is too complex to be captured effectively and when the perturbation leads to singularities.

Researchers working in fields related to the CMS will be able to spend more time investigating ideas and less effort performing tedious calculations. Formally intractable problems will become solvable, allowing advances in research. The primary goal of this system is to provide a tool that can advance the work of numerous other members of the research community.

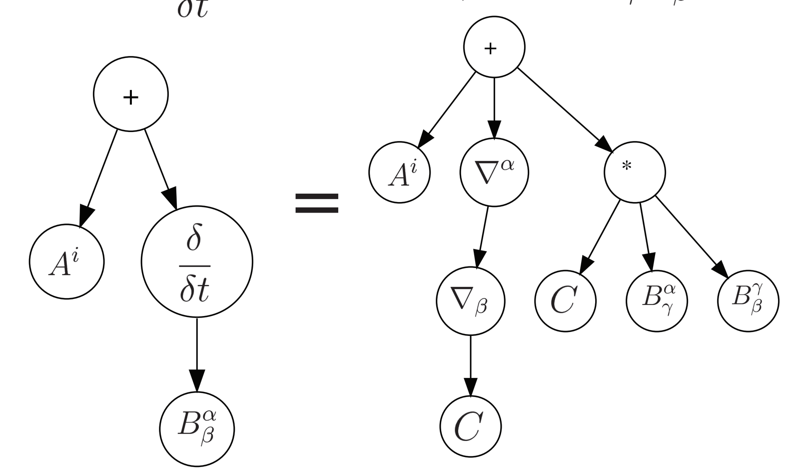
Method

We have built a prototype system to solve a selection of problems in the CMS that are currently relevant to researchers. New rules and objects are added to expand the types of problems the system can handle. Known solutions are compared to the systems output to determine if calculations are being handled correctly.

The over arching strategy of the system is to:
- Create an tree structure representing each expression.
- Walk the tree to find subtrees that match known rules in the CMS.
- Replace the subtrees with a new version representing the applied rule.
- Find equivalent subtrees to combine and cancel terms.
- Apply Rules towards a canonical form.
- Export the final expression in symbolic or Mathematica form.

Applying Rules

$$A^i + \frac{\delta B^\alpha_\beta}{\delta t} = A^i + \nabla^\alpha \nabla_\beta C + C B^\alpha_\gamma B^\gamma_\beta$$



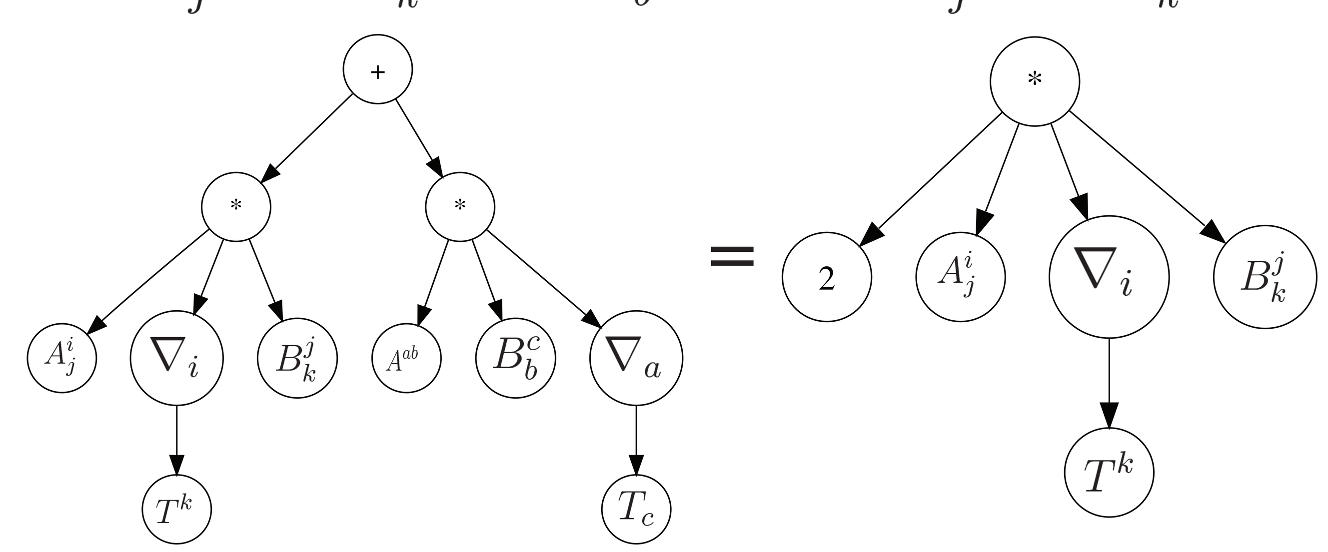
When a subtree that matches a rule is found, the existing subtree is pruned and a new subtree is created applying the rule.
The figure shows the tree manipulation required to apply one rule for the invariant time derivative to the curvature tensor.

$$\frac{\delta B^\alpha_\beta}{\delta t} = \nabla^\alpha \nabla_\beta C + C B^\alpha_\gamma B^\gamma_\beta$$

These operations can all be applied automatically improving the speed and reliability of calculations. Some elements of the rule are variable, in this case the indexes alpha and beta can have any names as long as they have the correct properties and positions.
Many rules increase the total size of the expression. This creates long and complex expressions when working symbolically.

Combining Terms

$$A^i_j \nabla_i T^k B^j_k + A^{ab} B^c_b \nabla_a T_c = 2 A^i_j \nabla_i T^k B^j_k$$



The two products are equal, which means they can be combined to simplify the expression.

Why are these two products equal? $A^i_j \nabla_i T^k B^j_k = A^{ab} B^c_b \nabla_a T_c$

Multiplication is commutative allowing term reordering. $A^i_j B^j_k \nabla_i T^k = A^{ab} B^c_b \nabla_a T_c$

Contracted (paired) indexes can be renamed. i=a,j=b, k=c $A^a_b B^b_c \nabla_a T^c = A^{ab} B^c_b \nabla_a T_c$

Contracted Indexes can be juggled. Flip the positions of the two b and c indexes. Possible positions are upper and lower. $A^{ab} B^c_b \nabla_a T_c = A^{ab} B^c_b \nabla_a T_c$

Results

The system has been shown to successfully solve a subset of problems in the CMS. Problems were selected that have known solutions and the calculations of the system were compared to the results. The output was also converted into Mathematica code for numerical comparisons. Problems were selected that could be solved using the subset of rules currently implemented in the prototype system.

One model problem that was successfully solved by the system involved Poisson's equation. We examined Poisson's problem with $\nabla_i \nabla^i u = 1$ on a regular polygon with N sides. This problem uses many of the implemented rules.

The first variation of the model problem is determined by hand calculations. This gives the system a starting point to determine the higher variations.

$$u_1 = -C N_i \nabla^i u$$

The initial formula for higher order variations can be described by known formulas. Converting these into simplified form is performed automatically by the system.

$$u_2 = -C N_i \nabla^i u_1 - \frac{\delta C N^i \nabla_i u}{\delta t}$$

Allowing the program to automatically simply the expression results in the following expanded formula.

$$u_2 = -2 C N^i \nabla_i u_1 - \frac{\delta C}{\delta t} N^i \nabla_i u + C Z^i_\alpha \nabla^\alpha C \nabla_i u - C^2 N^i N^j \nabla_i \nabla_j u$$

This expansion shows the number of terms is already increasing rapidly. The third variation can also be simplified in a similar manner.

$$u_3 = \left\{ \begin{array}{c} C^2 \nabla^\beta C Z^i_\beta N^j \nabla_i \nabla_j u + 2 C^2 \nabla^\alpha C Z^i_\alpha N^j \nabla_i \nabla_j u \\ + C^2 \nabla^\alpha C \nabla_i u Z^i_\beta B^\beta_\alpha - C^3 N^i N^j N^k \nabla_i \nabla_j \nabla_k u \\ + 3 C \nabla^\alpha C Z^i_\alpha \nabla_i u_1 + C \nabla^\beta \frac{\delta C}{\delta t} Z^i_\beta \nabla_i u \\ + 2 \frac{\delta C}{\delta t} \nabla^\alpha C Z^i_\alpha \nabla_i u - 3 C^2 N^i N^j \nabla_i \nabla_j u_1 \\ + C \nabla_\beta C \nabla^\beta C N^i \nabla_i u - 3 C \frac{\delta C}{\delta t} N^i N^j \nabla_i \nabla_j u \\ - 3 C N^i \nabla_i u_2 - \frac{\delta^2 C}{\delta t^2} N^i \nabla_i u - \frac{\delta C}{\delta t} N^i \nabla_i u_1 \end{array} \right\}$$

The third variation clearly makes the case for an automated approach to the symbolic computation. In this case, the third order has already become cumbersome to calculate by hand. Even higher orders, like the fifth and sixth can be calculated automatically by our prototype. Each of these results was converted to a general equation in polar coordinates to automatically compare the results to the known solutions. This model provides one of the many problems the system has already shown success on. The problem is also relevant to researchers in the field of CMS, giving a glimpse of the systems future potential.

Conclusions

The prototype system has been used to accurately solve a number of known problems. It has also been used to support ongoing research in the CMS. The system has been able to output the final results as Mathematica code allowing for numerical calculations.
The growth rate for many expressions has increased the number of terms to a size that overwhelms the system. Finding methods to better handle very large expressions will be a key factor moving forward.
Determining if expressions are equivalent is possible but currently takes factorial time. Finding more efficient methods for equivalence testing will improve system performance.
Overall, the system has shown significant promise and solved and number of significant model problems. Ongoing development will lead to a system that will simplify the research in the CMS.

Contact

Mark Boady mwb33@drexel.edu
Pavel Grinfeld pg77@drexel.edu http://pg.math.drexel.edu/
Jeremy Johnson jjohnson@cs.drexel.edu
http://www.cs.drexel.edu/~jjohnson/

References

M. Boady, P. Grinfeld, and J. Johnson. Boundary variation of Poisson's equation: a model problem for symbolic calculus of moving surfaces. *Submitted to Numer. Funct. Anal. Opt.*, 2010.
M. Boady, P. Grinfeld, and J. Johnson. The calculus of moving surfaces in mathematica. *In preparation*, 2010.
M. Boady, P. Grinfeld, and J. Johnson. Symbolic calculus of moving surfaces. *Submitted to Proc. of International Symposium on Symbolic and Algebraic Computation (ISSAC) 2011*, 2011.
P. Grinfeld and G. Strang. Laplace eigenvalues on regular polygons: a series in 1/N. *In preparation*, 2009.