Poster Presentation

Symbolic Computation of Conserved Densities for Systems of Nonlinear Evolution and Lattice Equations

Willy Hereman and Ünal Göktaş

Dept. Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80401-1887

Los Alamos Days 1998
University of Colorado at Boulder
Thursday, April 30, 1998
5:30-6:30p.m.
• **Purpose**

Design and implement algorithms for polynomial conservation laws of nonlinear systems of evolution and lattice equations.

• **Motivation**

– Conservation laws describe the conservation of fundamental physical quantities (linear momentum, energy, etc.).

– For nonlinear PDEs and lattices, the existence of a sufficiently large (in principal infinite) number of conservation laws assures complete integrability.

– Conservation laws provide a method to study quantitative and qualitative properties of equations and their solutions e.g. Hamiltonian structures.

– Conservation laws can be used to test numerical integrators.
PART I: Algorithm for Evolution Equations

Consider a system of evolution equations

\[ u_t = F(u, u_x, u_{2x}, ..., u_{mx}) \]

in a (single) space variable \( x \) and time \( t \), and with

\[ u = (u_1, u_2, ..., u_n), \quad F = (F_1, F_2, ..., F_n). \]

Notation:

\[ u_{mx} = u^{(m)} = \frac{\partial u}{\partial x^m}. \]

\( F \) is a polynomial function in \( u, u_x, ..., u_{mx} \).

PDEs of higher order in \( t \), should be recast as a first-order system.

• Conservation Law

\[ D_t \rho + D_x J = 0 \]

with conserved density \( \rho \) and flux \( J \).

Both are polynomial in \( u, u_x, u_{2x}, u_{3x}, ... \).

Consequently,

\[ P = \int_{-\infty}^{+\infty} \rho \, dx = \text{constant} \]

if \( J \) vanishes at infinity.

Conserved densities are equivalent if they differ by a total derivative.
• Example

Consider the Korteweg-de Vries (KdV) equation

\[ u_t + uu_x + u_{3x} = 0. \]

Conserved densities:

\[ \rho_1 = u, \quad D_t(u) + D_x\left(\frac{u^2}{2} + u_{2x}\right) = 0. \]

\[ \rho_2 = u^2, \quad D_t(u^2) + D_x\left(\frac{2u^3}{3} + 2uu_{2x} - u_x^2\right) = 0. \]

\[ \rho_3 = u^3 - 3u^2 x, \quad D_t\left(u^3 - 3u^2 x\right) + D_x\left(\frac{3}{4}u^4 - 6uu_x^2 + 3u^2u_{2x} + 3u_{2x}^2 - 6u_xu_{3x}\right) = 0. \]

\[ \vdots \]

\[ \rho_6 = u^6 - 60u^3u_x^2 - 30u_x^4 + 108u^2u_{2x}^2 + \frac{720}{7}u_{2x}^3 - \frac{648}{7}uu_{3x}^2 + \frac{216}{7}u_{4x}^2, \quad \text{...... long ......} \]

\[ \vdots \]

Time and space dependent conservation law:

\[ D_t\left(tu^2 - 2xu\right) + D_x\left(\frac{2}{3}tu^3 - xu^2 + 2tuu_{2x} - tu_x^2 - 2xu_{2x} + 2u_x\right) = 0. \]
• **Key Concept: Dilation Invariance**

The KdV equation and its conservation laws are invariant under the dilation (scaling) symmetry

\[(t, x, u) \rightarrow (\lambda^{-3}t, \lambda^{-1}x, \lambda^2 u).\]

\(u\) corresponds to two \(x\)-derivatives, \(u \sim D_x^2\). Similarly, \(D_t \sim D_x^3\).

The **weight**, \(w\), of a variable equals the number of \(x\)-derivatives that variable carries.

Weights are rational. Weights of dependent variables are nonnegative.

Set \(w(D_x) = 1\).

Due to dilation invariance: \(w(u) = 2\) and \(w(D_t) = 3\).

Consequently, \(w(x) = -1\) and \(w(t) = -3\).

The **rank** of a monomial is its total weight in terms of \(x\)-derivatives.

Every term (monomial) in the KdV equation has rank 5.

The KdV equation has rank 5.

This property is called **uniformity in rank**.

• **Steps of the Algorithm for Evolution Equations**

1. Determine weights (scaling properties) of variables & parameters.

2. Construct the form of the density (building blocks).

3. Determine the constant coefficients.
• **Example:** For the KdV equation, compute the density of rank 6.

**Step 1: Compute the weights.**

Require uniformity in rank to compute the weights of the dependent variables (solve a linear system).

With \( w(D_x) = 1 : w(u) + w(D_t) = 2w(u) + 1 = w(u) + 3 \).

Hence, \( w(u) = 2, w(D_t) = 3 \).

**Step 2: Determine the form of the density.**

List all possible powers of \( u \), up to rank 6 : \([u, u^2, u^3]\).

Introduce \( x \) derivatives to ‘complete’ the rank.

\( u \) has weight 2, introduce \( D_x^4 \).

\( u^2 \) has weight 4, introduce \( D_x^2 \).

\( u^3 \) has weight 6, no derivative needed.

Apply the derivatives.

Remove terms that are total \( x \)-derivatives with or total derivative up to terms kept earlier in the list.

\[
[u_{4x}] \rightarrow \emptyset \quad \text{empty list.}
\]

\[
[u_x^2, uu_{2x}] \rightarrow [u_x^2] \quad \text{since } uu_{2x} = (uu_x)_x - u_x^2.
\]

\[
[u^3] \rightarrow [u^3].
\]

Combine the ‘building blocks’:

\[
\rho = c_1 u^3 + c_2 u_x^2.
\]
Step 3: Determine the coefficients in the density.

Determine the coefficients \( c_1 \) and \( c_2 \).

- Compute \( D_t \rho = 3c_1 u^2 u_t + 2c_2 u_x u_{xt} \).
- Replace \( u_t \) by \(-(uu_x + u_{3x})\) and \( u_{xt} \) by \(-(uu_x + u_{3x})_x\).
- Integrate the result with respect to \( x \).

Carry out all integrations by parts (or use the Euler operator).

\[
D_t \rho = -D_x \left[ \frac{3}{4} c_1 u^4 -(3c_1-c_2)uu_x^2 + 3c_1 u^2 u_{2x} - c_2 u_{2x}^2 + 2c_2 u_x u_{3x} \right] \\
-(3c_1 + c_2)u_x^3.
\]

- The non-integrable (last) term must vanish. Thus, \( c_1 = -\frac{1}{3} c_2 \).
Set \( c_2 = -3 \), hence, \( c_1 = 1 \).

Result:

\[
\rho = u^3 - 3u_x^2.
\]

Expression [...] yields

\[
J = \frac{3}{4} u^4 - 6uu_x^2 + 3u_x^2 u_{2x} + 3u_{2x}^2 - 6u_x u_{3x}.
\]
Application

A Class of Fifth-Order Evolution Equations

\[ u_t + \alpha u^2 u_x + \beta u_x u_{2x} + \gamma uu_{3x} + u_{5x} = 0 \]

where \( \alpha, \beta, \gamma \) are nonzero parameters.

\[ u \sim D_x^2. \]

Special cases:

- \( \alpha = 30 \quad \beta = 20 \quad \gamma = 10 \quad \text{Lax.} \)
- \( \alpha = 5 \quad \beta = 5 \quad \gamma = 5 \quad \text{Sawada – Kotera.} \)
- \( \alpha = 20 \quad \beta = 25 \quad \gamma = 10 \quad \text{Kaup – Kupershmidt.} \)
- \( \alpha = 2 \quad \beta = 6 \quad \gamma = 3 \quad \text{Ito.} \)

What are the conditions for the parameters \( \alpha, \beta \) and \( \gamma \) so that the equation admits a density of fixed rank?

- **Rank 2:**
  No condition
  \[ \rho = u. \]

- **Rank 4:**
  Condition: \( \beta = 2\gamma \quad \text{(Lax and Ito cases)} \)
  \[ \rho = u^2. \]
– Rank 6:
Condition:

\[ 10\alpha = -2\beta^2 + 7\beta\gamma - 3\gamma^2 \]

(Lax, SK, and KK cases)

\[ \rho = u^3 + \frac{15}{(-2\beta + \gamma)} u_x^2. \]

– Rank 8:

1. \[ \beta = 2\gamma \quad \text{(Lax and Ito cases)} \]

\[ \rho = u^4 - \frac{6\gamma}{\alpha} uu_x^2 + \frac{6}{\alpha} u_{2x}^2. \]

2. \[ \alpha = -\frac{2\beta^2 - 7\beta\gamma - 4\gamma^2}{45} \quad \text{(SK, KK and Ito cases)} \]

\[ \rho = u^4 - \frac{135}{2\beta + \gamma} uu_x^2 + \frac{675}{(2\beta + \gamma)^2} u_{2x}^2. \]

– Rank 10:
Condition:

\[ \beta = 2\gamma \]

and

\[ 10\alpha = 3\gamma^2 \]

(Lax case)

\[ \rho = u^5 - \frac{50}{\gamma} u^2 u_x^2 + \frac{100}{\gamma^2} uu_{2x}^2 - \frac{500}{7\gamma^3} u_{3x}^2. \]
What are the necessary conditions for the parameters $\alpha$, $\beta$ and $\gamma$ so that the equation admits $\infty$ many polynomial conservation laws?

- If $\alpha = \frac{3}{10} \gamma^2$ and $\beta = 2\gamma$ then there is a sequence (without gaps!) of conserved densities (Lax case).

- If $\alpha = \frac{1}{5} \gamma^2$ and $\beta = \gamma$ then there is a sequence (with gaps!) of conserved densities (SK case).

- If $\alpha = \frac{1}{5} \gamma^2$ and $\beta = \frac{5}{2} \gamma$ then there is a sequence (with gaps!) of conserved densities (KK case).

- If
  \[ \alpha = \frac{2\beta^2 - 7\beta\gamma + 4\gamma^2}{45} \]
  or
  \[ \beta = 2\gamma \]
  then there is a conserved density of rank 8.

Combine both conditions: $\alpha = \frac{2\gamma^2}{9}$ and $\beta = 2\gamma$ (Ito case).
PART II: Algorithm for Lattice Equations

- Conservation Laws for Lattices.

Given: a lattice equation, continuous in time, discretized in space

\[
\dot{u}_n = F(..., u_{n-1}, u_n, u_{n+1}, ...)
\]

\(u_n\) and \(F\) are vector dynamical variables.

\(F\) is polynomial with constant coefficients.

No restrictions on the level of the shifts or the degree of nonlinearity.

**Conservation law:**

\[
\dot{\rho}_n = J_n - J_{n+1}
\]

density \(\rho_n\) and flux \(J_n\).

Both are polynomials in \(u_n\) and its shifts.

\[
\frac{d}{dt} \left( \sum_n \rho_n \right) = \sum_n \dot{\rho}_n = \sum_n (J_n - J_{n+1})
\]

if \(J_n\) is bounded for all \(n\).

With suitable boundary or periodicity conditions

\[
\sum_n \rho_n = \text{constant.}
\]
• **Example**

Consider the one-dimensional Toda lattice

\[ \ddot{y}_n = \exp (y_{n-1} - y_n) - \exp (y_n - y_{n+1}) \]

\(y_n\) is the displacement from equilibrium of the \(n\)th particle with unit mass under an exponential decaying interaction force between nearest neighbors.

Change of variables:

\[ u_n = \dot{y}_n, \quad v_n = \exp (y_n - y_{n+1}) \]

yields

\[ \dot{u}_n = v_{n-1} - v_n, \quad \dot{v}_n = v_n(u_n - u_{n+1}). \]

Toda system is completely integrable.

The first two density-flux pairs (computed by hand):

\[ \rho_n^{(1)} = u_n, \quad J_n^{(1)} = v_{n-1}, \quad \text{and} \quad \rho_n^{(2)} = \frac{1}{2}u_n^2 + v_n, \quad J_n^{(2)} = u_nv_{n-1}. \]

• **Key Concept: Dilation Invariance**

Toda system, as well as \(\rho_n^{(1)}, J_n^{(1)}, \) and \(\rho_n^{(2)}, J_n^{(2)}, \) are invariant under the dilation symmetry

\[ (t, u_n, v_n) \rightarrow (\lambda^{-1}t, \lambda u_n, \lambda^2 v_n). \]

Thus, \(u_n\) corresponds to one \(t\)-derivative: \(u_n \sim \frac{d}{dt}\). Similarly, \(v_n \sim \frac{d^2}{dt^2}\).

**Weight**, \(w\), of variables are defined in terms of \(t\)-derivatives.

Set \(w(\frac{d}{dt}) = 1\).
Weights of dependent variables are nonnegative, rational, and independent of \( n \).

Due to dilation invariance: \( w(u_n) = 1 \) and \( w(v_n) = 2 \).

The rank of a monomial is its total weight in terms of \( t \)-derivatives.

Require uniformity in rank for each equation to compute the weights (solve linear system):

\[
\begin{align*}
  w(u_n) + 1 &= w(v_n), \\
  w(v_n) + 1 &= w(u_n) + w(v_n),
\end{align*}
\]

yields \( w(u_n) = 1, w(v_n) = 2 \).

**Equivalence Criterion**

Define: \( D \) shift-down operator, and \( U \) shift-up operator, on the set of all monomials in \( u_n \) and its shifts.

For a monomial \( m \):

\[
Dm = m|_{n \rightarrow n-1}, \quad \text{and} \quad Um = m|_{n \rightarrow n+1}.
\]

For example

\[
Du_{n+2}v_n = u_{n+1}v_{n-1}, \quad Uu_{n-2}v_{n-1} = u_{n-1}v_n.
\]

Compositions of \( D \) and \( U \) define an equivalence relation. All shifted monomials are equivalent.

For example

\[
u_{n-1}v_{n+1} \equiv u_{n+2}v_{n+4} \equiv u_{n-3}v_{n-1}.
\]
**Equivalence criterion:**

Two monomials $m_1$ and $m_2$ are equivalent, $m_1 \equiv m_2$, if

$$m_1 = m_2 + [M_n - M_{n+1}]$$

for some polynomial $M_n$.

For example, $u_{n-2}u_n \equiv u_{n-1}u_{n+1}$ since

$$u_{n-2}u_n = u_{n-1}u_{n+1} + [u_{n-2}u_n - u_{n-1}u_{n+1}] = u_{n-1}u_{n+1} + [M_n - M_{n+1}].$$

**Main representative** of an equivalence class is the monomial with label $n$ on $u$ (or $v$).

For example, $u_nu_{n+2}$ is the main representative of the class with elements $u_{n-1}u_{n+1}, u_{n+1}u_{n+3}$, etc.

Use lexicographical ordering to resolve conflicts.

For example, $u_nv_{n+2}$ (not $u_{n-2}v_n$) is the main representative of the class with elements $u_{n-3}v_{n-1}, u_{n+2}v_{n+4}$, etc.

**Steps of the Algorithm for Lattices**

Three-step algorithm to find conserved densities:

1. Determine the weights.
2. Construct the form of density.
3. Determine the coefficients.
**Example:** For the Toda lattice, compute the density of rank 3.

**Step 1: Compute the weights.**

Here $w(u_n) = 1$ and $w(v_n) = 2$.

**Step 2: Construct the form of the density.**

List all monomials in $u_n$ and $v_n$ of rank 3 or less:

$$ \mathcal{G} = \{ u_n^3, u_n^2, u_nv_n, u_n, v_n \}.$$  

For each monomial in $\mathcal{G}$, introduce enough $t$-derivatives to complete its weight to 3. Use the lattice to remove $\dot{u}_n$ and $\dot{v}_n$:

$$ \frac{d^0}{dt^0}(u_n^3) = u_n^3, \quad \frac{d^0}{dt^0}(u_nv_n) = u_nv_n, $$

$$ \frac{d}{dt}(u_n^2) = 2u_nv_{n-1} - 2u_nv_n, \quad \frac{d}{dt}(v_n) = u_nv_n - u_{n+1}v_n, $$

$$ \frac{d^2}{dt^2}(u_n) = u_{n-1}v_{n-1} - u_nv_{n-1} - u_nv_n + u_{n+1}v_n.$$  

Gather the resulting terms in a set

$$ \mathcal{H} = \{ u_n^3, u_nv_{n-1}, u_nv_n, u_{n-1}v_{n-1}, u_{n+1}v_n \}. $$

Replace members in the same equivalence class by their main representatives.

For example, $u_nv_{n-1} \equiv u_{n+1}v_n$ are replaced by $u_nv_{n-1}$. Replace $\mathcal{H}$ by

$$ \mathcal{I} = \{ u_n^3, u_nv_{n-1}, u_nv_n \}$$

which has the building blocks of the conserved density.
Linearly combine the monomials in $I$

$$\rho_n = c_1 u_n^3 + c_2 u_n v_{n-1} + c_3 u_n v_n.$$ 

**Step 3: Determine the coefficients in the density.**

Require that $\dot{\rho}_n = J_n - J_{n+1}$, holds.

Compute $\dot{\rho}_n$ and use the lattice to remove $\dot{u}_n$ and $\dot{v}_n$.

Group the terms

$$\dot{\rho}_n = (3c_1 - c_2)u_n^2 v_{n-1} + (c_3 - 3c_1)u_n^2 v_n + (c_3 - c_2)v_{n-1}v_n$$

$$+ c_2 u_{n-1} u_n v_{n-1} + c_2 v_{n-1}^2 - c_3 u_n u_{n+1} v_n - c_3 v_n^2.$$ 

Use the equivalence criterion to modify $\dot{\rho}_n$.

Replace $u_{n-1} u_n v_{n-1}$ by $u_n u_{n+1} v_n + [u_{n-1} u_n v_{n-1} - u_n u_{n+1} v_n]$.

Introduce the main representatives. Thus

$$\dot{\rho}_n = (3c_1 - c_2)u_n^2 v_{n-1} + (c_3 - 3c_1)u_n^2 v_n$$

$$+ (c_3 - c_2)v_n v_{n+1} + [(c_3 - c_2)v_{n-1} v_n - (c_3 - c_2)v_n v_{n+1}]$$

$$+ c_2 u_n u_{n+1} v_n + [c_2 u_{n-1} u_n v_{n-1} - c_2 u_n u_{n+1} v_n]$$

$$+ c_2 v_n^2 + [c_2 v_{n-1}^2 - c_2 v_n^2] - c_3 u_n u_{n+1} v_n - c_3 v_n^2.$$ 

Group the terms outside of the square brackets and move the pairs inside the square brackets to the bottom.
Rearrange the latter terms so that they match the pattern \([J_n - J_{n+1}]\).

Hence

\[
\dot{\rho}_n = (3c_1 - c_2)u_n^2v_{n-1} + (c_3 - 3c_1)u_n^2v_n \\
+ (c_3 - c_2)v_nv_{n+1} + (c_2 - c_3)u_nv_{n+1}v_n + (c_2 - c_3)v_n^2 \\
\quad + \left\{ (c_3 - c_2)v_n^{-1}v_n + c_2u_{n-1}u_nv_{n-1} + c_2v_{n-1}^2 \right\} \\
\quad - \left\{ (c_3 - c_2)v_nv_{n+1} + c_2u_nv_{n+1}v_n + c_2v_n^2 \right\}.
\]

The terms inside the square brackets determine:

\[
J_n = (c_3 - c_2)v_{n-1}v_n + c_2u_{n-1}u_nv_{n-1} + c_2v_{n-1}^2.
\]

The terms outside the square brackets must vanish, thus

\[
\mathcal{S} = \{3c_1 - c_2 = 0, c_3 - 3c_1 = 0, c_2 - c_3 = 0\}.
\]

The solution is \(3c_1 = c_2 = c_3\), so choose \(c_1 = \frac{1}{3}\), and \(c_2 = c_3 = 1\):

\[
\rho_n = \frac{1}{3}u_n^3 + u_n(v_{n-1} + v_n), \quad J_n = u_{n-1}u_nv_{n-1} + v_{n-1}^2.
\]

Analogously, conserved densities of rank \(\leq 5\):

\[
\rho_n^{(1)} = u_n \quad \rho_n^{(2)} = \frac{1}{2}u_n^2 + v_n
\]

\[
\rho_n^{(3)} = \frac{1}{3}u_n^3 + u_n(v_{n-1} + v_n)
\]

\[
\rho_n^{(4)} = \frac{1}{4}u_n^4 + u_n^2(v_{n-1} + v_n) + u_nu_{n+1}v_n + \frac{1}{2}v_n^2 + v_nv_{n+1}
\]

\[
\rho_n^{(5)} = \frac{1}{5}u_n^5 + u_n^3(v_{n-1} + v_n) + u_nu_{n+1}v_n(u_n + u_{n+1}) \\
+ u_nv_{n-1}(v_{n-2} + v_{n-1} + v_n) + u_nv_n(v_{n-1} + v_n + v_{n+1}).
\]
• **Example: Nonlinear Schrödinger (NLS) equation**

Ablowitz and Ladik discretization of the NLS equation:

\[
i u_n = u_{n+1} - 2u_n + u_{n-1} + u_n^* u_n (u_{n+1} + u_{n-1}).
\]

\(u_n^*\) is the complex conjugate of \(u_n\).

Treat \(u_n\) and \(v_n = u_n^*\) as independent variables and add the complex conjugate equation. Absorb \(i\) in the scale on \(t\):

\[
\dot{u}_n = u_{n+1} - 2u_n + u_{n-1} + u_n v_n (u_{n+1} + u_{n-1}),
\]

\[
\dot{v}_n = -(v_{n+1} - 2v_n + v_{n-1}) - u_n v_n (v_{n+1} + v_{n-1}).
\]

Since \(v_n = u_n^*\), \(w(v_n) = w(u_n)\).

No uniformity in rank! Introduce an auxiliary parameter \(\alpha\) with weight.

\[
\dot{u}_n = \alpha (u_{n+1} - 2u_n + u_{n-1}) + u_n v_n (u_{n+1} + u_{n-1}),
\]

\[
\dot{v}_n = -\alpha (v_{n+1} - 2v_n + v_{n-1}) - u_n v_n (v_{n+1} + v_{n-1}).
\]

Uniformity in rank leads to

\[
w(u_n) + 1 = w(\alpha) + w(u_n) = 2w(u_n) + w(v_n) = 3w(u_n),
\]

\[
w(v_n) + 1 = w(\alpha) + w(v_n) = 2w(v_n) + w(u_n) = 3w(v_n).
\]

which yields

\[w(u_n) = w(v_n) = \frac{1}{2}, w(\alpha) = 1.\]

Uniformity in rank is essential for steps 1 and 2.

After Step 2, set \(\alpha = 1\). Step 3 leads to the result:

\[
\rho_n^{(1)} = c_1 u_n v_{n-1} + c_2 u_n v_{n+1}, \quad \text{etc.}
\]
• **Scope and Limitations of Algorithms & Software**

- Systems of evolution equations or lattice equations must be polynomial in dependent variables.
  No *explicitly* dependencies on the independent variables.
- Only one space variable (continuous or discretized) is allowed.
- Program only computes polynomial conservation laws.
- Program computes conservation laws that explicitly depend on the independent variables, if the degree of dependency is assigned.
- No limit on the number of equations in the system.
  In practice: time and memory constraints.
- Input systems may have (nonzero) parameters.
  Program computes the compatibility conditions for parameters such that conservation laws (of a given rank) exist.
- Systems can also have parameters with (unknown) weight.
  This allows one to test evolution and lattice equations of non-uniform rank.
- For systems where one or more of the weights are free, the program prompts the user for info.
- Fractional weights and ranks are permitted.
- Complex dependent variables are allowed.
- PDE or lattice must be first-order in $t$. 


• Publications – Software


