Towards a classification of evolution equations with Lax pairs over the octonions

2024 CMS Winter Meeting — Vancouver

T. Wolf, S. Anco, P. Lam

Brock University, St Catharines

December 2, 2024

Outline

About Lax Pairs

- An evolution equation $u_t = F(u, u_x, u_{xx}, \dots)$ is said to be integrable if it has a **Lax Pair** L and M that are linear differential operators in terms of ∂_x with coefficients in $u, u_x, u_{xx}, ..$ satisfying

$$L_t = [M, L]$$

identically in all $u, u_x, u_{xx}, ..$ iff $u_t = F$.

About Lax Pairs

- An evolution equation $u_t = F(u, u_x, u_{xx}, \dots)$ is said to be integrable if it has a **Lax Pair** L and M that are linear differential operators in terms of ∂_x with coefficients in $u, u_x, u_{xx}, ..$ satisfying

$$L_t = [M, L]$$

  identically in all $u, u_x, u_{xx}, ..$ iff $u_t = F$.

- Lax pair is used in the inverse scattering transform to generate multi-soliton solutions which have many physical applications.

About Lax Pairs

- An evolution equation $u_t = F(u, u_x, u_{xx}, \dots)$ is said to be integrable if it has a **Lax Pair** L and M that are linear differential operators in terms of ∂_x with coefficients in $u, u_x, u_{xx}, ..$ satisfying

$$L_t = [M, L]$$

identically in all $u, u_x, u_{xx}, ..$ iff $u_t = F$.

- Lax pair is used in the inverse scattering transform to generate multi-soliton solutions which have many physical applications.
- Two well known examples:

$u_t = uu_x + u_{xxx}$ Korteweg-De Vries (KdV) equation

$L = \partial_x^2 + \frac{1}{6}u, \qquad M = 4\partial_x^3 + u\partial_x + \frac{1}{2}u_x \qquad (w(L) = 2)$

$u_t = u^2 u_x + u_{xxx}$ modified Korteweg-De Vries (mKdV) equation

$L = \partial_x + u, \qquad M = -u_{xx} - \frac{1}{3}u^3 \qquad (w(L) = 1)$

$L = \partial_x^2 + 2u\partial_x + u^2 + u_x, \qquad M = -u_{xx} - \frac{1}{3}u^3 \qquad (w(L) = 2)$

About Lax Pairs

- An evolution equation $u_t = F(u, u_x, u_{xx}, \dots)$ is said to be integrable if it has a **Lax Pair** L and M that are linear differential operators in terms of ∂_x with coefficients in $u, u_x, u_{xx}, ..$ satisfying

$$L_t = [M, L]$$

identically in all $u, u_x, u_{xx}, ..$ iff $u_t = F$.

- Lax pair is used in the inverse scattering transform to generate multi-soliton solutions which have many physical applications.
- Two well known examples:

$u_t = uu_x + u_{xxx}$ Korteweg-De Vries (KdV) equation

$L = \partial_x^2 + \frac{1}{6}u, \qquad M = 4\partial_x^3 + u\partial_x + \frac{1}{2}u_x \qquad (w(L) = 2)$

$u_t = u^2u_x + u_{xxx}$ modified Korteweg-De Vries (mKdV) equation

$L = \partial_x + u, \qquad M = -u_{xx} - \frac{1}{3}u^3 \qquad (w(L) = 1)$

$L = \partial_x^2 + 2u\partial_x + u^2 + u_x, \qquad M = -u_{xx} - \frac{1}{3}u^3 \qquad (w(L) = 2)$

About Octonions I

- Cayley-Dickson construction: real, complex, quaternions, octonions, sedenions,... by introducing 1 new imaginary number each time

About Octonions I

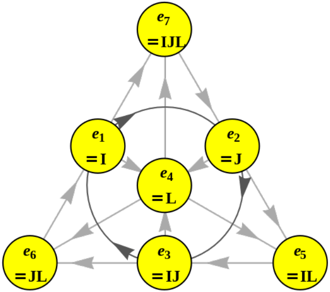
- Cayley-Dickson construction: real, complex, quaternions, octonions, sedenions,... by introducing 1 new imaginary number each time
- 8-dimensional algebra formed by 1 real and 7 imaginary basis elements

About Octonions I

- Cayley-Dickson construction: real, complex, quaternions, octonions, sedenions,... by introducing 1 new imaginary number each time
- 8-dimensional algebra formed by 1 real and 7 imaginary basis elements
- normed division algebra over the real numbers

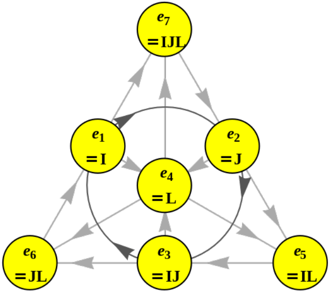
About Octonions II

- noncommutative, nonassociative



About Octonions II

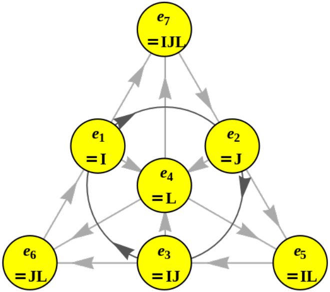
- noncommutative, nonassociative



- but
  alternative: $x(xy) = (xx)y, \quad (yx)x = y(xx),$

About Octonions II

- noncommutative, nonassociative



- but
  alternative: $x(xy) = (xx)y, \quad (yx)x = y(xx)$,
  as a consequence the
  associator $[x, y, z] := (xy)z - x(yz)$
  satisfies
  $[x, x, y] = [y, x, x] = 0$
  and as a consequence of that also
  $[x, y, x] = 0$.

Applications of Octonions

- appear in attempts to understand and extend the Standard Model of elementary particle physics and string theory C. Furey, Phys. Rev. D 86, 025024 (2012); T.P.Singh, Z.Naturforsch. A 75, 1051 (2020)

Applications of Octonions

- appear in attempts to understand and extend the Standard Model of elementary particle physics and string theory C. Furey, Phys. Rev. D 86, 025024 (2012); T.P.Singh, Z.Naturforsch. A 75, 1051 (2020)

- other applications in quantum logic, special relativity and supersymmetry John C. Baez `https://arxiv.org/abs/math/0105155` (2002)

Applications of Octonions

- appear in attempts to understand and extend the Standard Model of elementary particle physics and string theory C. Furey, Phys. Rev. D 86, 025024 (2012); T.P.Singh, Z.Naturforsch. A 75, 1051 (2020)
- other applications in quantum logic, special relativity and supersymmetry John C. Baez `https://arxiv.org/abs/math/0105155` (2002)
- little literature exists on specific integrable evolution equations over octonions A. Restuccia, A. Sotomayor, J.P. Veiro, arXiv:1609.05410v1 [math-ph] (2016)

Goal

- Integrable scalar evolution equations have been extended to multi-component (coupled) evolution equations (e.g. over vectors, matrices, Lie algebras, graded algebras).

Goal

- Integrable scalar evolution equations have been extended to multi-component (coupled) evolution equations (e.g. over vectors, matrices, Lie algebras, graded algebras).
- This project: Integrable evolution equations over the octonions

Goal

- Integrable scalar evolution equations have been extended to multi-component (coupled) evolution equations (e.g. over vectors, matrices, Lie algebras, graded algebras).
- This project: Integrable evolution equations over the octonions
- Start with KdV and mKdV type that have a Lax pair

Goal

- Integrable scalar evolution equations have been extended to multi-component (coupled) evolution equations (e.g. over vectors, matrices, Lie algebras, graded algebras).
- This project: Integrable evolution equations over the octonions
- Start with KdV and mKdV type that have a Lax pair
- Later goal: Classification

Outline

General Idea

- Select weights for $\partial_t, \partial_x, u, L$,
  e.g. KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, and start with $w(L) = 2$
  $\rightarrow\ w(M) = w(L_t) - w(L) = w(\partial_t) = 3$
  $\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$

General Idea

- Select weights for $\partial_t, \partial_x, u, L$,
  e.g. KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, and start with $w(L) = 2$
  $\rightarrow\ w(M) = w(L_t) - w(L) = w(\partial_t) = 3$
  $\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
- Make general homogenous polynomial ansatz for $F(= u_t), L, M$ with undetermined constant coefficients f_j, l_j, m_j and octonion variables $u, u_x, u_{xx}, ...$

General Idea

- Select weights for $\partial_t, \partial_x, u, L$,
  e.g. KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, and start with $w(L) = 2$
  $\rightarrow\ w(M) = w(L_t) - w(L) = w(\partial_t) = 3$
  $\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
- Make general homogenous polynomial ansatz for $F(= u_t), L, M$ with undetermined constant coefficients f_j, l_j, m_j and octonion variables $u, u_x, u_{xx}, ...$
- Formulate $L_t = [M, L]$

General Idea

- Select weights for $\partial_t, \partial_x, u, L$,
  e.g. KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, and start with $w(L) = 2$
  $\rightarrow\ w(M) = w(L_t) - w(L) = w(\partial_t) = 3$
  $\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
- Make general homogenous polynomial ansatz for $F(= u_t), L, M$ with undetermined constant coefficients f_j, l_j, m_j and octonion variables $u, u_x, u_{xx}, ...$
- Formulate $L_t = [M, L]$
- Split wrt. $u, u_x, u_{xx}, ...$

General Idea

- Select weights for $\partial_t, \partial_x, u, L$,
  e.g. KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, and start with $w(L) = 2$
  $\rightarrow\ w(M) = w(L_t) - w(L) = w(\partial_t) = 3$
  $\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
- Make general homogenous polynomial ansatz for $F(= u_t), L, M$ with undetermined constant coefficients f_j, l_j, m_j and octonion variables $u, u_x, u_{xx}, \ldots$
- Formulate $L_t = [M, L]$
- Split wrt. $u, u_x, u_{xx}, \ldots$
- Solve the overdetermined non-linear polynomial system for unknown coefficients f_j, l_j, m_j to obtain the integrable equation $u_t = F$ and Lax pair L, M.

Adapting to Non-commutativity and Non-Associativity

Instead of linear *differential operators* L, M

- Introduce zero weight $G(x)$

Adapting to Non-commutativity and Non-Associativity

Instead of linear *differential operators* L, M

- Introduce zero weight $G(x)$
- Make polynomial ansatz for LG, MG, linear homogeneous in $G, G_x, ...$

Adapting to Non-commutativity and Non-Associativity

Instead of linear *differential operators* L, M

- Introduce zero weight $G(x)$
- Make polynomial ansatz for LG, MG, linear homogeneous in $G, G_x, ...$
- Instead of $L_t G = [M, L]G$ formulate $(LG)_t = M(LG) - L(MG)$.

Adapting to Non-commutativity and Non-Associativity

Instead of linear *differential operators* L, M

- Introduce zero weight $G(x)$
- Make polynomial ansatz for LG, MG, linear homogeneous in $G, G_x, ...$
- Instead of $L_t G = [M, L]G$ formulate $(LG)_t = M(LG) - L(MG)$.
- To compute $(LG)_t$ replace $u_t = F, \;\; u_{tx} = dF/dx, ...$

Adapting to Non-commutativity and Non-Associativity

Instead of linear *differential operators* L, M

- Introduce zero weight $G(x)$
- Make polynomial ansatz for LG, MG, linear homogeneous in $G, G_x, ...$
- Instead of $L_tG = [M, L]G$ formulate $(LG)_t = M(LG) - L(MG)$.
- To compute $(LG)_t$ replace $u_t = F, \;\; u_{tx} = dF/dx, ...$
- To compute $M(LG)$ replace in MG each $G, G_x, ..$ 'in place' by $LG, (LG)_x, ...$
- To compute $L(MG)$ replace in LG each $G, G_x, ..$ 'in place' by $MG, (MG)_x, ...$

KdV Example

KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, try $w(LG) = w(L) = 2$
$\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
$\rightarrow\ w(MG) = w(M) = w(\partial_t) = 3$

$$
\begin{aligned}
F &= f_1 u_{xxx} + f_2 u u_x + f_3 u_x u \\
LG &= l_1 G_{xx} + l_2 G u + l_3 u G \\
MG &= m_1 G_{xxx} + m_2 G_x u + m_3 u G_x + m_4 G u_x + m_5 u_x G
\end{aligned}
$$

KdV Example

KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, try $w(LG) = w(L) = 2$
$\to \; w(F) = w(u) + w(\partial_t) = 5$
$\to \; w(MG) = w(M) = w(\partial_t) = 3$

$$
\begin{aligned}
F &= f_1 u_{xxx} + f_2 u u_x + f_3 u_x u \\
LG &= l_1 G_{xx} + l_2 G u + l_3 u G \\
MG &= m_1 G_{xxx} + m_2 G_x u + m_3 u G_x + m_4 G u_x + m_5 u_x G \\
(LG)_t &= l_2 G F + l_3 F G
\end{aligned}
$$

KdV Example

KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, try $w(LG) = w(L) = 2$
$\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
$\rightarrow\ w(MG) = w(M) = w(\partial_t) = 3$

$$\begin{aligned}
F &= f_1 u_{xxx} + f_2 u u_x + f_3 u_x u \\
LG &= l_1 G_{xx} + l_2 G u + l_3 u G \\
MG &= m_1 G_{xxx} + m_2 G_x u + m_3 u G_x + m_4 G u_x + m_5 u_x G \\
(LG)_t &= l_2 G F + l_3 F G \\
L(MG) &= l_1 (MG)_{xx} + l_2 (MG) u + l_3 u (MG)
\end{aligned}$$

KdV Example

KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, try $w(LG) = w(L) = 2$
$\rightarrow\ w(F) = w(u) + w(\partial_t) = 5$
$\rightarrow\ w(MG) = w(M) = w(\partial_t) = 3$

$$
\begin{aligned}
F &= f_1 u_{xxx} + f_2 u u_x + f_3 u_x u \\
LG &= l_1 G_{xx} + l_2 G u + l_3 u G \\
MG &= m_1 G_{xxx} + m_2 G_x u + m_3 u G_x + m_4 G u_x + m_5 u_x G \\
(LG)_t &= l_2 G F + l_3 F G \\
L(MG) &= l_1 (MG)_{xx} + l_2 (MG) u + l_3 u (MG) \\
M(LG) &= m_1 (LG)_{xxx} + m_2 (LG)_x u + m_3 u (LG)_x + \\
&\quad m_4 (LG) u_x + m_5 u_x (LG)
\end{aligned}
$$

KdV Example

KdV-scaling: $w(\partial_t) = 3$, $w(\partial_x) = 1$, $w(u) = 2$, try $w(LG) = w(L) = 2$
$\rightarrow$ $w(F) = w(u) + w(\partial_t) = 5$
$\rightarrow$ $w(MG) = w(M) = w(\partial_t) = 3$

$$
\begin{aligned}
F &= f_1 u_{xxx} + f_2 u u_x + f_3 u_x u \\
LG &= l_1 G_{xx} + l_2 G u + l_3 u G \\
MG &= m_1 G_{xxx} + m_2 G_x u + m_3 u G_x + m_4 G u_x + m_5 u_x G \\
(LG)_t &= l_2 G F + l_3 F G \\
L(MG) &= l_1 (MG)_{xx} + l_2 (MG) u + l_3 u (MG) \\
M(LG) &= m_1 (LG)_{xxx} + m_2 (LG)_x u + m_3 u (LG)_x + \\
&\quad m_4 (LG) u_x + m_5 u_x (LG)
\end{aligned}
$$

Problem: For high $w(\partial_t), w(L)$ and low $w(\partial_x), w(u)$ the number of terms goes into the 100s..1000s

Adapting to Identities for Octonions

Spliting wrt. $u, u_x, u_{xx}, ...$ is too restrictive because of **polynomial identities** of octonions, like
$0 = (wu)v - w(uv) + (vu)w - v(uw)$

Adapting to Identities for Octonions

Spliting wrt. $u, u_x, u_{xx}, \ldots$ is too restrictive because of **polynomial identities** of octonions, like
$0 = (wu)v - w(uv) + (vu)w - v(uw)$

Instead

- Replace $u, u_x, ..$ by their component form $u = \sum_{i=0}^{7} u_i(x,t) e_i, \ldots$

Adapting to Identities for Octonions

Spliting wrt. $u, u_x, u_{xx}, ...$ is too restrictive because of **polynomial identities** of octonions, like
$0 = (wu)v - w(uv) + (vu)w - v(uw)$

Instead

- Replace $u, u_x, ..$ by their component form $u = \sum_{i=0}^{7} u_i(x,t)e_i, ...$
- Split wrt. $e_i, u_i, u_{xi}, ...$

Adapting to Identities for Octonions

Spliting wrt. $u, u_x, u_{xx}, ...$ is too restrictive because of **polynomial identities** of octonions, like
$0 = (wu)v - w(uv) + (vu)w - v(uw)$

Instead

- Replace $u, u_x, ..$ by their component form $u = \sum_{i=0}^{7} u_i(x,t)e_i, ...$
- Split wrt. $e_i, u_i, u_{xi}, ...$

Consequences:

- Huge computational cost,
- A new necessary research project:
  Find and use **all** octonion identities, not only the known fully antisymmetric ones and also those with repeating factors in a term

Adapting to Identities for Octonions

Spliting wrt. $u, u_x, u_{xx}, ...$ is too restrictive because of **polynomial identities** of octonions, like
$0 = (wu)v - w(uv) + (vu)w - v(uw)$

Instead

- Replace $u, u_x, ..$ by their component form $u = \sum_{i=0}^{7} u_i(x,t)e_i, ...$
- Split wrt. $e_i, u_i, u_{xi}, ...$

Consequences:

- Huge computational cost,
- A new necessary research project:
  Find and use **all** octonion identities, not only the known fully antisymmetric ones and also those with repeating factors in a term

Outline

Outline

Algebraic Challenges

- Is a given octonion polynomial identically zero?

Algebraic Challenges

- Is a given octonion polynomial identically zero?
- How to simplify an octonion polynomial?

Algebraic Challenges

- Is a given octonion polynomial identically zero?
- How to simplify an octonion polynomial?
- How to decide ideal membership?

Algebraic Challenges

- Is a given octonion polynomial identically zero?
- How to simplify an octonion polynomial?
- How to decide ideal membership?
- Find all identically vanishing polynomials up to some degree.

Algebraic Challenges

- Is a given octonion polynomial identically zero?
- How to simplify an octonion polynomial?
- How to decide ideal membership?
- Find all identically vanishing polynomials up to some degree.
- Find all central (real) polynomials up to some degree.

Outline

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k
- Replace variables by their component form, e.g. $u = \sum_{i=0}^{7} u_i e_i$

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k
- Replace variables by their component form, e.g. $u = \sum_{i=0}^{7} u_i e_i$
- Split $P = 0$ w.r.t. $e_i, u_i, ...$

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k
- Replace variables by their component form, e.g. $u = \sum_{i=0}^{7} u_i e_i$
- Split $P = 0$ w.r.t. $e_i, u_i, ...$
- Solve the linear system for the undetermined coefficients c_k

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k
- Replace variables by their component form, e.g. $u = \sum_{i=0}^{7} u_i e_i$
- Split $P = 0$ w.r.t. $e_i, u_i, ...$
- Solve the linear system for the undetermined coefficients c_k
- Substitute general solution into P

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k
- Replace variables by their component form, e.g. $u = \sum_{i=0}^{7} u_i e_i$
- Split $P = 0$ w.r.t. $e_i, u_i, ...$
- Solve the linear system for the undetermined coefficients c_k
- Substitute general solution into P
- Get all IDs (identities) as coefficients of free parameters in P

Vanishing Identities $P = 0$

Outline

- Formulate a general polynomial P of degree d in n octonion variables $u, ..$ with undetermined coefficients c_k
- Replace variables by their component form, e.g. $u = \sum_{i=0}^{7} u_i e_i$
- Split $P = 0$ w.r.t. $e_i, u_i, ...$
- Solve the linear system for the undetermined coefficients c_k
- Substitute general solution into P
- Get all IDs (identities) as coefficients of free parameters in P
- Find linear combinations of identities and permutations of them that are short, highly symmetric to allow a compact formulation.

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t\ =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$
$e =$ # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $d! \times m(d)$ (multilinearity)
τ = # of terms of P in expanded form = $t \times 8^d$
c = # of real/imag. components of all octonion variables = $8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$
$e =$ # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $d! \times m(d)$ (multilinearity)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}
c	8	16	24	32	40	48	56	64

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $d! \times m(d)$ (multilinearity)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}
c	8	16	24	32	40	48	56	64
i	0	0						

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m =$ # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$
$e =$ # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}
c	8	16	24	32	40	48	56	64
i	0	0	5					

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $d! \times m(d)$ (multilinearity)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d=n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}
c	8	16	24	32	40	48	56	64
i	0	0	5	88				

The Computational Complexity of Multilinearity

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $d! \times m(d)$ (multilinearity)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}
c	8	16	24	32	40	48	56	64
i	0	0	5	88	1530			

The Computational Complexity of Multilinearity

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m =$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $d! \times m(d)$ (multilinearity)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$
$e =$ # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6	7	8
m	1	1	2	5	14	42	132	429
t	1	2	12	120	1680	30240	6.65×10^5	1.69×10^7
τ	8	128	6140	491520	55×10^6	7.9×10^9	1.39×10^{12}	2.84×10^{14}
c	8	16	24	32	40	48	56	64
i	0	0	5	88	1530	?	?	?
e	1	2	7	32	150	?	?	?

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)

The Computational Complexity of Repeating Factors

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m =$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)

The Computational Complexity of Repeating Factors

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $n^d \times m(d)$ (factors may repeat)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$

The Computational Complexity of Repeating Factors

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $n^d \times m(d)$ (factors may repeat)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$

The Computational Complexity of Repeating Factors

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $n^d \times m(d)$ (factors may repeat)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42

The Computational Complexity of Repeating Factors

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m =$ # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of $P : n^d \times m(d)$ (factors may repeat)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$
$e =$ # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form = $t \times 8^d$
c = # of real/imag. components of all octonion variables = $8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}
c	8	16	24	32	40	48

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form = $t \times 8^d$
c = # of real/imag. components of all octonion variables = $8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}
c	8	16	24	32	40	48
i	0	0				

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form = $t \times 8^d$
c = # of real/imag. components of all octonion variables = $8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}
c	8	16	24	32	40	48
i	0	0	26			

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form = $t \times 8^d$
c = # of real/imag. components of all octonion variables = $8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}
c	8	16	24	32	40	48
i	0	0	26	992		

The Computational Complexity of Repeating Factors

n = number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
d = degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
m= # of different ways to non-associative multiply the d factors of 1 term,
$m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
t = # of terms of P : $n^d \times m(d)$ (factors may repeat)
τ = # of terms of P in expanded form $= t \times 8^d$
c = # of real/imag. components of all octonion variables $= 8n$
i = # of identities = # of free coeff. in general solution of $P = 0$
e = # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}
c	8	16	24	32	40	48
i	0	0	26	992	40375	

The Computational Complexity of Repeating Factors

$n =$ number of octonion variables $u, v, w..$ (in application $u, u_x, u_{2x}, ..$)
$d =$ degree of polynomial $P(u, v, ..)$ (in the application $P = L, M$)
$m=$ # of different ways to non-associative multiply the d factors of 1 term, $m(1) = 1,\ m(d) = \sum_{i=1}^{d-1} m(i) \times m(d-i)$ (recursive formula summing over all $d-1$ options for the last of the $d-1$ multiplications)
$t =$ # of terms of P : $n^d \times m(d)$ (factors may repeat)
$\tau =$ # of terms of P in expanded form $= t \times 8^d$
$c =$ # of real/imag. components of all octonion variables $= 8n$
$i =$ # of identities = # of free coeff. in general solution of $P = 0$
$e =$ # of essential terms in P which is $t - z$

$d = n$	1	2	3	4	5	6
m	1	1	2	5	14	42
t	1	4	54	1280	43750	1.95×10^6
τ	8	256	9213	5.24×10^6	1.4336×10^9	5.13×10^{11}
c	8	16	24	32	40	48
i	0	0	26	992	40375	?
e	1	4	28	288	3375	?

Central Polynomials

A polynomial $P = P(x, y, \ldots)$ is a *central* polynomial if P is real for any octonion variables $x, y, ...$ and thus commutes with any other octonian variable u:

$$[P, u] = 0$$

and thus also satisfies the vanishing identity

$$[P, u, v] = (Pv)w - P(vw) = P(vw) - P(vw) = 0$$

for any octonions u, v.

Central Polynomials

A polynomial $P = P(x, y, \ldots)$ is a *central* polynomial if P is real for any octonion variables $x, y, \ldots$ and thus commutes with any other octonian variable u:

$$[P, u] = 0$$

and thus also satisfies the vanishing identity

$$[P, u, v] = (Pv)w - P(vw) = P(vw) - P(vw) = 0$$

for any octonions u, v.

Same procedure to compute them, only ignore coefficient of e_0 after splitting w.r.t. e_i.

Outline

Known Minimal Degree Central Polynomials

Racine (1986) [3], Hentzel, Peresi (1996) [4], Shestakov, Zhukavet (2009) [5]:

degree 1,2,3: None

degree 4:
$$[a, b] \circ [c, d], \tag{1}$$

where $x \circ y := xy + yx$,

Known Minimal Degree Central Polynomials

Racine (1986) [3], Hentzel, Peresi (1996) [4],
Shestakov, Zhukavet (2009) [5]:

degree 1,2,3: None

degree 4:
$$[a,b]\circ[c,d], \tag{1}$$

where $x\circ y := xy + yx$,

degree 5:
$$\sum_{\text{alt}}\{24a(b(c(de))) + 8a([b,c,d]e) - 11[a,b,[c,d,e]]\}, \tag{2}$$

where $\sum$ is the alternating sum over the arguments.

Known Minimal Degree Central Polynomials

Racine (1986) [3], Hentzel, Peresi (1996) [4],
Shestakov, Zhukavet (2009) [5]:

degree 1,2,3: None

degree 4:
$$[a, b] \circ [c, d], \tag{1}$$

where $x \circ y := xy + yx$,

degree 5:
$$\sum_{\text{alt}} \{24a(b(c(de))) + 8a([b, c, d]e) - 11[a, b, [c, d, e]]\}, \tag{2}$$

where $\sum$ is the alternating sum over the arguments.

degree 6: No new ones.

Known Minimal Degree Identities

degree 1, 2: None
degree 3: Just the alternative laws
degree 4: No new ones
degree 5:

$$[[a, b] \circ [c, d], e] = 0, \tag{3}$$

Known Minimal Degree Identities

degree 1, 2: None
degree 3: Just the alternative laws
degree 4: No new ones
degree 5:

$$[[a, b] \circ [c, d], e] = 0, \tag{3}$$

$$\overline{P}_3(x^2) - \overline{P}_3(x) \circ x = 0, \tag{4}$$

where $V_x(y) := x \circ y$ and $\overline{P}_3$ is defined by

$$\overline{P}_3 = V_a V_b V_c + V_c V_a V_b + V_b V_c V_a - V_b V_a V_c - V_a V_c V_b - V_c V_b V_a$$

Known Minimal Degree Identities

degree 1, 2: None
degree 3: Just the alternative laws
degree 4: No new ones
degree 5:

$$[[a,b]\circ[c,d],e]=0, \tag{3}$$

$$\overline{P}_3(x^2)-\overline{P}_3(x)\circ x=0, \tag{4}$$

where $V_x(y) := x\circ y$ and $\overline{P}_3$ is defined by

$$\overline{P}_3 = V_aV_bV_c + V_cV_aV_b + V_bV_cV_a - V_bV_aV_c - V_aV_cV_b - V_cV_bV_a$$

degree 6:

$$\left[\sum_{\text{alt}}\{24a(b(c(de)))+8a([b,c,d]e)-11[a,b,[c,d,e]]\},f\right]=0, \tag{5}$$

Outline

Degree 3 Vanishing Identities with Repeating Factors

Alternative laws $[u, u, v] = 0, [v, u, u] = 0$ give

$$[u, v, w] = [u, v, w] - [u + w, v, u + w] = ... = -[w, v, u]$$

and further total antisymmetry:

$$[u, v, w] = [v, w, u] = [w, u, v] = -[v, u, w] = -[u, w, v] = -[w, v, u]$$

Degree 3 Vanishing Identities with Repeating Factors

Alternative laws $[u, u, v] = 0, [v, u, u] = 0$ give

$$[u, v, w] = [u, v, w] - [u + w, v, u + w] = ... = -[w, v, u]$$

and further total antisymmetry:

$$[u, v, w] = [v, w, u] = [w, u, v] = -[v, u, w] = -[u, w, v] = -[w, v, u]$$

This is an example for equivalence of a (not fully skey symmetric 3-variable ID to a 2-variable IDs.

Degree 3 Vanishing Identities with Repeating Factors

Alternative laws $[u,u,v]=0,[v,u,u]=0$ give

$$[u,v,w]=[u,v,w]-[u+w,v,u+w]=...=-[w,v,u]$$

and further total antisymmetry:

$$[u,v,w]=[v,w,u]=[w,u,v]=-[v,u,w]=-[u,w,v]=-[w,v,u]$$

This is an example for equivalence of a (not fully skey symmetric 3-variable ID to a 2-variable IDs.

Such IDs of degree >3 are not systematically investigated sofar but needed for reducing polynomials.

Degree 3 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
$n = d = 3$ with *repeating factors*

Reductions	t	i	e
none	54	26	28
alternative laws	33	5	28
$(wu)v = ..., w > v, [w,u,v] = -[v,u,w]$	30	2	28
$(wu)v = ..., u \geq v, [w,u,v] = +[u,v,w]$	29	1	28
$(wu)v = ..., w \geq u, [w,u,v] = +[v,w,u]$	28	0	28

Degree 3 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
$n = d = 3$ with *repeating factors*

Reductions	t	i	e
none	54	26	28
alternative laws	33	5	28
$(wu)v = ..., w > v, [w, u, v] = -[v, u, w]$	30	2	28
$(wu)v = ..., u \geq v, [w, u, v] = +[u, v, w]$	29	1	28
$(wu)v = ..., w \geq u, [w, u, v] = +[v, w, u]$	28	0	28

The 26 identities included permutations of non-(skew)symmetric identities.

Degree 3 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
$n = d = 3$ with *repeating factors*

Reductions	t	i	e
none	54	26	28
alternative laws	33	5	28
$(wu)v = ..., w > v, [w, u, v] = -[v, u, w]$	30	2	28
$(wu)v = ..., u \geq v, [w, u, v] = +[u, v, w]$	29	1	28
$(wu)v = ..., w \geq u, [w, u, v] = +[v, w, u]$	28	0	28

The 26 identities included permutations of non-(skew)symmetric identities.

e=28=const. $\rightarrow$ All terms dropped from P were redundant.

Degree 3 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
$n = d = 3$ with *repeating factors*

Reductions	t	i	e
none	54	26	28
alternative laws	33	5	28
$(wu)v = ..., w > v, [w,u,v] = -[v,u,w]$	30	2	28
$(wu)v = ..., u \geq v, [w,u,v] = +[u,v,w]$	29	1	28
$(wu)v = ..., w \geq u, [w,u,v] = +[v,w,u]$	28	0	28

The 26 identities included permutations of non-(skew)symmetric identities.

e=28=const. $\rightarrow$ All terms dropped from P were redundant.

Finally $i = 0 \rightarrow$ All redundant terms from P were dropped.

Degree 3 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
$n = d = 3$ with *repeating factors*

Reductions	t	i	e
none	54	26	28
alternative laws	33	5	28
$(wu)v = ..., w > v, [w, u, v] = -[v, u, w]$	30	2	28
$(wu)v = ..., u \geq v, [w, u, v] = +[u, v, w]$	29	1	28
$(wu)v = ..., w \geq u, [w, u, v] = +[v, w, u]$	28	0	28

The 26 identities included permutations of non-(skew)symmetric identities.

e=28=const. $\rightarrow$ All terms dropped from P were redundant.

Finally $i = 0 \rightarrow$ All redundant terms from P were dropped.

List of used identities is necessary and sufficient for this purpose.

Degree 4 Minimal General Polynomials

Identities satisfied by Moufang loops (Ruth Moufang 1935) [1]

$$\begin{aligned} z(x(zy)) &= ((zx)z)y \\ x(z(yz)) &= ((xz)y)z \\ (zx)(yz) &= (z(xy))z \\ (zx)(yz) &= z((xy)z) \end{aligned}$$

Degree 4 Minimal General Polynomials

Identities satisfied by Moufang loops (Ruth Moufang 1935) [1]

$$\begin{aligned} z(x(zy)) &= ((zx)z)y \\ x(z(yz)) &= ((xz)y)z \\ (zx)(yz) &= (z(xy))z \\ (zx)(yz) &= z((xy)z) \end{aligned}$$

Equivalent formulations in terms of associators:

$$w[u,v,w] = [u,vu,w] = [u,v,wu]$$
$$[u,v,w]u = [u,uv,w] = [u,v,uw]$$

Reverse Polynomials

Lemma: If P is a polynomial of octonion variables vanishing identically $P = 0$ then the reverse polynomial $R(P)$ vanishes too, $R(P) = 0$.

Example:

$$\begin{aligned} 0 &= (v[z,u,w] + [u,v,wz])_{\{v,z\}} \\ 0 &= ([zw,v,u] + [w,u,z]v)_{\{v,z\}} \end{aligned}$$

Reverse Polynomials

Lemma: If P is a polynomial of octonion variables vanishing identically $P = 0$ then the reverse polynomial $R(P)$ vanishes too, $R(P) = 0$.
Example:

$$
\begin{aligned}
0 &= (v[z,u,w] + [u,v,wz])_{\{v,z\}} \\
0 &= ([zw,v,u] + [w,u,z]v)_{\{v,z\}}
\end{aligned}
$$

are equivalent to

$$0 = [u,v,wz]_{\{u,w\}\{v,z\}}$$

modulo anti-symmetry of associators despite being the result of another symmetrization.

An Identity for General Non-associative Algebras

Qualitatively different:
Associator identity not using alternating property, valid for any non-associative algebra

$$0 = u[v,w,z] - [uv,w,z] + [u,vw,z] - [u,v,wz] + [u,v,w]z$$

Not useful to remove terms but for manual proofs

An Identity for General Non-associative Algebras

Qualitatively different:
Associator identity not using alternating property, valid for any non-associative algebra

$$0 = u[v, w, z] - [uv, w, z] + [u, vw, z] - [u, v, wz] + [u, v, w]z$$

Not useful to remove terms but for manual proofs

Palindrome identity after $u \leftrightarrow z,\ y \leftrightarrow w$.

Degree 4 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
Example: $n = d = 4$ *multilinear* case

Reductions	t	i	e
none	120	88	32
$(wu)v = ..., w > v, [w, u, v] = -[v, u, w]$	72	40	32
$(wu)v = ..., u \geq v,\ [w, u, v] = +[u, v, w]$	56	24	32
$(wu)v = ..., w \geq u, [w, u, v] = +[v, w, u]$	40	8	32
$(uv)(wx) = ..., v \geq x, 0 = [u, v, wz]_{\{v,w\}\{u,z\}}$	32	0	32

Degree 4 Minimal General Polynomials

Reductions require all identities, not only alternative laws.
Example: $n = d = 4$ *repeating factors*

Reductions	t	i	e
none	1280	992	288
alternative laws	784	496	288
identity in 2 factor products	712	424	288
$(wu)v = ..., w > v, [w,u,v] = -[v,u,w]$	520	232	288
$(wu)v = ..., u \geq v,\ [w,u,v] = +[u,v,w]$	432	144	288
$(wu)v = ..., w \geq u, [w,u,v] = +[v,w,u]$	344	56	288
$(uv)(wx) = ..., v \geq x, 0 = [u,v,wz]_{\{v,w\}\{u,z\}}$	288	0	288

Degree 5 Minimal General Polynomials I

Reductions require all identities, not only alternative laws.
Example: $n = d = 5$ *multiliear* polynomial

Reductions	t	i	e
none	1680	1530	150
$(wu)v = ..., w > v, [w,u,v] = -[v,u,w]$	790	640	150
$(wu)v = ..., u \geq v,\ [w,u,v] = +[u,v,w]$	525	375	150
$(wu)v = ..., w \geq u, [w,u,v] = +[v,w,u]$	330	180	150
$(uv)(wx) = ..., v \geq x, 0 = [u,v,wz]_{\{v,w\}\{u,z\}}$	226	76	150

Degree 5 Minimal General Polynomials II

$n = d = 5$ *multiliear* polynomial

Reductions	t	i	e
$(pr)(u(qs)) = ..., p < q, r < s$ $0 = ([pr(u(qs))] - p(r[uqs]))_{\{pq\}\{rs\}}$	211	61	150
$0 = [p, \text{real of degree } 4]$	186	36	150
$(rp)((qs)u) = ..., p < q, r < s$ $0 = (-(rp)[qsu] + p(r[qsu]) - (ps)[rqu] + s(p[rqu]))_{\{pq\}}$	170	20	150
$(pr)(q(su)) = ..., p < q, q < r, r < s$ $0 = (+[pr(q(su))] + [pr(u(sq))]$ $-[pr(s(qu))] + p(r[qus])))_{\{pq\}\{rs\}}$	169	19	150
$(pr)((sq)u) = ..., p < q, q < r$ $0 = (-[pr((sq)u)] + [pr(q(su))] + p(u[rsq])$ $-u[(pq)rs] + u(p[qrs]))_{\{pq\}}$	167	17	150

Degree 5 Minimal General Polynomials III

$n = d = 5$ *multiliear* polynomial

Reductions	t	i	e
$(pr)((qs)u) = ..., p < q < r < s < u$ $0 = (-[pr(s(qu))] + [pr((qs)u)] + p(u[rsq]) - u[pr(qs)])_{\{pq\}}$	166	16	150
$(qr)((ps)u), (qr)(u(ps)), (qr)(s(up)),$ based on 6 longer $(qr)(s(pu)), (qs)(u(pr)), (qu)(r(sp))$ identities	160	10	150
$p(q(r(us))), p(q(u(rs))), p(r(s(qu))), p(r(u(qs))), p(u(q(sr)))$ $p(u(r(sq))), p(s(u(qr))), q(r(p(su))), q(r(s(pu))), q(r(u(ps)))$ $0 = (q(r(u(ps))) + r(u(q(sp))) + u(q(p(sr))))_{[uq]\{qrs\}}$	150	0	150

Degree 5 Minimal General Polynomials III

$n = d = 5$ *multiliear* polynomial

Reductions	t	i	e
$(pr)((qs)u) = ..., p < q < r < s < u$ $0 = (-[pr(s(qu))] + [pr((qs)u)] + p(u[rsq]) - u[pr(qs)])_{\{pq\}}$	166	16	150
$(qr)((ps)u), (qr)(u(ps)), (qr)(s(up)),$ based on 6 longer $(qr)(s(pu)), (qs)(u(pr)), (qu)(r(sp))$ identities	160	10	150
$p(q(r(us))), p(q(u(rs))), p(r(s(qu))), p(r(u(qs))), p(u(q(sr)))$ $p(u(r(sq))), p(s(u(qr))), q(r(p(su))), q(r(s(pu))), q(r(u(ps)))$ $0 = (q(r(u(ps))) + r(u(q(sp))) + u(q(p(sr))))_{[uq]\{qrs\}}$	150	0	150

- Last reduction uses 10 identities each with 36 terms $*(*(*(**)))$:

$$0 = \big(q(r(u(ps))) + r(u(q(sp))) + u(q(p(sr)))\big)_{[uq]\{qrs\}}$$

$$0 = \big(q(r(u(ps))) + r(u(s(qp))) + u(q(p(sr)))\big)_{[ps]\{qrs\}}$$

Degree 5 Minimal General Polynomials III

$n = d = 5$ *multiliear* polynomial

Reductions	t	i	e
$(pr)((qs)u) = ..., p < q < r < s < u$ $0 = (-[pr(s(qu))] + [pr((qs)u)] + p(u[rsq]) - u[pr(qs)])_{\{pq\}}$	166	16	150
$(qr)((ps)u), (qr)(u(ps)), (qr)(s(up)),$ based on 6 longer $(qr)(s(pu)), (qs)(u(pr)), (qu)(r(sp))$ identities	160	10	150
$p(q(r(us))), p(q(u(rs))), p(r(s(qu))), p(r(u(qs))), p(u(q(sr)))$ $p(u(r(sq))), p(s(u(qr))), q(r(p(su))), q(r(s(pu))), q(r(u(ps)))$ $0 = (q(r(u(ps))) + r(u(q(sp))) + u(q(p(sr))))_{[uq]\{qrs\}}$	150	0	150

- Last reduction uses 10 identities each with 36 terms $*(*(*(**)))$:

$$0 = \big(q(r(u(ps))) + r(u(q(sp))) + u(q(p(sr)))\big)_{[uq]\{qrs\}}$$

$$0 = \big(q(r(u(ps))) + r(u(s(qp))) + u(q(p(sr)))\big)_{[ps]\{qrs\}}$$

- Only left multiplications, associativity does not matter, valid for any non-associative algebra

Degree 4 Central Multilinear Polynomials

Apart from the known $[a, b] \circ [c, d]$ also this is real:

$$
\begin{aligned}
&+p(q(rs)) + p(r(qs)) + s(r(qp)) + s(q(rp)) \\
&-p(q(sr)) - p(r(sq)) - s(r(pq)) - s(q(pr)) \\
&-q(p(rs)) - r(p(qs)) - r(s(qp)) - q(s(rp)) \\
&+q(p(sr)) + r(p(sq)) + r(s(pq)) + q(s(pr))
\end{aligned}
$$

Degree 4 Central Multilinear Polynomials

Apart from the known $[a,b]\circ[c,d]$ also this is real:

$$
\begin{aligned}
& +p(q(rs))+p(r(qs))+s(r(qp))+s(q(rp))\\
& -p(q(sr))-p(r(sq))-s(r(pq))-s(q(pr))\\
& -q(p(rs))-r(p(qs))-r(s(qp))-q(s(rp))\\
& +q(p(sr))+r(p(sq))+r(s(pq))+q(s(pr))\\
= \quad & +p(q[rs])+p(r[qs])+s(r[qp])+s(q[rp])\\
& -q(p[rs])-r(p[qs])-r(s[qp])-q(s[rp])
\end{aligned}
$$

Degree 4 Central Multilinear Polynomials

Apart from the known $[a,b]\circ[c,d]$ also this is real:

$$\begin{aligned}
& +p(q(rs))+p(r(qs))+s(r(qp))+s(q(rp)) \\
& -p(q(sr))-p(r(sq))-s(r(pq))-s(q(pr)) \\
& -q(p(rs))-r(p(qs))-r(s(qp))-q(s(rp)) \\
& +q(p(sr))+r(p(sq))+r(s(pq))+q(s(pr)) \\
= \quad & +p(q[rs])+p(r[qs])+s(r[qp])+s(q[rp]) \\
& -q(p[rs])-r(p[qs])-r(s[qp])-q(s[rp]) \\
= \quad & +[p(q][rs])+[p(r][qs])+[s(r][qp])+[s(q][rp])
\end{aligned}$$

Degree 4 Central Multilinear Polynomials

Apart from the known $[a,b] \circ [c,d]$ also this is real:

$$
\begin{aligned}
& +p(q(rs)) + p(r(qs)) + s(r(qp)) + s(q(rp)) \\
& -p(q(sr)) - p(r(sq)) - s(r(pq)) - s(q(pr)) \\
& -q(p(rs)) - r(p(qs)) - r(s(qp)) - q(s(rp)) \\
& +q(p(sr)) + r(p(sq)) + r(s(pq)) + q(s(pr)) \\
= \; & +p(q[rs]) + p(r[qs]) + s(r[qp]) + s(q[rp]) \\
& -q(p[rs]) - r(p[qs]) - r(s[qp]) - q(s[rp]) \\
= \; & +[p(q][rs]) + [p(r][qs]) + [s(r][qp]) + [s(q][rp]) \\
= \; & \big([p(q][rs]) + [s(r][qp])\big)_{\{q,r\}}
\end{aligned}
$$

Degree 4 Central Multilinear Polynomials

Apart from the known $[a,b]\circ[c,d]$ also this is real:

$$
\begin{aligned}
&+p(q(rs)) + p(r(qs)) + s(r(qp)) + s(q(rp)) \\
&-p(q(sr)) - p(r(sq)) - s(r(pq)) - s(q(pr)) \\
&-q(p(rs)) - r(p(qs)) - r(s(qp)) - q(s(rp)) \\
&+q(p(sr)) + r(p(sq)) + r(s(pq)) + q(s(pr)) \\
= \;&+p(q[rs]) + p(r[qs]) + s(r[qp]) + s(q[rp]) \\
&-q(p[rs]) - r(p[qs]) - r(s[qp]) - q(s[rp]) \\
= \;&+[p(q][rs]) + [p(r][qs]) + [s(r][qp]) + [s(q][rp]) \\
= \;&\big([p(q][rs]) + [s(r][qp])\big)_{\{q,r\}} \\
= \;&\big((1+R)[p(q][rs])\big)_{\{q,r\}} = \big((1+R)(p(q(rs)))\big)_{[p,q],[rs],\{q,r\}}
\end{aligned}
$$

Degree 4 Central Multilinear Polynomials

Apart from the known $[a,b]\circ[c,d]$ also this is real:

$$
\begin{aligned}
& +p(q(rs)) + p(r(qs)) + s(r(qp)) + s(q(rp)) \\
& -p(q(sr)) - p(r(sq)) - s(r(pq)) - s(q(pr)) \\
& -q(p(rs)) - r(p(qs)) - r(s(qp)) - q(s(rp)) \\
& +q(p(sr)) + r(p(sq)) + r(s(pq)) + q(s(pr)) \\
= \; & +p(q[rs]) + p(r[qs]) + s(r[qp]) + s(q[rp]) \\
& -q(p[rs]) - r(p[qs]) - r(s[qp]) - q(s[rp]) \\
= \; & +[p(q][rs]) + [p(r][qs]) + [s(r][qp]) + [s(q][rp]) \\
= \; & \big([p(q][rs]) + [s(r][qp])\big)_{\{q,r\}} \\
= \; & \big((1+R)[p(q][rs])\big)_{\{q,r\}} = \big((1+R)(p(q(rs)))\big)_{[p,q],[rs],\{q,r\}}, \text{ also} \\
= \; & \big([p(q][rs]) + [s(q][rp])\big)_{\{q,r\}}
\end{aligned}
$$

Degree 4 Central Multilinear Polynomials

Apart from the known $[a,b]\circ[c,d]$ also this is real:

$$\begin{aligned}
& +p(q(rs))+p(r(qs))+s(r(qp))+s(q(rp)) \\
& -p(q(sr))-p(r(sq))-s(r(pq))-s(q(pr)) \\
& -q(p(rs))-r(p(qs))-r(s(qp))-q(s(rp)) \\
& +q(p(sr))+r(p(sq))+r(s(pq))+q(s(pr)) \\
= \; & +p(q[rs])+p(r[qs])+s(r[qp])+s(q[rp]) \\
& -q(p[rs])-r(p[qs])-r(s[qp])-q(s[rp]) \\
= \; & +[p(q][rs])+[p(r][qs])+[s(r][qp])+[s(q][rp]) \\
= \; & \big([p(q][rs])+[s(r][qp])\big)_{\{q,r\}} \\
= \; & \big((1+R)[p(q][rs])\big)_{\{q,r\}} = \big((1+R)(p(q(rs)))\big)_{[p,q],[rs],\{q,r\}}, \text{ also} \\
= \; & \big([p(q][rs])+[s(q][rp])\big)_{\{q,r\}} \\
= \; & \big(p(q(rs))\big)_{[p,q],[rs],\{q,r\}\{p,s\}}
\end{aligned}$$

Degree 4 Central Multilinear Polynomials continued

Similarly to $\big(p(q(rs))\big)_{[p,q],[rs],\{q,r\}\{p,s\}}$

also $\big(p(q(rs))\big)_{[p,q],[rs],\{p,r\}\{q,s\}}$

is a different real degree 4 polynomial.

Similarly to $\big(p(q(rs))\big)_{[p,q],[rs],\{q,r\}\{p,s\}}$

also $\big(p(q(rs))\big)_{[p,q],[rs],\{p,r\}\{q,s\}}$

is a different real degree 4 polynomial.

Commutators of the 3 real polynomials with any octonion result in a total of 25 identities of degree 5 (table above).

Degree 4 Central Multilinear Polynomials continued

Similarly to $\big(p(q(rs))\big)_{[p,q],[rs],\{q,r\}\{p,s\}}$

also $\big(p(q(rs))\big)_{[p,q],[rs],\{p,r\}\{q,s\}}$

is a different real degree 4 polynomial.

Commutators of the 3 real polynomials with any octonion result in a total of 25 identities of degree 5 (table above).

Changing $p(q(rs))$ to $((pq)r)s$, $(pq)(rs)$, $(p(qr))s$, $p((qr)s)$ does not give new real polynomials.

Outline

General

- CA systems know non-commutativity but not non-associativity.

General

- CA systems know non-commutativity but not non-associativity.
- Why not using existing packages, like 'DifferentialGeometry' in MAPLE?

General

- CA systems know non-commutativity but not non-associativity.
- Why not using existing packages, like 'DifferentialGeometry' in MAPLE?
- Total ordering of octonion products needed to define leading terms of IDs to reduce P.

General

- CA systems know non-commutativity but not non-associativity.
- Why not using existing packages, like 'DifferentialGeometry' in MAPLE?
- Total ordering of octonion products needed to define leading terms of IDs to reduce P.
- How to avoid the extremely time-costly splitting of polynomials with, e.g. 250 million terms?

General

- CA systems know non-commutativity but not non-associativity.
- Why not using existing packages, like 'DifferentialGeometry' in MAPLE?
- Total ordering of octonion products needed to define leading terms of IDs to reduce P.
- How to avoid the extremely time-costly splitting of polynomials with, e.g. 250 million terms?
- How to lower cubic cost of solving lin. alg. system with 10^5 equations?

Algorithmic Changes

- For each row in the tables do not do 1 run but a sequencee of them. Start with smaller number of components than $8n$ and increase it successively.

Algorithmic Changes

- For each row in the tables do not do 1 run but a sequencee of them. Start with smaller number of components than $8n$ and increase it successively.
  After each run update relations between coeffs in P.

Algorithmic Changes

- For each row in the tables do not do 1 run but a sequencee of them. Start with smaller number of components than $8n$ and increase it successively.
  After each run update relations between coeffs in P.
- For multilinear P, when adding more new components to an octonian variable for the next run, set the other components of the variable temporarily to zero.

Algorithmic Changes

- For each row in the tables do not do 1 run but a sequencee of them. Start with smaller number of components than $8n$ and increase it successively.
  After each run update relations between coeffs in P.
- For multilinear P, when adding more new components to an octonian variable for the next run, set the other components of the variable temporarily to zero.
- Fine tune the number of new components per run.

Outline

Result for KdV-Weights

- $w(\partial_t) = 3,\ w(\partial_x) = 1,\ w(u) = 2,\ w(L) = 2$
- Two solutions have the same evolution equation

$$u_t = u_{xxx} + uu_x + u_x u = u_{xxx} + (u^2)_x$$

- Two slightly different Lax pairs

$$LG = G_{xx} + \tfrac{1}{3}uG, \quad MG = 4G_{xxx} + 2uG_x + u_x G$$

$$LG = G_{xx} + \tfrac{1}{3}Gu, \quad MG = 4G_{xxx} + 2G_x u + Gu_x$$

Result for mKdV-Weights I

- Weights: $w(\partial_t) = 3, w(\partial_x) = 1, w(u) = 1$ (< 2) first try: $w(L) = 1$,
- More unknowns: f_i $(i = 1, \ldots, 7)$, l_i $(i = 1, 2, 3)$, m_i $(i = 1, \ldots, 21)$

Result for mKdV-Weights I

- Weights: $w(\partial_t) = 3, w(\partial_x) = 1, w(u) = 1$ (< 2) first try: $w(L) = 1$,
- More unknowns: f_i $(i = 1, \ldots, 7)$, l_i $(i = 1, 2, 3)$, m_i $(i = 1, \ldots, 21)$
- Six solutions with 5 free parameters each,

Result for mKdV-Weights I

- Weights: $w(\partial_t) = 3, w(\partial_x) = 1, w(u) = 1$ (< 2) first try: $w(L) = 1$,
- More unknowns: f_i $(i = 1, \ldots, 7)$, l_i $(i = 1, 2, 3)$, m_i $(i = 1, \ldots, 21)$
- Six solutions with 5 free parameters each,
- Removing non-relevant gauge terms in MG,
- Scaling u and L,
- Rewrite solution by using product commutators $[A, B] = AB - BA$ and associators $[A, B, C] = (AB)C - A(BC)$

Result for mKdV-Weights I

- Weights: $w(\partial_t) = 3, w(\partial_x) = 1, w(u) = 1$ (< 2) first try: $w(L) = 1$,
- More unknowns: f_i $(i = 1, \ldots, 7)$, l_i $(i = 1, 2, 3)$, m_i $(i = 1, \ldots, 21)$
- Six solutions with 5 free parameters each,
- Removing non-relevant gauge terms in MG,
- Scaling u and L,
- Rewrite solution by using product commutators $[A, B] = AB - BA$ and associators $[A, B, C] = (AB)C - A(BC)$
- Result: 2 evolution equations each with 3 Lax pairs

Result for mKdV-Weights II

$$u_t = u_{xxx} + \alpha(u^2 u_x + u u_x u + u_x u^2) + [u, [u_x, u]]$$

$$LG = G_x - [G, u] + 2Gu$$
$$MG = [G, u_{xx}] - 2Gu_{xx} + 2[G, u_x, u] - [G, [u, u_x] + \alpha([G, u^3] - 2Gu^3)$$

$$LG = G_x + uG$$
$$MG = -u_{xx}G + 2[u_x, u, G] + [u, u_x]G - \alpha u^3 G$$

and a similar Lax pair with G on the left in all products.

Result for mKdV-Weights III

$$u_t = u_{xxx} + [u_{xx}, u] + \alpha(u^2 u_x + u u_x u + u_x u^2) + 2[u, [u_x, u]]$$

$$LG = G_x - [G, u] \cancel{+2Gu}$$
$$MG = [G, u_{xx}] + 6[G, u_x, u] + 2[G, [u_x, u]] + \alpha[G, u^3]$$

$$LG = G_x - 2uG - Gu$$
$$MG = -[u_{xx}, G] + 3u_{xx}G + 6[u_x, u, G] - 2[[u_x, u], G] - \alpha([u^3, G] - 3u^3 G)$$

and a similar Lax pair with G on the left in all products.

Result for mKdV-Weights IV

New try with next higher $w(L) = 2 \ (> 1)$

Result for mKdV-Weights IV

New try with next higher $w(L) = 2$ (> 1)

Results:

- The same evelutionary equations with higher degree Lax pairs

Result for mKdV-Weights IV

New try with next higher $w(L) = 2$ (> 1)

Results:

- The same evelutionary equations with higher degree Lax pairs
- 2 additional evolution equations each with 2 Lax pairs:

$$u_t = u_{xxx} - 3(u^2 u_x + u_x u^2) \qquad \text{(mKdV equation)}$$

$$LG = G_{xx} + (u_x - u^2)G,$$

$$MG = 4G_{xxx} + 6(u_x - u^2)G_x + 3(u_{xx} - (u^2)_x)G$$

and a similar Lax pair with G on the left in all products.

Result for mKdV-Weights IV

New try with next higher $w(L) = 2$ (> 1)

Results:

- The same evelutionary equations with higher degree Lax pairs
- 2 additional evolution equations each with 2 Lax pairs:

$$u_t = u_{xxx} - 3(u^2 u_x + u_x u^2) \qquad \text{(mKdV equation)}$$

$$LG = G_{xx} + (u_x - u^2)G,$$

$$MG = 4G_{xxx} + 6(u_x - u^2)G_x + 3(u_{xx} - (u^2)_x)G$$

and a similar Lax pair with G on the left in all products.

$$u_t = u_{xxx} + 3u_x^2 \qquad \text{(Potential KdV equation)}$$

$$LG = G_{xx} + u_x G, \quad MG = 4G_{xxx} + 6u_x G_x + 3u_{xx} G$$

and a similar Lax pair with G on the left in all products.

Outline

Conclusion I on Octonion Identities

We obtained

- minimal general octonion polynomials *multilinear* and with *repeating* factors, both cases for degree 3, 4, 5

Conclusion I on Octonion Identities

We obtained

- minimal general octonion polynomials *multilinear* and with *repeating* factors, both cases for degree 3, 4, 5
- new compact multilinear vanishing identities and central polynomials of degree 4, 5,

Conclusion I on Octonion Identities

We obtained

- minimal general octonion polynomials *multilinear* and with *repeating* factors, both cases for degree 3, 4, 5
- new compact multilinear vanishing identities and central polynomials of degree 4, 5,
- term dropping rules applicable also to higher degree polynomials

Conclusion I on Octonion Identities

We obtained

- minimal general octonion polynomials *multilinear* and with *repeating* factors, both cases for degree 3, 4, 5
- new compact multilinear vanishing identities and central polynomials of degree 4, 5,
- term dropping rules applicable also to higher degree polynomials
- efficient algorithms for computations with octonions

Conclusion I on Octonion Identities

We obtained

- minimal general octonion polynomials *multilinear* and with *repeating* factors, both cases for degree 3, 4, 5
- new compact multilinear vanishing identities and central polynomials of degree 4, 5,
- term dropping rules applicable also to higher degree polynomials
- efficient algorithms for computations with octonions
- insight into using reverse multiplication to formulate new types of symmetries (multifactor and non-associative generalizations of the commutator and the Jordan product)

Conclusion II on Equations and Methods

- We found the octonion KdV, mKdV and potential KdV equation that possess Lax pairs.

Conclusion II on Equations and Methods

- We found the octonion KdV, mKdV and potential KdV equation that possess Lax pairs.
- The method is (mostly) algorithmic.
  - Inputs are $w(u), w(\partial_x), w(\partial_t), w(L)$.
  - The scaling homogeneous ansatz polynomials for F, LG, MG are generated by a separate program, which automatically uses octonion identities up to degree 4 to eliminate redundant terms of degree ≥ 4.
  - Currently Maple formulates the overdetermined systems.
  - Solution is done by Maple (simple cases) or 'Crack' (larger cases).
  - Start at the lowest possible $w(L)$ and later increase the weight to search for additional variants of evolution equation finally limited by complexity.

Conclusion III on Outlook

- Different Lax pairs may be useful for different purposes.
  - Inverse scattering transformation (IST)
  - Darboux transformation

Conclusion III on Outlook

- Different Lax pairs may be useful for different purposes.
  - Inverse scattering transformation (IST)
  - Darboux transformation
- Future work:
  - Find and use degree > 5 identities of octonions to produce general polynomial ansatz for F, LG, MG with minimal number of terms.

Conclusion III on Outlook

- Different Lax pairs may be useful for different purposes.
  - Inverse scattering transformation (IST)
  - Darboux transformation
- Future work:
  - Find and use degree > 5 identities of octonions to produce general polynomial ansatz for F, LG, MG with minimal number of terms.
  - ($\checkmark$) Rewrite in REDUCE what is currently in MAPLE

Conclusion III on Outlook

- Different Lax pairs may be useful for different purposes.
  - Inverse scattering transformation (IST)
  - Darboux transformation
- Future work:
  - Find and use degree > 5 identities of octonions to produce general polynomial ansatz for F, LG, MG with minimal number of terms.
  - (✓) Rewrite in Reduce what is currently in Maple
  - (✓) Run 5th order evolution equations

Conclusion III on Outlook

- Different Lax pairs may be useful for different purposes.
  - Inverse scattering transformation (IST)
  - Darboux transformation
- Future work:
  - Find and use degree > 5 identities of octonions to produce general polynomial ansatz for F, LG, MG with minimal number of terms.
  - ($\checkmark$) Rewrite in REDUCE what is currently in MAPLE
  - ($\checkmark$) Run 5th order evolution equations
  - ($\checkmark$) Run other scaling weights: $w(u) = \frac{1}{2}$ (e.g. Ibragimov-Shabat equation)

Conclusion III on Outlook

- Different Lax pairs may be useful for different purposes.
  - Inverse scattering transformation (IST)
  - Darboux transformation
- Future work:
  - Find and use degree > 5 identities of octonions to produce general polynomial ansatz for F, LG, MG with minimal number of terms.
  - ($\checkmark$) Rewrite in REDUCE what is currently in MAPLE
  - ($\checkmark$) Run 5th order evolution equations
  - ($\checkmark$) Run other scaling weights: $w(u) = \frac{1}{2}$ (e.g. Ibragimov-Shabat equation)
  - Add complex conjugation $u, \bar{u}$ (e.g. NLS equation)

Outline

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  $\rightarrow$ allowing to check 5th order equations with low L weight

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  $\rightarrow$ allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  $\rightarrow$ allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:
  - Closing a 40 year old, publicly unknown, memory drain in the computer algebra system REDUCE when loading data files,

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  → allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:
  - Closing a 40 year old, publicly unknown, memory drain in the computer algebra system REDUCE when loading data files,
    → now unlimited many case distinctions

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  → allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:
  - Closing a 40 year old, publicly unknown, memory drain in the computer algebra system REDUCE when loading data files,
    → now unlimited many case distinctions
    → new bottlenecks showing up

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  → allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:
  - Closing a 40 year old, publicly unknown, memory drain in the computer algebra system REDUCE when loading data files,
    → now unlimited many case distinctions
    → new bottlenecks showing up
  - Detailed heuristic to minimize # of case distinctions

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  → allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:
  - Closing a 40 year old, publicly unknown, memory drain in the computer algebra system REDUCE when loading data files,
    → now unlimited many case distinctions
    → new bottlenecks showing up
  - Detailed heuristic to minimize # of case distinctions
  - Adapting solution strategy to new module strengths

Computational Progress

- Complete re-write in REDUCE and using CRACK for solving bi-linear overdetermined system
  $\rightarrow$ allowing to check 5th order equations with low L weight
- For solving larger bi-linear overdetermined system (eg. 2000 equations) for (eg. 100) unknown coefficients:
  - Closing a 40 year old, publicly unknown, memory drain in the computer algebra system REDUCE when loading data files,
    $\rightarrow$ now unlimited many case distinctions
    $\rightarrow$ new bottlenecks showing up
  - Detailed heuristic to minimize # of case distinctions
  - Adapting solution strategy to new module strengths
- Apply scaling freedom of L, t, x, u and $M \rightarrow M + \alpha L^n$ to reduce # of unknown coefficients

About Reverse Duality

- Reversing factors:

$$\leftrightarrow \begin{array}{c} a \cdot (b \cdot c) \\ c \cdot (b \cdot a) \end{array} \quad \text{or} \quad \leftrightarrow \begin{array}{c} (a \cdot b) \cdot (c \cdot (d \cdot e)) \\ (e \cdot d) \cdot (c \cdot (b \cdot a)) \end{array}$$

About Reverse Duality

- Reversing factors:

$$\leftrightarrow \quad \begin{matrix} a\cdot(b\cdot c) \\ c\cdot(b\cdot a) \end{matrix} \qquad \text{or} \qquad \leftrightarrow \quad \begin{matrix} (a\cdot b)\cdot(c\cdot(d\cdot e)) \\ (e\cdot d)\cdot(c\cdot(b\cdot a)) \end{matrix}$$

- If an octonion polynomial is zero then the factor reversed version is also zero.

About Reverse Duality

- Reversing factors:

  $\leftrightarrow \begin{array}{c} a\cdot(b\cdot c) \\ c\cdot(b\cdot a) \end{array}$ or $\leftrightarrow \begin{array}{c} (a\cdot b)\cdot(c\cdot(d\cdot e)) \\ (e\cdot d)\cdot(c\cdot(b\cdot a)) \end{array}$

- If an octonion polynomial is zero then the factor reversed version is also zero.

  $\rightarrow$ An octonion identity is either palindromic or there exists a reverse dual version of it.

About Reverse Duality

- Reversing factors:

$$\leftrightarrow \begin{array}{c} a\cdot(b\cdot c) \\ c\cdot(b\cdot a) \end{array} \quad \text{or} \quad \leftrightarrow \begin{array}{c} (a\cdot b)\cdot(c\cdot(d\cdot e)) \\ (e\cdot d)\cdot(c\cdot(b\cdot a)) \end{array}$$

- If an octonion polynomial is zero then the factor reversed version is also zero.
  $\rightarrow$ An octonion identity is either palindromic or there exists a reverse dual version of it.
  $\rightarrow$ For each evol. eqn. + Lax pair there is a reverse version unless it is palindromic.

About Reverse Duality

- Reversing factors:

$$\leftrightarrow \begin{array}{c} a\cdot(b\cdot c) \\ c\cdot(b\cdot a) \end{array} \quad \text{or} \quad \leftrightarrow \begin{array}{c} (a\cdot b)\cdot(c\cdot(d\cdot e)) \\ (e\cdot d)\cdot(c\cdot(b\cdot a)) \end{array}$$

- If an octonion polynomial is zero then the factor reversed version is also zero.
  $\rightarrow$ An octonion identity is either palindromic or there exists a reverse dual version of it.
  $\rightarrow$ For each evol. eqn. + Lax pair there is a reverse version unless it is palindromic.
- Symmetry broken use of identities $\rightarrow$ non-symmetric Lax pairs

$w(u)/w(\partial_x) = 2$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
2	1	3	2	KdV: $u_t = u_x u + u u_x + u_{3x}$ 3 sol: 2 reverse dual L, M, 1 palindromic L, M

$w(u)/w(\partial_x) = 2$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
2	1	3	2	KdV: $u_t = u_x u + u u_x + u_{3x}$ 3 sol: 2 reverse dual L, M, 1 palindromic L, M
			4,6,8	same $u_t, L_4 = {L_2}^2, L_6 = {L_2}^3, L_8 = {L_2}^4$, both L not sym. anymore due to non-sym. use of identities

$w(u)/w(\partial_x) = 2$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
2	1	3	2	KdV: $u_t = u_x u + u u_x + u_{3x}$ 3 sol: 2 reverse dual L, M, 1 palindromic L, M
			4,6,8	same $u_t, L_4 = {L_2}^2, L_6 = {L_2}^3, L_8 = {L_2}^4$, both L not sym. anymore due to non-sym. use of identities
		5	2	$u_t = 5u_{3x}u + 10u_{2x}u_x + 2u_x(uu) + 10u_x u_{2x}$ $+2u(u_x u) + 2u(uu_x) + 5uu_{3x} + 5u_{5x}$ palindromic u_t, 2 symmetric L, both MG not symm., due to non-symm. identity usage real limit: $u_t = 5u_{5x} + 10uu_{3x} + 20u_x u_{2x} + 6u^2 u_x$ real $u_t = u_{3x} + 2uu_x$ has symmetry $u_\tau = 3u_{5x} + 10uu_{3x} + 20u_x u_{2x} + 10u^2 u_x$ scaling u, x, t merge both, Kaup-Kupershmidt: $u_t = .. + 10uu_{3x} + 25u_x u_{2x} + ..$ Sawata-Kotera: $u_t = .. + 15uu_{3x} + 15u_x u_{2x} + ..$

$w(u)/w(\partial_x) = 2$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
2	1	3	2	KdV: $u_t = u_x u + u u_x + u_{3x}$ 3 sol: 2 reverse dual L, M, 1 palindromic L, M
			4,6,8	same $u_t, L_4 = {L_2}^2, L_6 = {L_2}^3, L_8 = {L_2}^4$, both L not sym. anymore due to non-sym. use of identities
		5	2	$u_t = 5u_{3x}u + 10u_{2x}u_x + 2u_x(uu) + 10u_x u_{2x}$ $+2u(u_x u) + 2u(uu_x) + 5uu_{3x} + 5u_{5x}$ palindromic u_t, 2 symmetric L, both MG not symm., due to non-symm. identity usage real limit: $u_t = 5u_{5x} + 10uu_{3x} + 20u_x u_{2x} + 6u^2 u_x$ real $u_t = u_{3x} + 2uu_x$ has symmetry $u_\tau = 3u_{5x} + 10uu_{3x} + 20u_x u_{2x} + 10u^2 u_x$ scaling u, x, t merge both, Kaup-Kupershmidt: $u_t = .. + 10uu_{3x} + 25u_x u_{2x} + ..$ Sawata-Kotera: $u_t = .. + 15uu_{3x} + 15u_x u_{2x} + ..$
			4,6	same u_t, $L_4 = {L_2}^2, L_6 = {L_2}^3$

$w(u)/w(\partial_x) = 2$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
2	1	3	2	KdV: $u_t = u_x u + u u_x + u_{3x}$ 3 sol: 2 reverse dual L, M, 1 palindromic L, M
			4,6,8	same $u_t, L_4 = {L_2}^2, L_6 = {L_2}^3, L_8 = {L_2}^4$, both L not sym. anymore due to non-sym. use of identities
		5	2	$u_t = 5u_{3x}u + 10u_{2x}u_x + 2u_x(uu) + 10u_x u_{2x}$ $+2u(u_x u) + 2u(uu_x) + 5uu_{3x} + 5u_{5x}$ palindromic u_t, 2 symmetric L, both MG not symm., due to non-symm. identity usage real limit: $u_t = 5u_{5x} + 10uu_{3x} + 20u_x u_{2x} + 6u^2 u_x$ real $u_t = u_{3x} + 2uu_x$ has symmetry $u_\tau = 3u_{5x} + 10uu_{3x} + 20u_x u_{2x} + 10u^2 u_x$ scaling u, x, t merge both, Kaup-Kupershmidt: $u_t = .. + 10uu_{3x} + 25u_x u_{2x} + ..$ Sawata-Kotera: $u_t = .. + 15uu_{3x} + 15u_x u_{2x} + ..$
			4,6	same u_t, $L_4 = {L_2}^2, L_6 = {L_2}^3$
		7	2,4	palindromic $u_t = 21$ terms, $L_4 = {L_2}^2$, hypothesis: $LG = Gu + w(\partial_t)G_{2x}, \quad LG = uG + w(\partial_t)G_{2x}$

$w(u)/w(\partial_x) = 1$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
1	1	3	1	6 sol, all generalized mKdV: palindromic 1. sol: $u_t = a(u_x(uu) + u(u_x u)(1+3a) + u(uu_x)) + u_{3x}$ $LG = auG + G_x$, 2. sol with reverse L, M 3. sol: $u_t = a(u_x(uu) + u(u_x u)(1+6ab^2)$ $+u(uu_x) + b[u, u_{2x}] + u_{3x}$, $LG = b[G, u] + G_x$ 4. sol like 1.,2. but slightly different L, M 5. sol like 3. but slightly different L, M 6. sol like 3. but again slightly different L, M

$w(u)/w(\partial_x) = 1$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
1	1	3	1	6 sol, all generalized mKdV: palindromic 1. sol: $u_t = a(u_x(uu) + u(u_xu)(1+3a) + u(uu_x)) + u_{3x}$ $LG = auG + G_x$, 2. sol with reverse L, M 3. sol: $u_t = a(u_x(uu) + u(u_xu)(1+6ab^2)$ $+u(uu_x) + b[u, u_{2x}] + u_{3x}$, $LG = b[G, u] + G_x$ 4. sol like 1.,2. but slightly different L, M 5. sol like 3. but slightly different L, M 6. sol like 3. but again slightly different L, M
			2	10 sol, 6 soln like for $w(L) = 1$ only with higher weight L. 4 sol for potential KdV $u_t = 3{u_x}^2 + u_{3x}$

$w(u)/w(\partial_x) = 1$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
1	1	3	1	6 sol, all generalized mKdV: palindromic 1. sol: $u_t = a(u_x(uu) + u(u_x u)(1+3a) + u(uu_x)) + u_{3x}$ $LG = auG + G_x$, 2. sol with reverse L, M 3. sol: $u_t = a(u_x(uu) + u(u_x u)(1+6ab^2)$ $+u(uu_x) + b[u, u_{2x}] + u_{3x}$, $LG = b[G,u] + G_x$ 4. sol like 1.,2. but slightly different L, M 5. sol like 3. but slightly different L, M 6. sol like 3. but again slightly different L, M
			2	10 sol, 6 soln like for $w(L) = 1$ only with higher weight L. 4 sol for potential KdV $u_t = 3{u_x}^2 + u_{3x}$
			3	6 sol, same u_t as for $w(L) = 1$, higher weight L

$w(u)/w(\partial_x) = 1$

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
1	1	3	1	6 sol, all generalized mKdV: palindromic 1. sol: $u_t = a(u_x(uu) + u(u_xu)(1+3a) + u(uu_x)) + u_{3x}$ $LG = auG + G_x$, 2. sol with reverse L, M 3. sol: $u_t = a(u_x(uu) + u(u_xu)(1+6ab^2)$ $+u(uu_x) + b[u, u_{2x}] + u_{3x}$, $LG = b[G, u] + G_x$ 4. sol like 1.,2. but slightly different L, M 5. sol like 3. but slightly different L, M 6. sol like 3. but again slightly different L, M
			2	10 sol, 6 soln like for $w(L) = 1$ only with higher weight L. 4 sol for potential KdV $u_t = 3{u_x}^2 + u_{3x}$
			3	6 sol, same u_t as for $w(L) = 1$, higher weight L
			4	14 sol, apart from all above, also solns with $u_t = -a^2(u_x(uu) + u(uu_x)) + 3u_x$ and $u_t = au_{2x}u + bu_x(uu) + cu(uu_x) + u_{3x}$ with symmetries of real limit up to order 13: $u_\tau = 3tu_{3x} + 3atuu_{2x} + 3btu^2u_x + xu_x + u$ $u_\tau = u_x$

$w(u)/w(\partial_x) = 1$ continued

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
1	1	5	1	6 soln, need to be re-computed using higher degree identities to be simpler and usable

$w(u)/w(\partial_x) = 1$ continued

$w(u)$	$w(\partial_x)$	$w(\partial_t)$	$w(L)$	Comments
1	1	5	1	6 soln, need to be re-computed using higher degree identities to be simpler and usable
1	1	5	2	10 soln, 8 of them need to use higher degree identities in a re-computation to get simpler result, 2 soln have palindromic u_t: $u_t = u_{3x}u_x + {u_{2x}}^2 + \frac{2}{5}{u_x}^3 + u_x u_{3x} + u_{5x}$ $LG = u_x G + 5G_x$, 2. sol with reverse L, M Hypothesis: for odd $w(\partial_t)$: $LG = u_x G + w(\partial_t) G_{2x}$ + reverse L, M

$w(u)/w(\partial_x) = 1/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
1	2	3	1	3 sol all like $u_t = au_x u + buu_x$. The special case of anti-palindromiC $u_t = [u, u_x]$ has palindromic $LG = uG + Gu$ and anti-palin. $MG = [G, u_x]$

$w(u)/w(\partial_x) = 1/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
1	2	3	1	3 sol all like $u_t = au_x u + buu_x$. The special case of anti-palindromiC $u_t = [u, u_x]$ has palindromic $LG = uG + Gu$ and anti-palin. $MG = [G, u_x]$
			2..5	3 sol, all like for $w(L) = 1$ with same symmetry properties, only with L having one more u as factor for each increased $w(L)$. Example: $L_2 = u^2G$ or $= Gu^2$ or $= a(u^2G + Gu^2) + buGu$
		4	1..5	no solutions

$w(u)/w(\partial_x) = 1/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
1	2	3	1	3 sol all like $u_t = au_x u + buu_x$. The special case of anti-palindromiC $u_t = [u, u_x]$ has palindromic $LG = uG + Gu$ and anti-palin. $MG = [G, u_x]$
			2..5	3 sol, all like for $w(L) = 1$ with same symmetry properties, only with L having one more u as factor for each increased $w(L)$. Example: $L_2 = u^2G$ or $= Gu^2$ or $= a(u^2G + Gu^2) + buGu$
		4	1..5	no solutions
		5	1	anti-palindromic: $u_t = [u, u_{2x}]$ with arbitrary multiples of products $u_x u^3$ in all multiplication orders of u, u_x being compatible

$w(u)/w(\partial_x) = 1/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
1	2	3	1	3 sol all like $u_t = au_x u + buu_x$. The special case of anti-palindromiC $u_t = [u, u_x]$ has palindromic $LG = uG + Gu$ and anti-palin. $MG = [G, u_x]$
			2..5	3 sol, all like for $w(L) = 1$ with same symmetry properties, only with L having one more u as factor for each increased $w(L)$. Example: $L_2 = u^2 G$ or $= Gu^2$ or $= a(u^2 G + Gu^2) + buGu$
		4	1..5	no solutions
		5	1	anti-palindromic: $u_t = [u, u_{2x}]$ with arbitrary multiples of products $u_x u^3$ in all multiplication orders of u, u_x being compatible
		6	1..4	(Ibragimov-Shabat) no solutions

$w(u)/w(\partial_x) = 1/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
1	2	3	1	3 sol all like $u_t = au_xu + buu_x$. The special case of anti-palindromiC $u_t = [u, u_x]$ has palindromic $LG = uG + Gu$ and anti-palin. $MG = [G, u_x]$
			2..5	3 sol, all like for $w(L) = 1$ with same symmetry properties, only with L having one more u as factor for each increased $w(L)$. Example: $L_2 = u^2G$ or $= Gu^2$ or $= a(u^2G + Gu^2) + buGu$
		4	1..5	no solutions
		5	1	anti-palindromic: $u_t = [u, u_{2x}]$ with arbitrary multiples of products u_xu^3 in all multiplication orders of u, u_x being compatible
		6	1..4	(Ibragimov-Shabat) no solutions
		7	1	$u_t = [u_{3x}, u], LG = uG, MG = u_{3x}G + 2Gu_{3x}$ and 5 other solution

$w(u)/w(\partial_x) = 1/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
1	2	3	1	3 sol all like $u_t = au_xu + buu_x$. The special case of anti-palindromiC $u_t = [u, u_x]$ has palindromic $LG = uG + Gu$ and anti-palin. $MG = [G, u_x]$
			2..5	3 sol, all like for $w(L) = 1$ with same symmetry properties, only with L having one more u as factor for each increased $w(L)$. Example: $L_2 = u^2G$ or $= Gu^2$ or $= a(u^2G + Gu^2) + buGu$
		4	1..5	no solutions
		5	1	anti-palindromic: $u_t = [u, u_{2x}]$ with arbitrary multiples of products u_xu^3 in all multiplication orders of u, u_x being compatible
		6	1..4	(Ibragimov-Shabat) no solutions
		7	1	$u_t = [u_{3x}, u], LG = uG, MG = u_{3x}G + 2Gu_{3x}$ and 5 other solution
		7	2	6 sol, needs re-run using higher deree identities 1 sol: $u_t = [u_{3x}, u], LG = uG, MG = u_{3x}G + 2Gu_{3x}$

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_x u + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_x u + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_x u + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic
				2 sol with $u_t = {u_x}^2$ both reverse sym.

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic
				2 sol with $u_t = {u_x}^2$ both reverse sym.
			11,13,14	1 sol u_t like 3. sol of $w(L) = 3$

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic
				2 sol with $u_t = {u_x}^2$ both reverse sym.
			11,13,14	1 sol u_t like 3. sol of $w(L) = 3$
		9	3,6,9	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u, u_{3x}]$

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic
				2 sol with $u_t = {u_x}^2$ both reverse sym.
			11,13,14	1 sol u_t like 3. sol of $w(L) = 3$
		9	3,6,9	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u, u_{3x}]$
			5,10	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u_x, u_{2x}]$

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic
				2 sol with $u_t = {u_x}^2$ both reverse sym.
			11,13,14	1 sol u_t like 3. sol of $w(L) = 3$
		9	3,6,9	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u, u_{3x}]$
			5,10	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u_x, u_{2x}]$
		11	3,6,9	3 sol with $u_t = [u, u_{4x}] + au_xu^3 + buu_xu^2$ $+cu^2u_xu + du^3u_x$

$w(u)/w(\partial_x) = 3/2$

$w(u)$	$w(\partial_x)$	$w(\partial_x)$	$w(L)$	Comments
3	2	5	3	3 sol, 1: $u_t = au_xu + buu_x$, 2: rev sym soln 3: special case: anti-palindromic $u_t = [u, u_x]$ palin. $LG = uG + Gu$, anti-pal. $MG = [G, u_x]$
			6,9,12	as above with 1 extra u in LG
			11	1 sol, same as 3. sol of $w(L) = 3$ but with higher degree LG
		7	3,6,9,12	3 sol, all $u_t = [u, u_{2x}]$, same MG 2 sol reverse sym, the 3rd palindromic
				2 sol with $u_t = {u_x}^2$ both reverse sym.
			11,13,14	1 sol u_t like 3. sol of $w(L) = 3$
		9	3,6,9	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u, u_{3x}]$
			5,10	3 sol like $w(\partial_t) = 5, w(L) = 3$ only with $u_t = [u_x, u_{2x}]$
		11	3,6,9	3 sol with $u_t = [u, u_{4x}] + au_xu^3 + buu_xu^2$ $+cu^2u_xu + du^3u_x$
		11	11	1 sol with same u_t

Conclusion IV on Observations

- Increasing $w(L)$ can show new integrable equations with same other weights (eg. mKdV $\rightarrow$ potential KdV).
- If a set of weights of $u, \partial_x, \partial_t, L$ includes an integrable equation then increasing $w(L)$ in fixed intervals gives more lax pairs for same equation.
- Increasing $w(\partial_t)$ in fixed intervals show higher symmetries of integrable equations.

Outstanding Computational Challenges

- Goal: Find octonion version of Ibragimov-Shabat, Kaup Kupershmidt, Sawata Kotera if they exist
  $\rightarrow$ Checking higher weights of L

Outstanding Computational Challenges

- Goal: Find octonion version of Ibragimov-Shabat, Kaup Kupershmidt, Sawata Kotera if they exist
  $\rightarrow$ Checking higher weights of L
- Trying to determine and use all 6th degree octonion identities
- Last resort in solving non-lin. alg. system: Groebner basis computation

Outstanding Computational Challenges

- Goal: Find octonion version of Ibragimov-Shabat, Kaup Kupershmidt, Sawata Kotera if they exist
  $\rightarrow$ Checking higher weights of L
- Trying to determine and use all 6th degree octonion identities
- Last resort in solving non-lin. alg. system: Groebner basis computation
  $\rightarrow$ implement dynamically changing ordering of variables

Outstanding Computational Challenges

- Goal: Find octonion version of Ibragimov-Shabat, Kaup Kupershmidt, Sawata Kotera if they exist
  $\rightarrow$ Checking higher weights of L
- Trying to determine and use all 6th degree octonion identities
- Last resort in solving non-lin. alg. system: Groebner basis computation
  $\rightarrow$ implement dynamically changing ordering of variables
  $\rightarrow$ implement dynamical detection of overdet. sub-systems

Outstanding Computational Challenges

- Goal: Find octonion version of Ibragimov-Shabat, Kaup Kupershmidt, Sawata Kotera if they exist
  $\rightarrow$ Checking higher weights of L
- Trying to determine and use all 6th degree octonion identities
- Last resort in solving non-lin. alg. system: Groebner basis computation
  $\rightarrow$ implement dynamically changing ordering of variables
  $\rightarrow$ implement dynamical detection of overdet. sub-systems
  $\rightarrow$ add probabilistic module selection instead of rigid heuristic

Outstanding Computational Challenges

- Goal: Find octonion version of Ibragimov-Shabat, Kaup Kupershmidt, Sawata Kotera if they exist
  $\rightarrow$ Checking higher weights of L
- Trying to determine and use all 6th degree octonion identities
- Last resort in solving non-lin. alg. system: Groebner basis computation
  $\rightarrow$ implement dynamically changing ordering of variables
  $\rightarrow$ implement dynamical detection of overdet. sub-systems
  $\rightarrow$ add probabilistic module selection instead of rigid heuristic

Outstanding Mathematical Challenges

- Use Lax pairs to compute other features of integrability.
- Are multiple Lax pairs useful for anything?
- Is there some upper bound for the weight (degree) of L for a Lax Pair to exit for a given integrable octonion equation?

Outline

References I

- Ruth Moufang (1935), "Zur Struktur von Alternativkörpern", Math. Ann., 110: 416–430, doi:10.1007/bf01448037
- John C. Baez "The Octonions", `https://arxiv.org/abs/math/0105155`
- Michel L. Racine, "Minimal identities of octonion algebras", Journal of Algebra Volume 115, Issue 1, 15 May 1988, Pages 251-260 `https://www.sciencedirect.com/science/article/pii/0021869388902943`
- Irvin R. Hentzel, Luiz A. Peresi "Identities of Cayley-Dickson Algebras" Journal of Algebra 188, 292-309 (1997)
- Ivan Shestakov, Natalia Zhukavets, "Skew-symmetric identities of octonions", Journal of Pure and Applied Algebra 213 (2009) 479–492

References II

- A.Restuccia, A.Sotomayor, J.P.Veiro, “Integrability of the Korteweg-de Vries equation valued on a Cayley-Dickson algebra”, arXiv:1609.05410v1 [math-ph] 18 Sep 2016

- A. Restuccia, A. Sotomayor (2017). Singular Lagrangians and Its Corresponding Hamiltonian Structures. In H. Canbolat (Ed.), Lagrangian Mechanics (pp. Ch. 1). IntechOpen.

- A. Restuccia, A. Sotomayor, J.P. Veiro (2018). A new integrable equation valued on a Cayley-Dicksonalgebra. Journal Physics A: Mathematical and Theoretical 51 (34), 345203.

- M. Fernandez, A. Restuccia, A. Sotomayor. (2019). On the hamiltonian formulation of an octonionic integrableextension for the Korteweg-de Vries equation. Journal of Physics: Conference Series 1391 (1), 012151.

- Julia Cen, Andreas Fring, “ Multicomplex solitons”. Journal of Nonlinear Mathematical Physics, Vol. 27, No. 1 (2020) 17–35 https://link.springer.com/content/pdf/10.1080/14029251.2020.1683963.pdf

The End

Thank you!