Minimal General Octonion Polynomials and Octonion Identities

OCNMP - Bad Ems 2024

T. Wolf, P. Lam, S. Anco

Brock U, Ontario, Canada

June 25, 2024

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.

Differential Challenges

To classify integrable evolutionary PDEs over octonions

Differential Challenges

To classify integrable evolutionary PDEs over octonions

- decide on homogeneity weights of $u, \partial_t, \partial_x, L$ where $u = u(t,x)$ is octonion valued

Differential Challenges

To classify integrable evolutionary PDEs over octonions

- decide on homogeneity weights of $u, \partial_t, \partial_x, L$ where $u = u(t,x)$ is octonion valued
- make a general ansatz (with minimal number of terms) for homogeneous polynomials F, L, M in $u, u_x, u_{xx}, \ldots$ with undetermined coefficients satisfying $u_t = F, \quad L_t = [L, M]$

Differential Challenges

To classify integrable evolutionary PDEs over octonions

- decide on homogeneity weights of $u, \partial_t, \partial_x, L$ where $u = u(t, x)$ is octonion valued
- make a general ansatz (with minimal number of terms) for homogeneous polynomials F, L, M in $u, u_x, u_{xx}, \ldots$ with undetermined coefficients satisfying $u_t = F, \quad L_t = [L, M]$
- replace u by its component form $u = \sum_{i=0}^{7} u_i(x, t) e_i$

Differential Challenges

To classify integrable evolutionary PDEs over octonions

- decide on homogeneity weights of $u, \partial_t, \partial_x, L$ where $u = u(t, x)$ is octonion valued
- make a general ansatz (with minimal number of terms) for homogeneous polynomials F, L, M in $u, u_x, u_{xx}, \ldots$ with undetermined coefficients satisfying $u_t = F, \quad L_t = [L, M]$
- replace u by its component form $u = \sum_{i=0}^{7} u_i(x, t) e_i$
- split $L_t = [L, M]$ w.r.t. $e_i, u_i, u_{ix}, ...$

Differential Challenges

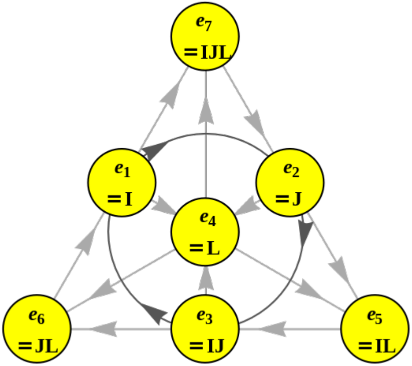
To classify integrable evolutionary PDEs over octonions

- decide on homogeneity weights of $u, \partial_t, \partial_x, L$ where $u = u(t,x)$ is octonion valued
- make a general ansatz (with minimal number of terms) for homogeneous polynomials F, L, M in $u, u_x, u_{xx}, \ldots$ with undetermined coefficients satisfying $u_t = F, \quad L_t = [L, M]$
- replace u by its component form $u = \sum_{i=0}^{7} u_i(x,t)e_i$
- split $L_t = [L, M]$ w.r.t. $e_i, u_i, u_{ix}, \ldots$
- solve the system of bilinear algebraic conditions for the undetermined coefficients in F, L, M to obtain integrable evolution equations $u_t = F$ together with their Lax-pair L, M.

Outline

About Octonions I

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



About Octonions II

- normed division algebra over the real numbers
- noncommutative, nonassociative but
  alternative: $x(xy) = (xx)y, \quad (yx)x = y(xx),$

About Octonions II

- normed division algebra over the real numbers
- 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
  $[x, y, x] = 0$.

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)
τ = # 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)
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

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)
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}

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)
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					

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)
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	?	?	?
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)
$\tau =$ # 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

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

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)
$\tau =$ # 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

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				

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)
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	?
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 $x, y, \ldots$ 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.

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:

$$[[a, b] \circ [c, d], e] = 0, \qquad (3)$$

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

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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	30	2	28
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	29	1	28
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	30	2	28
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	29	1	28
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	30	2	28
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	29	1	28
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	30	2	28
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	29	1	28
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	30	2	28
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	29	1	28
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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:

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

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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	72	40	32
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	56	24	32
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	40	8	32
$(uv)(wx) = ..., v \geq x, 0 = [u, v, wz]_{\{v,w\}\{u,z\}}$	32	0	32

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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	520	232	288
$(wu)v = ..., u \geq v,\ [wuv] = +[uvw]$	432	144	288
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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
$(wu)v = ..., w > v, [wuv] = -[vuw]$	790	640	150
$(wu)v = ..., u \geq v, [wuv] = +[uvw]$	525	375	150
$(wu)v = ..., w \geq u, [wuv] = +[vwu]$	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:

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

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 deree 4 polynomial.

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 deree 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 deree 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

Implementation

- *Question:* How can one do computations in a computer algebra system that does not know about non-associative variables and always simplifies $(ab)c - a(bc) = 0$?

Implementation

- *Question:* How can one do computations in a computer algebra system that does not know about non-associative variables and always simplifies $(ab)c - a(bc) = 0$?

  Answer: Label each product of octonions by a number and consider each product having 2 factors, i.e. each number refers to 2 numbers, its first and second factor.

Implementation

- *Question:* How can one do computations in a computer algebra system that does not know about non-associative variables and always simplifies $(ab)c - a(bc) = 0$?

  Answer: Label each product of octonions by a number and consider each product having 2 factors, i.e. each number refers to 2 numbers, its first and second factor.

  This also provides a total ordering of all terms which is needed when using identities to remove terms from P.

General Procedure I

- Formulate most general polynomial P for given d, v. Use known octonion identities of lower degree to remove terms in the ansatz for P that could be eliminated by those identities.
- Iterate
  - When formulating P use known relations between the undetermined coefficients, initially none.
  - Expand all octonions in terms of their 8 components (initially less later increasingly more) to get in the last run τ terms.
  - Split the expanded P wrt. the 8 e_i and the (max. c) components.
  - Solve the homogeneaous linear algebraic system $P = 0$ for the unknown coefficients and substitute the solution into P.
  - Append the list of known relations between the coefficients of P.

until $P \equiv 0$ or each octonion had 8 components in the last run.

General Procedure II

- For each free coefficient c_i in the general solution of $P = 0$ print the coefficient of c_i in P which is an identically vanishing polynomial in octonion variables. There are many identities, each can have 10s to 100s of terms and may have no symmetry because any linear combination of identities is an identity.
- Pick the shortest identity, add it in a list of identities and identify the leading term (product) in this identity.
- Formulate a pattern rule to identify this leading term and similar leading terms from similar identities after permutations of varibles.
- Set the coefficients of these terms in the general ansatz for P to zero and repeat the whole procedure.
- Repeat the whole process until P is identically zero.

Discussion

- We do not stop after the first run and use all computed identities because we would not be sure whether the leading terms of these identities are independent of each other because we do not only want an (ugly, non-intelligent) list of all identities but we want a complete, necessary and sufficient set of substitution rules to be usable to simplify a general P in applications when $P = L$ or $P = M$.
- If at any stage a very short identity results that is compactifyable due to its (anti-)symmetry then start from scratch with this being the first used identity to reduce P. As a result, the identities in the final list will be shorter, more symmetric and easier to understand.

Implementation Issues II

- A first implementation as a Pascal program to generate the most general ansatz of a quaternion or octonion polynomial modulo known identities for degreee 3 and 4 is highly efficient and was used for the work reported by Philic Lam. This program does not have to expand octonions in terms of their components and is therefore not capabale to compute or verify identities.
- The first program used to do that, the Maple module 'DifferentialGeometry' evaluating a degree 4 polynomial with 1280 terms had to be stopped after 6 days.
- An implementation from scratch in the computer algebra system Reduce was able to perform such a computation in about 2 hours which is progress but still too slow for deriving degreee 5 identities with 100 times as many terms (55 million) and an even higher factor of computation time.

Implementation Issues III

- The computer algebra system Reduce has an internal total ordering of all variables. A polynomial is stored as a polynomial in the leading variables with coefficients being polynomials in the 2nd variable and so on recursively. After computing the expanded P one wants to split with respect to e_i and the with respect to components u_i. If variables c_i would have a high priority then this splitting would be an exponentially slow process for large expressions. By giving e_i highest priority and c_i lowest priority, the splitting process has only linear complexity and is extremely fast.

Efficiency Measures I

- Starting with a smaller number of components, i.e. less than $8 \times v$, and setting the other components to 0 and thus deriving just some dependencies between coefficients c_i, the total number of terms of the expanded P is even increasing because terms of P do not have a single c_i as coefficient anymore but a whole linear sum of them. But there still is a benefit. As octonion terms in P are expanded and added one by one to P, they start to cancel as the identities vanish identically.

Efficiency Measures II

- When gradually increasing the number of non-zero components of one octonion variable u then for multilinear P in the next loop the already non-zero investigated components u_j can be set to zero because due to linearity of P in u the new non-zero components do not interfere with the new non-zero components. This reduces the efford by a factor of 8. Important is a good balance between on one hand breaking up the whole computation into many small increases in the number of non-zero components in order to gather as many as possible zero c_i and relations between non-zero c_i before the last loop with the most terms and on the other hand keeping redundancy low by computing many times P that does ot provide new information on the c_j. For example, for $n = d = 5$ a good strategy is to start with 14 randomly selected non-zero components and then in the next loops to add 7,6,5,4,3,2,2,... new non-zero components until the maximum 40 is reached.

Efficiency Measures III

- The extreme version of breaking up computation would be to go through all 8^d combinations of each octonion variable having only 1 component. The number of terms at any time would then be only t and not τ. But computing ans splitting P and solving the resulting conditions 8^d times involves too much redundancy ans is too slow, although it solves any memory problems. This would be beneficial under parallel execution.

Correctness

Question: Why are not all i identities obtained after the first run, used at once to drop i terms from the general polynomial P?

Correctness

Question: Why are not all i identities obtained after the first run, used at once to drop i terms from the general polynomial P?

Answer: One purpose is to avoid dropping terms that prevent the same identity. This goal is achieved if the number e of essential terms in P is unchanged after each run, i.e. after dropping the next term(s) from P.

Correctness

Question: Why are not all i identities obtained after the first run, used at once to drop i terms from the general polynomial P?

Answer: One purpose is to avoid dropping terms that prevent the same identity. This goal is achieved if the number e of essential terms in P is unchanged after each run, i.e. after dropping the next term(s) from P.

Another purpose is to get identities in their shortest, most symmetric and thus compactifyable form.

Outline

Summary

We obtained

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

Summary

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,

Summary

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

Summary

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

Summary

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)

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!