

Multiplication formula of twisted Jacobian algebras in terms of Hessians

Sangwook Lee (Joint work with Bo Jin, Kyeonghwan Lee)

2026 New Uzbekistan University - Soongsil University G-LAMP International Workshop

Plan

Setup

The two operators

Diagonal vs off-diagonal maps

The Hessian candidates

The ideal from off-diagonal maps

The result

Setup

Anticommutative variables

Consider three variables $\theta_1, \theta_2, \theta_3$ which *anticommute* with each other:

$$\theta_i \theta_j = -\theta_j \theta_i, \quad \text{in particular } \theta_i^2 = 0.$$

Such variables appear naturally in calculus of several variables, exterior algebras etc.

For $S = \mathbb{C}[x_1, x_2, x_3, y_1, y_2, y_3]$, we work with

$$S[\theta_1, \theta_2, \theta_3] = S \oplus \bigoplus_i S \cdot \theta_i \oplus \bigoplus_{i < j} S \cdot \theta_i \theta_j \oplus S \cdot \theta_1 \theta_2 \theta_3.$$

Differential operators

Adjoin differential operators $\partial_1, \partial_2, \partial_3$ subject to:

$$\theta_i \theta_j = -\theta_j \theta_i, \quad \partial_i \partial_j = -\partial_j \partial_i, \quad \partial_i \theta_j = -\theta_j \partial_i + \delta_{ij}$$

Example. $\partial_1 \theta_1 = 1 - \theta_1 \partial_1$.

Hence $\partial_1 \theta_1$ is itself a differential operator, and it acts on the left of θ_1 as zero (because $\theta_1^2 = 0$).

The full operator algebra is generated by polynomials in θ_i, ∂_i .

The two operators

The objects of main interest

Let g_{ij} , f_{ij} , $\bar{g}_{ij}$, $\bar{f}_{ij}$ be polynomials in the ζ_k 's. Define two operators of degree 3 in $\theta_1\theta_2\theta_3$:

$$\begin{aligned} e^{\eta\rho}(\theta_1\theta_2\theta_3) = & \theta_1\theta_2\theta_3 - g_{11}\theta_2\theta_3\partial_1 + g_{12}\theta_1\theta_3\partial_1 - g_{13}\theta_1\theta_2\partial_1 \\ & + g_{22}\theta_1\theta_3\partial_2 - g_{23}\theta_1\theta_2\partial_2 - g_{33}\theta_1\theta_2\partial_3 - f_{12}\theta_3 + f_{13}\theta_2 - f_{23}\theta_1 \\ & - g_{22}g_{11}\theta_3\partial_2\partial_1 + g_{23}g_{11}\theta_2\partial_2\partial_1 + g_{33}g_{11}\theta_2\partial_3\partial_1 + f_{23}g_{11}\partial_1 \\ & - g_{23}g_{12}\theta_1\partial_2\partial_1 - g_{33}g_{12}\theta_1\partial_3\partial_1 - f_{13}g_{12}\partial_1 + g_{22}g_{13}\theta_1\partial_2\partial_1 + f_{12}g_{13}\partial_1 \\ & - g_{22}g_{33}\theta_1\partial_3\partial_2 - f_{13}g_{22}\partial_2 + f_{12}g_{23}\partial_2 + f_{12}g_{33}\partial_3 + g_{33}g_{22}g_{11}\partial_3\partial_2\partial_1 \end{aligned} \quad (2.14)$$

$$\begin{aligned} e^{\eta\rho^{-1}}(\theta_1\theta_2\theta_3) = & \theta_1\theta_2\theta_3 - \bar{g}_{11}\theta_2\theta_3\partial_1 + \bar{g}_{12}\theta_1\theta_3\partial_1 - \bar{g}_{13}\theta_1\theta_2\partial_1 \\ & + \bar{g}_{22}\theta_1\theta_3\partial_2 - \bar{g}_{23}\theta_1\theta_2\partial_2 - \bar{g}_{33}\theta_1\theta_2\partial_3 - \bar{f}_{12}\theta_3 + \bar{f}_{13}\theta_2 - \bar{f}_{23}\theta_1 \\ & - \bar{g}_{22}\bar{g}_{11}\theta_3\partial_2\partial_1 + \bar{g}_{23}\bar{g}_{11}\theta_2\partial_2\partial_1 + \bar{g}_{33}\bar{g}_{11}\theta_2\partial_3\partial_1 + \bar{f}_{23}\bar{g}_{11}\partial_1 \\ & - \bar{g}_{23}\bar{g}_{12}\theta_1\partial_2\partial_1 - \bar{g}_{33}\bar{g}_{12}\theta_1\partial_3\partial_1 - \bar{f}_{13}\bar{g}_{12}\partial_1 + \bar{g}_{22}\bar{g}_{13}\theta_1\partial_2\partial_1 + \bar{f}_{12}\bar{g}_{13}\partial_1 \\ & - \bar{g}_{22}\bar{g}_{33}\theta_1\partial_3\partial_2 - \bar{f}_{13}\bar{g}_{22}\partial_2 + \bar{f}_{12}\bar{g}_{23}\partial_2 + \bar{f}_{12}\bar{g}_{33}\partial_3 + \bar{g}_{33}\bar{g}_{22}\bar{g}_{11}\partial_3\partial_2\partial_1 \end{aligned} \quad (2.15)$$

We are interested in the **product** of (2.14) and (2.15) due to its relevance in G -Frobenius algebra structure (emerging as a mirror to the G -covering of a symplectic manifold.)

The product: operator-by-operator expansion

The product is a long sum of monomials of the form

$$(\text{coefficient in } g, f, \bar{g}, \bar{f}) \cdot O$$

for various operators $O \in \{\theta_I \partial_J\}$.

After computer-algebra simplification, the product expands as 32 blocks indexed by operators O :

$$\theta_1 \theta_2 \theta_3 \partial_1, \theta_1 \theta_2 \theta_3 \partial_2, \dots, \theta_2 \theta_3, \theta_1 \theta_2, \dots, 1, \partial_i \partial_j.$$

We list the coefficient of each O as a polynomial in $g_{ij}, f_{ij}, \bar{g}_{ij}, \bar{f}_{ij}$.

Code used: generate.js — formal product of (2.14) and (2.15) (1/2)

```
1      const A = [
2        { c: "", o: "\\theta_1\\theta_2
3          \\theta_3" },
4        { c: "- g_{11}", o: "\\theta_2
5          \\theta_3\\partial_1" },
6        { c: "+ g_{12}", o: "\\theta_1
7          \\theta_3\\partial_1" },
8        { c: "- g_{13}", o: "\\theta_1
9          \\theta_2\\partial_1" },
10       { c: "+ g_{22}", o: "\\theta_1
11         \\theta_3\\partial_2" },
12       { c: "- g_{23}", o: "\\theta_1
13         \\theta_2\\partial_2" },
14       { c: "- g_{33}", o: "\\theta_1
15         \\theta_2\\partial_3" },
16     ],
17     { c: "- f_{12}", o: "\\theta_3"
18     },
19     { c: "+ f_{13}", o: "\\theta_2"
20     },
21     { c: "- f_{23}", o: "\\theta_1"
22     },
23     { c: "- g_{22}g_{11}", o: "\\
24       theta_3\\partial_2\\partial_1" },
25     { c: "+ g_{23}g_{11}", o: "\\
26       theta_2\\partial_2\\partial_1" },
27     { c: "+ g_{33}g_{11}", o: "\\
28       theta_2\\partial_3\\partial_1" },
29     { c: "+ f_{23}g_{11}", o: "\\
30       partial_1" },
31     { c: "- g_{23}g_{12}", o: "\\
32       theta_1\\partial_2\\partial_1" },
33     { c: "- g_{33}g_{12}", o: "\\
34       theta_1\\partial_3\\partial_1" },
35     { c: "- f_{13}g_{12}", o: "\\
36       partial_1" },
37     { c: "+ g_{22}g_{13}", o: "\\
38       theta_1\\partial_2\\partial_1" },
39     { c: "+ f_{12}g_{13}", o: "\\
40       partial_1" }
41   ]
42
43   { c: "- \\bar{g}_{23}", o: "\\
44     theta_1\\theta_2\\partial_2" },
45   { c: "- \\bar{g}_{33}", o: "\\
46     theta_1\\theta_2\\partial_3" },
47   { c: "- \\bar{f}_{12}", o: "\\
48     theta_3" },
49   { c: "+ \\bar{f}_{13}", o: "\\
50     theta_2" },
51   { c: "- \\bar{f}_{23}", o: "\\
52     theta_1" },
53   { c: "- \\bar{g}_{22}\\bar{g}_{
54     11}", o: "\\theta_3\\partial_2\\partial_1" },
55   { c: "+ \\bar{g}_{23}\\bar{g}_{
56     11}", o: "\\theta_2\\partial_2\\partial_1" },
57   { c: "+ \\bar{g}_{33}\\bar{g}_{
58     11}", o: "\\theta_2\\partial_3\\partial_1" },
59   { c: "+ \\bar{f}_{23}\\bar{g}_{
60     11}", o: "\\partial_1" },
61   { c: "- \\bar{g}_{23}\\bar{g}_{
62     12}", o: "\\theta_1\\partial_2\\partial_1" },
63   { c: "- \\bar{g}_{33}\\bar{g}_{
64     12}", o: "\\theta_1\\partial_3\\partial_1" },
65   { c: "- \\bar{f}_{13}\\bar{g}_{
66     12}", o: "\\partial_1" },
67   { c: "+ \\bar{g}_{22}\\bar{g}_{
68     13}", o: "\\theta_1\\partial_2\\partial_1" },
69   { c: "+ \\bar{f}_{12}\\bar{g}_{
70     13}", o: "\\partial_1" },
71   { c: "- \\bar{g}_{22}\\bar{g}_{
72     22}", o: "\\partial_2" },
73   { c: "+ \\bar{f}_{12}\\bar{g}_{
74     23}", o: "\\partial_2" },
75   { c: "+ \\bar{f}_{12}\\bar{g}_{
76     33}", o: "\\partial_3" },
77   { c: "+ \\bar{g}_{33}\\bar{g}_{
78     22}\\bar{g}_{11}", o: "\\partial_3\\partial_2\\partial_1" }
79 ]
```

Code used: `generate.js` — formal product of (2.14) and (2.15) (2/2)

```

59 .replace(/ /g, '');
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1
```

Code used: simplify.js — normal ordering and collecting terms (1/3)

```
1         import fs from 'fs';
2
3         const terms = JSON.parse(fs.
4         readFileSync('src/terms.json', 'utf8'));
5
6         function parseOp(opStr) {
7             const tokens = [];
8             let i = 0;
9             while (i < opStr.length
10             ) {
11                 if (opStr.
12                 startsWith('\theta_', i)) {
13                     tokens.
14                     push(opStr.substring(i, i + 8));
15                     i += 8;
16                 } else if (
17                 opStr.startsWith('\partial_', i)) {
18                     tokens.
19                     push(opStr.substring(i, i + 10));
20                     i +=
21                     10;
22                 } else {
23                     i++;
24                 }
25             }
26             return tokens;
27         }
28
29         function simplifyTokens(tokens)
30         {
31             let results = [{ sign:
32             1, tokens: [...tokens] }];
33
34             let changed = true;
35
36             while (changed) {
37                 changed = false
38
39                 let newResults
40
41                 if (tA.startsWith('\partial_') && tB.startsWith('\theta_')) {
42                     let idx1 = tA.substring(9);
43                     let idx2 = tB.substring(7);
44
45                     if (idx1 === idx2) {
46                         let t1 = [...res.tokens];
47                         t1.splice(i, 2);
48                         newResults.push({ sign: res.sign, tokens
49                         : t1 });
50
51                         let t2 = [...res.tokens];
52                         t2[i] = tB;
53                         t2[i+1] = tA;
54                         newResults.push({ sign: -res.sign,
55                         tokens: t2 });
56                     } else {
57                         let t2 = [...res.tokens];
58                         t2[i] = tB;
59                         t2[i+1] = tA.
```

Code used: simplify.js — normal ordering and collecting terms (2/3)

```
66         changed = true;
67         break;
68     } else if (idx1 === idx2) {
69         // zero
70         found = true;
71         changed = true;
72         break;
73     }
74     } else if (tA.startsWith('\\partial_') && tB.startsWith(
75     '\\partial_')) {
76         let idx1 = parseInt(tA.substring(9));
77         let idx2 = parseInt(tB.substring(9));
78         if (idx1 > idx2) {
79             let t2 = [...res.tokens];
80             t2[i] = tB;
81             t2[i+1] = tA;
82             newResults.push({ sign: -res.sign,
83             tokens: t2 });
84             found = true;
85             changed = true;
96         }
97         return results;
98     }
99     const simplifiedTerms = [];
100     for (const term of terms) {
101         const tokens = parseOp(
102             term.op);
103         simplifyTokens(tokens);
104         for (const res of
105             results) {
106             let newSign =
107                 if (res.sign
108                     newSign
109                     === -1) {
110                         = newSign === '+' ? '-' : '+';
111                     }
112             let newOp = res
113                 simplifiedTerms
114                 sign:
115                 coeff:
116                 op:
117             });
118         }
119         // Group terms with the same
120         // coeff and op
```

Code used: simplify.js — normal ordering and collecting terms (3/3)

```
131         grouped[key] += (term.  
132         sign === '+' ? 1 : -1);  
133     }  
134     const finalTerms = [];  
135     for (const key in grouped) {  
136         const count = grouped[  
137         key];  
138         if (count !== 0) {  
139             const [coeff,  
140             let sign =  
141             count > 0 ? '+' : '-';  
142             let absCount =  
143             Math.abs(count);  
144             let newCoeff =  
145             coeff;  
146             if (absCount >  
147             1) {  
148                 // If  
149                 coeff is empty, just use count  
150                 if (  
151                 newCoeff === '') {  
152                     newCoeff = absCount.toString();  
153                 } else  
154                 {  
155                     newCoeff = absCount.toString() + newCoeff;  
156                 }  
157             }  
158             finalTerms.push  
159             ({  
160                 sign,  
161                 coeff:  
162                 newCoeff,  
163                 op  
164             }));  
165         }
```

An example coefficient

For $O = \theta_2\theta_3$:

$$\text{coeff} = -\bar{f}_{12}f_{13} + \bar{f}_{13}f_{12} + \bar{f}_{23}g_{11} + f_{12}g_{13} - f_{13}g_{12} + f_{23}g_{11} + g_{11}g_{23}.$$

For $O = \theta_1\theta_2\theta_3\partial_3$:

$$\text{coeff} = \bar{f}_{12}\bar{g}_{33} - \bar{f}_{12}g_{33} + \bar{g}_{33}f_{12} + \bar{g}_{33}g_{12} - f_{12}g_{33} - g_{12}g_{33}.$$

Every block is a homogeneous polynomial in $g, f, \bar{g}, \bar{f}$.

An example coefficient

For $O = \theta_2\theta_3$:

$$\text{coeff} = -\bar{f}_{12}f_{13} + \bar{f}_{13}f_{12} + \bar{f}_{23}g_{11} + f_{12}g_{13} - f_{13}g_{12} + f_{23}g_{11} + g_{11}g_{23}.$$

For $O = \theta_1\theta_2\theta_3\partial_3$:

$$\text{coeff} = \bar{f}_{12}\bar{g}_{33} - \bar{f}_{12}g_{33} + \bar{g}_{33}f_{12} + \bar{g}_{33}g_{12} - f_{12}g_{33} - g_{12}g_{33}.$$

Every block is a homogeneous polynomial in $g, f, \bar{g}, \bar{f}$.

Bookkeeping question. How do we read off the multiplicative structure of this product?

Diagonal vs off-diagonal maps

Two kinds of components

The product naturally splits into two kinds of pieces.

(1) Diagonal maps — preserve some θ -component:

$$S \theta_i \mapsto S \theta_i, \quad S \theta_i \theta_j \mapsto S \theta_i \theta_j, \quad \dots$$

(2) Off-diagonal maps — do not preserve any θ -component:

$$S \theta_3 \mapsto S \theta_1, \quad S 1 \mapsto S \theta_1 \theta_2, \quad \dots$$

Two kinds of components

The product naturally splits into two kinds of pieces.

(1) **Diagonal maps** — preserve some θ -component:

$$S \theta_i \mapsto S \theta_i, \quad S \theta_i \theta_j \mapsto S \theta_i \theta_j, \quad \dots$$

(2) **Off-diagonal maps** — do not preserve any θ -component:

$$S \theta_3 \mapsto S \theta_1, \quad S 1 \mapsto S \theta_1 \theta_2, \quad \dots$$

Fact. Each diagonal component is what we take into account as the result of multiplication:
though the components look different, they are equivalent *modulo an ideal*.

Code used: simplify_diagonal.js — diagonal maps $\Delta(x)$ (1/2)

```
1         import fs from 'fs';
2
3         const terms = JSON.parse(fs.
4         readFileSync('src/terms_simplified.json', 'utf8'));
5
6         const basis = [
7         { name: '1', indices: [] },
8         { name: '\\\\theta_1', indices:
9         [1] },
10        { name: '\\\\theta_2', indices:
11        [2] },
12        { name: '\\\\theta_3', indices:
13        [3] },
14        { name: '\\\\theta_1\\\\theta_2',
15        indices: [1, 2] },
16        { name: '\\\\theta_1\\\\theta_3',
17        indices: [1, 3] },
18        { name: '\\\\theta_2\\\\theta_3',
19        indices: [2, 3] },
20        { name: '\\\\theta_1\\\\theta_2\\\\
21        theta_3', indices: [1, 2, 3] }
22        ];
23
24        function parseOp(opStr) {
25            const theta = [];
26            const del = [];
27            const regex = /\\\\(theta
28            |partial)_(\\d)/g;
29
30            let match;
31            while ((match = regex.
32            exec(opStr)) !== null) {
33                if (match[1]
34                === 'theta') theta.push(parseInt(match[2]));
35                if (match[1]
36                === 'partial') del.push(parseInt(match[2]));
37            }
38            return { theta, del };
39        }
40
41        for (let i = op.del.
42        length - 1; i >= 0; i--) {
43            const d = op.
44            del[i];
45            const idx =
46            current.indexOf(d);
47            if (idx === -1)
48                return { sign: 0, indices: [] };
49            if (idx % 2 ===
50            1) sign *= -1;
51            current.splice(
52            idx, 1);
53        }
54
55        for (let i = op.theta.
56        length - 1; i >= 0; i--) {
57            const t = op.
58            theta[i];
59            if (current.
60            includes(t)) return { sign: 0, indices: [] };
61            let insertIdx =
62            0;
63            while (
64            insertIdx < current.length && current[insertIdx] < t) {
65                insertIdx++;
66            }
67            const
68            inversions = insertIdx;
69            if (inversions
70            % 2 === 1) sign *= -1;
71            current.splice(
72            insertIdx, 0, t);
73        }
74
75        return { sign, indices:
76        current };
77    }
```

Code used: simplify_diagonal.js — diagonal maps $\Delta(x)$ (2/2)

```

64         for (const b of basis) {
65             const combined = {};
66
67             for (const term of
68                 terms) {
69                 const op =
70                 parseOp(term.op);
71                 const applied =
72                 applyOp(op, b.indices);
73                 sign !== 0) {
74                     const
75                     targetName = basis.find(x => x.indices.join(',') === applied.
76                     indices.join(',')).name;
77                     if (
78                     targetName === b.name) {
79
80                         const normCoeff = normalizeCoeff(term.coeff);
81
82                         let termSign = term.sign === '-' ? -1 : 1;
83
84                         let totalSign = termSign * applied.sign;
85
86                         if (!combined[normCoeff]) combined[normCoeff] = 0;
87
88                         combined[normCoeff] += totalSign;
89                     }
90                 }
91             }
92
93             let expr = '';
94             const sortedCoeffs =
95             Object.keys(combined).sort();
96
97             sortedCoeffs) {
98                 const count =
99                 for (const c of
100                     combined[c]
101                     if (expr === ''
102                     if (
103                     expr +=
104                     signStr === '-') expr += '- ';
105                     countStr + termStr;
106                     } else {
107                     expr +=
108                     ' ${signStr} ${countStr}${termStr}';
109                     }
110                 if (expr === '') expr =
111                 '0';
112                 diagonalResults.push({
113                     generator: b.
114                     name,
115                     expr: expr
116                 });
117             }
118             fs.writeFileSync('src/
119             diagonal_maps_simplified.json', JSON.stringify(diagonalResults,
120             null, 2));
121             console.log('Simplified
122             diagonal maps.');
```

The diagonal maps Δ

For each generator $x \in \{1, \theta_1, \theta_2, \theta_3, \theta_1\theta_2, \theta_1\theta_3, \theta_2\theta_3, \theta_1\theta_2\theta_3\}$ we record $\Delta(x)$, the diagonal component.

Two simple examples after simplification:

$$\begin{aligned}\Delta(1) &= -\bar{f}_{12}f_{12}g_{33} + \bar{f}_{13}f_{12}g_{23} - \bar{f}_{13}f_{13}g_{22} \\ &\quad - \bar{f}_{23}f_{12}g_{13} + \bar{f}_{23}f_{13}g_{12} - \bar{f}_{23}f_{23}g_{11} + g_{11}g_{22}g_{33}, \\ \Delta(\theta_3) &= -\bar{f}_{12}\bar{g}_{33}f_{12} + \bar{f}_{13}f_{12}g_{23} - \bar{f}_{13}f_{13}g_{22} \\ &\quad - \bar{f}_{23}f_{12}g_{13} + \bar{f}_{23}f_{13}g_{12} - \bar{f}_{23}f_{23}g_{11} + \bar{g}_{33}g_{11}g_{22}.\end{aligned}$$

The remaining six components have analogous (more or less symmetric) shapes.

$$\Delta(\theta_2\theta_3)$$

The map of central interest in this talk:

$$\begin{aligned}\Delta(\theta_2\theta_3) = & \bar{f}_{12}\bar{g}_{23}f_{13} - \bar{f}_{12}\bar{g}_{33}f_{12} - \bar{f}_{13}\bar{g}_{22}f_{13} \\ & - \bar{f}_{23}f_{12}g_{13} + \bar{f}_{23}f_{13}g_{12} - \bar{f}_{23}f_{23}g_{11} - \bar{f}_{23}g_{11}g_{23} \\ & + \bar{g}_{22}\bar{g}_{33}g_{11} \\ & - \bar{g}_{23}f_{12}g_{13} + \bar{g}_{23}f_{13}g_{12} - \bar{g}_{23}f_{23}g_{11} - \bar{g}_{23}g_{11}g_{23}.\end{aligned}$$

Twelve terms; each a degree-3 monomial in $g, f, \bar{g}, \bar{f}$.

The Hessian candidates

Can one of the diagonal maps $\Delta(x)$
(or a linear combination) be realised as a determinant?

Can one of the diagonal maps $\Delta(x)$
(or a linear combination) be realised as a determinant?

We have the *Hessian-type* determinant in mind — there are several natural 3×3 candidates whose determinants are degree 3 polynomials in our generators. Each g_{ij} or f_{ij} is a second *difference* of a polynomial W , so we do not expect it just as a coincidence.

Four candidate matrices

$$\begin{aligned} M_1 &= \begin{pmatrix} g_{11} & -f_{12} & -f_{13} \\ -g_{12} - f_{12} & g_{22} & f_{23} \\ -\bar{f}_{13} & -\bar{f}_{23} & \bar{g}_{33} \end{pmatrix}, & M_2 &= \begin{pmatrix} g_{11} & -f_{12} & -f_{13} \\ -\bar{f}_{12} & \bar{g}_{22} & \bar{g}_{23} + \bar{f}_{23} \\ -\bar{f}_{13} & -\bar{f}_{23} & \bar{g}_{33} \end{pmatrix}, \\ M_3 &= \begin{pmatrix} \bar{g}_{11} & \bar{g}_{12} + \bar{f}_{12} & \bar{g}_{13} + \bar{f}_{13} \\ -g_{12} - f_{12} & g_{22} & f_{23} \\ g_{13} - f_{13} & g_{23} - f_{23} & -g_{33} \end{pmatrix}, & M_4 &= \begin{pmatrix} \bar{g}_{11} & \bar{g}_{12} + \bar{f}_{12} & \bar{g}_{13} + \bar{f}_{13} \\ -\bar{f}_{12} & \bar{g}_{22} & \bar{g}_{23} + \bar{f}_{23} \\ g_{13} - f_{13} & g_{23} - f_{23} & -g_{33} \end{pmatrix}. \end{aligned}$$

Task. Check whether $\det(M_i)$ is equivalent to one of the diagonal maps $\Delta(x)$ modulo an ideal $\mathcal{I}$. We came up with these matrices since each row gives rise to a partial derivative of W by certain linear combination.

The ideal from off-diagonal maps

The obstacle, and the idea

Problem. Diagonal maps and determinants are just formal polynomials in $g_{ij}, f_{ij}, \bar{g}_{ij}, \bar{f}_{ij}$. There is no natural way of identifying diagonals and determinants.

The obstacle, and the idea

Problem. Diagonal maps and determinants are just formal polynomials in $g_{ij}, f_{ij}, \bar{g}_{ij}, \bar{f}_{ij}$. There is no natural way of identifying diagonals and determinants.

Idea. The off-diagonal components are exactly the pieces that are used for the identification!

Quadratic off-diagonal maps

Among the off-diagonal maps, the ones whose right-hand side is a homogeneous **degree-2** polynomial in the generators are exactly six — those going from θ -degree d to θ -degree $d + 2$, with $d \in \{0, 1\}$.

$$O(1 \rightarrow \theta_1 \theta_2) = \bar{f}_{12}g_{33} - \bar{f}_{13}f_{23} - \bar{f}_{13}g_{23} + \bar{f}_{23}f_{13} + \bar{f}_{23}g_{13} + f_{12}g_{33} + g_{12}g_{33},$$

$$O(1 \rightarrow \theta_1 \theta_3) = \bar{f}_{12}f_{23} + \bar{f}_{13}g_{22} - \bar{f}_{23}f_{12} - \bar{f}_{23}g_{12} - f_{12}g_{23} + f_{13}g_{22} - g_{12}g_{23} + g_{13}g_{22},$$

$$O(1 \rightarrow \theta_2 \theta_3) = -\bar{f}_{12}f_{13} + \bar{f}_{13}f_{12} + \bar{f}_{23}g_{11} + f_{12}g_{13} - f_{13}g_{12} + f_{23}g_{11} + g_{11}g_{23},$$

and three more $O(\theta_i \rightarrow \theta_1 \theta_2 \theta_3)$, $i = 1, 2, 3$.

The result

Theorem (computer-verified identity)

$$\Delta(\theta_2\theta_3) = \det(M_2) - (\bar{f}_{23} + \bar{g}_{23}) O(1 \rightarrow \theta_2\theta_3)$$

or, equivalently,

$$\det(M_2) \equiv \Delta(\theta_2\theta_3) \pmod{\mathcal{I}}.$$

- Among M_1, M_2, M_3, M_4 , only M_2 admits such a relation with $\Delta(\theta_2\theta_3)$.
- The multiplier $\bar{f}_{23} + \bar{g}_{23}$ is a linear element — this is the cleanest correction possible.

Code used: get_m4_combination.js — ideal membership problem (1/4)

```
1         import fs from 'fs';
2
3         const diagMaps = JSON.parse(fs.
4         readFileSync('src/diagonal_maps_simplified.json', 'utf8'));
5         const allOffDiagMaps = JSON.
6         parse(fs.readFileSync('src/all_off_diagonal_maps.json', 'utf8'));
7
8         function parseExprToMap(expr) {
9             const map = {};
10             if (expr === '0')
11                 return map;
12
13             let s = expr.replace(/
14             /g, '');
15             if (!s.startsWith('+')) {
16                 s = '+' + s;
17             }
18             const regex = /[+-]([
19             a-zA-Z0-9_{\\}*)/g;
20             let match;
21             while ((match = regex.
22             exec(s)) !== null) {
23                 const sign =
24                 match[1] === '+' ? 1 : -1;
25                 match[2];
26
27                 ) continue;
28
29                 let coeff = 1;
30                 const
31                 if (coeffMatch)
32                     if (coeffMatch
33                     coeff =
34
35             ) term = '1';
36
37             map[term] = (
38                 map[term] || 0) + sign * coeff;
39             }
40             return map;
41         }
42
43         const variables = [
44             "f_{12}", "f_{13}", "f_{23}",
45             "\\bar{f}_{12}", "\\bar{f}_{13}",
46             "\\bar{f}_{23}",
47             "g_{11}", "g_{12}", "g_{13}", "
48             g_{22}", "g_{23}", "g_{33}",
49             "\\bar{g}_{11}", "\\bar{g}_{12}",
50             "\\bar{g}_{13}", "\\bar{g}_{22}", "\\bar{g}_{23}", "\\bar{g}_{
51             33}"
52         ];
53
54         const vectors = [];
55         const vectorNames = [];
56
57         diagMaps.forEach((m, i) => {
58             vectors.push(
59                 vectorNames.push('D_{$i
60                 +1}');
61             });
62
63         allOffDiagMaps.forEach((m) => {
64             if (m.expr === '0')
65                 const baseMap =
66                 parseExprToMap(m.expr);
67
68             const firstTerm =
69                 Object.keys(baseMap)[0];
70             const factors =
```

Code used: get_m4_combination.js — ideal membership problem (2/4)

```
57         const degree = factors. 84
length;
58
59         if (degree === 2) { 85
60             variables. 86
61
62             for ( 87
63                 const 88
64                 multipliedMap = {}; 89
65                 for ( 90
66                 const [term, coeff] of Object.entries(baseMap)) { 91
67                     let f = term === '1' ? [] : term.match(/\{\d+\}/g) || []; 92
68                     f.push(v); 93
69                     f.sort(); 94
70                     const newTerm = f.join(''); 95
71                     multipliedMap[newTerm] = (multipliedMap[newTerm] || 0) + 96
coeff; 97
72                     } 98
73                     vectors 99
74                     .push(multipliedMap); 100
75                     vectorNames.push(`${v} * 0(${m.source}->${m.target})`); 101
76                     } else if (degree === 102
77                     3) { 103
78                         vectors.push( 104
79                         baseMap); 105
80                         vectorNames. 106
81                         push(`${0(${m.source}->${m.target})}`); 107
82                     } 108
83                     } 109
84                     } 110
85                     } 111
86                     } 112
87                     } 113
88                     } 114
89                     } 115
90                     } 116
91                     } 117
92                     } 118
93                     } 119
94                     } 120
95                     } 121
96                     } 122
97                     } 123
98                     } 124
99                     } 125
100                     } 126
101                     } 127
102                     } 128
103                     } 129
104                     } 130
105                     } 131
106                     } 132
107                     } 133
108                     } 134
109                     } 135
110                     } 136
111                     } 137
112                     } 138
113                     } 139
114                     } 140
115                     } 141
116                     } 142
117                     } 143
118                     } 144
119                     } 145
120                     } 146
121                     } 147
122                     } 148
123                     } 149
124                     } 150
125                     } 151
126                     } 152
127                     } 153
128                     } 154
129                     } 155
130                     } 156
131                     } 157
132                     } 158
133                     } 159
134                     } 160
135                     } 161
136                     } 162
137                     } 163
138                     } 164
139                     } 165
140                     } 166
141                     } 167
142                     } 168
143                     } 169
144                     } 170
145                     } 171
146                     } 172
147                     } 173
148                     } 174
149                     } 175
150                     } 176
151                     } 177
152                     } 178
153                     } 179
154                     } 180
155                     } 181
156                     } 182
157                     } 183
158                     } 184
159                     } 185
160                     } 186
161                     } 187
162                     } 188
163                     } 189
164                     } 190
165                     } 191
166                     } 192
167                     } 193
168                     } 194
169                     } 195
170                     } 196
171                     } 197
172                     } 198
173                     } 199
174                     } 200
175                     } 201
176                     } 202
177                     } 203
178                     } 204
179                     } 205
180                     } 206
181                     } 207
182                     } 208
183                     } 209
184                     } 210
185                     } 211
186                     } 212
187                     } 213
188                     } 214
189                     } 215
190                     } 216
191                     } 217
192                     } 218
193                     } 219
194                     } 220
195                     } 221
196                     } 222
197                     } 223
198                     } 224
199                     } 225
200                     } 226
201                     } 227
202                     } 228
203                     } 229
204                     } 230
205                     } 231
206                     } 232
207                     } 233
208                     } 234
209                     } 235
210                     } 236
211                     } 237
212                     } 238
213                     } 239
214                     } 240
215                     } 241
216                     } 242
217                     } 243
218                     } 244
219                     } 245
220                     } 246
221                     } 247
222                     } 248
223                     } 249
224                     } 250
225                     } 251
226                     } 252
227                     } 253
228                     } 254
229                     } 255
230                     } 256
231                     } 257
232                     } 258
233                     } 259
234                     } 260
235                     } 261
236                     } 262
237                     } 263
238                     } 264
239                     } 265
240                     } 266
241                     } 267
242                     } 268
243                     } 269
244                     } 270
245                     } 271
246                     } 272
247                     } 273
248                     } 274
249                     } 275
250                     } 276
251                     } 277
252                     } 278
253                     } 279
254                     } 280
255                     } 281
256                     } 282
257                     } 283
258                     } 284
259                     } 285
260                     } 286
261                     } 287
262                     } 288
263                     } 289
264                     } 290
265                     } 291
266                     } 292
267                     } 293
268                     } 294
269                     } 295
270                     } 296
271                     } 297
272                     } 298
273                     } 299
274                     } 300
275                     } 301
276                     } 302
277                     } 303
278                     } 304
279                     } 305
280                     } 306
281                     } 307
282                     } 308
283                     } 309
284                     } 310
285                     } 311
286                     } 312
287                     } 313
288                     } 314
289                     } 315
290                     } 316
291                     } 317
292                     } 318
293                     } 319
294                     } 320
295                     } 321
296                     } 322
297                     } 323
298                     } 324
299                     } 325
300                     } 326
301                     } 327
302                     } 328
303                     } 329
304                     } 330
305                     } 331
306                     } 332
307                     } 333
308                     } 334
309                     } 335
310                     } 336
311                     } 337
312                     } 338
313                     } 339
314                     } 340
315                     } 341
316                     } 342
317                     } 343
318                     } 344
319                     } 345
320                     } 346
321                     } 347
322                     } 348
323                     } 349
324                     } 350
325                     } 351
326                     } 352
327                     } 353
328                     } 354
329                     } 355
330                     } 356
331                     } 357
332                     } 358
333                     } 359
334                     } 360
335                     } 361
336                     } 362
337                     } 363
338                     } 364
339                     } 365
340                     } 366
341                     } 367
342                     } 368
343                     } 369
344                     } 370
345                     } 371
346                     } 372
347                     } 373
348                     } 374
349                     } 375
350                     } 376
351                     } 377
352                     } 378
353                     } 379
354                     } 380
355                     } 381
356                     } 382
357                     } 383
358                     } 384
359                     } 385
360                     } 386
361                     } 387
362                     } 388
363                     } 389
364                     } 390
365                     } 391
366                     } 392
367                     } 393
368                     } 394
369                     } 395
370                     } 396
371                     } 397
372                     } 398
373                     } 399
374                     } 400
375                     } 401
376                     } 402
377                     } 403
378                     } 404
379                     } 405
380                     } 406
381                     } 407
382                     } 408
383                     } 409
384                     } 410
385                     } 411
386                     } 412
387                     } 413
388                     } 414
389                     } 415
390                     } 416
391                     } 417
392                     } 418
393                     } 419
394                     } 420
395                     } 421
396                     } 422
397                     } 423
398                     } 424
399                     } 425
400                     } 426
401                     } 427
402                     } 428
403                     } 429
404                     } 430
405                     } 431
406                     } 432
407                     } 433
408                     } 434
409                     } 435
410                     } 436
411                     } 437
412                     } 438
413                     } 439
414                     } 440
415                     } 441
416                     } 442
417                     } 443
418                     } 444
419                     } 445
420                     } 446
421                     } 447
422                     } 448
423                     } 449
424                     } 450
425                     } 451
426                     } 452
427                     } 453
428                     } 454
429                     } 455
430                     } 456
431                     } 457
432                     } 458
433                     } 459
434                     } 460
435                     } 461
436                     } 462
437                     } 463
438                     } 464
439                     } 465
440                     } 466
441                     } 467
442                     } 468
443                     } 469
444                     } 470
445                     } 471
446                     } 472
447                     } 473
448                     } 474
449                     } 475
450                     } 476
451                     } 477
452                     } 478
453                     } 479
454                     } 480
455                     } 481
456                     } 482
457                     } 483
458                     } 484
459                     } 485
460                     } 486
461                     } 487
462                     } 488
463                     } 489
464                     } 490
465                     } 491
466                     } 492
467                     } 493
468                     } 494
469                     } 495
470                     } 496
471                     } 497
472                     } 498
473                     } 499
474                     } 500
475                     } 501
476                     } 502
477                     } 503
478                     } 504
479                     } 505
480                     } 506
481                     } 507
482                     } 508
483                     } 509
484                     } 510
485                     } 511
486                     } 512
487                     } 513
488                     } 514
489                     } 515
490                     } 516
491                     } 517
492                     } 518
493                     } 519
494                     } 520
495                     } 521
496                     } 522
497                     } 523
498                     } 524
499                     } 525
500                     } 526
501                     } 527
502                     } 528
503                     } 529
504                     } 530
505                     } 531
506                     } 532
507                     } 533
508                     } 534
509                     } 535
510                     } 536
511                     } 537
512                     } 538
513                     } 539
514                     } 540
515                     } 541
516                     } 542
517                     } 543
518                     } 544
519                     } 545
520                     } 546
521                     } 547
522                     } 548
523                     } 549
524                     } 550
525                     } 551
526                     } 552
527                     } 553
528                     } 554
529                     } 555
530                     } 556
531                     } 557
532                     } 558
533                     } 559
534                     } 560
535                     } 561
536                     } 562
537                     } 563
538                     } 564
539                     } 565
540                     } 566
541                     } 567
542                     } 568
543                     } 569
544                     } 570
545                     } 571
546                     } 572
547                     } 573
548                     } 574
549                     } 575
550                     } 576
551                     } 577
552                     } 578
553                     } 579
554                     } 580
555                     } 581
556                     } 582
557                     } 583
558                     } 584
559                     } 585
560                     } 586
561                     } 587
562                     } 588
563                     } 589
564                     } 590
565                     } 591
566                     } 592
567                     } 593
568                     } 594
569                     } 595
570                     } 596
571                     } 597
572                     } 598
573                     } 599
574                     } 600
575                     } 601
576                     } 602
577                     } 603
578                     } 604
579                     } 605
580                     } 606
581                     } 607
582                     } 608
583                     } 609
584                     } 610
585                     } 611
586                     } 612
587                     } 613
588                     } 614
589                     } 615
590                     } 616
591                     } 617
592                     } 618
593                     } 619
594                     } 620
595                     } 621
596                     } 622
597                     } 623
598                     } 624
599                     } 625
600                     } 626
601                     } 627
602                     } 628
603                     } 629
604                     } 630
605                     } 631
606                     } 632
607                     } 633
608                     } 634
609                     } 635
610                     } 636
611                     } 637
612                     } 638
613                     } 639
614                     } 640
615                     } 641
616                     } 642
617                     } 643
618                     } 644
619                     } 645
620                     } 646
621                     } 647
622                     } 648
623                     } 649
624                     } 650
625                     } 651
626                     } 652
627                     } 653
628                     } 654
629                     } 655
630                     } 656
631                     } 657
632                     } 658
633                     } 659
634                     } 660
635                     } 661
636                     } 662
637                     } 663
638                     } 664
639                     } 665
640                     } 666
641                     } 667
642                     } 668
643                     } 669
644                     } 670
645                     } 671
646                     } 672
647                     } 673
648                     } 674
649                     } 675
650                     } 676
651                     } 677
652                     } 678
653                     } 679
654                     } 680
655                     } 681
656                     } 682
657                     } 683
658                     } 684
659                     } 685
660                     } 686
661                     } 687
662                     } 688
663                     } 689
664                     } 690
665                     } 691
666                     } 692
667                     } 693
668                     } 694
669                     } 695
670                     } 696
671                     } 697
672                     } 698
673                     } 699
674                     } 700
675                     } 701
676                     } 702
677                     } 703
678                     } 704
679                     } 705
680                     } 706
681                     } 707
682                     } 708
683                     } 709
684                     } 710
685                     } 711
686                     } 712
687                     } 713
688                     } 714
689                     } 715
690                     } 716
691                     } 717
692                     } 718
693                     } 719
694                     } 720
695                     } 721
696                     } 722
697                     } 723
698                     } 724
699                     } 725
700                     } 726
701                     } 727
702                     } 728
703                     } 729
704                     } 730
705                     } 731
706                     } 732
707                     } 733
708                     } 734
709                     } 735
710                     } 736
711                     } 737
712                     } 738
713                     } 739
714                     } 740
715                     } 741
716                     } 742
717                     } 743
718                     } 744
719                     } 745
720                     } 746
721                     } 747
722                     } 748
723                     } 749
724                     } 750
725                     } 751
726                     } 752
727                     } 753
728                     } 754
729                     } 755
730                     } 756
731                     } 757
732                     } 758
733                     } 759
734                     } 760
735                     } 761
736                     } 762
737                     } 763
738                     } 764
739                     } 765
740                     } 766
741                     } 767
742                     } 768
743                     } 769
744                     } 770
745                     } 771
746                     } 772
747                     } 773
748                     } 774
749                     } 775
750                     } 776
751                     } 777
752                     } 778
753                     } 779
754                     } 780
755                     } 781
756                     } 782
757                     } 783
758                     } 784
759                     } 785
760                     } 786
761                     } 787
762                     } 788
763                     } 789
764                     } 790
765                     } 791
766                     } 792
767                     } 793
768                     } 794
769                     } 795
770                     } 796
771                     } 797
772                     } 798
773                     } 799
774                     } 800
775                     } 801
776                     } 802
777                     } 803
778                     } 804
779                     } 805
780                     } 806
781                     } 807
782                     } 808
783                     } 809
784                     } 810
785                     } 811
786                     } 812
787                     } 813
788                     } 814
789                     } 815
790                     } 816
791                     } 817
792                     } 818
793                     } 819
794                     } 820
795                     } 821
796                     } 822
797                     } 823
798                     } 824
799                     } 825
800                     } 826
801                     } 827
802                     } 828
803                     } 829
804                     } 830
805                     } 831
806                     } 832
807                     } 833
808                     } 834
809                     } 835
810                     } 836
811                     } 837
812                     } 838
813                     } 839
814                     } 840
815                     } 841
816                     } 842
817                     } 843
818                     } 844
819                     } 845
820                     } 846
821                     } 847
822                     } 848
823                     } 849
824                     } 850
825                     } 851
826                     } 852
827                     } 853
828                     } 854
829                     } 855
830                     } 856
831                     } 857
832                     } 858
833                     } 859
834                     } 860
835                     } 861
836                     } 862
837                     } 863
838                     } 864
839                     } 865
840                     } 866
841                     } 867
842                     } 868
843                     } 869
844                     } 870
845                     } 871
846                     } 872
847                     } 873
848                     } 874
849                     } 875
850                     } 876
851                     } 877
852                     } 878
853                     } 879
854                     } 880
855                     } 881
856                     } 882
857                     } 883
858                     } 884
859                     } 885
860                     } 886
861                     } 887
862                     } 888
863                     } 889
864                     } 890
865                     } 891
866                     } 892
867                     } 893
868                     } 894
869                     } 895
870                     } 896
871                     } 897
872                     } 898
873                     } 899
874                     } 900
875                     } 901
876                     } 902
877                     } 903
878                     } 904
879                     } 905
880                     } 906
881                     } 907
882                     } 908
883                     } 909
884                     } 910
885                     } 911
886                     } 912
887                     } 913
888                     } 914
889                     } 915
890                     } 916
891                     } 917
892                     } 918
893                     } 919
894                     } 920
895                     } 921
896                     } 922
897                     } 923
898                     } 924
899                     } 925
900                     } 926
901                     } 927
902                     } 928
903                     } 929
904                     } 930
905                     } 931
906                     } 932
907                     } 933
908                     } 934
909                     } 935
910                     } 936
911                     } 937
912                     } 938
913                     } 939
914                     } 940
915                     } 941
916                     } 942
917                     } 943
918                     } 944
919                     } 945
920                     } 946
921                     } 947
922                     } 948
923                     } 949
924                     } 950
925                     } 951
926                     } 952
927                     } 953
928                     } 954
929                     } 955
930                     } 956
931                     } 957
932                     } 958
933                     } 959
934                     } 960
935                     } 961
936                     } 962
937                     } 963
938                     } 964
939                     } 965
940                     } 966
941                     } 967
942                     } 968
943                     } 969
944                     } 970
945                     } 971
946                     } 972
947                     } 973
948                     } 974
949                     } 975
950                     } 976
951                     } 977
952                     } 978
953                     } 979
954                     } 980
955                     } 981
956                     } 982
957                     } 983
958                     } 984
959                     } 985
960                     } 986
961                     } 987
962                     } 988
963                     } 989
964                     } 990
965                     } 991
966                     } 992
967                     } 993
968                     } 994
969                     } 995
970                     } 996
971                     } 997
972                     } 998
973                     } 999
974                     } 1000
975                     } 1001
976                     } 1002
977                     } 1003
978                     } 1004
979                     } 1005
980                     } 1006
981                     } 1007
982                     } 1008
983                     } 1009
984                     } 1010
985                     } 1011
986                     } 1012
987                     } 1013
988                     } 1014
989                     } 1015
990                     } 1016
991                     } 1017
992                     } 1018
993                     } 1019
994                     } 1020
995                     } 1021
996                     } 1022
997                     } 1023
998                     } 1024
999                     } 1025
1000                     } 1026
1001                     } 1027
1002                     } 1028
1003                     } 1029
1004                     } 1030
1005                     } 1031
1006                     } 1032
1007                     } 1033
1008                     } 1034
1009                     } 1035
1010                     } 1036
1011                     } 1037
1012                     } 1038
1013                     } 1039
1014                     } 1040
1015                     } 1041
1016                     } 1042
1017                     } 1043
1018                     } 1044
1019                     } 1045
1020                     } 1046
1021                     } 1047
1022                     } 1048
1023                     } 1049
1024                     } 1050
1025                     } 1051
1026                     } 1052
1027                     } 1053
1028                     } 1054
1029                     } 1055
1030                     } 1056
1031                     } 1057
1032                     } 1058
1033                     } 1059
1034                     } 1060
1035                     } 1061
1036                     } 1062
1037                     } 1063
1038                     } 1064
1039                     } 1065
1040                     } 1066
1041                     } 1067
1042                     } 1068
1043                     } 1069
1044                     } 1070
1045                     } 1071
1046                     } 1072
1047                     } 1073
1048                     } 1074
1049                     } 1075
1050                     } 1076
1051                     } 1077
1052                     } 1078
1053                     } 1079
1054                     } 1080
1055                     } 1081
1056                     } 1082
1057                     } 1083
1058                     } 1084
1059                     } 1085
1060                     } 1086
1061                     } 1087
1062                     } 1088
1063                     } 1089
1064                     } 1090
1065                     } 1091
1066                     } 1092
1067                     } 1093
1068                     } 1094
1069                     } 1095
1070                     } 1096
1071                     } 1097
1072                     } 1098
1073                     } 1099
1074                     } 1100
1075                     } 1101
1076                     } 1102
1077                     } 1103
1078                     } 1104
1079                     } 1105
1080                     } 1106
1081                     } 1107
1082                     } 1108
1083                     } 1109
1084                     } 1110
1085                     } 1111
1086                     } 1112
1087                     } 1113
1088                     } 1114
1089                     } 1115
1090                     } 1116
1091                     } 1117
1092                     } 1118
1093                     } 1119
1094                     } 1120
1095                     } 1121
1096                     } 1122
1097                     } 1123
1098                     } 1124
1099                     } 1125
1100                     } 1126
1101                     } 1127
1102                     } 1128
1103                     } 1129
1104                     } 1130
1105                     } 1131
1106                     } 1132
1107                     } 1133
1108                     } 1134
1109                     } 1135
1110                     } 1136
1111                     } 1137
1112                     } 1138
1113                     } 1139
1114                     } 1140
1115                     } 1141
1116                     } 1142
1117                     } 1143
1118                     } 1144
1119                     } 1145
1120                     } 1146
1121                     } 1147
1122                     } 1148
1123                     } 1149
1124                     } 1150
1125                     } 1151
1126                     } 1152
1127                     } 1153
1128                     } 1154
1129                     } 1155
1130                     } 1156
1131                     } 1157
1132                     } 1158
1133                     } 1159
1134                     } 1160
1135                     } 1161
1136                     } 1162
1137                     } 1163
1138                     } 1164
1139                     } 1165
1140                     } 1166
1141                     } 1167
1142                     } 1168
1143                     } 1169
1144                     } 1170
1145                     } 1171
1146                     } 1172
1147                     } 1173
1148                     } 1174
1149                     } 1175
1150                     } 1176
1151                     } 1177
1152                     } 1178
1153                     } 1179
1154                     } 1180
1155                     } 1181
1156                     } 1182
1157                     } 1183
1158                     } 1184
1159                     } 1185
1160                     } 1186
1161                     } 1187
1162                     } 1188
1163                     } 1189
1164                     } 1190
1165                     } 1191
1166                     } 1192
1167                     } 1193
1168                     } 1194
1169                     } 1195
1170                     } 1196
1171                     } 1197
1172                     } 1198
1173                     } 1199
1174                     } 1200
1175                     } 1201
1176                     } 1202
1177                     } 1203
1178                     } 1204
1179                     } 1205
1180                     } 1206
1181                     } 1207
1182                     } 1208
1183                     } 1209
1184                     } 1210
1185                     } 1211
1186                     } 1212
1187                     } 1213
1188                     } 1214
1189                     } 1215
1190                     } 1216
1191                     } 1217
1192                     } 1218
1193                     } 1219
1194                     } 1220
1195                     } 1221
1196                     } 1222
1197                     } 1223
1198                     } 1224
1199                     } 1225
1200                     } 1226
1201                     } 1227
1202                     } 1228
1203                     } 1229
1204                     } 1230
1205                     } 1231
1206                     } 1232
1207                     } 1233
1208                     } 1234
1209                     } 1235
1210                     } 1236
1211                     } 1237
1212                     } 1238
1213                     } 1239
1214                     } 1240
1215                     } 1241
1216                     } 1242
1217                     } 1243
1218                     } 1244
1219                     } 1245
1220                     } 1246
1221                     } 1247
1222                     } 1248
1223                     } 1249
1224                     } 1250
1225                     } 1251
1226                     } 1252
1227                     } 1253
1228                     } 1254
1229                     } 1255
1230                     } 1256
123
```

Code used: `get_m4_combination.js` — ideal membership problem (3/4)

```

114   && !s.startsWith('-')) s = '+' + s;
115   a-zA-Z0-9_{}\\|*)/g;
116   exec(s) !== null) {
117       const sign =
118       match[1] === '+' ? 1 : -1;
119       match[2];
120       if (term === ''
121       ) continue;
122       map[term] || 0) + sign;
123   }
124   return map;
125   }
126   function det3x3(M) {
127       const m = M.map(row =>
128           row.map(parseEntry));
129       const term1 = multiply(
130           m[0][0], subtract(multiply(m[1][1], m[2][2]), multiply(m[1][2],
131           m[2][1])));
132       const term2 = multiply(
133           m[0][1], subtract(multiply(m[1][0], m[2][2]), multiply(m[1][2],
134           m[2][0])));
135       const term3 = multiply(
136           m[0][2], subtract(multiply(m[1][0], m[2][1]), multiply(m[1][1],
137           m[2][0])));
138       return add(subtract(
139           term1, term2), term3);
140   }
141   const M4 = [
142       ["g_{11}", "-f_{12}", "-f_{13}"
143       ],
144       ["-\\bar{f}_1", "\\bar{g}_1",
145       "-\\bar{f}_2", "\\bar{g}_2",
146       "-\\bar{f}_3", "\\bar{g}_3",
147       "-\\bar{f}_4", "\\bar{g}_4",
148       "-\\bar{f}_5", "\\bar{g}_5",
149       "-\\bar{f}_6", "\\bar{g}_6",
150       "-\\bar{f}_7", "\\bar{g}_7",
151       "-\\bar{f}_8", "\\bar{g}_8",
152       "-\\bar{f}_9", "\\bar{g}_9",
153       "-\\bar{f}_{10}", "\\bar{g}_{10}",
154       "-\\bar{f}_{11}", "\\bar{g}_{11}",
155       "-\\bar{f}_{12}", "\\bar{g}_{12}",
156       "-\\bar{f}_{13}", "\\bar{g}_{13}"]
157   ];
158   const allMonomials = new Set();
159   vectors.forEach(v => Object.
160   keys(v).forEach(k => allMonomials.add(k)));
161   Object.keys(detM4).forEach(k =>
162   allMonomials.add(k));
163   const monomials = Array.from(
164   allMonomials).sort();
165   const matrix = vectors.map(v =>
166   monomials.map(m => v[m] || 0));
167   const targetM4 = monomials.map(
168   m => detM4[m] || 0);
169   function solve(matrix, target)
170   {
171       const m = matrix.length;
172       const n = matrix[0].
173       length;
174       const A = [];
175       for (let i = 0; i < n;
176       i++) {
177           A.push([]);
178           for (let j = 0; j < m;
179           j++) {
180               A[i].
181               push(matrix[j][i]);
182           }
183           A[i].push(
184               target[i]);
185       }
186       let row = 0;
187       const pivotCols = [];
188       for (let col = 0; col <

```

Code used: get_m4_combination.js — ideal membership problem (4/4)

```
171                                     201
172                                     202
173         row];                                     203
174         pivotRow];                               204
175         temp;                                     205
176                                     206
177         ({row, col});                             207
178                                     208
179         = A[row][col];                           209
180         col; j <= m; j++) A[row][j] /= pivotVal; 210
181                                     211
182         i < n; i++) {                             212
183                                     213
184         != row && Math.abs(A[i][col]) > 1e-9) {    214
185                                     215
186         const factor = A[i][col];                 216
187                                     217
188         for (let j = col; j <= m; j++) {           218
189                                     219
190         A[i][j] -= factor * A[row][j];             220
191                                     221
192         }                                           222
193         }                                           223
194                                     224
195         let consistent = true;                     225
196         for (let i = row; i < n                     226
197         ; i++) {
198         i][m]) > 1e-9) {
201         const temp = A[
202
203         A[row] = A[
204         A[pivotRow] =
205
206         pivotCols.push
207
208
209         const pivotVal
210
211         for (let j =
212
213         for (let i = 0;
214         if (i
215
216
217
218
219
220
221         row++;
222
223
224         let consistent = true;
225         for (let i = row; i < n
226         if (Math.abs(A[
201         const solution = new
202
203         Array(m).fill(0);
204
205         for (const p of
206         pivotCols) {
207         solution[p.col]
208         = A[p.row][m];
209         }
210         return solution;
211
212         }
213
214         const solM4 = solve(matrix,
215         targetM4);
216
217         function formatSolution(sol) {
218         if (!sol) return "No
219         linear combination found.";
220         const parts = [];
221         for (let i = 0; i < sol
222         .length; i++) {
223         if (Math.abs(
224         sol[i]) > 1e-9) {
225         let val
226         = Math.round(sol[i] * 10000) / 10000;
227         parts.
228         push(`${val} * ${vectorNames[i]}');
229         }
230         return parts.length > 0
231         ? parts.join('\n+ ') : "0";
232         }
233         console.log("det(M_4)
234         combination:");
235         console.log(formatSolution(
236         solM4));
```

Verification in symbols

Expand the right-hand side $\det(M_2) - (\bar{f}_{23} + \bar{g}_{23}) O(1 \rightarrow \theta_2 \theta_3)$.

$$\begin{aligned}\det(M_2) = & -\bar{f}_{12}\bar{f}_{23}f_{13} - \bar{f}_{12}\bar{g}_{33}f_{12} + \bar{f}_{13}\bar{f}_{23}f_{12} - \bar{f}_{13}\bar{g}_{22}f_{13} \\ & + \bar{f}_{13}\bar{g}_{23}f_{12} + \bar{f}_{23}^2g_{11} + \bar{f}_{23}\bar{g}_{23}g_{11} + \bar{g}_{22}\bar{g}_{33}g_{11},\end{aligned}$$

$$\begin{aligned}(\bar{f}_{23} + \bar{g}_{23}) O(1 \rightarrow \theta_2 \theta_3) = & -\bar{f}_{12}\bar{f}_{23}f_{13} - \bar{f}_{12}\bar{g}_{23}f_{13} + \bar{f}_{13}\bar{f}_{23}f_{12} + \bar{f}_{13}\bar{g}_{23}f_{12} \\ & + \bar{f}_{23}^2g_{11} + \bar{f}_{23}\bar{g}_{23}g_{11} + \bar{f}_{23}f_{12}g_{13} - \bar{f}_{23}f_{13}g_{12} \\ & + \bar{f}_{23}f_{23}g_{11} + \bar{f}_{23}g_{11}g_{23} + \bar{g}_{23}f_{12}g_{13} - \bar{g}_{23}f_{13}g_{12} \\ & + \bar{g}_{23}f_{23}g_{11} + \bar{g}_{23}g_{11}g_{23}.\end{aligned}$$

Subtracting cancels eight terms and reproduces the twelve terms of $\Delta(\theta_2 \theta_3)$ exactly. □

Summary

Summary

- Our problem is nothing but the ideal membership problem which is accessible by pre-existing tools. Nevertheless, thanks to AI tools, computer illiterates such as myself might be able to call on the aid of computers much more easily.
- In this project major breakthroughs were due to HI(Human Intelligences): the role of AI was to deal with complications which take too much time for humans. We wondered how much the AI could suggest theoretical breakthroughs towards our solution, and at this moment it seems much more probable that it can do quite a lot.

Thank you very much!