Module-intersection Integration-by-parts (IBP) method

Yang Zhang

Johannes Gutenberg University of Mainz
ETH Zurich

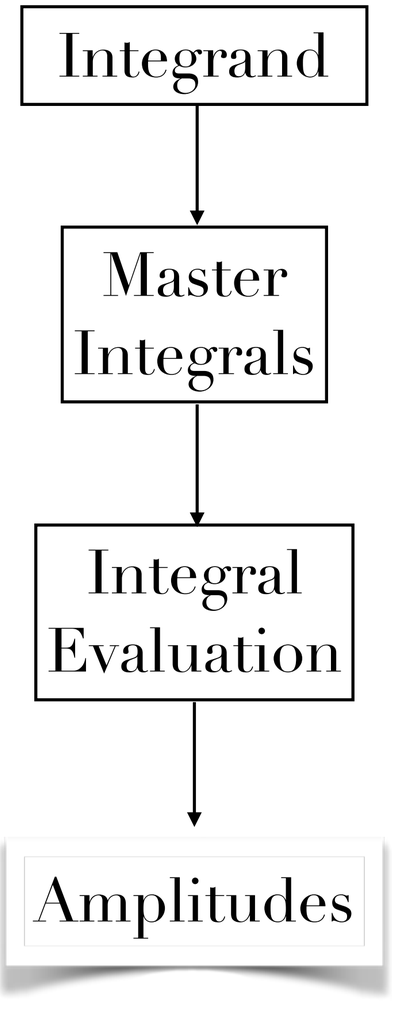
1805.01873

with

Janko Boehm, Hans Schoenemann (Kaiserslautern)
Alessandro Georgoudis (Uppsala), Kasper Larsen (Southampton)

Humboldt university, Berlin
May 31, 2018

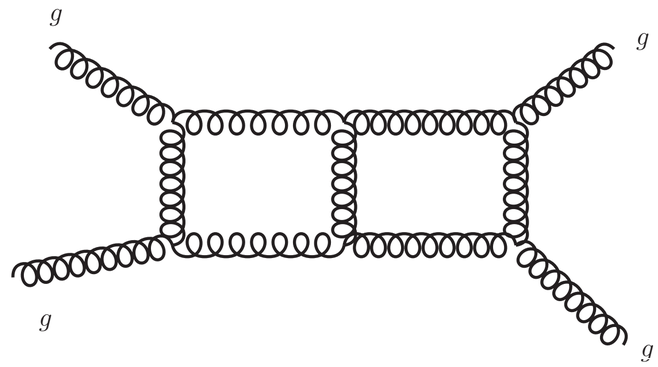
Integration-by-Parts (IBP) reduction



Feynman Rules,
Unitarity of Quantum theory

Integration-by-parts (IBP)

Numeric: Mellin-Barnes, secdec …
Analytic: Differential Equation,
Dimensional recursion



Multi-loop Feynman integral
for high-precision scattering amplitudes

D: space-time dimension, usually as a free parameter
instead of D=4 (Dimensional Regularization)

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \ldots D_k^{\alpha_k}}, \quad \alpha_i \in \mathbb{Z}$$

Integration-by-parts reduction can be a bottleneck
of high energy physics computations

Integration-by-Parts (IBP) reduction

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \ldots D_k^{\alpha_k}}, \quad \alpha_i \in \mathbb{Z}$$

$$\boxed{\int \frac{dl_1^D}{i\pi^{D/2}} \cdots \int \frac{dl_L^D}{i\pi^{D/2}} \frac{\partial}{\partial l_i^\mu} \left(\frac{v_i^\mu}{D_1 \ldots D_k} \right) = 0}$$

IBP: Chetyrkin, Tkachov 1981
Laporta 2001

IBP → could remove "most" integrals for multi-loop cases

→ master integrals (proven to be a finite set)

IBP reduction for multi-loop integrals with many parameters can be difficult

two-loop 2 to 3 scattering …

state-of-art IBP programs

FIRE (Smirnov), Reduze (von Manteuffel, Studerus), LiteRed (Lee)
Kira (Maierhofer, Usovitsch, Uwer)

Problems with multi-loop IBP reduction

Large linear system

Many parameters in IBPs

Problems with multi-loop IBP reduction

Large linear system

trim linear system dramatically
by unitarity cuts and no-double-propagator condition

Many parameters in IBPs

localization trick
in monomial ordering

Integration-by-Parts (IBP) reduction integral class selection

Gluza, Kajda, Kosower 1009.0472

Roman Lee, 1405.5616

Smaller linear system, fewer integrals

I. Integrals without "doubled propagators" , (Baikov rep.)

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \ldots D_m^{\alpha_m} D_{m+1}^{\alpha_{m+1}} \ldots D_k^{\alpha_k}}, \quad \begin{cases} \alpha_i \leq 1, & 1 \leq i \leq m \\ \alpha_i \leq 0, & m < i \leq k \end{cases}$$

Small subset of integrals

II. Integrals without "numerators", (parametric rep.)

Bitoun, Bogner, Klausen, Panzer 1712.09215

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \ldots D_m^{\alpha_m} D_{m+1}^{\alpha_{m+1}} \ldots D_k^{\alpha_k}}, \quad \begin{cases} \alpha_i \geq 0, & 1 \leq i \leq m \\ \alpha_i = 0, & m < i \leq k \end{cases}$$

Small subset of integrals

New IBP reduction:
Only use a chosen smaller integral class
Hopefully to get a much smaller linear system

a subset of IBPs satisfying algebraic constraints

Computational Algebraic Geometry

Baikov representation without "doubled propagators"

Ita 1510.05626

K. Larsen and YZ, 1511.01071

When $k = LE + L(L+1)/2$, (E is the number of independent legs), the Baikov rep. is

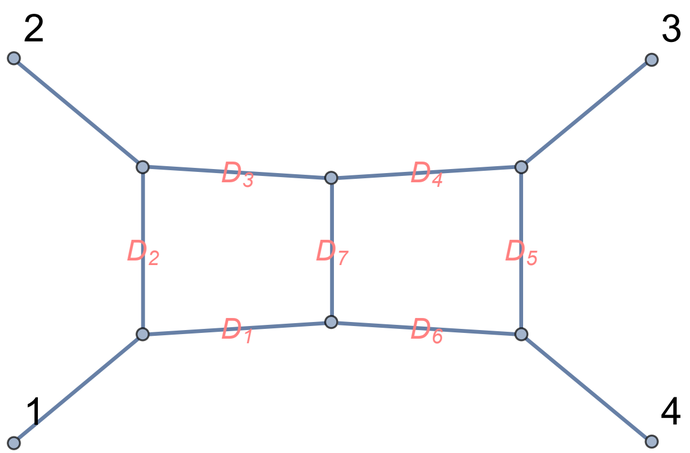
$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \ldots D_k^{\alpha_k}} \propto \prod_{1\le i\le L+E, \max\{i,E+1\}\le j\le L+E} \left(\int dx_{ij} \right) \det(S)^{\frac{D-L-E-1}{2}} \frac{1}{D_1^{\alpha_1} \ldots D_k^{\alpha_k}}$$

$$\propto \prod_{i=1}^{k} \left(\int dz_i \right) \det(S)^{\frac{D-L-E-1}{2}} \frac{1}{z_1^{\alpha_1} \ldots z_k^{\alpha_k}} \qquad z_i \equiv D_i$$

Baikov representation

where $\{v_1, \ldots v_{L+E}\} \equiv \{k_1, \ldots k_E, l_1, \ldots l_L\}$ and $x_{ij} \equiv v_i \cdot v_j$. S is a $(L+E)\times(L+E)$ matrix with $S_{ij} = x_{ij}$ (Gram matrix).

$L = 2, E = 3, m = 7$ and $k = 9$. $\{v_1, \ldots, v_5\} \equiv \{k_1, k_2, k_4, l_1, l_2\}$.



$$S = \begin{pmatrix} 0 & \frac{s}{2} & \frac{t}{2} & x_{11} & x_{21} \\ \frac{s}{2} & 0 & \frac{1}{2}(-s-t) & x_{12} & x_{22} \\ \frac{t}{2} & \frac{1}{2}(-s-t) & 0 & x_{13} & x_{23} \\ x_{11} & x_{12} & x_{13} & x_{44} & x_{45} \\ x_{21} & x_{22} & x_{23} & x_{45} & x_{55} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \frac{s}{2} & \frac{t}{2} & \frac{z_1}{2} - \frac{z_2}{2} & \frac{z_9}{2} - \frac{z_6}{2} \\ \frac{s}{2} & 0 & \frac{1}{2}(-s-t) & \frac{1}{2}(s - z_3) + \frac{z_2}{2} & \frac{1}{2}(z_4 - s) - \frac{z_9}{2} \\ \frac{t}{2} & \frac{1}{2}(-s-t) & 0 & \frac{z_8}{2} - \frac{z_1}{2} & \frac{z_6}{2} - \frac{z_5}{2} \\ \frac{z_1}{2} - \frac{z_2}{2} & \frac{1}{2}(s - z_3) + \frac{z_2}{2} & \frac{z_8}{2} - \frac{z_1}{2} & z_1 & -\frac{z_1}{2} - \frac{z_6}{2} + \frac{z_7}{2} \\ \frac{z_9}{2} - \frac{z_6}{2} & \frac{1}{2}(z_4 - s) - \frac{z_9}{2} & \frac{z_6}{2} - \frac{z_5}{2} & -\frac{z_1}{2} - \frac{z_6}{2} + \frac{z_7}{2} & z_6 \end{pmatrix}$$

Baikov representation without "doubled propagators"

$$R = \mathbb{Q}(\text{parameters})[z_1, \ldots z_k]$$

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \ldots D_m^{\alpha_m} D_{m+1}^{\alpha_{m+1}} \ldots D_k^{\alpha_k}}, \quad \begin{cases} \alpha_i \leq 1, & 1 \leq i \leq m \\ \alpha_i \leq 0, & m < i \leq k \end{cases}$$

Just consider IBPs

$$0 = \left(\prod_{i=1}^{k} \int dz_i \right) \sum_{j=1}^{k} \frac{\partial}{\partial z_j} \left(a_j(z) \det(S)^{\frac{D-L-E-1}{2}} \frac{1}{z_1 \ldots z_m} \right)$$

Polynomials!

Further require $F \equiv \det(S)$

"Affine varieties and Lie algebras of vector fields"
Hauser, Müller 1993

1. no shifted exponent: $\sum_{j=1}^{k} a_j(z) \frac{\partial F}{\partial z_j} + \beta(z) F = 0$ These $\big(a_1(z), \ldots a_k(z)\big)$ form a module $M_1 \subset R^k$.

2. no doubled propagator: $a_i(z) \in \langle z_i \rangle, \quad 1 \leq i \leq m$ These $\big(a_1(z), \ldots a_k(z)\big)$ form a module $M_2 \subset R^k$.

Both M_1 and M_2 are pretty simple ...

Y.Z. 1612.02249 $\boxed{M_1 \cap M_2}$ Intersection of two modules

Computation of the first module M_1

In principle, the syzygy can be computed by Schreyer's theorem, but the computation is heavy …

$$\sum_{j=1}^{k} a_j(z)\frac{\partial F}{\partial z_j} + \beta(z)F = 0$$

- syzygy for the $\{\frac{\partial F}{\partial z_1}, \ldots, \frac{\partial F}{\partial z_k}, F\}$
- $\mathrm{Ann}(F^s)$, annihilator of F^s in Weyl algebra.

Computation of the first module M_1

In principle, the syzygy can be computed by Schreyer's theorem, but the computation is heavy …

$$\sum_{j=1}^{k} a_j(z)\frac{\partial F}{\partial z_j} + \beta(z)F = 0$$

- syzygy for the $\{\frac{\partial F}{\partial z_1}, \ldots, \frac{\partial F}{\partial z_k}, F\}$
- $\mathrm{Ann}(F^s)$, annihilator of F^s in Weyl algebra.

If F is a determinant matrix whose elements are free variables, this kind of syzygy module is simple.

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}$$

Roman Lee's idea

No computation is needed

syzygy generators (Laplace expansion)

$$\sum_j a_{k,j}\frac{\partial(\det A)}{\partial a_{i,j}} - \delta_{k,i}\cdot \det A = 0$$ provides all syzygy generators

The Gram matrix in Baikov rep. is symmetric and not all entries are free variables …
"weighted" Laplace expansion

Computation of the first module M_1

1712.09737 Boehm, Georgoudis, Larsen, Schulze, YZ

By Gulliksen–Negard and Jozefiak exact sequences

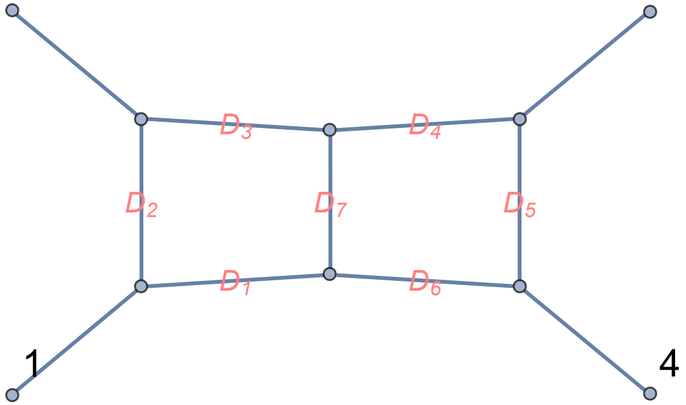
$$
\begin{aligned}
\mathrm{Sl}_n(\mathscr{O}_X)^{\oplus 2} &\longrightarrow \mathrm{Mat}_n(\mathscr{O}_X) \longrightarrow I_{n-1}(M) \longrightarrow 0\,,\\
(A,B) &\longmapsto MA - BM\,,\\
C &\longmapsto \mathrm{tr}(M^\star C)\,,\\
\mathrm{Sl}_n(\mathscr{O}_{X'}) &\longrightarrow \mathrm{Sym}_n(\mathscr{O}_{X'}) \longrightarrow I_{n-1}(S) \longrightarrow 0\,,\\
A &\longmapsto SA + A^t S\,,\\
D &\longmapsto \mathrm{tr}(S^\star D)\,,
\end{aligned}
$$

The syzygies found from Laplace expansion are complete.

Example, massless double box

$\mathbb{Q}(s,t)[z_1,\ldots z_9]$: 2 parameters, 9 variables (Each row is a module generator)

$$M_1 = \begin{pmatrix}
z_1 - z_2 & z_1 - z_2 & -s + z_1 - z_2 & 0 & 0 & 0 & z_1 - z_2 - z_6 + z_9 & t + z_1 - z_2 & 0 \\
0 & 0 & 0 & s - z_6 + z_9 & -t - z_6 + z_9 & -z_6 + z_9 & z_1 - z_2 - z_6 + z_9 & 0 & -z_6 + z_9 \\
s + z_2 - z_3 & z_2 - z_3 & z_2 - z_3 & 0 & 0 & 0 & z_2 - z_3 + z_4 - z_9 & -t + z_2 - z_3 & 0 \\
0 & 0 & 0 & z_4 - z_9 & t + z_4 - z_9 & -s + z_4 - z_9 & z_2 - z_3 + z_4 - z_9 & 0 & z_4 - z_9 \\
-z_1 + z_8 & -t - z_1 + z_8 & s - z_1 + z_8 & 0 & 0 & 0 & -z_1 - z_5 + z_6 + z_8 & -z_1 + z_8 & 0 \\
0 & 0 & 0 & -s - z_5 + z_6 & -z_5 + z_6 & -z_5 + z_6 & -z_1 - z_5 + z_6 + z_8 & 0 & t - z_5 + z_6 \\
2\,z_1 & z_1 + z_2 & -s + z_1 + z_3 & 0 & 0 & 0 & z_1 - z_6 + z_7 & z_1 + z_8 & 0 \\
0 & 0 & 0 & s - z_3 - z_6 + z_7 & -z_6 + z_7 - z_8 & -z_1 - z_6 + z_7 & z_1 - z_6 + z_7 & 0 & -z_2 - z_6 + z_7 \\
-z_1 - z_6 + z_7 & -z_1 + z_7 - z_9 & s - z_1 - z_4 + z_7 & 0 & 0 & 0 & -z_1 + z_6 + z_7 & -z_1 - z_5 + z_7 & 0 \\
0 & 0 & 0 & -s + z_4 + z_6 & z_5 + z_6 & 2\,z_6 & -z_1 + z_6 + z_7 & 0 & z_6 + z_9
\end{pmatrix}$$



$$M_2 = \begin{pmatrix}
z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & z_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & z_4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & z_5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & z_6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & z_7 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$M_1 \cap M_2$ (even without cut) is computed in ~4 seconds, with Singular 4.1.0,

intersect(M1,M2,"std")

Bonus, master integrals from Baikov representation

Package Azurite, **A ZUR**ich-bred **InTE**gral-determination method

Georgoudis, Larsen, YZ

1612.04252

Lee Pomeransky, 1308.6676

Bitoun, Bogner, Klausen, Panzer 1712.09215

Consider the maximal cuts of inequivalent sector in Baikov representation

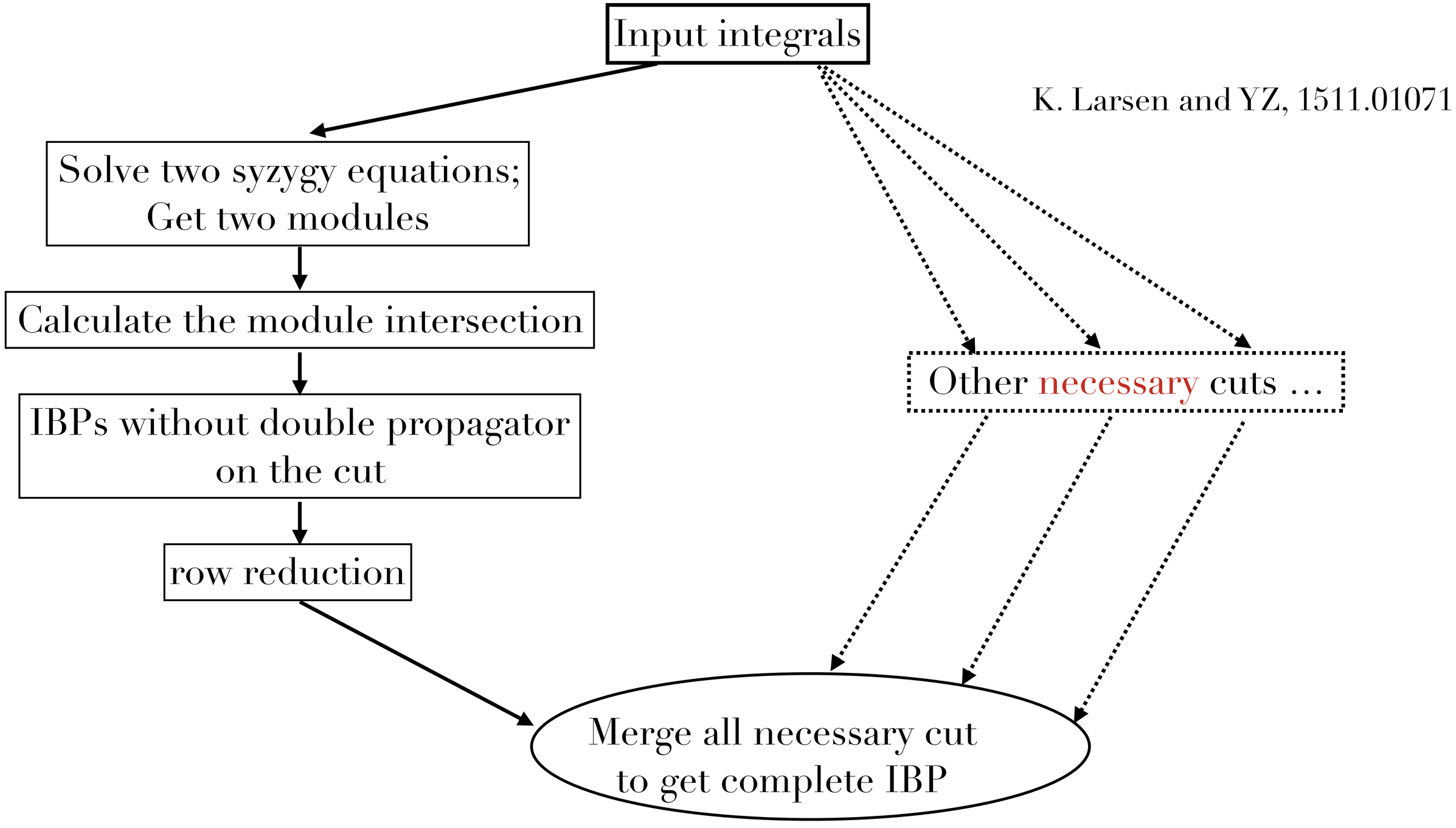
Find the syzygies and reduce max-cut IBPs over a finite field

	Syzygy algorithm	Reduction program
Azurite 1.x.x	Schreyer	Mathematica
Azurite 2.x.x	Laplace expansion	SpaSM

For three loop examples tested, Azurite 2.x.x is 5~10 times faster than Azurite 1.x.x

to appear 2018

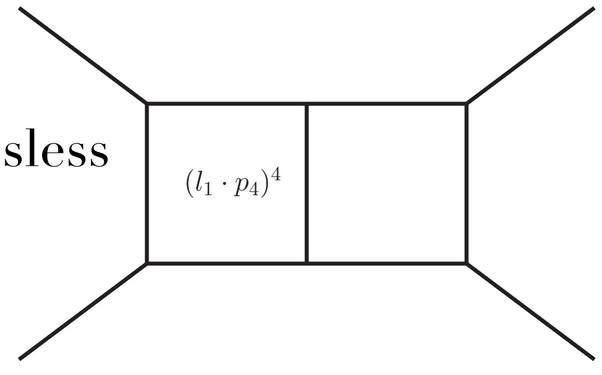
Complete IBP from cut construction



Example, massless double box with cut

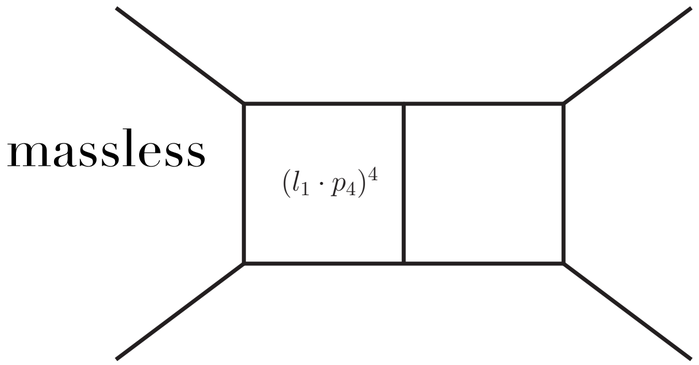
code powered by
Mathematica/Macaulay2/Singular

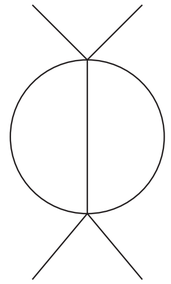
massless



Example, massless double box with cut

code powered by
Mathematica/Macaulay2/Singular

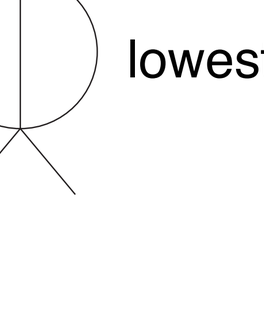




lowest MI

lowest MI

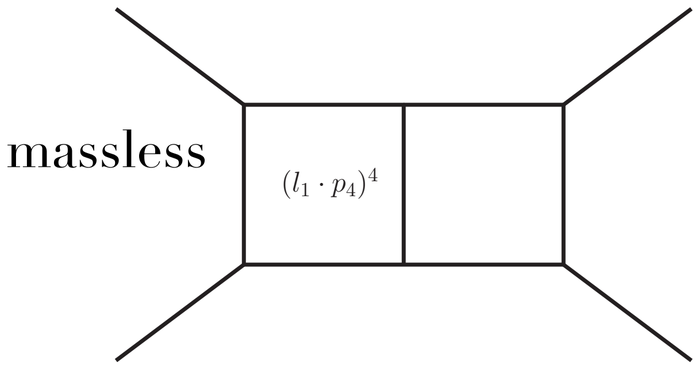
lowest MI

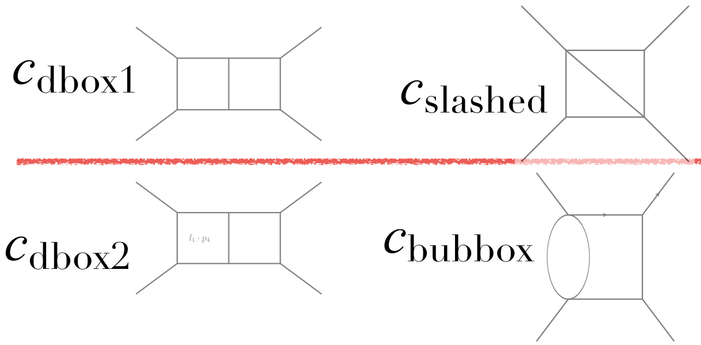


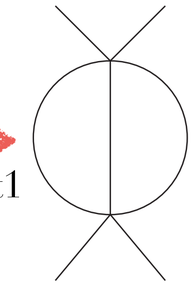
lowest MI

Example, massless double box with cut

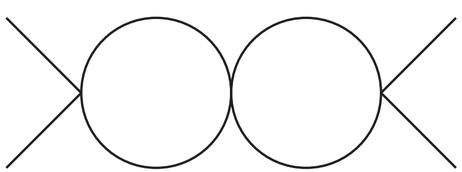
code powered by
Mathematica/Macaulay2/Singular



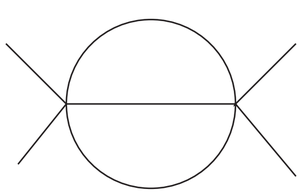




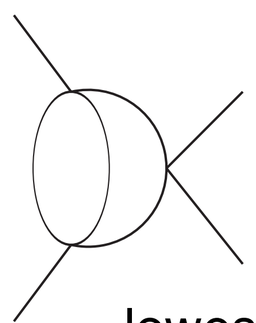
lowest MI



lowest MI



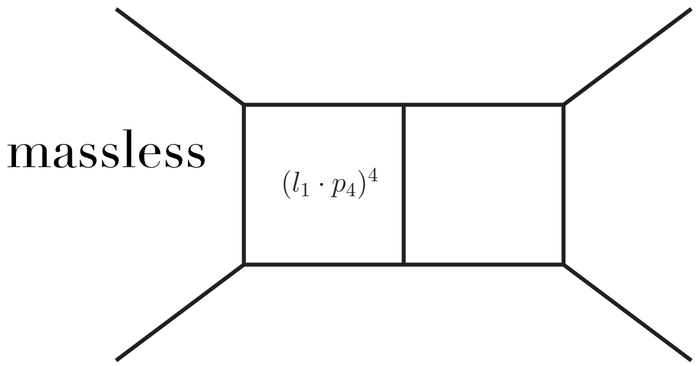
lowest MI

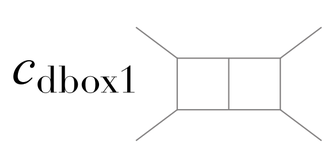


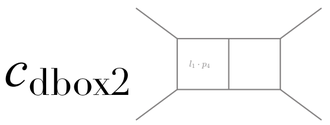
lowest MI

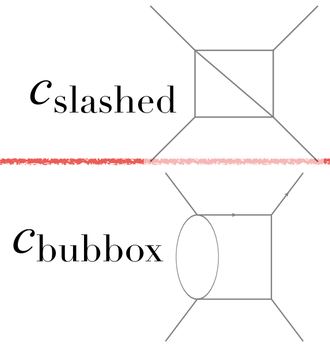
Example, massless double box with cut

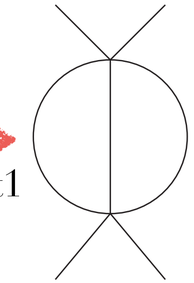
code powered by
Mathematica/Macaulay2/Singular



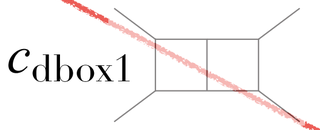


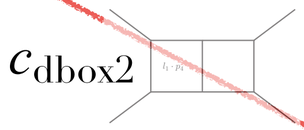


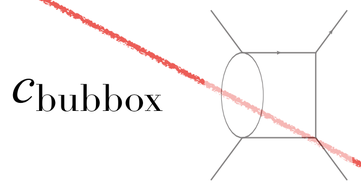


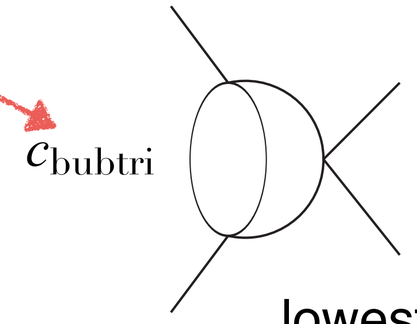


lowest MI

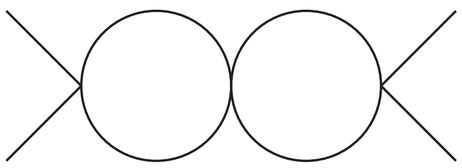




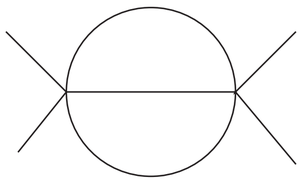




lowest MI



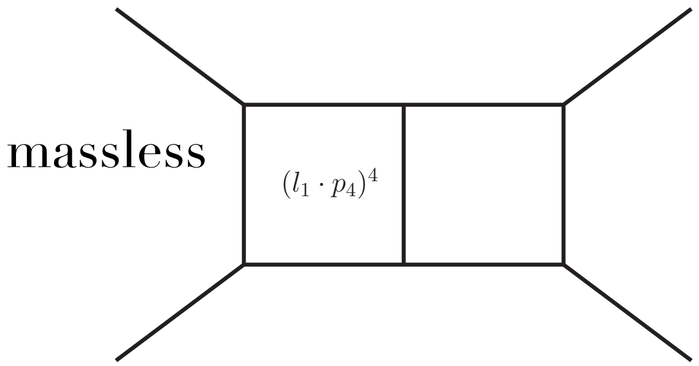
lowest MI

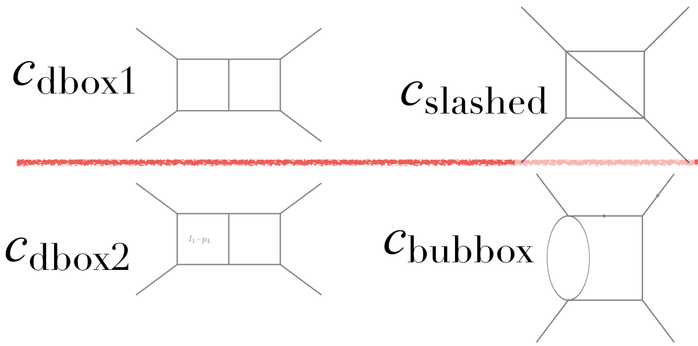


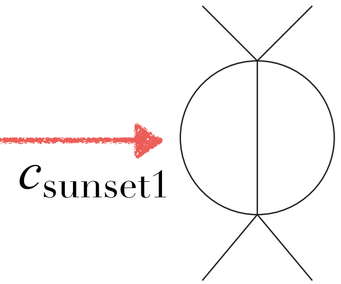
lowest MI

Example, massless double box with cut

code powered by
Mathematica/Macaulay2/Singular

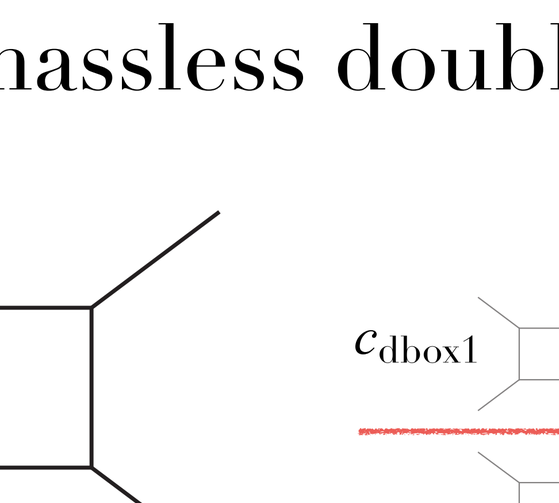


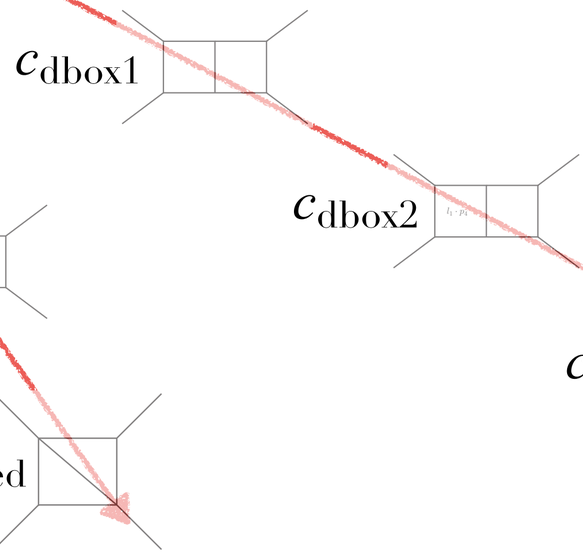


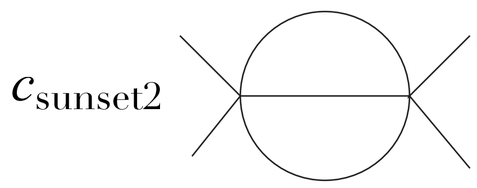


lowest MI

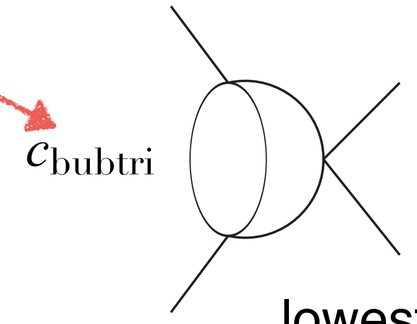
lowest MI







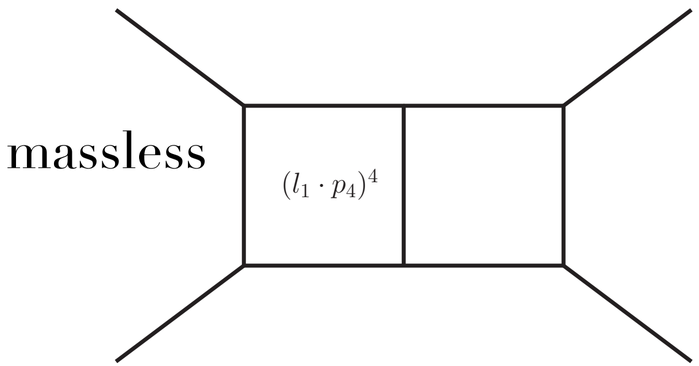
lowest MI

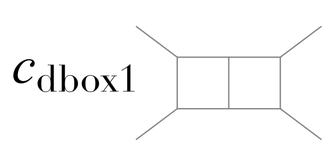


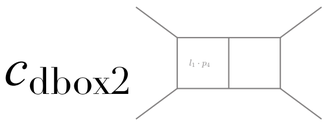
lowest MI

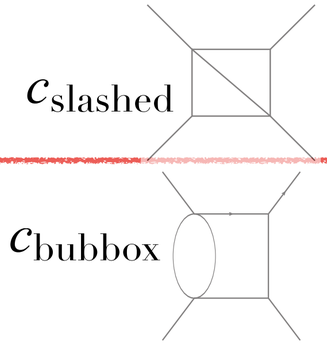
Example, massless double box with cut

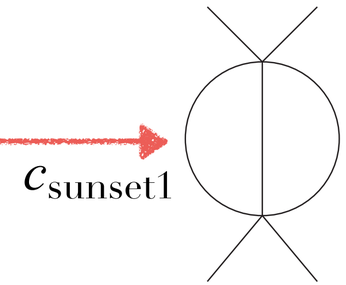
code powered by
Mathematica/Macaulay2/Singular



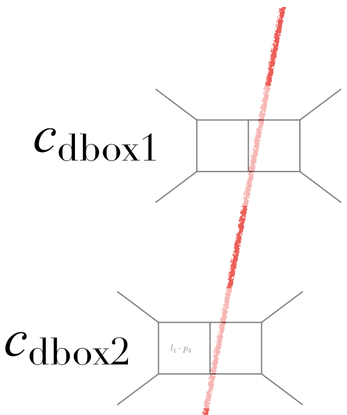


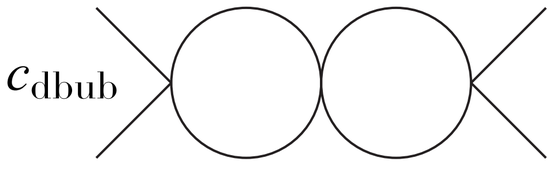




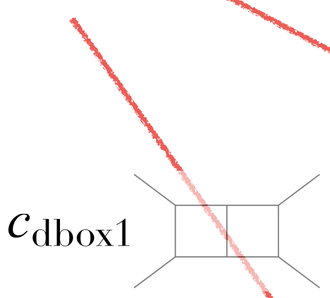


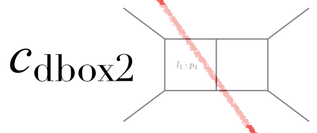
lowest MI

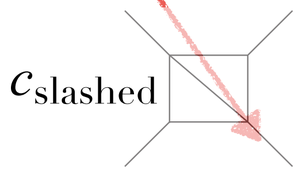


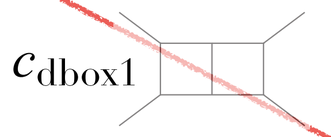


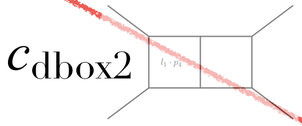
lowest MI

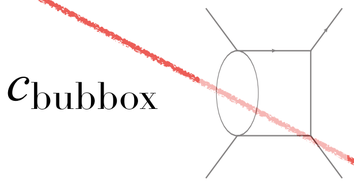


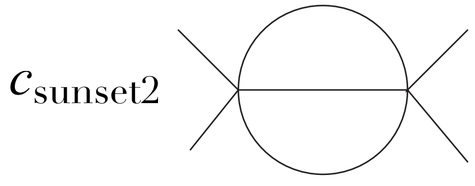




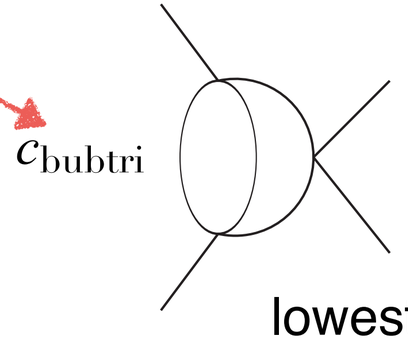








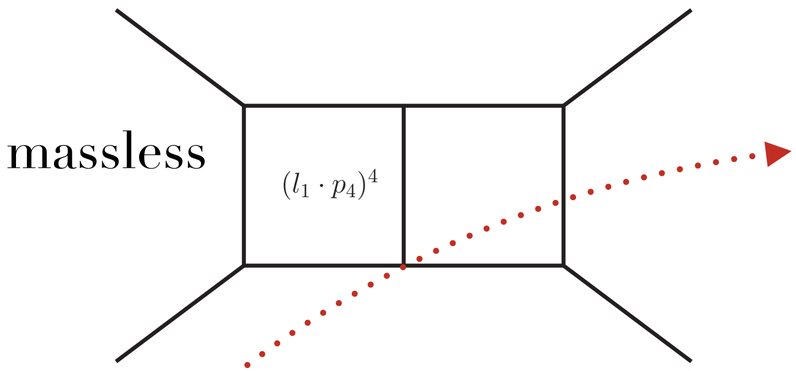
lowest MI

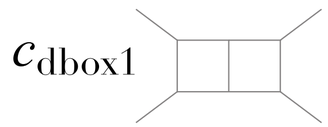


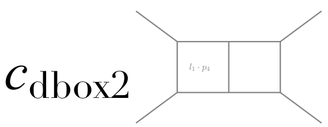
lowest MI

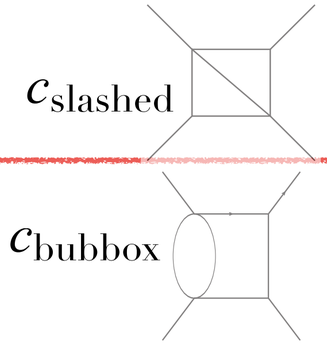
Example, massless double box with cut

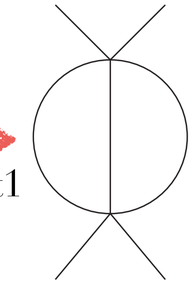
code powered by
Mathematica/Macaulay2/Singular



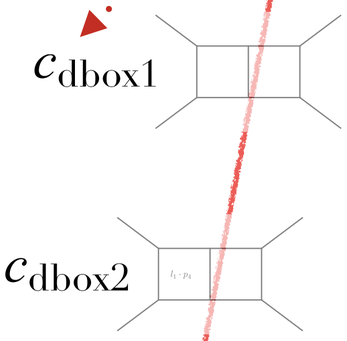


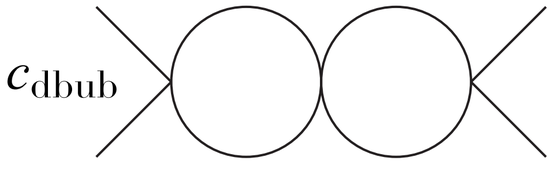




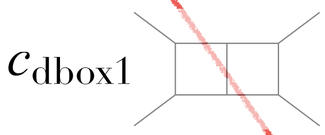


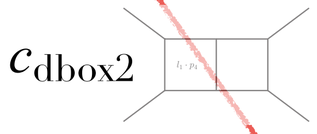
lowest MI

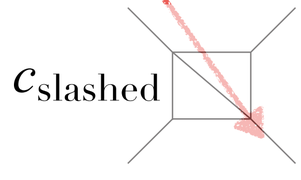


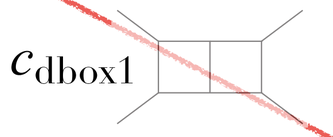


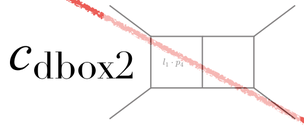
lowest MI

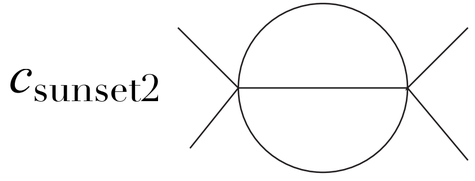




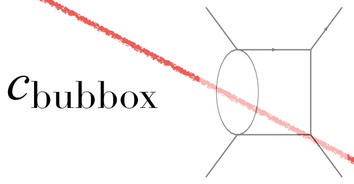


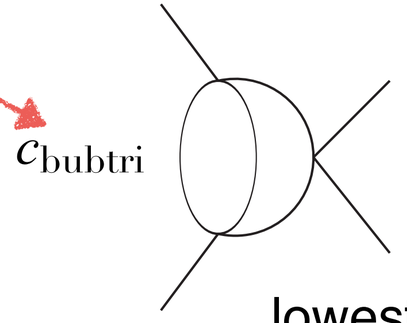






lowest MI





lowest MI

Non-trivial example

Nonplanar hexagon-box

(3,6,7) cut

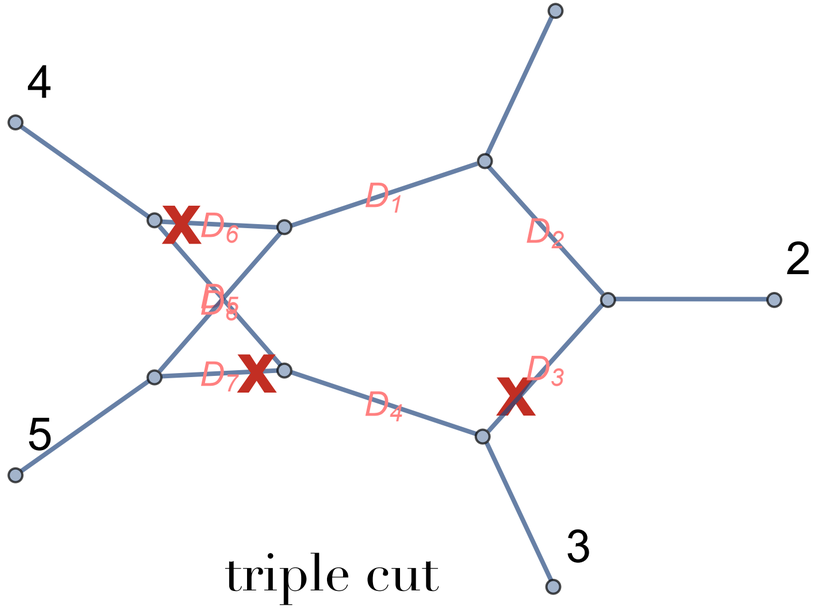
10 different cuts needed (8 triple cuts, 2 quadruple cuts)

$\mathbb{Q}(s_{12}, s_{13}, s_{14}, s_{23}, s_{24})[z_1, z_2, z_4, z_5, z_8, z_9, z_{10}, z_{11}]$: 5 parameters, 11-3=8 variables

$z_3 \to 0, z_6 \to 0, z_7 \to 0$

$M_1' =$

$$\begin{pmatrix}
z_1 - z_2 & z_1 - z_2 & -s12 + z_1 - z_2 & -s12 - s13 + z_1 - z_2 & s14 + z_1 - z_2 - z_8 + z_{10} & z_1 - z_2 - z_8 + z_{10} & 0 & 0 & -s12 - s13 - s14 + z_1 - z_2 & 0 & 0 \\
0 & 0 & 0 & 0 & s14 + z_1 - z_2 - z_8 + z_{10} & z_1 - z_2 - z_8 + z_{10} & s12 + s13 + s14 - z_8 + z_{10} & -z_8 + z_{10} & 0 & -z_8 + z_{10} & s12 - z_8 + z_{10} \\
s12 + z_2 & z_2 & z_2 & -s23 + z_2 & s12 + s24 + z_2 - z_8 + z_{11} & s12 + z_2 - z_8 + z_{11} & 0 & 0 & -s23 - s24 + z_2 & 0 & 0 \\
0 & 0 & 0 & 0 & s12 + s24 + z_2 - z_8 + z_{11} & s12 + z_2 - z_8 + z_{11} & s12 + s23 + s24 - z_8 + z_{11} & -z_8 + z_{11} & 0 & s12 - z_8 + z_{11} & -z_8 + z_{11} \\
s13 + s23 - z_4 & s23 - z_4 & -z_4 & -z_4 & -2\,s12 - s13 - s14 - s23 - s24 - z_5 + z_8 - z_9 - z_{10} - z_{11} & -s12 - z_5 + z_8 - z_9 - z_{10} - z_{11} & 0 & 0 & s12 + s13 + s14 + s23 + s24 - z_4 & 0 & 0 \\
0 & 0 & 0 & 0 & -2\,s12 - s13 - s14 - s23 - s24 - z_5 + z_8 - z_9 - z_{10} - z_{11} & -s12 - z_5 + z_8 - z_9 - z_{10} - z_{11} & -2\,s12 - s13 - s14 - s23 - s24 + z_4 - z_5 + z_8 - z_9 - z_{10} - z_{11} & -s12 - s13 - s23 + z_4 - z_5 + z_8 - z_9 - z_{10} - z_{11} & 0 & -s12 - s13 - s14 - s23 - s24 + z_4 - z_5 + z_8 - z_9 - z_{10} - z_{11} & -s12 - s13 + z_4 - z_5 + z_8 - z_9 - z_{10} - z_{11} \\
-s12 - s13 - s23 + z_4 - z_9 & -s12 - s13 - s14 - s23 + z_4 - z_9 & -s12 - s13 - s14 - s23 - s24 + z_4 - z_9 & z_4 - z_9 & z_5 & z_5 & 0 & 0 & z_4 - z_9 & 0 & 0 \\
0 & 0 & 0 & 0 & z_5 & z_5 & -z_4 + z_5 + z_9 & s12 + s13 + s23 - z_4 + z_5 + z_9 & 0 & z_4 + z_9 & s12 + s13 + s23 - z_4 + z_5 + z_9 \\
2\,z_1 & z_1 + z_2 & -s12 + z_1 & -s12 - s13 - s23 + z_1 + z_4 & -s12 - s13 - s23 + z_1 + z_4 - z_8 - z_9 & z_1 - z_8 & -z_8 - z_9 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -s12 - s13 - s23 + z_1 + z_4 - z_8 - z_9 & z_1 - z_8 & 0 & -z_1 - z_8 & 0 & -z_1 & s12 - z_1 + z_2 - z_8 \\
-z_1 - z_8 & -z_1 - z_{10} & -z_1 + z_8 - z_{10} - z_{11} & s12 + s13 + s23 - z_1 - z_4 + z_5 + z_8 + z_9 & s12 + s13 + s23 - z_1 - z_4 + z_5 + z_8 + z_9 & -z_1 + z_8 & z_8 & 2\,z_8 & 0 & z_8 + z_{10} & z_8 + z_{11} \\
0 & 0 & 0 & s12 + s13 + s23 - z_1 - z_4 + z_5 + z_9 & 0 & -z_1 + z_8 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$



triple cut

$M_2' =$

$$\begin{pmatrix}
z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & z_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & z_5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & z_8 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

$M_1' \cap M_2' = ?$

Numerically (and over finite field) the intersection can be found in several minutes

Analytically the intersection seems very difficult

Localization trick for computational algebraic geometry

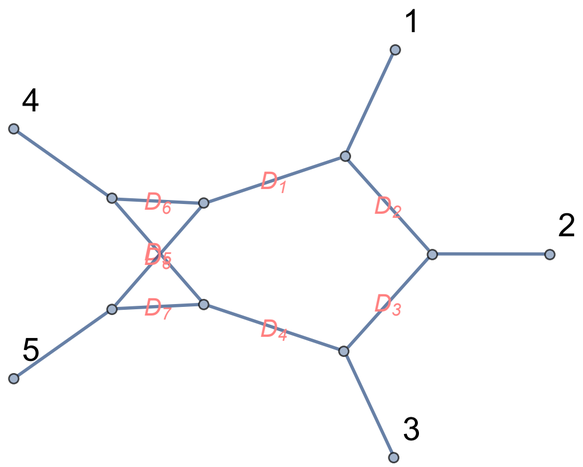
Boehm, Schoeneman, University of Kaiserslautern

Analytic Mandelstam variables (parameters) slow down the computation

> Localization trick:
> Treat parameters as variables, and compute in a particular monomial ordering

[variables] > [parameters]

this trick makes all polynomials homogeneous



Module intersection for (1,4,6,7)

$\mathbb{Q}(s_{12}, s_{13}, s_{14}, s_{23}, s_{24})[z_2, z_3, z_5, z_8, z_9, z_{10}, z_{11}]$ with
$z_2 > z_3 > z_5 > z_8 > z_9 > z_{10} > z_{11}$

$\mathbb{Q}[z_2, z_3, z_5, z_8, z_9, z_{10}, z_{11}, s_{12}, s_{13}, s_{14}, s_{23}, s_{24}]$ with
$[z_2, z_3, z_5, z_8, z_9, z_{10}, z_{11}] > [s_{12}, s_{13}, s_{14}, s_{23}, s_{24}]$

? (does not finish)

minutes with Singular

It is not a good idea to take $s_{12} \to 1$!

All module intersections for 10 necessary cuts found analytically

Module intersections to get algebraic constraint

Hexagon-box

efficient implement in Singular by Boehm

cuts	timing (sec)	RAM (GB)	generator size (raw, MB)	generator size (trimmed, MB)
1,5,7	218	4.3	68	10
2,5,7	43	1.1	25	1.4
2,5,8	303	6.7	49	3.1
2,6,7	743	9.8	100	2.8
3,5,8	404	7.4	97	3.7
3,6,7	699	11	80	3.6
3,6,8	24	1	10	1.6
4,6,8	797	13.7	21	1.6
1,4,5,8	53	1.7	4.4	3.6
1,4,6,7	196	3.0	9.4	4.1

Heuristic generator trimming algorithm, Y.Z.

trimmed IBPs

Hexagon-box, all rank-4 numerators

cuts	# IBPs	# Integrals	Bytes (MB)	Density
1,5,7	1144	1177	1.2	1.4%
2,5,7	1170	1210	0.99	1.3%
2,5,8	1152	1190	1.1	1.5%
2,6,7	1118	1155	1.0	1.5%
3,5,8	1160	1202	1.2	1.5%
3,6,7	1173	1217	1.3	1.7%
3,6,8	1135	1176	0.77	1.2%
4,6,8	1140	1176	0.94	1.2%
1,4,5,8	700	723	0.69	1.7%
1,4,6,7	683	706	0.66	1.6%

Heuristic linear trimming algorithm, Y.Z.

- remove linearly dependent IBPs
- remove IBPs irrelevant to targets
- prefer IBPs with small byte counting

powered by SpaSM package (Charles Bouillaguet)

Linear Reduction

efficient linear algebra code under development …

(photo from my hometown, Hefei, China)

Linear Reduction

Hexagon-box, all rank-4 numerators

Preliminary Mathematica code, YZ

- Sparse RREF (row reduced echelon form) with weighted Markowitz pivoting strategy
- Finite field numeric zero guessing
- Integral basis change
- Interpolation if necessary: heuristic multivariate rational function interpolation algorithm

More efficient implement in Singular by Bendle and Boehm
to be available

Linear Reduction

Hexagon-box, all rank-4 numerators

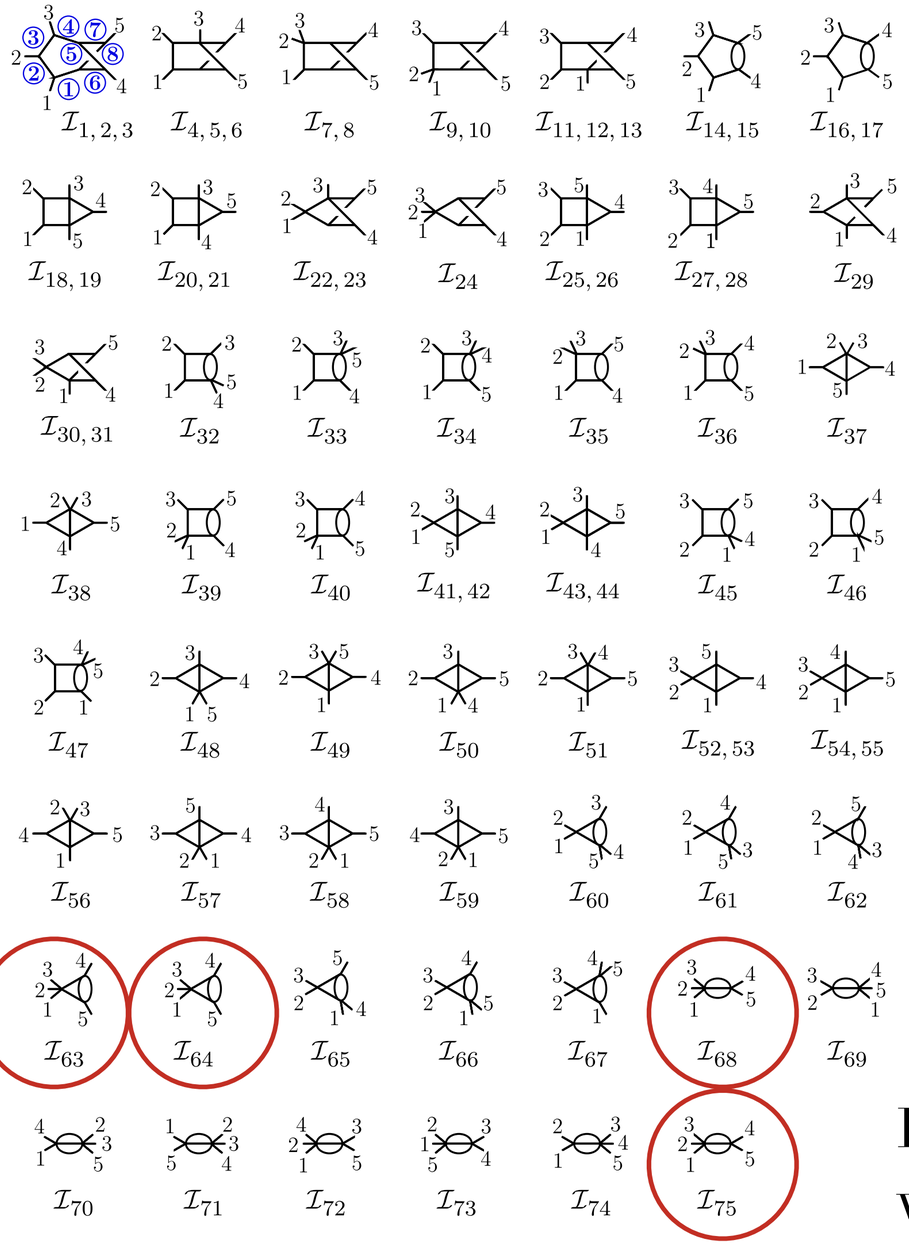
cuts	# IBPs	# Integrals
1,5,7	1144	1177
2,5,7	1170	1210
2,5,8	1152	1190
2,6,7	1118	1155
3,5,8	1160	1202
3,6,7	1173	1217
3,6,8	1135	1176
4,6,8	1140	1176
1,4,5,8	700	723
1,4,6,7	683	706

Preliminary Mathematica code

bivariate interpolation
RREF: 2.5 hours, one core, 1.8 GB RAM (440 times)
Interpolation: 23 minutes, one core, 15 GB RAM

31 minutes, one core, 1.5 GB RAM

Merge all the cuts



Reduce to

75 “pre”-master integrals

↓

73 master integrals

Remove integrals related by global symmetries with Azurite

Towards an “industry”-level implement of this IBP algorithm

- more flexible choice of cuts
- adaptive kinematic variable choice for specific cuts
- more efficient parallelization strategy
- automatic selection of the pivot strategy
- all codes rewritten in Singular/C++

Dominik Bendle, Janko Boehm, Alessandro Georgoudis, YZ,
and
more students/postdocs from TU Kaiserslautern and MPI Munich

Summary

- Using unitarity cuts, syzygies and module intersection method to decrease the number of integrals dramatically in IBP systems
- “Localization trick” is surprisingly powerful for module intersection computation with multiple parameters
- An efficient way of Gaussian(-Bareiss) elimination of the small linear system of IBPs needed

towards an automatic program

of complicated IBP reductions for multi-loop LHC processes