New version of NeatIBP

A novel tool for Feynman integral computation

Zihao Wu

Based on:
NeatIBP 1.1: ZW, Janko Boehm, Rourou Ma, Johann Usovitsch, Yingxuan Xu, Yang Zhang, arXiv: 2502.20778
NeatIBP 1.0: ZW, Janko Boehm, Rourou Ma, Hefeng Xu, Yang Zhang, Comput.Phys.Commun. 295 (2024) 108999

What is NeatIBP for?

Computing fixed-order amplitudes for scattering processes remains a key obstacle to producing precise predictions for the LHC and HL-LHC. (…) we divide the computation of multi-loop amplitudes into two broad categories:1. (…) 2. Calculating the integrals which appear in the amplitudes.

…

The use of integration-by-parts (IBP) identities (…) remains a critically important technique in modern loop calculations, but also presents a major bottleneck.

…

*The **NeatIBP** tool uses syzygy and module intersection techniques to provide IBP systems in which the propagator degrees are limited.*

Les Houches 2023 - Physics at TeV Colliders:
Report on the Standard Model Precision Wishlist

Alexander Huss[1], Joey Huston[2], Stephen Jones[3], Mathieu Pellen[4], Raoul Röntsch[5]

[1] *Theoretical Physics Department, CERN, 1211 Geneva 23, Switzerland*

[2] *Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA*

[3] *Institute for Particle Physics Phenomenology, Durham University, Durham DH1 3LE, United Kingdom*

[4] *Albert-Ludwigs-Universität Freiburg, Physikalisches Institut, Hermann-Herder-Straße 3, D-79104 Freiburg, Germany*

[5] *University of Milan and INFN Milan, Via Celoria 20133, Milan, Italy*

Multi-loop Feynman integrals in particle physics

Multi-loop Feynman integrals

↓

Multi-loop Feynman diagrams

→

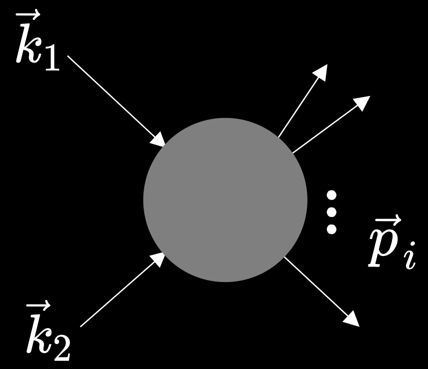
Action of the theory in QFT

↓

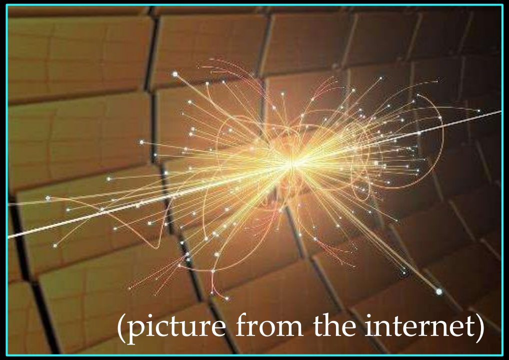
High-order scattering amplitudes

↓

High-precision experiments



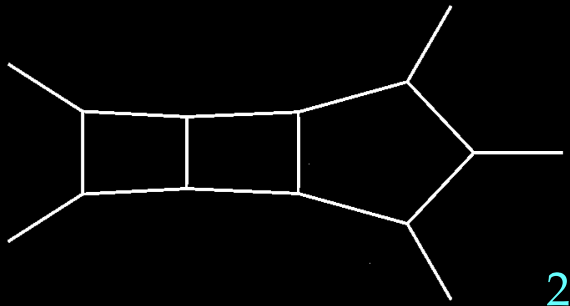
$$_{\text{out}}\langle \vec{p}_1 \vec{p}_2 \cdots | \vec{k}_1 \vec{k}_2 \rangle_{\text{in}}.$$



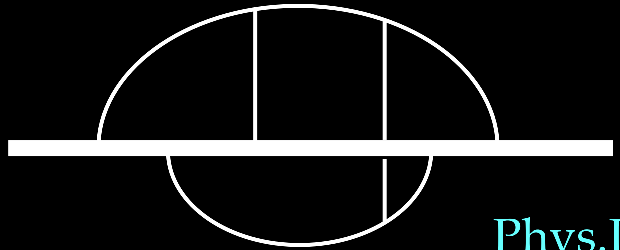
(picture from the internet)

Multi-loop Feynman integrals, frontiers

More loops:

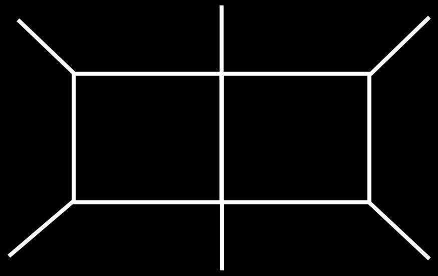


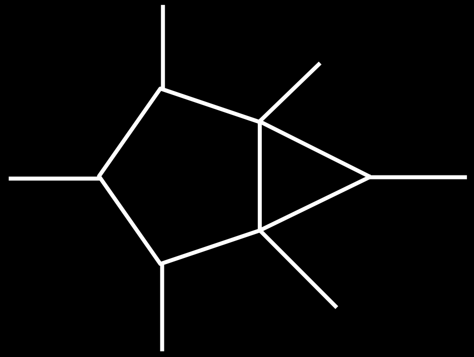
2411.18697

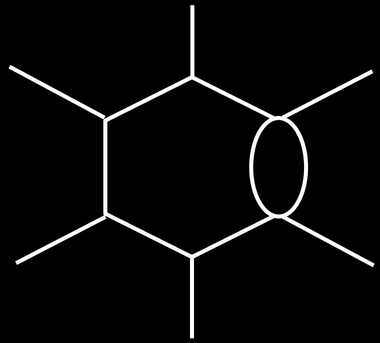


Phys.Rev.D 109 (2024) 7, L071503

More legs:

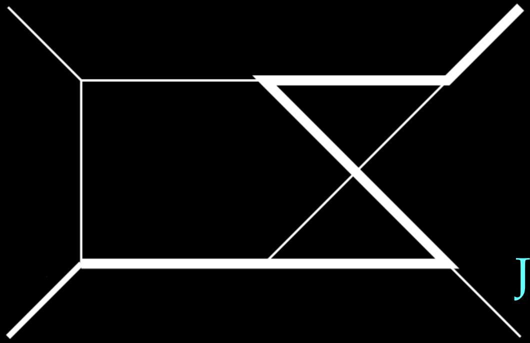




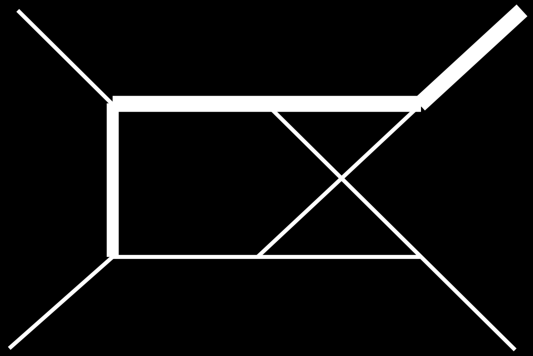


JHEP 08 (2024) 027

More masses:



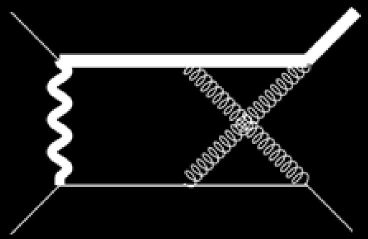
JHEP 07 (2023) 089



JHEP 06 (2023) 144

Workflow of analytic computation of Feynman integrals

Feynman diagrams



↓ Feynman rules

Feynman integrals

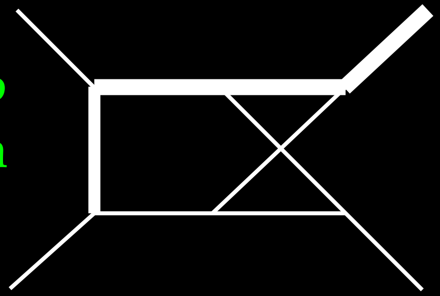
$$\int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{\mathcal{N}}{D_1^{\delta_1} \cdots D_m^{\delta_m}}.$$

↓ Tensor reduction

Scalar integrals

$$\int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

Scalar integrals keep only the information of propagators



↓ Integration-by-parts (IBP) reduction

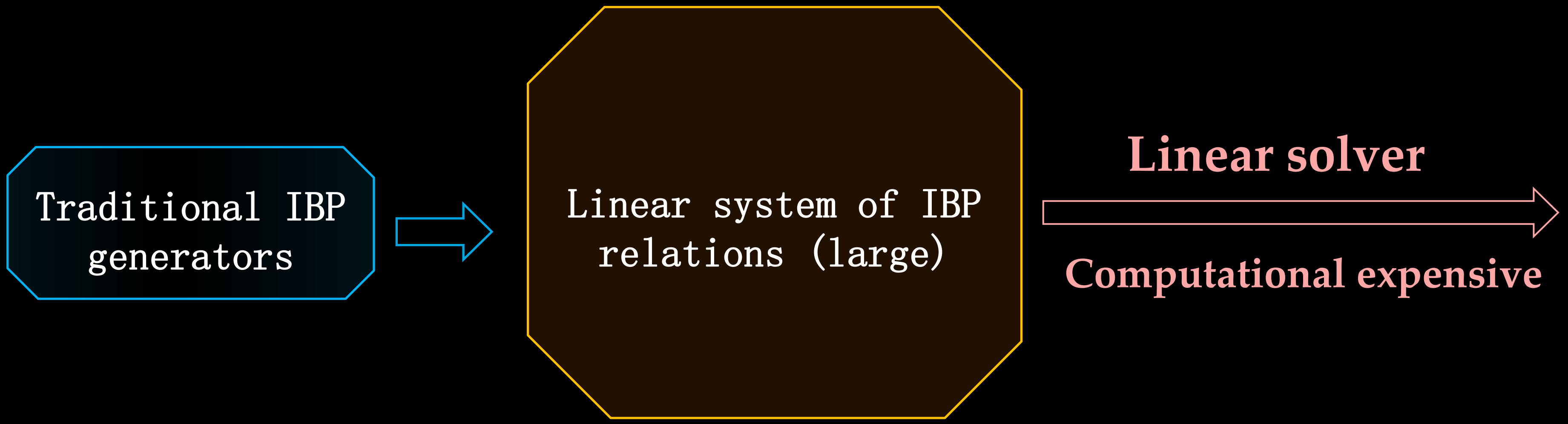
Master Integrals

A linear independent basis of the scalar integrals

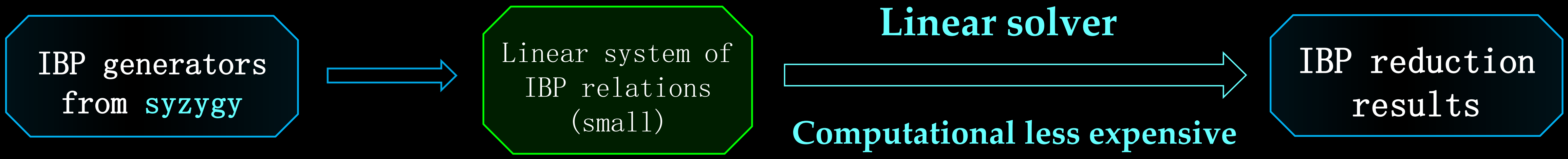
→ Analytic evaluation methods

Analytic functions

Traditional IBP algorithm

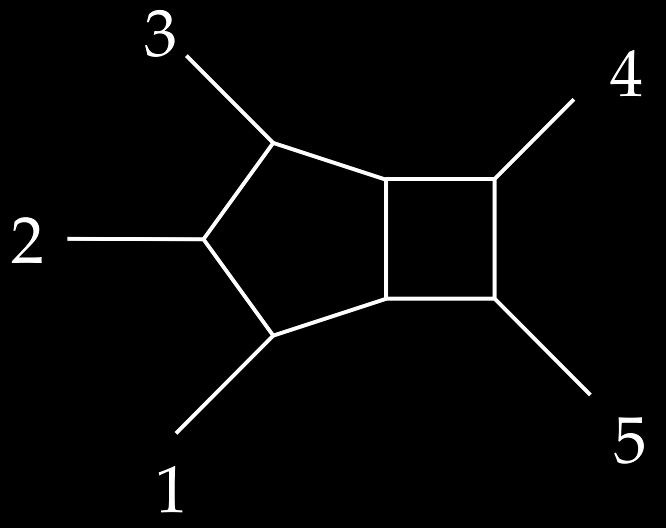


NeatIBP:



Performance of NeatIBP

2L5P Example



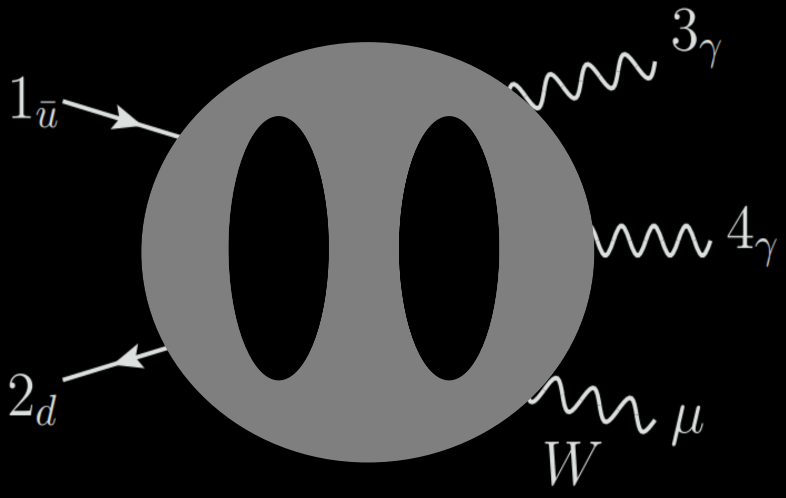
Target integrals:
Max numerator degree: 5
Max denominator power: 1
Quantity: 2483

# of IBP (FIRE6): **11207942**

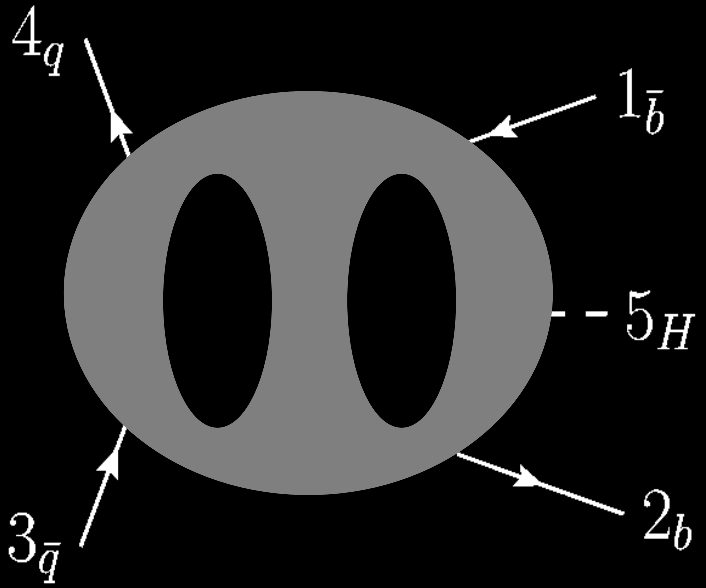
# of IBP (NeatIBP): **14120**

Time used: 27m at  CPU threads: 20
RAM: 128GB

Application of NeatIBP:
Two-loop-five-point amplitudes

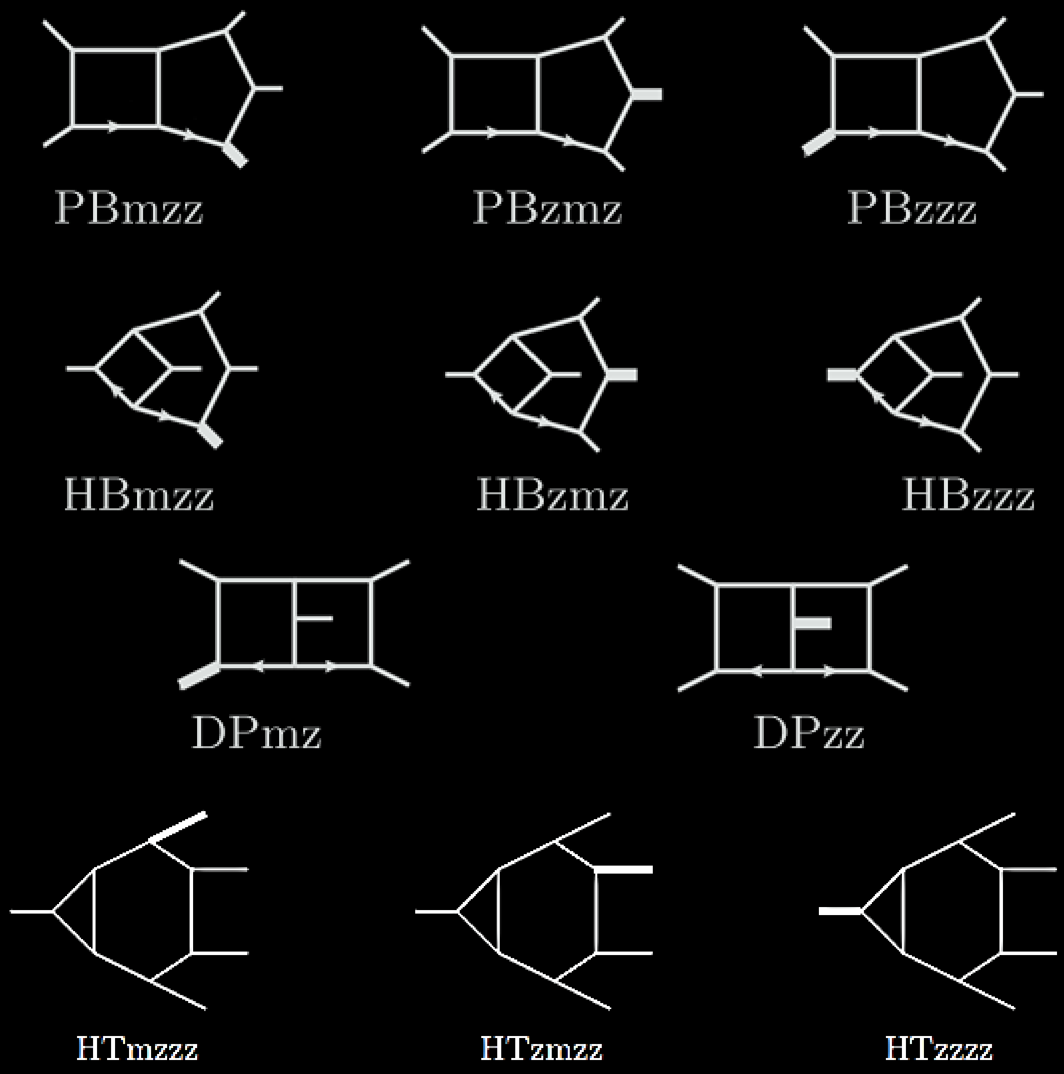


Simon Badger, Heribertus Bayu Hartanto, ZW, Yang Zhang and Simone Zoia, *JHEP* 03 (2025) 066



Simon Badger, Heribertus Bayu Hartanto, Rene Poncelet, ZW, Yang Zhang and Simone Zoia, *JHEP* 12 (2025) 221

Relevant Feynman diagrams



NeatIBP generated IBP relations within hours for each diagram

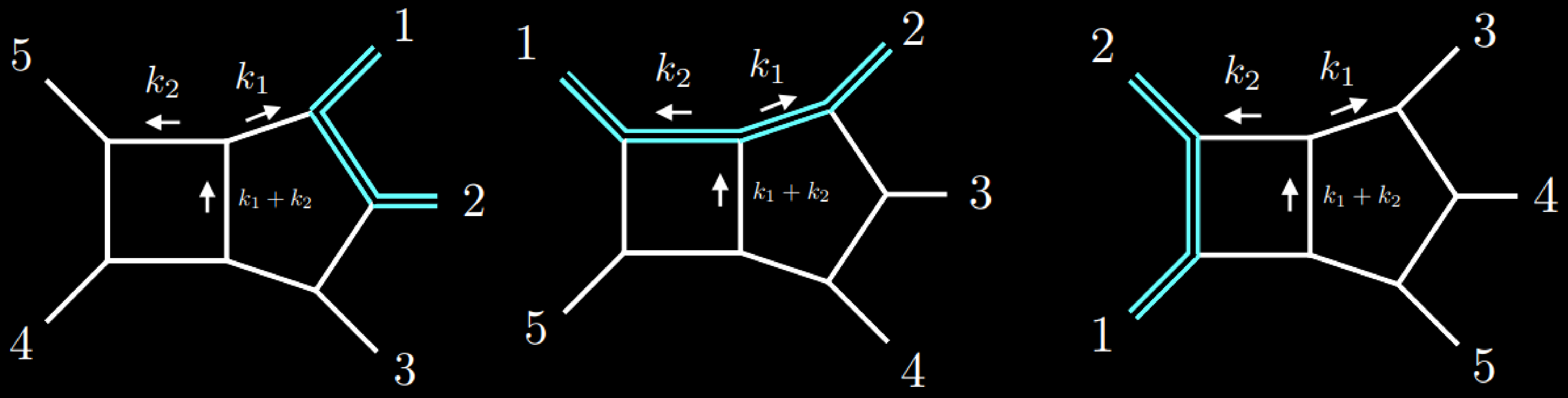
JHEP 03 (2025) 066:

"This (using NeatIBP) improves the evaluation of the solution to the IBP relations, resulting in both a speed up in the finite-field sampling of the rational coefficients of the amplitudes and in a reduction of its memory footprint (by around 8 times and 3 times, respectively, for the leading colour two-loop five-particle amplitudes)."

Application of NeatIBP:

Two-loop-five-point amplitudes for $p\,p \to t\,\bar{t}\,j$ at leading color

Simon Badger, Matteo Becchetti, Nicolò Giraudo, Simone Zoia, JHEP 07 (2024) 073



(diagrams from JHEP 07 (2024) 073)

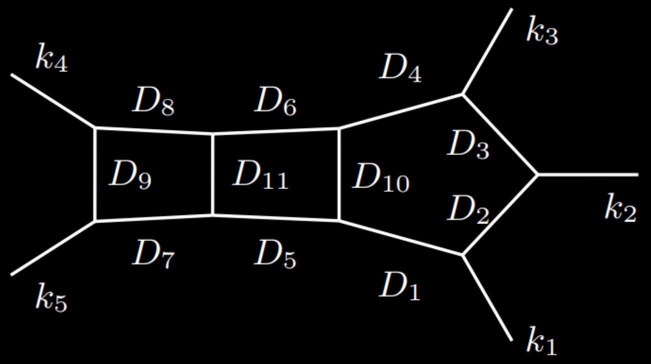
"The latter package (NeatIBP), in particular, allows us to obtain optimized systems of IBP relations through the solution of syzygy equations, this way making their solution substantially simpler."

Application of NeatIBP:
Three-loop-five-point diagrams

Yuanche Liu, Antonela Matijašić, Julian Miczajka, Yingxuan Xu, Yongqun Xu, Yang Zhang, arXiv:2411.18697

Analytic evaluation of the master integrals for the Pentagon-box-box diagram via differential equations

Target integrals: integrals needed for DE of master integrals



(diagram from arXiv:2411.18697)

"On a laptop, NeatIBP finds about 85000 IBP identities in full kinematics dependence within several hours. These relations are sufficient to derive the differential equation. As a comparison, standard IBP tools would generate three orders of magnitude more IBP relations, which make the reduction computation impossible."

Applications of NeatIBP:

Studies that used NeatIBP	Reference	Date
Two-loop QCD helicity amplitudes for $g\, g \to g\, t\, \bar{t}$ at leading color	*JHEP* 03 (2025) 070	2025.3.11
Full-color double-virtual amplitudes for $q\, \bar{q} \to b\, \bar{b}\, H$	*JHEP* 03 (2025) 066	2025.3.11
Three-loop five-point pentagon-box-box Feynman diagram	arXiv: 2411.18697	2024.11.27
Two-loop QCD corrections for $p\, p \to t\, \bar{t}\, j$	arXiv: 2411.10856	2024.11.16
Two-loop amplitudes for $W\, \gamma\, \gamma$ production at LHC	*JHEP* 12 (2025) 221	2024.12.30
NLO corrections to $J/\Psi\; c\, \bar{c}$ photoproduction	*Phys.Rev.D* 110 (2024) 9, 094047	2024.11.11
Two-loop five-point two-mass planar integrals	*JHEP* 10 (2024) 167	2024.10.23
Two-loop integrals for $t\, \bar{t}\, j$ production at hadron colliders in the leading color approximation	*JHEP* 07 (2024) 073	2024.7.9

You may also try NeatIBP in your projects

https://github.com/yzhphy/NeatIBP

Contents

Algorithms and features in NeatIBP 1.0

Main new features in the NeatIBP 1.1

2025.02

1. The Kira+NeatIBP interface

2. The spanning cuts method

3. Algorithm of syzygy vector simplification

Integration by parts reduction

Feynman integral family

$$I[\alpha_1, \cdots, \alpha_n] = \int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

IBP relations

$$0 = \int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{\partial}{\partial l_k^\mu} \frac{v^\mu}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}} \qquad 0 = \sum_{\{\alpha_i\}} c^i_{\alpha_1 \cdots \alpha_n} I[\alpha_1, \cdots, \alpha_n]$$

IBP reduction

Linear system of IBP relations

Gaussian Elimination

$$I[\alpha_1, \cdots, \alpha_n] = \sum_{i=1}^{N} c_i I_i$$

Master integrals

Coefficients

IBP relations in with multiple propagators

Target integrals

$G[1,1,1,\cdots,1,-1,-4]$

$G[1,1,1,\cdots,1,-5,0]$

$\ldots$

IBP-Related integrals

$G[1,1,0,\cdots,1,-1,-2]$ $G[0,1,1,\cdots,2,0,-1]$

$G[2,1,0,\cdots,1,-1,-1]$ $G[0,1,1,\cdots,0,0,-2]$

$G[1,1,0,\cdots,1,-3,-1]$ $G[1,2,0,\cdots,1,-3,0]$

$\ldots$

Master integrals

$G[1,1,1,\cdots,1,-1,0]$

$G[1,1,1,\cdots,1,0,0]$

$\ldots$

Contains **redundant integrals** with denominator indices lifted

$$0=\int\frac{\mathrm{d}^Dl_1}{i\pi^{D/2}}\cdots\frac{\mathrm{d}^Dl_L}{i\pi^{D/2}}\frac{\partial}{\partial l_k^\mu}\frac{v^\mu}{D_1^{\alpha_1}\cdots D_n^{\alpha_n}}=\int\frac{\mathrm{d}^Dl_1}{i\pi^{D/2}}\cdots\frac{\mathrm{d}^Dl_L}{i\pi^{D/2}}\frac{\frac{\partial v^\mu}{\partial l_k^\mu}-v^\mu\sum_{i=1}^n\frac{\partial D_i}{\partial l_k^\mu}\frac{\alpha_k}{D_i}}{D_1^{\alpha_1}\cdots D_n^{\alpha_n}}$$

Introducing multiple propagators

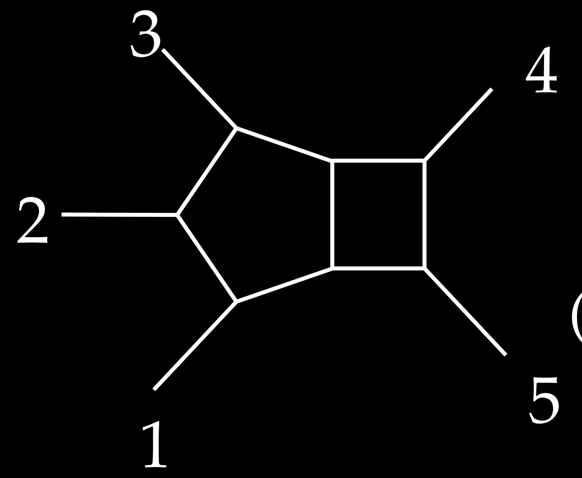
IBP relations from syzygy method

$$0 = \int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{\partial}{\partial l_k^\mu} \frac{v^\mu}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}} = \int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{\frac{\partial v^\mu}{\partial l_k^\mu} - v^\mu \sum_{i=1}^{n} \frac{\partial D_i}{\partial l_k^\mu} \frac{\alpha_k}{D_i}}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

Introducing multiple propagators

Janusz Gluza, Krzysztof Kajda, David A. Kosower, Phys.Rev.D 83 (2011) 045012

What if we find a good combination of v such that the additional Di cancels?



(top sector)

Syzygy equations

Indeed decreases the number of equations a lot

NeatIBP: syzygy method in Baikov representation

Feynman integrals in momentum space:

$$I[\alpha_1, \cdots, \alpha_n] = \int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

⇩ Variable transformation

Baikov representation: integrates directly over propagators z_i

$$I[\alpha_1, \cdots, \alpha_n] = C \int \mathrm{d}z_1 \cdots \mathrm{d}z_n P(z)^\alpha \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}}$$

IBP relations in Baikov representation

$$0 = \int \mathrm{d}z_1 \cdots \mathrm{d}z_n \sum_{i=1}^{n} \frac{\partial}{\partial z_i} \left(a_i(z) P^\alpha \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}} \right)$$

$$= \int \mathrm{d}z_1 \cdots \mathrm{d}z_n \left(\sum_{i=1}^{n} \frac{\partial a_i}{\partial z_i} P^\alpha + \sum_{i=1}^{n} \alpha a_i \boxed{\frac{\partial P}{\partial z_i} P^{\alpha-1}} - P^\alpha \sum_{i=1}^{n} \alpha_i \frac{a_i}{\boxed{z_i}} \right) \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}}$$

dimension shift

multiple propagators

$$\boxed{\left(\sum_{i=1}^{n} a_i(z) \frac{\partial P}{\partial z_i} \right) + b(z) P = 0}$$

$$\boxed{a_i(z) = b_i(z) z_i \quad \text{for } i \in \{j | \alpha_j > 0\}}$$

$$0 = \int \mathrm{d}z_1 \cdots \mathrm{d}z_n \left(\sum_{i=1}^{n} \left(\frac{\partial a_i}{\partial z_i} - \alpha_i b_i \right) + \alpha b \right) P^\alpha \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}}$$

Without multiple propagators

Polynomial equations in Baikov IBP relations

$$\left(\sum_{i=1}^{n} a_i(z)\frac{\partial P}{\partial z_i}\right) + b(z)P = 0$$

$$a_i(z) = b_i(z)z_i \quad \text{for } i \in \{j|\alpha_j > 0\}$$

Syzygy Equations

SINGULAR

Generators of the solution module

$$\begin{pmatrix} a_i \\ b \end{pmatrix} \in M \;\; =< f_1, f_2, \cdots >$$

Syzygy generator

$$\begin{pmatrix} a_i \\ b \end{pmatrix} \in M$$

Symbolic IBP relation

$$0 = \int \mathrm{d}z_1 \cdots \mathrm{d}z_n \left(\sum_{i=1}^{n} \left(\frac{\partial a_i}{\partial z_i} - \alpha_i b_i \right) + \alpha b \right) P^{\alpha} \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}}$$

Seeding

$$\vec{\alpha} \to (1, \cdots, 1, -2, -3)$$
$$\vec{\alpha} \to (1, \cdots, 1, 0, -5)$$
$$\vdots$$

Specific IBP relation

(# generators) × (# seeds) = (# IBP relations)

IBP relation selection

An enough IBP system $$0 = \sum_j c_{ij} I_j$$

Column reduction (numeric + finite field)

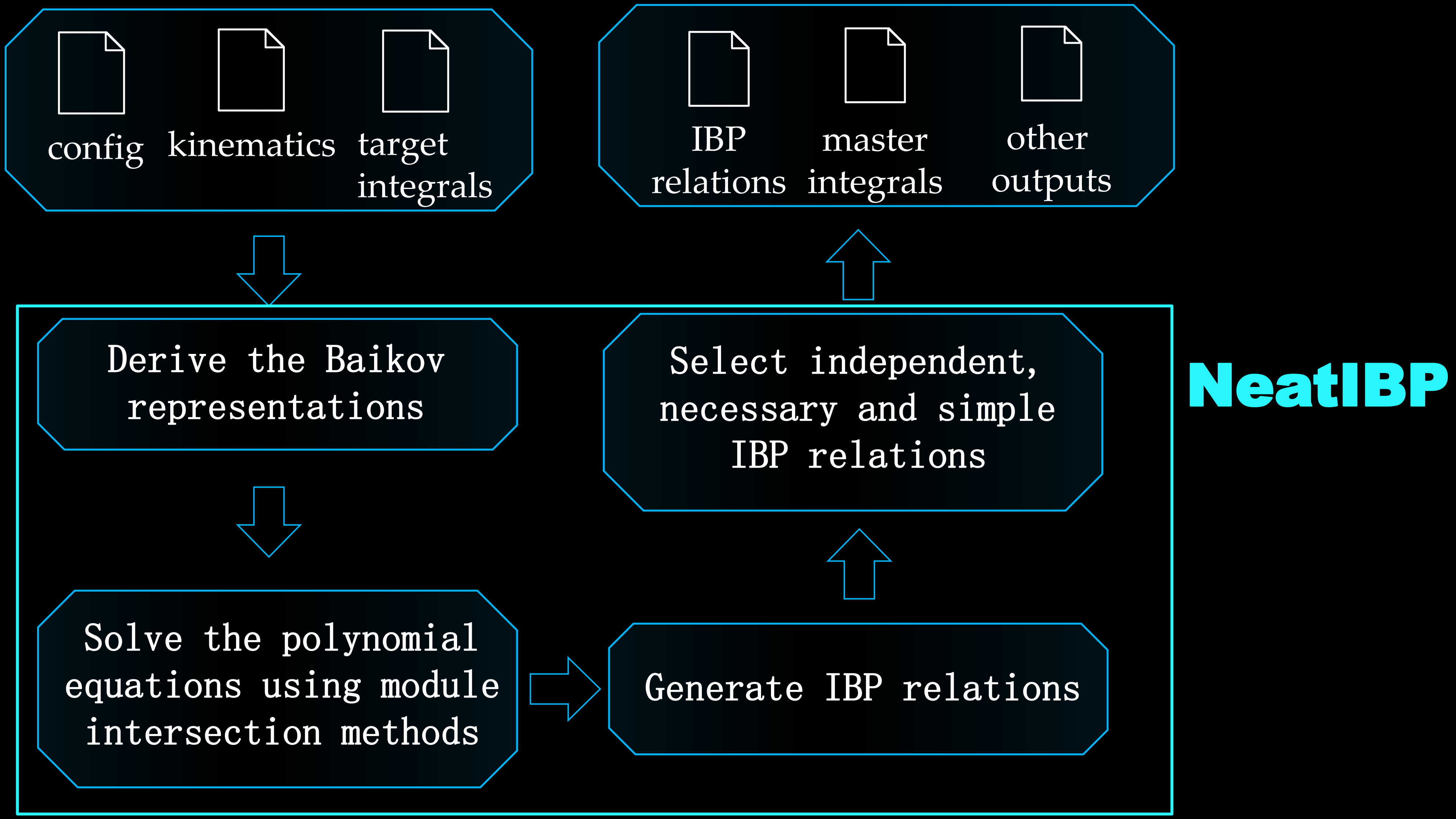
Linearly independent system $$0 = \sum_j \tilde{c}_{ij} I_j$$

Row reduction (numeric + finite field) $$R_{ik} = L_{ij} \tilde{c}_{jk}$$

Remove the unneeded relations for reducing the targets

Small-size IBP system minimally needed

The work flow and program package

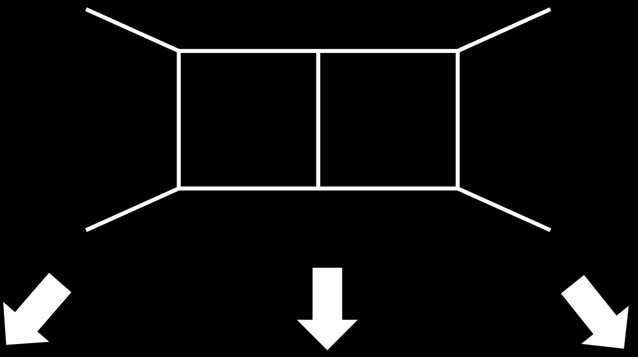


Parallelization between sectors

$$\int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

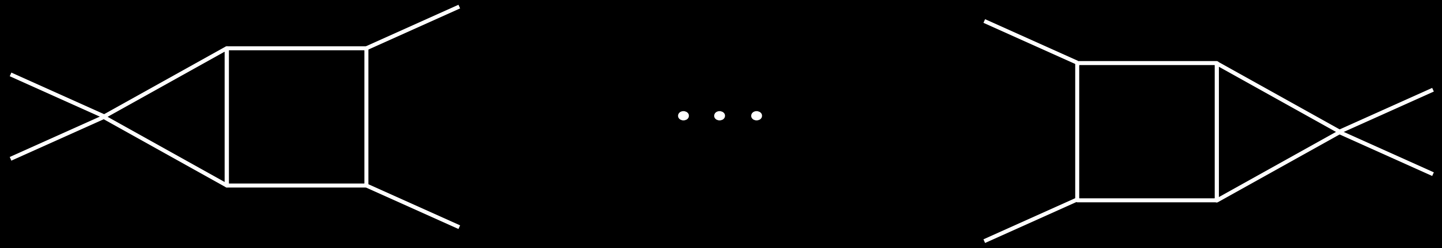
Tail mask strategy

current sector:



$$0 = \sum_j c_{ij}^{\mathrm{h}} I_j^{\mathrm{h}} + \sum_k c_{ik}^{\mathrm{t}} I_k^{\mathrm{t}}$$

sub sectors:



Contents

Algorithms and features in NeatIBP 1.0

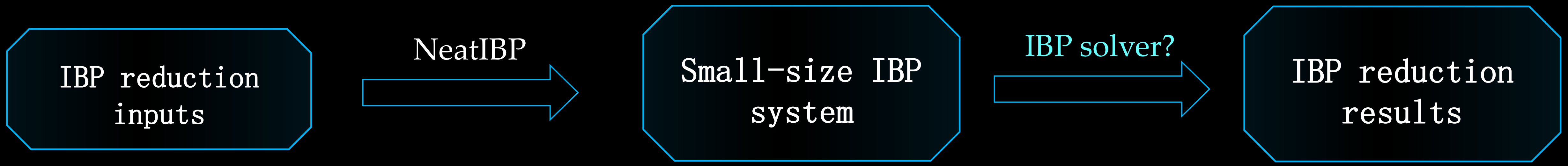
Main new features in the NeatIBP 1.1

1. The Kira+NeatIBP interface

2. The spanning cuts method

3. Algorithm of syzygy vector simplification

Currently, NeatIBP itself dose not perform IBP reduction

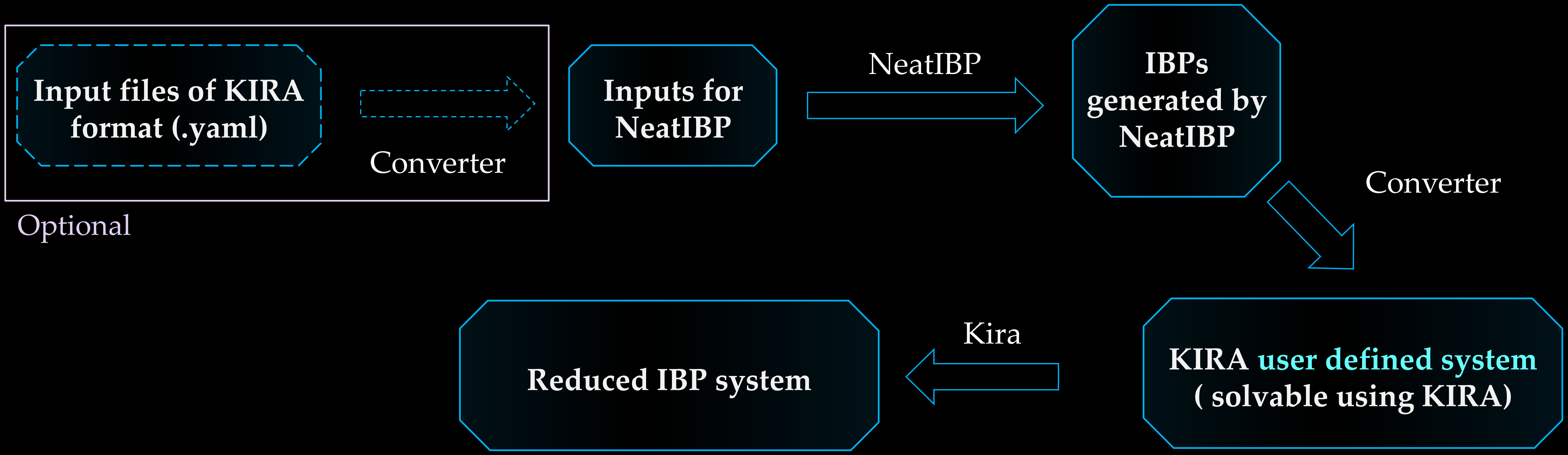


KIRA Comput.Phys.Commun. 230 (2018) 99-112, Comput.Phys.Commun. 266 (2021) 108024, etc.
as a popular integral reduction software, is a very nice IBP solver for NeatIBP output.

Kira allows user to feed in linear systems of IBP relations (called user defined system) and then reduce them .

The Kira+NeatIBP interface is included in NeatIBP 1.1

The Kira+NeatIBP interface (normal mode)



In NeatIBP 1.1, to use Kira to reduce the IBP generated by NeatIBP, add the following commands into NeatIBP "config.txt"

```
PerformIBPReduction=True;
IBPReductionMethod="Kira";
KiraCommand="/some/path/kira"
FermatPath="/some/path/fer64"
```

Contents

Generalized unitarity cuts in Baikov representation

$$I_{\alpha_1,\cdots,\alpha_n}|_{\mathcal{C}-\mathrm{cut}} \propto \oint_0 \prod_{i\in\mathcal{C}} \mathrm{d}z_i \int \prod_{i\notin\mathcal{C}} \mathrm{d}z_i P^\alpha \frac{1}{z_1^{\alpha_1}\dots z_n^{\alpha_n}}$$

Cuts change the integrals but preserve IBP relations

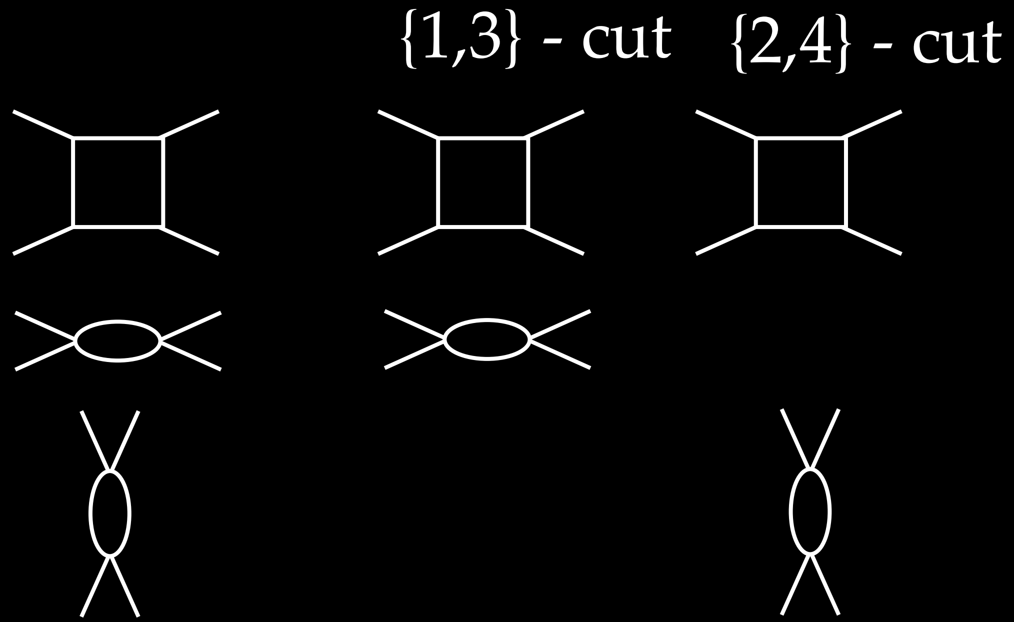
$$\sum_i c_i I_i = 0 \longrightarrow \sum_i c_i (I_i|_{\mathcal{C}-\mathrm{cut}}) = 0$$

Cuts increases the number of *zero sectors* in a family

$$\mathcal{C} \not\subseteq S \Rightarrow I|_{\mathcal{C}-\mathrm{cut}} = 0, \forall I \in \mathrm{Sector}\ S$$

Spanning cuts method

A spanning cuts $\{\mathcal{C}_i\}$ is defined so that for all nonzero sector S in a family, $\exists \mathcal{C}_i$, such that $C_i \subseteq S$



Step 1: Run NeatIBP on each cut

Current version of NeatIBP avoids cutting multiple propagators.
Thus, for integrals such that $\alpha_i = 1, i \in \mathcal{C}$, $\mathcal{C}$-cut means

$$P \to P|_{z_i \to 0, i \in \mathcal{C}}$$

Step 2: Reduce the IBP system on each cut   Kira

Step 3: Merge the reduced IBP system of all cuts

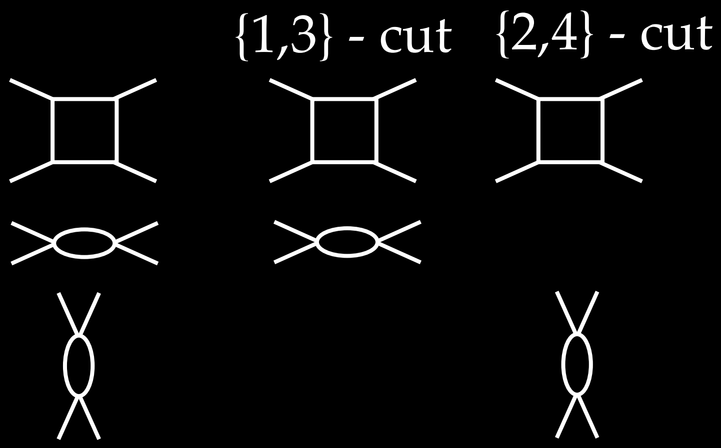
Benefit:

1. Split a difficult problem into simpler pieces
2. Parallelizable

Merging the spanning cut systems

$$I|_{\{1,3\}\text{-cut}} = \sum_i c^{1234}_{\{1,3\},i} I^{1234}_i|_{\{1,3\}\text{-cut}} + \sum_i c^{13}_{\{1,3\},i} I^{13}_i|_{\{1,3\}\text{-cut}}$$

$$I|_{\{2,4\}\text{-cut}} = \sum_i c^{1234}_{\{2,4\},i} I^{1234}_i|_{\{2,4\}\text{-cut}} + \sum_i c^{24}_{\{2,4\},i} I^{24}_i|_{\{2,4\}\text{-cut}}$$



⇩ merge

$$I = \sum_i c^{1234}_i I^{1234}_i + \sum_i c^{13}_i I^{13}_i + \sum_i c^{24}_i I^{24}_i$$

Currently, NeatIBP uses a “direct” merging strategy:

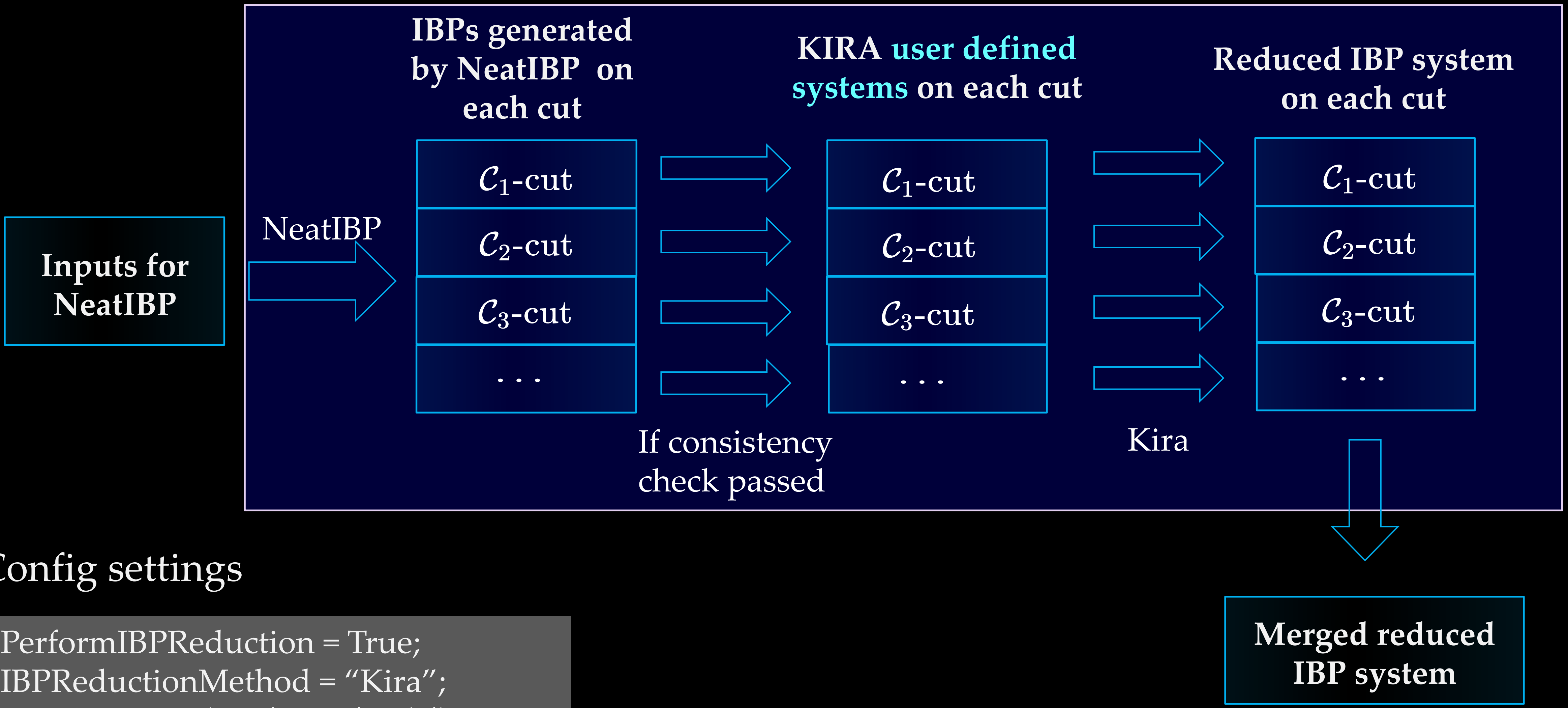
$$\begin{cases} c^{13}_i = c^{13}_{\{1,3\},i} \\ c^{24}_i = c^{24}_{\{2,4\},i} \\ c^{1234}_i = \boxed{c^{1234}_{\{1,3\},i} = c^{1234}_{\{2,4\},i}} \end{cases}$$

Consistency condition

NeatIBP checks consistency conditions before merging the spanning cut systems. This step cannot be omitted because the condition is NOT always guaranteed according to our observation. Inconsistency often observed in integrals with massive propagators.

The Kira+NeatIBP interface (spanning cuts mode)

Parallelized by default

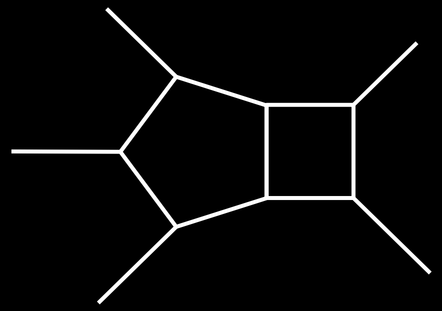


Config settings

```
PerformIBPReduction = True;
IBPReductionMethod = "Kira";
KiraCommand = "/some/path/kira";
FermatPath = "/some/path/fer64";
SpanningCutsMode = True;
```

A baby example of Kira+NeatIBP performance (with spanning cuts)

Two-loop five-point, with numerator degree 4



CPU threads: 20
RAM: 128GB

Running NeatIBP + Kira in spanning cuts mode:

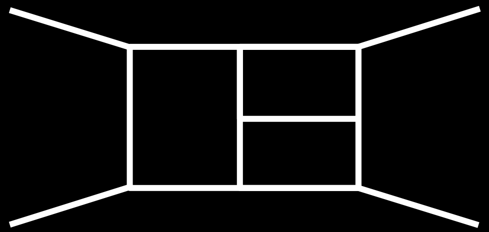
1. NeatIBP: Generates equations on 10 spanning cuts, time used: 8m
2. Number of equations: vary from 310 ~ 1583, on different cuts
3. Kira reduction time used: 5h

Running NeatIBP + Kira in normal mode:

1. NeatIBP: Generates 3708 equations spanning cuts, using 18m
2. Kira reduction time used: 24h

Another example of Kira+NeatIBP performance (with spanning cuts)

Three-loop four-point, with numerator degree 5



CPU threads: 20
RAM: 128GB

Running NeatIBP + Kira in spanning cuts mode:

1. NeatIBP: Generates equations on 22 spanning cuts, time used: 1h22m
2. Number of equations: 2k ~ 10k+ or so, at most 35k, on different cuts
3. Kira reduction time used : 35m

Running NeatIBP + Kira in normal mode:

1. NeatIBP: Generates 113k equations spanning cuts, using 13h
2. Kira reduction time used: 2h30m

Contents

Generator vector of the solution module of the syzygy equations

$$\begin{pmatrix} a_i \\ b \end{pmatrix} \in M$$

Symbolic IBP relation

$$0 = \int \mathrm{d}z_1 \cdots \mathrm{d}z_n \left(\sum_{i=1}^{n} \left(\frac{\partial a_i}{\partial z_i} - \alpha_i b_i \right) + \alpha b \right) P^{\alpha} \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}}$$

Seeding

$$\vec{\alpha} \to (1, \cdots, 1, -2, -3)$$

$$\vec{\alpha} \to (1, \cdots, 1, 0, -5)$$

$$\vdots$$

Specific IBP relation

(# generators) × (# seeds) = (# IBP relations)

Generator vector of the solution module of the syzygy equations

$$\begin{pmatrix} a_i \\ b \end{pmatrix} \in M$$

Symbolic IBP relation

$$0 = \int \mathrm{d}z_1 \cdots \mathrm{d}z_n \left(\sum_{i=1}^{n} \left(\frac{\partial a_i}{\partial z_i} - \alpha_i b_i \right) + \alpha b \right) P^{\alpha} \frac{1}{z_1^{\alpha_1} \cdots z_n^{\alpha_n}}$$

Seeding

$$\vec{\alpha} \to (1, \cdots, 1, -2, -3)$$
$$\vec{\alpha} \to (1, \cdots, 1, 0, -5)$$
$$\vdots$$

Specific IBP relation

(# generators) × (# seeds) = (# IBP relations)

~ hundreds

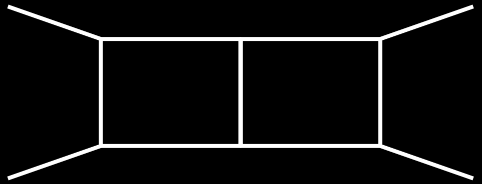
large combinatorial numbers

Seeding cost ↑

Idea of the algorithm (taking no-dot integrals as example)

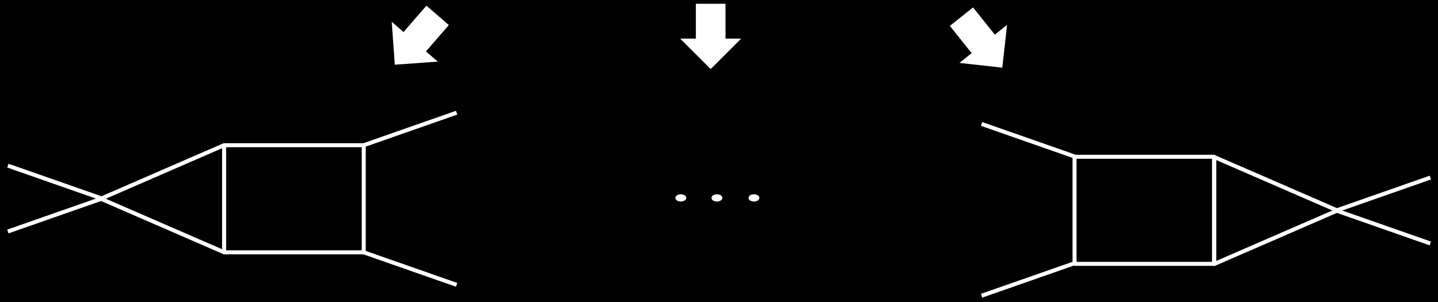
$$\int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

current sector:



$$0 = \sum_j c_{ij}^{\mathrm{h}} I_j^{\mathrm{h}} + \sum_k c_{ik}^{\mathrm{t}} I_k^{\mathrm{t}}$$

sub sectors:



Treating tail integrals as zeros:

1. IBP relations containing only sub sector integrals will be discarded
2. This is equivalent to maximal cut (mc):

$$\text{for } I \in \text{sector } S\,, \quad I|_{\mathrm{mc}} := I|_{S\text{-cut}}$$

Simplification of syzygy generators via maximal cut

Delete generators that dose not change the “maximal cut” module $M|_{\mathrm{mc}}$ defined as

$$M|_{\mathrm{mc}} := \langle f_1|_{\mathrm{mc}}, f_2|_{\mathrm{mc}}, \cdots \rangle \qquad \text{for} \qquad M = \langle f_1, f_2, \cdots \rangle$$

Step 1: Using “maximal cut” Groebner basis

$$G|_{\mathrm{mc}} := \mathrm{GB}(M|_{\mathrm{mc}}) = \langle g_1, g_2, \cdots \rangle \qquad \text{lifting} \qquad g_i = \sum_j c_{ij} f_i|_{\mathrm{mc}}$$

Delete f_j in generators of M if $c_{ij} = 0, \forall i$

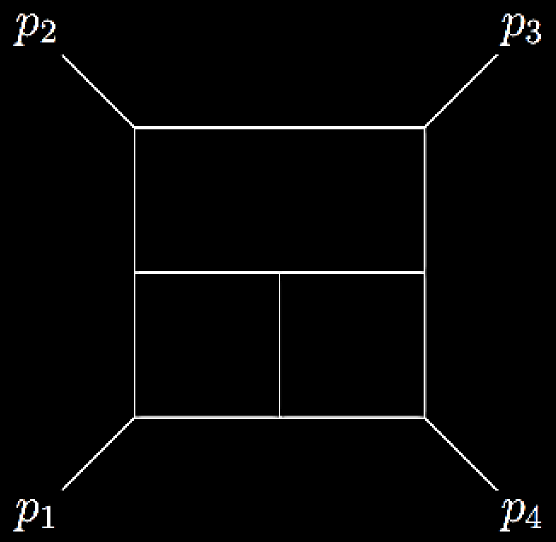
Because this means $f_j|_{\mathrm{mc}}$ does not contribute to $M|_{\mathrm{mc}}$

Step 2: Scanning

After step 1, sort the remaining generators from complex to simple $M' = \langle f'_1, \cdots, f'_n \rangle$

Scanning i from 1 to n, if deleting f'_i does not change the GB of maximal cut module, then delete it.

Example



Degree-5 targets

	without syzygy simplification			with syzygy simplification		
sector	#gen	time used	memory used	#gen	time used	memory used
1023	666	1h8m	20.2G	16	15m	6.7G
1022	665	4h4m	31.2G	39	1h4m	18.8G
1021	536	1h21m	22.0G	25	13m	4.0G
1019	541	50m	14.6G	30	9m	2.9G
1015	654	1h17m	16.9G	21	16m	5.0G
1013	602	2h1m	17.1G	32	37m	11.3G
981	432	2h24m	11.8G	61	47m	9.2G
949	520	2h1m	13.6G	514	3h37m	20.1G
719	404	56m	5.3G	398	1h19m	5.6G
511	721	38m	10.8G	31	13m	4.4G
379	657	39m	8.4G	657	1h44m	8.9G
351	778	2h13m	13.4G	778	2h33m	14.0G
251	510	46m	6.8G	35	14m	4.1G
...	...	...	...	...	...	...

Summary

Feynman integrals are important objects in particle physics.

The integration-by-parts (IBP) reduction is one of the bottle-neck steps in the evaluation of multi-loop Feynman integrals.

NeatIBP is an automated program generating small-size IBP system. It helps to reduce the computation cost of IBP reduction.

NeatIBP has been successfully applied to frontier problems, including two-loop amplitude computation in phenomenology and three-loop five-point Feynman diagrams.

We have displayed some new features of NeatIBP of the new version. They include:

1. The spanning cuts method：
   Splitting the hard problem into simpler pieces.
2. The Kira interface:
   Providing automated IBP solver.
3. Simplification algorithm of syzygy generators

https://github.com/yzhphy/NeatIBP