Recent developments of NeatIBP

A package generating small sized integration-by-parts relations for Feynman integral reduction

Zihao Wu

Based on: Rourou Ma, Johann Usovitsch, ZW, Yingxuan Xu, Yang Zhang, 24xx.xxxx,
related previous work: ZW, Janko Boehm, Rourou Ma, Hefeng Xu, Yang Zhang, Comput.Phys.Commun. 295 (2024) 108999
and important discussions with: David Kosower, Fabian Lange, Roman Lee, Yan-Qing Ma, Simone Zoia.

Multi-loop Feynman integrals in particle physics

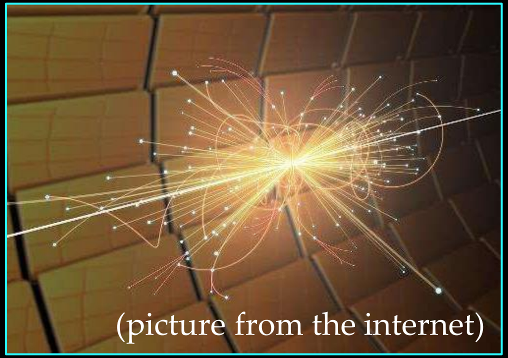
Multi-loop Feynman integrals

Multi-loop Feynman diagrams

Action of the theory in QFT

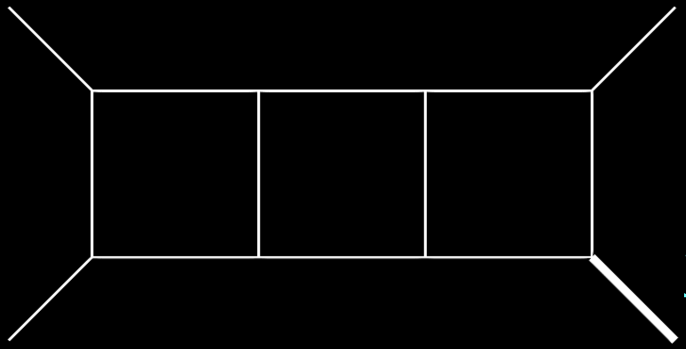
High-order scattering amplitudes

High-precision experiments

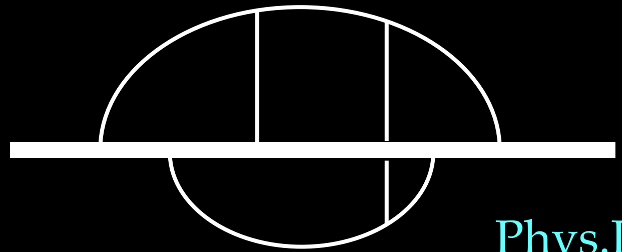

(picture from the internet)

Multi-loop Feynman integrals, frontiers

More loops:

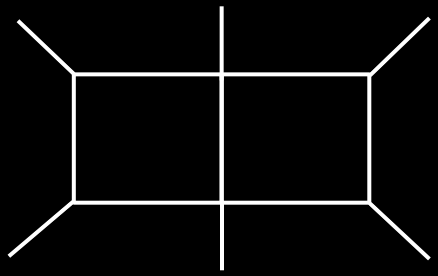


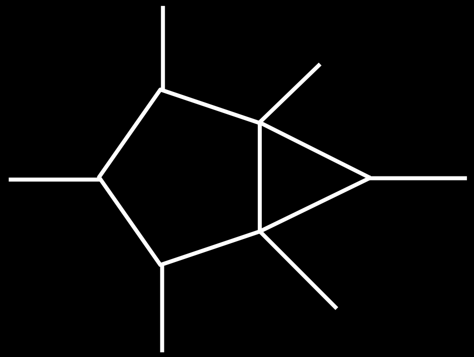
JHEP 05 (2023) 026

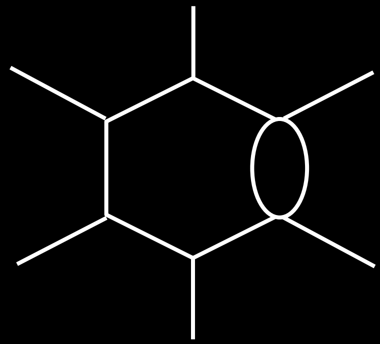


Phys.Rev.D 109 (2024) 7, L071503

More legs:

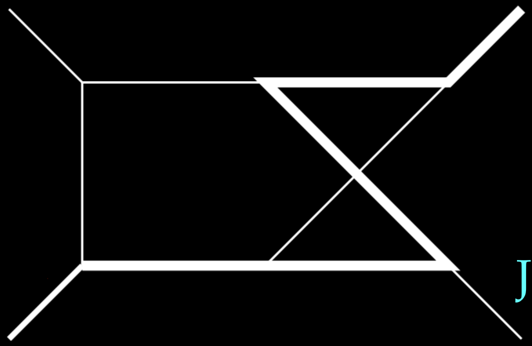




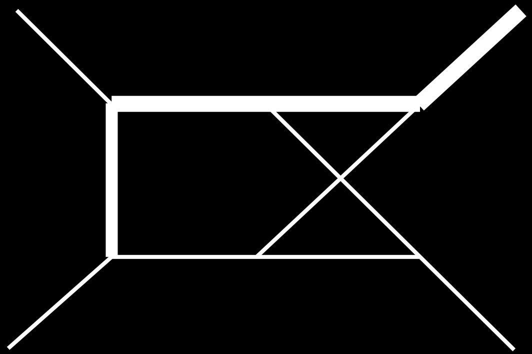


JHEP 08 (2024) 027

More masses:



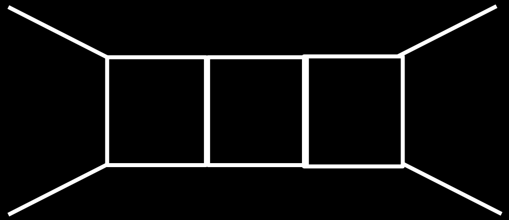
JHEP 07 (2023) 089



JHEP 06 (2023) 144

Feynman integral reduction

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

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

Integration-by-parts (IBP) reduction ↓

Master Integrals → Analytic functions

Analytic evaluation methods including dfferential equations

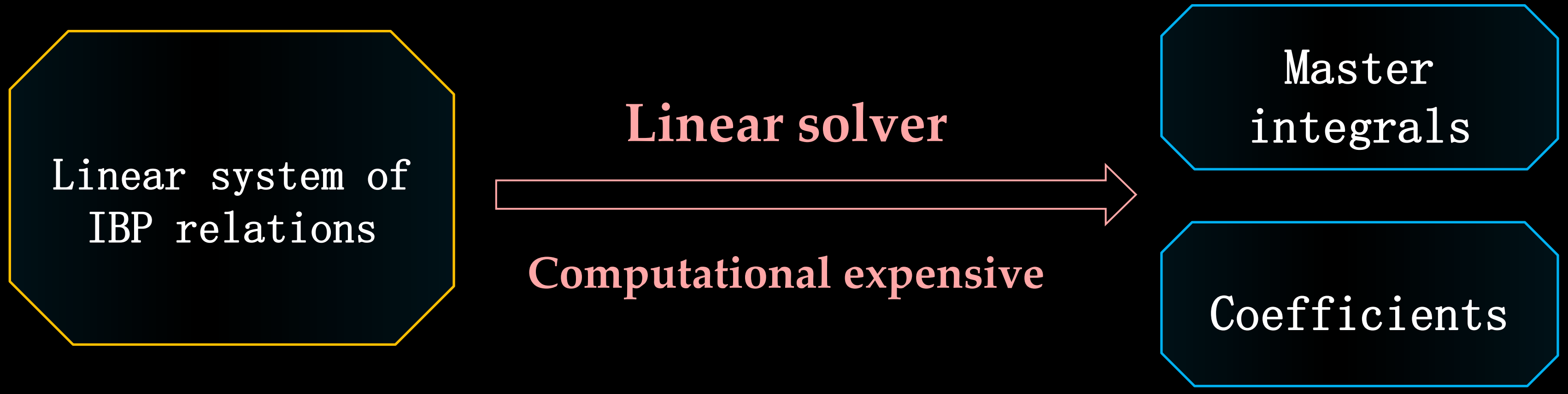
A linear independent basis of the scalar integrals

Computational bottle necks in integration-by-parts (IBP) reductions

A traditional IBP reductio method such as Laporta's algorithm Int. J. Mod. Phys. A, 2000, 15: 5087-5159.

Computational bottle necks in frontier problems

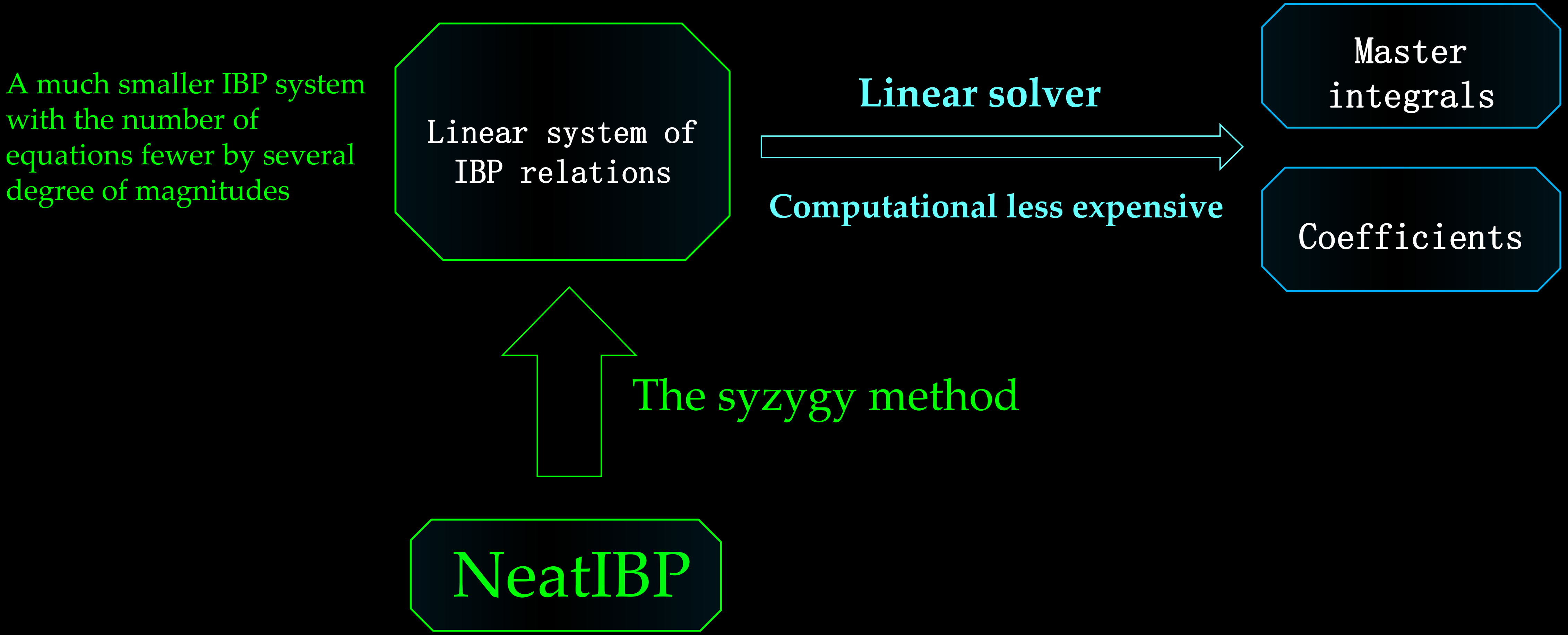
Multivariate, large-sized matrix (can be up to ~ 10^7)



Costing days/weeks on clusters for a cutting-edge problem

Costing hundreds of GB / several TB RAM or more

Generating small-size IBP system using NeatIBP



Contents

A brief review of NeatIBP

Review of NeatIBP, till its v1.0.2.4, the last published version (Comput.Phys.Commun. 295 (2024) 108999) at 14th May 2023

Recent developments and new features

Progress of NeatIBP from its v1.0.2.4 to the latest version, v1.0.5.1, at 6th Sep. 2024.

1. The spanning cuts method

2. The Kira+NeatIBP interface

3. Simplification of syzygy vectors using maximal cut

Contents

A brief review of NeatIBP

Review of NeatIBP, till its v1.0.2.4, the last published version (Comput.Phys.Commun. 295 (2024) 108999) at 14th May 2023

Recent developments and new features

Progress of NeatIBP from its v1.0.2.4 to the latest version, v1.0.5.1, at 6th Sep. 2024.

1. The spanning cuts method

2. The Kira+NeatIBP interface

3. Simplification of syzygy vectors using maximal cut

IBP relations in with multiple propagators

Target integrals

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

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

$\dots$

IBP-Related integrals

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

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

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

$\dots$

Master integrals

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

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

$\dots$

Contains **redundant integrals** with denominator indices lifted

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

Introducing multiple propagators

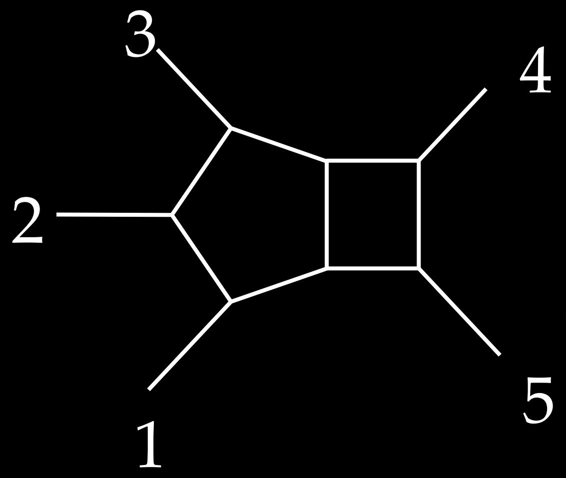
IBP relations from syzygy method

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

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) $\times$ (# 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

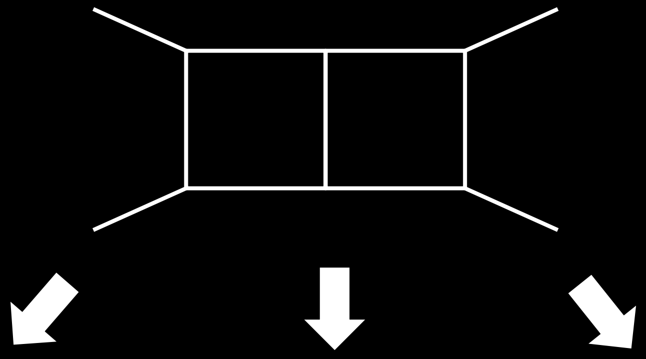
Small-size IBP system minimally needed

Tail mask strategy

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

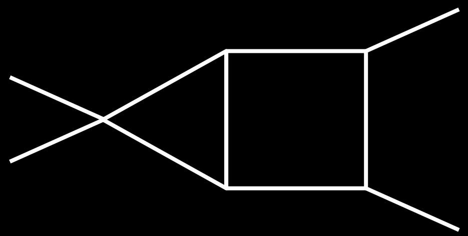
Treating tail integrals as zeros

current sector:

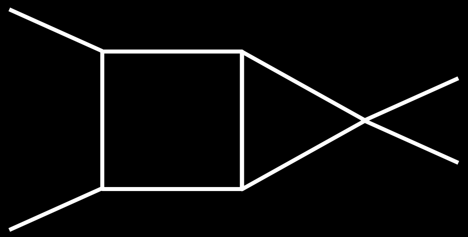


$$0 = \sum_j c_{ij}^{\mathrm{h}} I_j^{\mathrm{h}} + \sum_k c_{ik}^{\mathrm{t}} I_k^{\mathrm{t}}$$

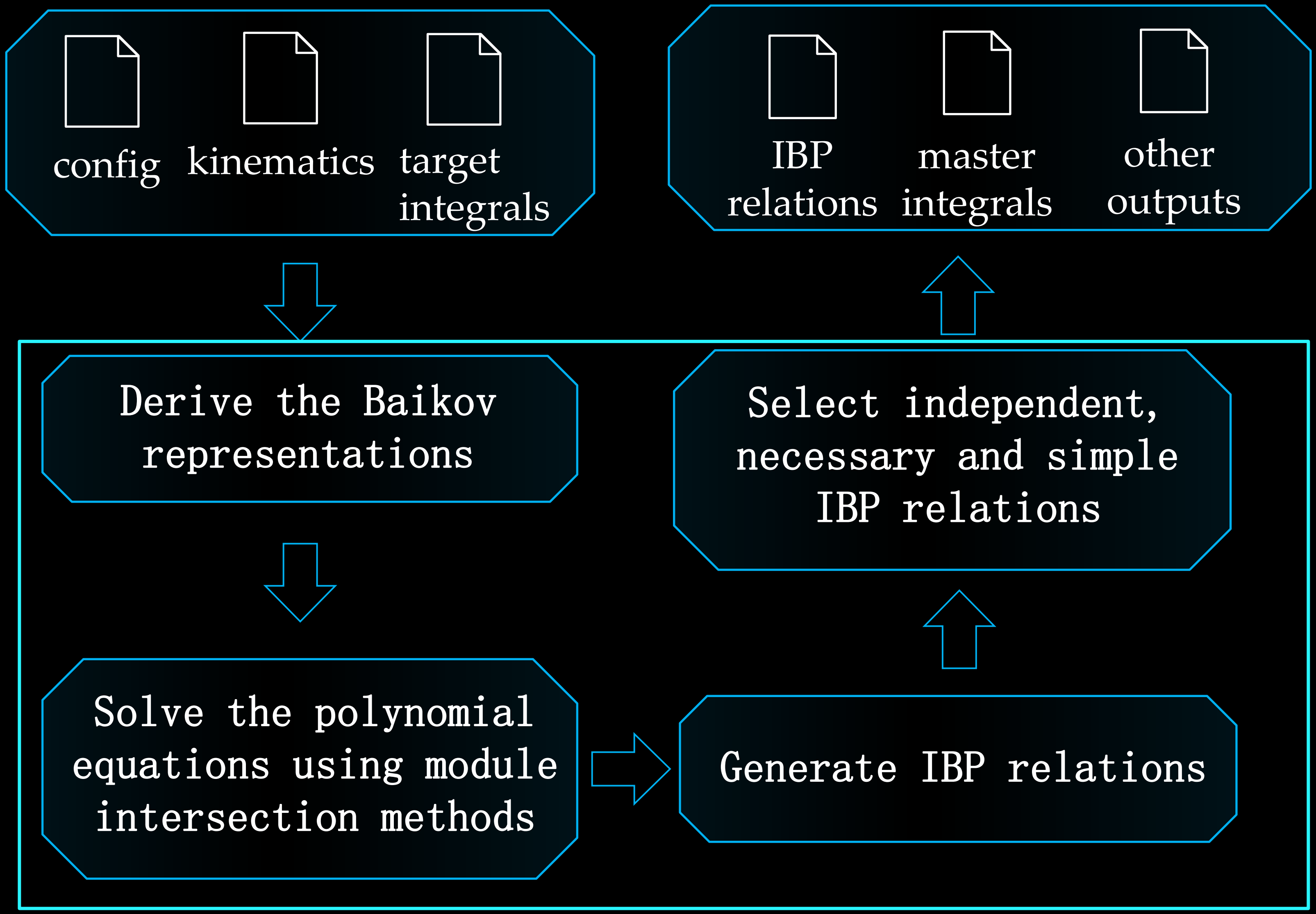
sub sectors:



$\ldots$



The work flow and program package



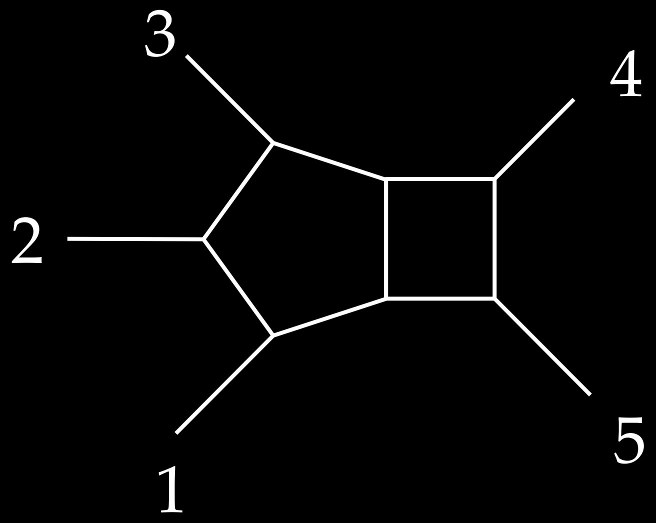
NeatIBP

Github:

https://github.com/yzhphy/NeatIBP

Performance of NeatIBP

2L5P Example



Target integrals with high-degree numerators
Quantity: 2483

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

Max numerator degree: 5
Max denominator power: 1
# of MI: 61
# of IBP: **14120**

Time used: 27m at 

CPU threads: 20
RAM: 128GB

Contents

A brief review of NeatIBP

Review of NeatIBP, till its v1.0.2.4, the last published version (Comput.Phys.Commun. 295 (2024) 108999) at 14th May 2023

Recent developments and new features

Progress of NeatIBP from its v1.0.2.4 to the latest version, v1.0.5.1, at 6th Sep. 2024.

1. The spanning cuts method

2. The Kira+NeatIBP interface

3. Simplification of syzygy vectors using maximal cut

Generalized unitarity cuts in Baikov representation

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

Cuts change the integrals but preserve IBP relations

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

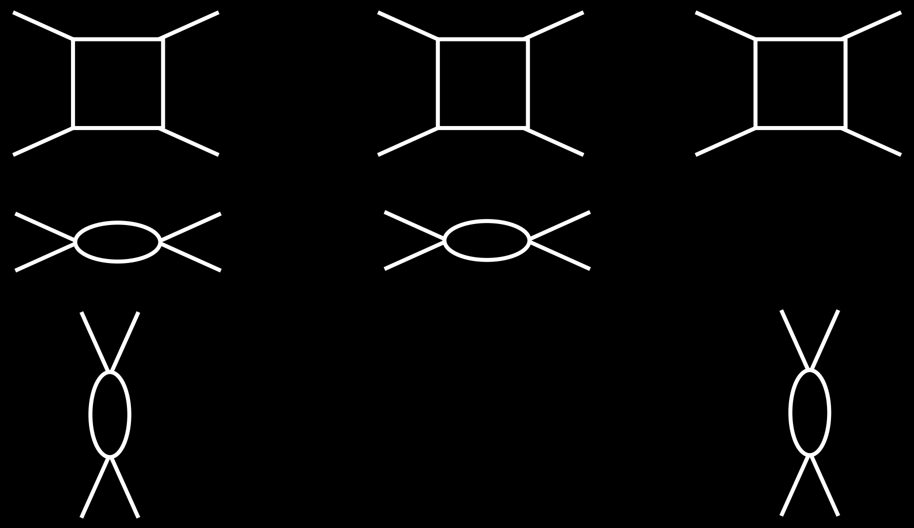
Cuts increases the number of *zero sectors* in a family

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

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 $C_i \subseteq S$

{1,3} - cut {2,4} - cut



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

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

Step 2: Reduce the IBP system on each cut

Step 3: Merge the reduced IBP system of all cuts

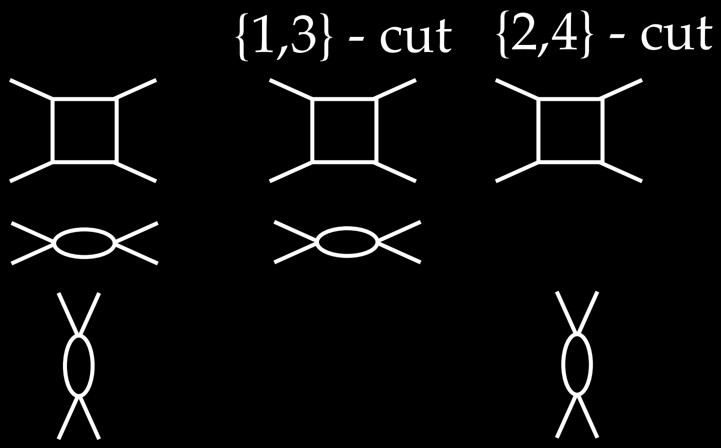
Benefit:

1. Σ difficulty of evaluation on spanning cuts < evaluation on the family as a whole
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:

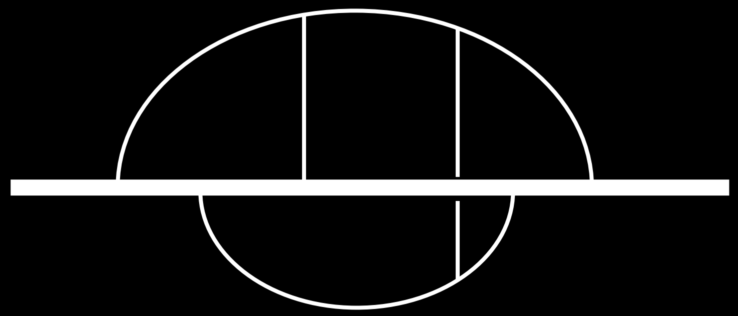
$$\begin{cases} c^{13}_i = c^{13}_{\{1,3\},i} \\ c^{24}_i = c^{24}_{\{2,4\},i} \\ c^{1234}_i = \boxed{c^{1234}_{\{1,3\},i} = c^{1234}_{\{2,4\},i}} \end{cases}$$

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.

Automated evaluation on spanning cuts in NeatIBP

Example:



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

Plainly running NeatIBP:

No, I cannot finish this computation. I stuck at syzygy vectors evaluation for days…. ☹

Running NeatIBP with spanning cuts

```
CutIndices = "spanning cuts";
```

Yes, after splitting the problem into 29 pieces of different cuts, I finished the IBP relation generation on spanning cuts in 10+ hours. ☺

Contents

A brief review of NeatIBP

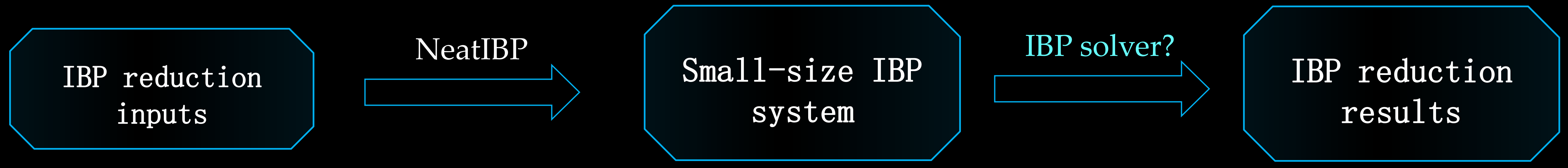
Review of NeatIBP, till its v1.0.2.4, the last published version (Comput.Phys.Commun. 295 (2024) 108999) at 14^{th} May 2023

Recent developments and new features

Progress of NeatIBP from its v1.0.2.4 to the latest version, v1.0.5.1, at 6^{th} Sep. 2024.

1. **The spanning cuts method**
2. **The Kira+NeatIBP interface**
3. **Simplification of syzygy vectors using maximal cut**

Currently, NeatIBP itself dose not perform IBP reduction

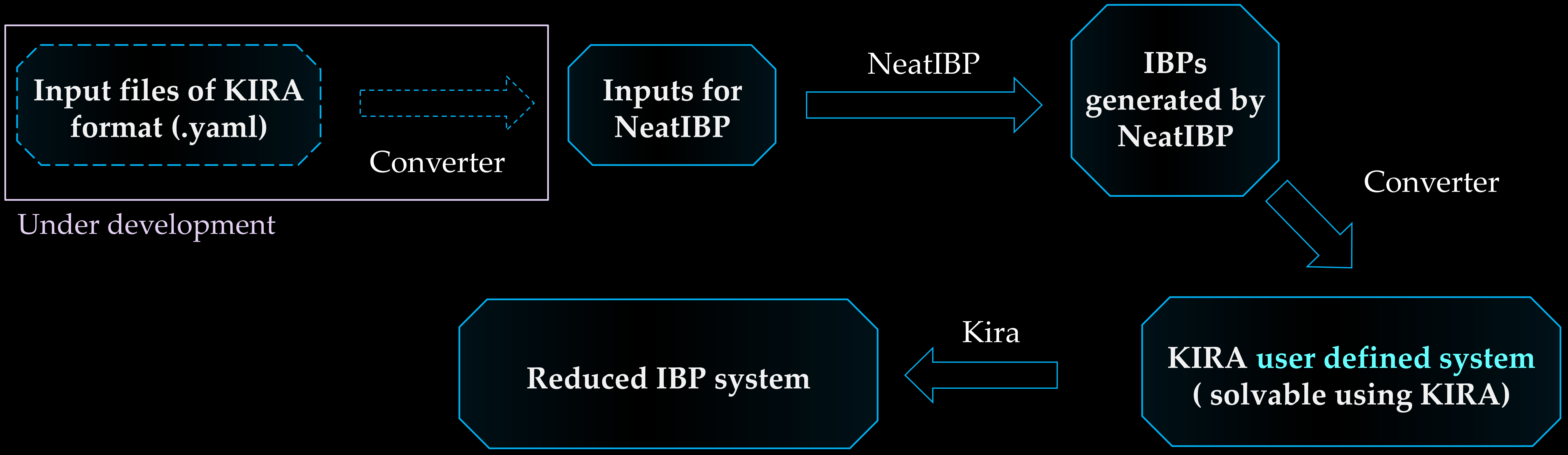


KIRA Comput.Phys.Commun. 230 (2018) 99-112, Comput.Phys.Commun. 266 (2021) 108024, etc.
as a popular integral reduction software, is a very nice IBP solver for NeatIBP output.

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 the latest version of NeatIBP (v1.0.5.1, at 6th Sep. 2024)

The Kira+NeatIBP interface (normal mode)

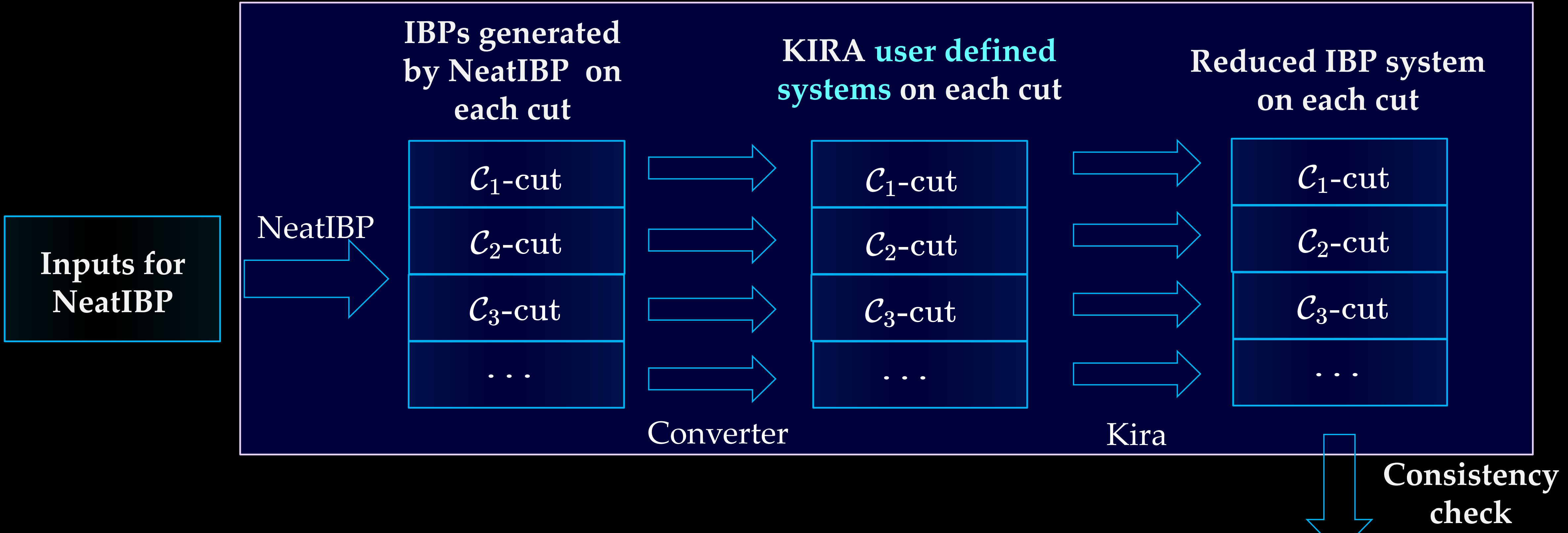


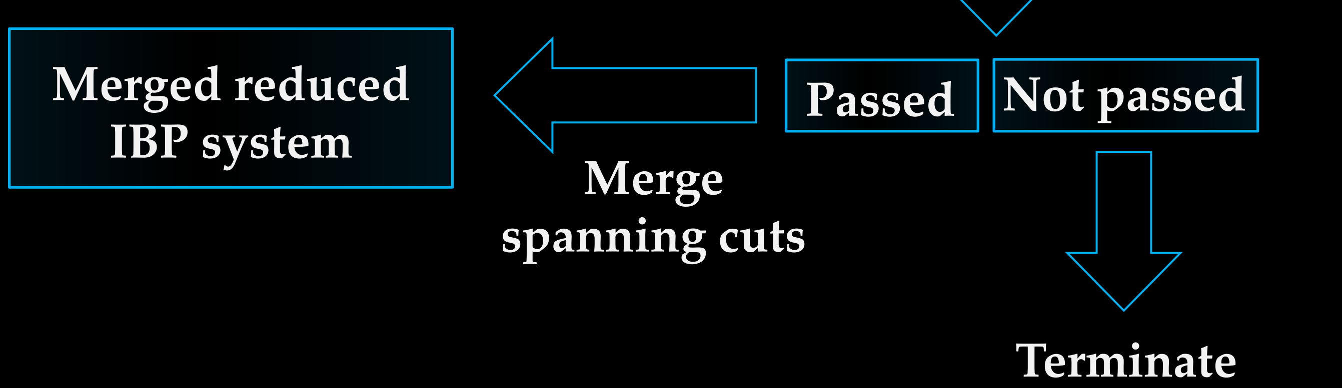
In NeatIBP (since v1.0.5.0), to use Kira to reduce the IBP generated by NeatIBP, add the following commands into NeatIBP “config.txt”

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

The Kira+NeatIBP interface (spanning cuts mode)

Parallelized by default



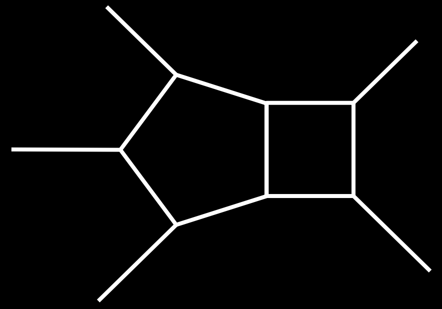


Config settings (NeatIBP v1.0.5.1):

```
PerformIBPReduction = True;
IBPReductionMethod = “Kira”;
KiraCommand = “/some/path/kira”;
FermatPath = “/some/path/fer64”;
CutIndices = “spanning cuts”;
```

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

Two-loop five-point, with numerator degree 3



CPU threads: 20
RAM: 128GB

Running NeatIBP + Kira in spanning cuts mode:

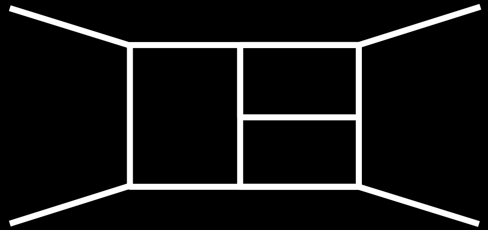
1. NeatIBP: Generates equations on 10 spanning cuts, using 5m
2. Number of equations: vary from 130 ~ 624, on different cuts
3. Kira running time: 10m (including converting, checking consistency, reducing and merging, in parallel mode)

Running NeatIBP + Kira in normal mode:

1. NeatIBP: Generates 1358 equations spanning cuts, using 16m
2. Kira reduction time used: 27m

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, using 6h21m
2. Number of equations: 2k ~ 10k+ or so, at most 35k, on different cuts
3. Kira running time: 7h40m (including converting, checking consistency, reducing and merging, in parallel mode)

Running NeatIBP + Kira in normal mode:

1. NeatIBP: Generates 114k equations spanning cuts, using 18h44m
2. Kira reduction: Testing in progress. Timing unknown yet. My 128GB RAM limit on the current machine is exceeded.

Contents

A brief review of NeatIBP

Review of NeatIBP, till its v1.0.2.4, the last published version (Comput.Phys.Commun. 295 (2024) 108999) at 14th May 2023

Recent developments and new features

Progress of NeatIBP from its v1.0.2.4 to the latest version, v1.0.5.1, at 6th Sep. 2024.

1. The spanning cuts method

2. The Kira+NeatIBP interface

3. Simplification of syzygy vectors using maximal cut

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

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

⇒

Formal 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

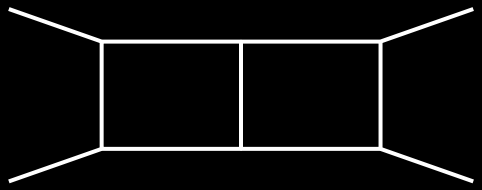
large binomial numbers

Seeding cost ↑

Tail mask strategy

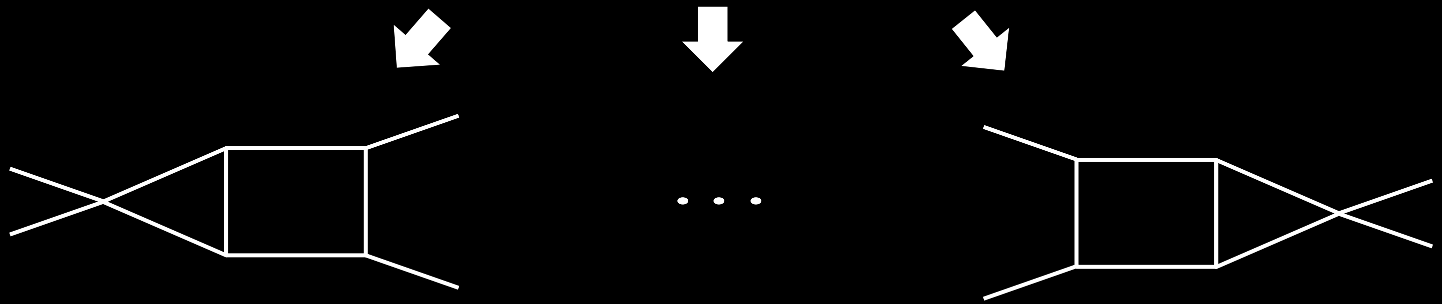
$$\int \frac{\mathrm{d}^D l_1}{i\pi^{D/2}} \cdots \frac{\mathrm{d}^D l_L}{i\pi^{D/2}} \frac{1}{D_1^{\alpha_1} \cdots D_n^{\alpha_n}}$$

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):

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

Maximal cut

Consider IBP relations with "no-multiple-propagator" integrals in a sector S such that $\begin{cases} \alpha_i = 1, & i \in S, \\ \alpha_i \leq 0, & i \notin S, \end{cases}$

Define maximal cut on a syzygy vector

$$f := \begin{pmatrix} a_i \\ b \end{pmatrix} \in M \qquad\qquad f|_{\mathrm{mc}} := f|_{z_i \to 0, i \in S}$$

From the IBP formula

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

We have

$f|_{\mathrm{mc}} = 0 \quad \Leftrightarrow$ Corresponding IBP relation contains only integrals in sub sectors

Simplification of syzygy generators via maximal cut

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

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

Step 1: Using "maximal cut" Groebner basis

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

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

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

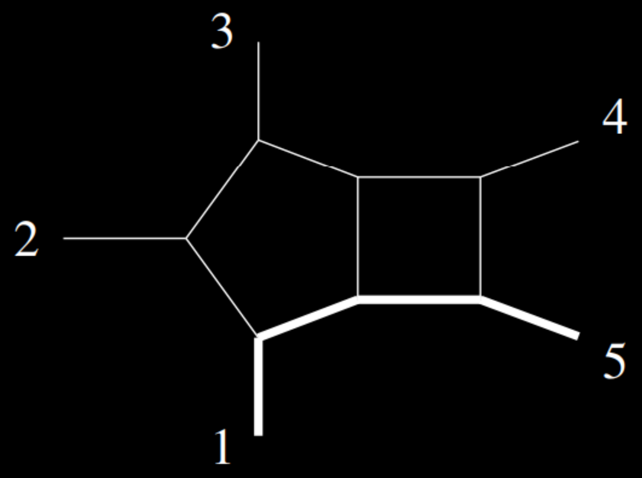
Step 2: Scanning

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

Top sector of a pentagon-box diagram



Before simplification: (# generators) = 374

After simplification: (# generators) = 14

~ 25 times shortened

More general cases

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

Consider IBP relations with generic integrals in a sector S such that $\begin{cases} \alpha_i > 0, & i \in S \\ \alpha_i \leq 0, & i \notin S \end{cases}$

modification

$$f := \begin{pmatrix} a_i \\ b \end{pmatrix} \in M \longrightarrow f := \begin{pmatrix} \tilde{a}_i \\ b \end{pmatrix} \text{ where } \tilde{a}_i := \begin{cases} \frac{a_i}{z_i}, & i \in S \\ a_i, & i \notin S \end{cases}$$

Summary

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

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

We have introduced some new features of NeatIBP till its latest 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 of syzygy generators using maximal cut method:
   Making the seeding process computational cheaper.