Feynman integrals reduction using Blade

Yan-Qing Ma

Peking University

Mainly based on works in collaboration with:
Xin Guan, Xiao Liu, Zhi-Feng Liu, Xiang Li, and Wen-Hao Wu

MathemAmplitudes 2023: QFT at the Computational Frontier
Padova, Italy, 2023/09/25-27

Outline

I. Introduction

II. Block-triangular form and Blade

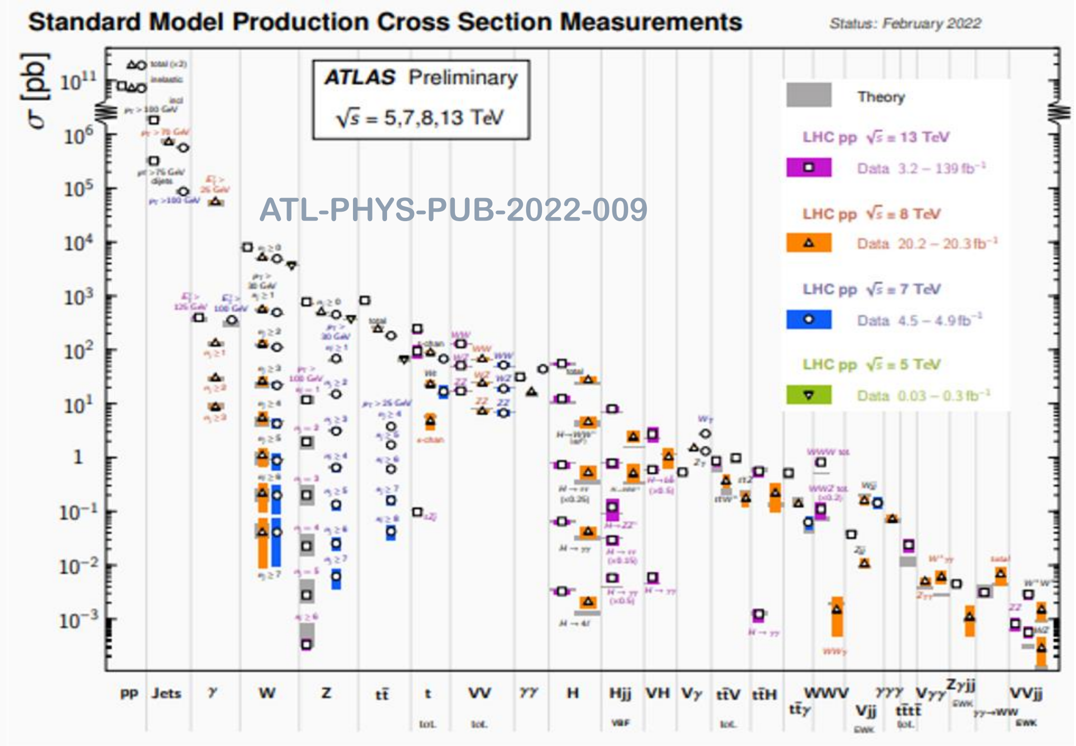
III. Applications of Blade

V. Summary and outlook

Era of precision physics at the LHC

➢ High-precision data

- Many observables probed at precent-level precision



- At least NNLO QCD corrections generally required (plus NLO EW, parton shower, resummation, etc.)

Automatic higher order perturbative calculation is highly demanded

Note: Automatic NLO correction obtained 15 years ago: MadGraph, Helac, etc

Efforts from PKU group towards automatic computation

	Generate amplitudes	Manipulate amplitudes	Integral reduction	Master integrals calculation
Package	qgraf (FeynArts, UFO, …)	CalcLoop (FeynCalc, and much more)	Blade (AIR,FIRE,Reduze,Kira,LiteRed,FiniteFlow, NeatIBP, …)	AMFlow (FIESTA,SecDec,Feyntrop, DiffExp, SeaSyde,…)
Link	http://cfif.ist.utl.pt/~paulo/qgraf.html	https://gitlab.com/multiloop-pku/calcloop	https://gitlab.com/multiloop-pku/blade	https://gitlab.com/multiloop-pku/amflow

- **Valid to any-loop order**
- **Key for automation is AMFlow: fully automatic, high precision**
- **Bottleneck is integral reduction: time/resource-consuming for complicated multiloop processes**

① **What is AMFlow?**

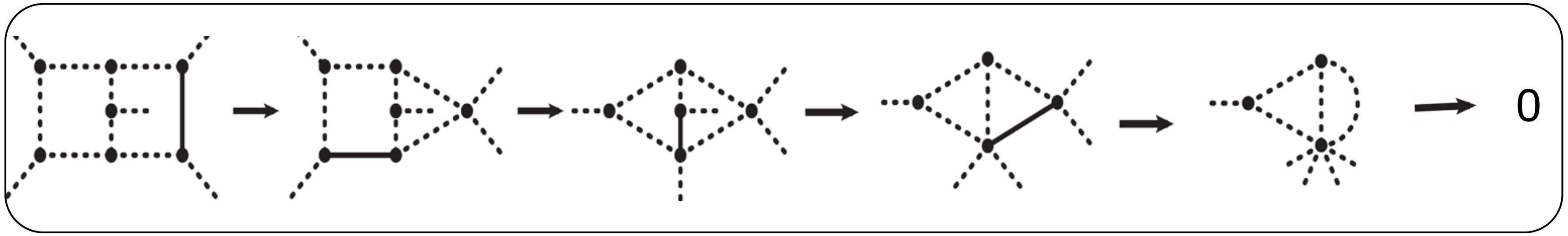
② **Why do we need Blade?**

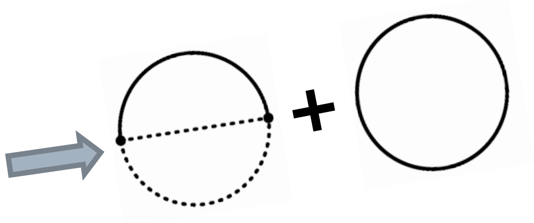
AMFlow: FIs to vacuum integrals

➢ Introducing auxiliary mass and taking it to infinity

$$I^{\text{aux}}_{\vec{\nu}}(D,\vec{s},\eta) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\mathcal{D}_{K+1}^{-\nu_{K+1}} \cdots \mathcal{D}_N^{-\nu_N}}{(\mathcal{D}_1 - \lambda_1 \eta + \mathrm{i}0^+)^{\nu_1} \cdots (\mathcal{D}_K - \lambda_K \eta + \mathrm{i}0^+)^{\nu_K}}$$

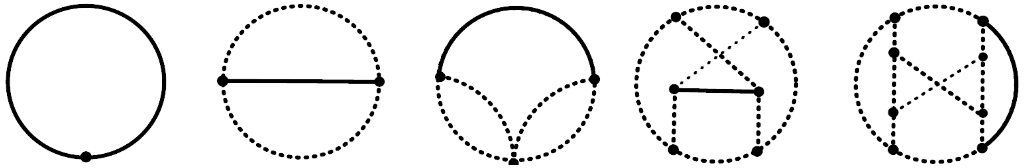
Liu, YQM, Wang, 1711.09572
Liu, YQM, 2107.01864





- Eventually, leaving only (single-mass) vacuum integrals as input
- Original FIs obtained by solving the DEs w.r.t. η

➢ Typical single-mass vacuum MIs: much simpler



Baikov, Chetyrkin, 1004.1153
Lee, Smirnov, Smirnov, 1108.0732
Georgoudis, et. al., 2104.08272

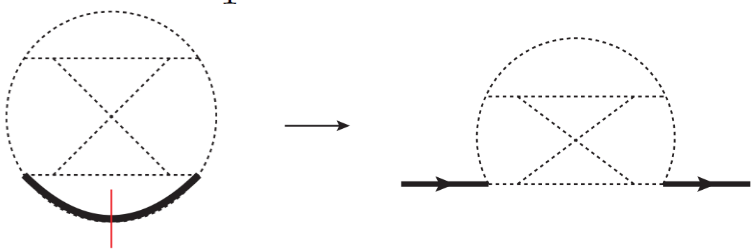
AMFlow: Vacuum integrals to vacuum integrals

➢ **A family of single-mass vacuum integrals**

Liu, YQM, 2201.11637

$$I_{\vec{\nu}}(D, m^2) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\mathcal{D}_{K+1}^{-\nu_{K+1}} \cdots \mathcal{D}_N^{-\nu_N}}{(\mathcal{D}_1 + \mathrm{i}0^+)^{\nu_1} \cdots (\mathcal{D}_K + \mathrm{i}0^+)^{\nu_K}}$$

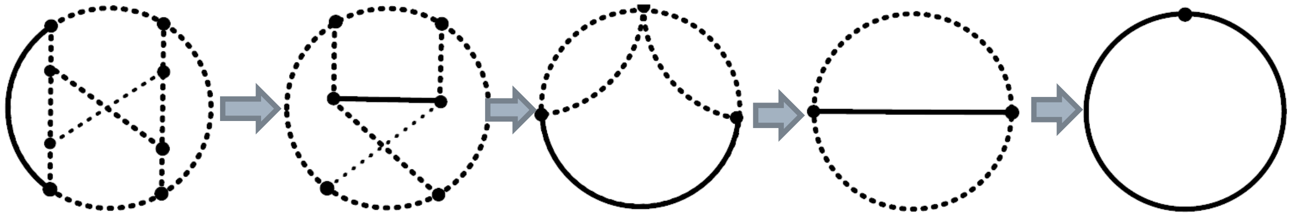
$$\mathcal{D}_1 = \ell_1^2 - m^2 + \mathrm{i}0^+$$



$$\widehat{I}_{\vec{\nu}'}(\ell_1^2) = \int \left(\prod_{i=2}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \right) \frac{\mathcal{D}_{K+1}^{-\nu_{K+1}} \cdots \mathcal{D}_N^{-\nu_N}}{\mathcal{D}_2^{\nu_2} \cdots \mathcal{D}_K^{\nu_K}}$$

$$I_{\vec{\nu}} = \int \frac{\mathrm{d}^D \ell_1}{\mathrm{i}\pi^{D/2}} \frac{(-\ell_1^2)^{\frac{(L-1)D}{2} - \nu + \nu_1}}{(\ell_1^2 - 1 + \mathrm{i}0^+)^{\nu_1}} \widehat{I}_{\vec{\nu}'}(-1) = \frac{\Gamma(\nu - LD/2)\Gamma(LD/2 - \nu + \nu_1)}{(-1)^{\nu_1}\Gamma(\nu_1)\Gamma(D/2)} \widehat{I}_{\vec{\nu}'}(-1)$$

- L-loop vacuum integrals expressed by $(L-1)$-loop p-integrals
- Using AMFLow: L-loop vacuum integrals reduced to $(L-1)$-loop vacuum integrals



Zero input; valid for any loop, any space-time dimension

AMFlow: Package

➢ Download

Liu, YQM, 2201.11669

Link: https://gitlab.com/multiloop-pku/amflow

Name	Last commit	Last update
diffeq_solver	update	5 months ago
examples	update	3 months ago
ibp_interface	fix_a_bug_for_mpi_version	1 week ago
AMFlow.m	fix mass mode	2 months ago
CHANGELOG.md	update changelog	1 week ago
FAQ.md	update	6 months ago
LICENSE.md	test	7 months ago
README.md	update	3 months ago
options_summary	update	3 months ago

➢ Feature

- **The first package that can calculate any FI (with any number of loops, any D and $\vec{s}$) to arbitrary precision,** ***given sufficient resource***

Why needs Blade: efficiency

➢ Difficulties of IBP reduction based on Laporta's algorithm

- Complicated intermediate expression
- Many auxiliary integrals, waste of resource

Air, Anastasiou, Lazopoulos, 0404258,
Fire , Smirnov, et al, 0807.3243, 1302.5885, 1408.2372,1901.07808
Reduze, Manteuffel, Studerus, 0912.2546, 1201.4330
LiteRed, Lee, 1212.2685, 1310.1145
Kira, Maierhöfer, et al, 1705.05610, 1812.01491, 2008.06494

➢ Some improvements

- Finite field: avoid intermediate expression swell

Manteuffel, Schabinger, 1406.4513
FiniteFlow, Peraro, 1905.08019
FireFly, Klappert, Lange, 1904.00009, 2004.01463
Reconstruction.m, Belitsky, Smirnov, Yakovlev, 2303.02511

- Syzygy equations: trimming IBP system

Gluza, Kajda, Kosower, 1009.0472; Larsen, et. al., 1511.01071, …
NeatIBP, Wu, et al. 2305.08783

- Block-triangular form: very small IBP system, usually improve the efficiency by 1-2 orders

Liu, YQM, 1801.10523, Guan, Liu, YQM, 1912.09294

- Better choices of basis: UT basis / D-factorized

Henn, 1304.1806; Abreu, et al., 1812.04586 ;
Smirnov, Smirnov, 2002.08042; Usovitsch, 2002.08173

➢ Ways to bypass IBP

- $1/D$ expansion;

Baikov, Chetyrkin, Kuhn, 0108197, …

$1/\eta$ expansion;

Liu, YQM, 1801.10523, …

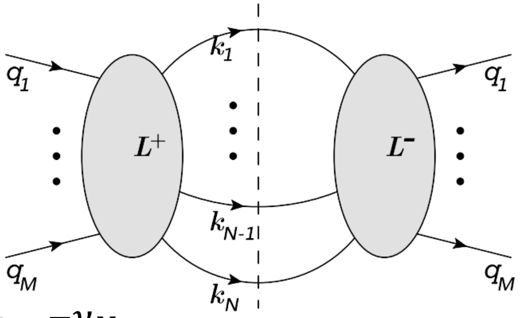
Intersection theory;

Frellesvig, et. al., 1901.11510, …

…

Why needs Blade: new features

- **Symmetries within amplitudes (or its c.c.)**



- **General integrand**

$$I(\vec{v}) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \mathcal{F} \frac{\mathcal{D}_{K+1}{}^{-\nu_{K+1}} \dots \mathcal{D}_N{}^{-\nu_N}}{(\mathcal{D}_1 + \mathrm{i}0)^{\nu_1} \dots (\mathcal{D}_K + \mathrm{i}0)^{\nu_K}}$$

$$\int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \left(\frac{\partial q^\mu}{\partial \ell_j^\mu} + q^\mu \frac{1}{\mathcal{F}} \frac{\partial \mathcal{F}}{\partial \ell_j^\mu} + q^\mu \sum_{m=1}^{N} \frac{-\nu_m}{D_m} \frac{\partial D_m}{\partial \ell_j^\mu} \right) \mathcal{F} \frac{\mathcal{D}_{K+1}{}^{-\nu_{K+1}} \dots \mathcal{D}_N{}^{-\nu_N}}{(\mathcal{D}_1 + \mathrm{i}0)^{\nu_1} \dots (\mathcal{D}_K + \mathrm{i}0)^{\nu_K}} = 0$$

  - Useful if $\frac{1}{\mathcal{F}} \frac{\partial \mathcal{F}}{\partial \ell_j^\mu}$ can be expressed as linear combinations of $\prod_{i=1}^{L} \mathcal{D}_i^{a_i}$
  - E.g.: $\mathcal{F} = \mathcal{D}_i^{a_i}$, $\mathcal{F} = e^{\sum x_i \mathcal{D}_i}$ See also Johann Usovitsch's talk

- **Complex variables** $m_1^2 = 1 + \mathrm{i}\,, s = 3 + 2\mathrm{i}$

New features of Blade

Outline

I. Introduction

II. Block-triangular form and Blade

III. Applications of Blade

V. Summary and outlook

The idea of block-triangular form

➢ Difficulty

- Reduction coefficients: very complicated rational functions, hard to reconstruct

➢ Why?

- Each relation involves minimal #integrals (a target integral and MIs), (spurious) poles of the target integrals must explicitly appear in the denominator, thus also complicated numerators

➢ Solution: find good relations among a larger set of FIs

$$
\begin{aligned}
Q_{11}\, I_1 + Q_{12}\, I_2 + Q_{13}\, I_3 + Q_{14}\, I_4 + \ldots + Q_{1N}\, I_N &= 0\\
Q_{21}\, I_1 + Q_{22}\, I_2 + Q_{23}\, I_3 + Q_{24}\, I_4 + \ldots + Q_{2N}\, I_N &= 0\\
Q_{33}\, I_3 + Q_{34}\, I_4 + \ldots + Q_{3N}\, I_N &= 0\\
Q_{43}\, I_3 + Q_{44}\, I_4 + \ldots + Q_{4N}\, I_N &= 0\\
\ldots
\end{aligned}
$$

- Recover complicated poles by the determinant of a simple matrix!
- Relations to be simple: easy to find, fast to use
- Relations to be efficient: in block-(upper)-triangular form; when solving the relations: no forward elimination, only backward substitution

How to find relations?

- **Decomposition of $Q_i(\vec{s},\epsilon)$** $\sum Q_i(\vec{s},\epsilon)\, I_i(\vec{s},\epsilon) = 0$

$$Q_i(\vec{s},\epsilon) = \sum_{\mu_0=0}^{\epsilon_{max}} \sum_{\mu} \tilde{Q}_i^{\mu_0\mu_1\ldots\mu_r} \epsilon^{\mu_0} s_1^{\mu_1} \ldots s_r^{\mu_r}$$

- $\tilde{Q}_i^{\mu_0\mu_1\ldots\mu_r}$ are unknowns
- $\mu_1 + \cdots + \mu_r = d_i$

- **Input from numerical IBP** $I_i(\vec{s}_0,\epsilon_0) = \sum_{j=1}^{n} C_{ij}(\vec{s}_0,\epsilon_0) M_j(\vec{s}_0,\epsilon_0)$

$$\Rightarrow \sum_{\mu_0,\mu} \sum_{j=1}^{n} \tilde{Q}_i^{\mu_0\ldots\mu_r} \epsilon_0^{\mu_0} s_{1,0}^{\mu_1} \ldots s_{r,0}^{\mu_r} C_{ij}(\vec{s}_0,\epsilon_0) M_j(\vec{s}_0,\epsilon_0) = 0$$

- $\vec{s}_0, \epsilon_0$ and $C_{ij}(\vec{s}_0,\epsilon_0)$ are numbers under finite field

- **Linear equations:** $\sum_{\mu_0,\mu} \tilde{Q}_i^{\mu_0\ldots\mu_r} \epsilon_0^{\mu_0} s_{1,0}^{\mu_1} \ldots s_{r,0}^{\mu_r} C_{ij}(\vec{s}_0,\epsilon_0) = 0$

  - With enough input: determine $\tilde{Q}_i^{\mu_0\ldots\mu_r}$

How to find good relations?

- **With $G = G_1 \cup G_2$, satisfy**
  - G_1 is more complicated than G_2
  - G_1 can be reduced to G_2

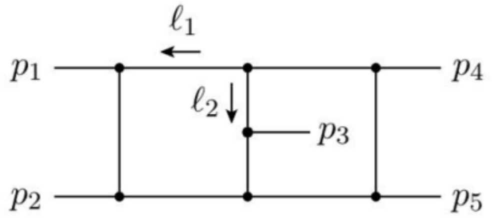
E.g.: $G = \{I(1,1,1,1,-3,0), I(1,1,1,1,0,-3), \ldots, I(1,1,1,1,0,0),$
$I(0,1,1,1,-2,0), \ldots, I(1,1,1,0,0,-2), \ldots, I(1,1,1,0,0,0),$
$I(0,0,1,1,-1,0), \ldots,$
$I(0,0,0,1,0,0), \ldots\}$

- **Search Algorithm**
  1. Set degree bound
  2. Search relations among G
  3. If obtained relations are enough to determine G_1 by G_2, stop;
     else, increase degree bound and go to step 2

Example of block-triangular form

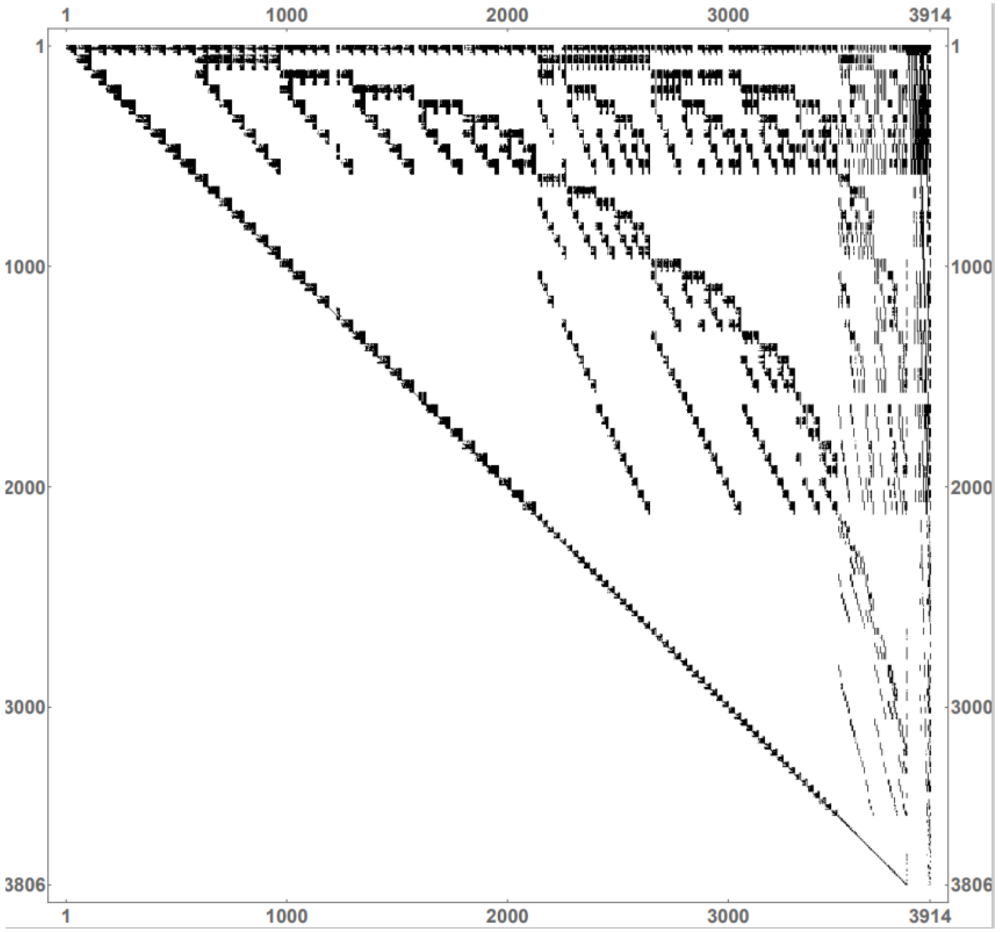
➢ Massless double pentagon

- $\epsilon, s_{12}, s_{23}, s_{34}, s_{45}, s_{51}$
- Degree-5 numerators



➢ Comparison

- IBP system
  - $\sim 3 * 10^5$ relations
  - Numeric sampling under finite field : 7s per phase point
- Block-triangular form
  - 3806 relations to reduce 3914 integrals to 108 MIs
  - 0.17s per phase point



➢ Framework

Integral extension

- Easy to search
- Delicate scheme
- User-defined scheme
- …

Determine master integrals (maximal-cut)

→ **Generate IBP**

- Automatic seeding
- Trim seed integrals
- …

→ **Adaptive search strategy**

- Delicate scheme
- Flexible polynomial ansatz
- …

→ **Reconstruction**

Using FiniteFlow

Usage of Blade

➢ Download and install

Link: https://gitlab.com/multiloop-pku/blade

B

Blade

Project ID: 42197761

75 Commits **1** Branch **0** Tags **2.4 MB** Project Storage

Block-triangular form improved Feynman integral decomposition .

```
chmod +x auto_install
./auto_install
```

➢ Examples

- 1_automatic - introduction to automatic reduction of Feynman loop integrals;
- 1_preferred_masters - introduction to automatic reduction with user-defined master integrals;
- 1_userdefined_target - introduction to automatic reduction of user defined target integrals;

...

Usage of Blade

➢ Basic usage

```
family = dbox;
dimension = 4-2*eps;
loop = {l1,l2};
leg = {p1,p2,p3,p4};
conservation = {p4->-p1-p2-p3};
replacement = {p1^2 -> 0, p2^2 -> 0, p3^2 -> msq, p1 p2 -> s/2, p1 p3 -> t/2, p2 p3 -> (-s-t)/2};
propagator = {(l1)^2,(l1+p1)^2,(l1+p1+p2)^2,l2^2-msq, (l1+l2)^2-msq, (l2-p1-p2)^2-msq, (l2-p3)^2, (l1-p3)^2,(l2+p1)^2};
topsector = {1,1,1,1,1,1,1,0,0};
numeric = {s -> 1};
```

```
BLFamilyDefine[family,dimension, propagator,loop,leg,conservation, replacement,topsector,numeric]
```

```
target={BL[dbox,{1,1,1,1,1,1,1,0,-3}],BL[dbox,{1,1,1,1,1,1,1,-1,-2}],
BL[dbox,{1,1,1,1,1,1,1,-2,-1}],BL[dbox,{1,1,1,1,1,1,1,-3,0}],
BL[dbox,{2,1,1,1,1,1,0,0,-2}], BL[dbox,{2,1,1,1,1,1,0,-2,0}],
BL[dbox,{0,1,2,2,1,1,0,0,-2}], BL[dbox,{0,1,2,2,1,1,0,-2,0}]};
```

```
res = BLReduce[target];
```

```
de = BLDifferentialEquation[{t}]
```

- Define the integral family using "BLFamilyDefine"
- Reduce target integrals using "BLReduce"
- Construct the differential equations using "BLDifferentialEquation"

Other improvements

➢ Divide to sub-families

- Reduce target integrals of different structure separately

  e.g. $I(1,1,1,1,-5,0)$ and $I(2,2,1,1,0,0)$

➢ Trim seed integrals

- The rank of seed integrals of sub-sectors can be smaller than that of top-sector, see “FilterLevel”

➢ Simple of Symmetry relations

$$D_a \to D'_a = \sum_b A_{ab} D_b$$

$$\int \mathrm{d}\mu_L \frac{(D_{k+1}^{(l)})^{-\nu_{k+1}} \dots (D_N^{(l)})^{-\nu_N}}{D_1^{\nu_1} \dots D_k^{\nu_k}} = \int \mathrm{d}\mu_L \frac{(D_{k+1}^{(r)})^{-\nu_{k+1}} \dots (D_N^{(r)})^{-\nu_N}}{D_{\sigma(1)}^{\nu_{\sigma(1)}} \dots D_{\sigma(k)}^{\nu_{\sigma(k)}}}$$

- Choose $\mathcal{D}_a^{(l)}$ and $\mathcal{D}_a^{(r)}$ to find an equation with the minimal number of terms

Outline

I. Introduction

II. Block-triangular form and Blade

III. Applications of Blade

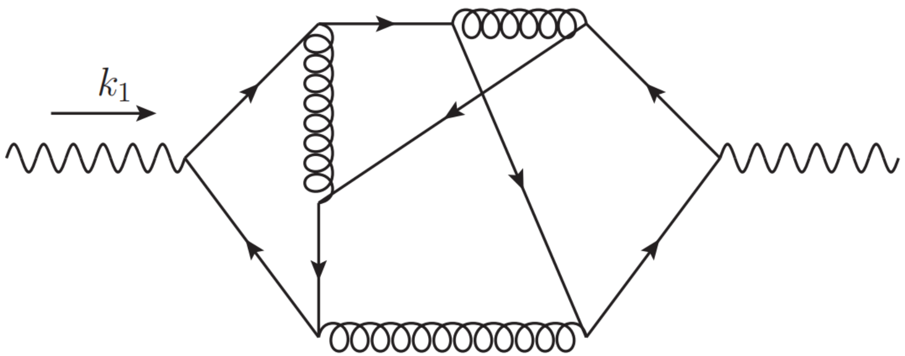
V. Summary and outlook

Example 1： 4-loop 2-point one massive internal line

- $e^+ e^- \to \gamma^*/Z^* \to t\,\bar{t}$ @ N^3LO_{QCD}

  Chen, Guan, He, Liu, YQM, 2209.14259

  - Two variables: $\epsilon\,, m_t^2. (k_1^2 \to 1)$



369 MIs

- Improvement for IBP: 2 orders

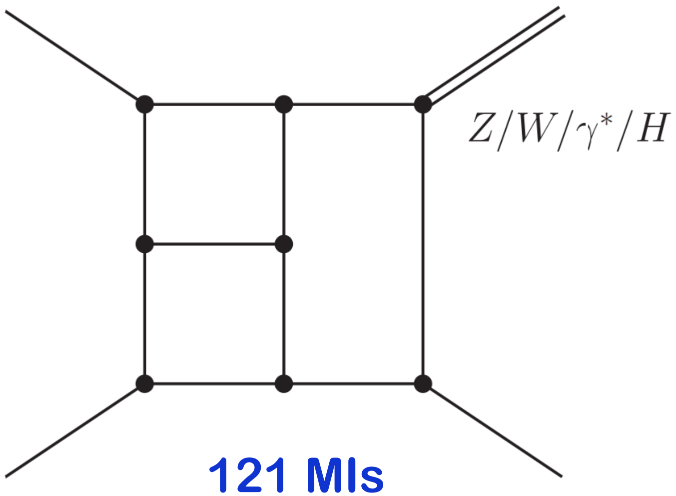
s_{max}	t_{IBP}	t_{BL}	Probes for search per prime	Probes for reconstruction per prime	#primes	CPU·h
4	439s	1.1s	64	6041	8	280

- s_{max} denotes the maximal degree of numerators

Example 2: 3-loop 4-point one massive external line

- $pp \to V + j$ @N^3LO, $pp \to H + j$ @N^3LO_{HTL}
  - Three variables: ϵ, m^2, t $(s \to 1)$
  - Leading color approximation

  Gehrmann, Jakubčík, Mella et al. 2307.15405



121 MIs

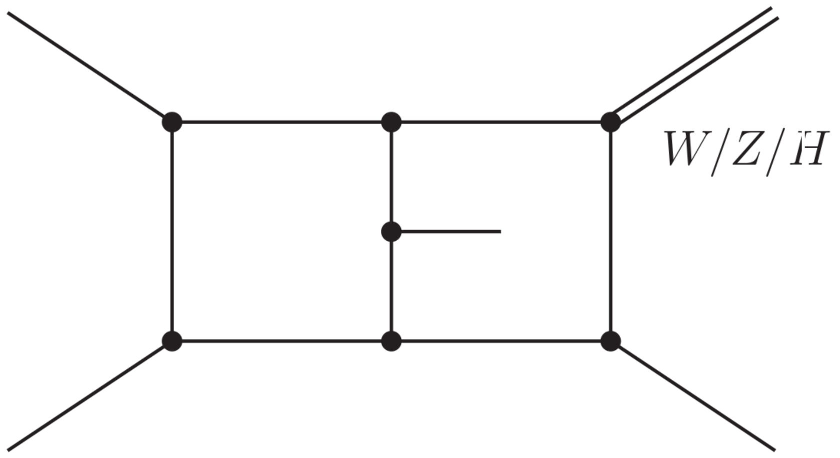
- **Improvement for IBP: more than 2 orders**

s_{max}	t_{IBP}	t_{BL}	Probes for search per prime	Probes for reconstruction per prime	#primes	CPU·h
3	44s	0.07s	150	42000	4	60
4	170s	0.23s	200	68442	4	180

Example 3： 2-loop 5-point one massive external line

- $pp \to Wjj,\ pp \to Hb\bar{b}, \ldots @N^2LO_{QCD}$
  - **Six variables:** $\epsilon, m^2, s_{23}, s_{34}, s_{45}, s_{51}, (s_{12} \to 1)$

    Abreu, Chicherin, Ita, et al. 2306.15431
    Hartanto, Poncelet, Popescu, et al. 2205.01687
    Abreu, Febres Cordero, Ita, et al, 2110.07541
    Badger, Hartanto, Kryś et al, 2107.14733, 2201.04075
    Badger, Hartanto, Zoia, 2102.02516



142 MIs

- **Improvement for IBP: 2 orders**
  - **Trick:** $\epsilon = \epsilon_0$

s_{max}	t_{IBP}	t_{BL}	Probes for search per prime	Probes for reconstruction per prime
5	12.4s	0.16s	1000	10^6?

Example 4: Generating Functions(GFs)

Guan, Li, YQM, 2306.02927

- $\mathcal{F} = e^{\sum_{i=K+1}^{N} x_i \mathcal{D}_i}$

$$G(\{\eta_i\},\{x_i\}) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\exp\{\sum_{i=K+1}^{N} x_i \mathcal{D}_i\}}{(\mathcal{D}_1-\eta_1)\cdots(\mathcal{D}_K-\eta_K)}$$

- **Analytic functions of η_i and x_i**
- **Derivatives w.r.t η_i and x_i give integrals with high dots and rank**
- **Original Feynman integrals obtained by:**

$$I(\vec{\nu}) = \lim_{\vec{\eta}\to 0} \lim_{\vec{x}\to 0} \partial_{\eta}^{\vec{\nu}} \partial_{x}^{\vec{\nu}} G(\{\eta_i\},\{x_i\})$$

$$\partial_{\eta}^{\vec{\nu}} \equiv \prod_{i=1}^{K} \frac{1}{(\nu_i-1)!} \frac{\partial^{\nu_i-1}}{\partial {\eta_i}^{\nu_i-1}},$$

$$\partial_{x}^{\vec{\nu}} \equiv \prod_{i=K+1}^{N} \frac{\partial^{\nu_i}}{\partial {x_i}^{\nu_i}}.$$

- **$\eta_i \to 0$ and $x_i \to 0$ means to pick the Taylor branch**

Example 4: Complete reduction rules

➢ Differential equations of GFs

$$\frac{\partial}{\partial \eta_i} \vec{G} = A_{\eta_i} \vec{G} \qquad \frac{\partial}{\partial x_j} \vec{G} = A_{x_j} \vec{G}$$

- High order derivates w.r.t η_i and x_j can be easily obtained by solving Des
- Solution: reduction relations for original FIs

➢ Complete reduction for general powers $\vec{\nu}$

- 1) Setup DEs of GFs using Blade;
- 2) Solve DEs to obtain reduction of integrals with arbitrary power
- Reduce infinite number of Feynman integrals within reasonable computational complexity

Summary and outlook

➢ Summary

- The block-triangular form is a way to improve the efficiency of IBP reduction
- Blade is a fully automated package for FIs reduction, armed with the method of block-triangular form
- Blade has many distinctive features, making it applicable in many general cases

➢ Improvements in future

- Combine syzygy equations (NeatIBP)
- Reduction with floating numbers?

Thank you!

Construction of blocks

➢ Integral extension

$$Q_{11}\,I_1 + Q_{12}\,I_2 + Q_{13}\,I_3 + Q_{14}\,I_4 + \ \dots + Q_{1N}\,I_N = 0$$
$$Q_{21}\,I_1 + Q_{22}\,I_2 + Q_{23}\,I_3 + Q_{24}\,I_4 + \ \dots + Q_{2N}\,I_N = 0$$
$$Q_{33}\,I_3 + Q_{34}\,I_4 + \ \dots + Q_{3N}\,I_N = 0$$
$$Q_{43}\,I_3 + Q_{44}\,I_4 + \ \dots + Q_{4N}\,I_N = 0$$
$$\dots$$

- A proper set of integrals $\Rightarrow$ simple relations, easy to search
- Basic construction

$$3^{\ominus} I(1,1,1,1,0,0) = \{I(1,1,1,1,-3,0), I(1,1,1,1,0,-3), \dots, I(1,1,1,1,0,0),$$
$$I(0,1,1,1,-2,0), \dots, I(1,1,1,0,0,-2), \dots, I(1,1,1,0,0,0),$$
$$I(0,0,1,1,-1,0), \dots,$$
$$I(0,0,0,1,0,0), \dots\}$$

- More schemes can be explored

➢ Constructing G_1 and G_2 for each block

- G_1: target integrals from some sector
- G_2: integrals in its subsectors

Adaptive search strategy

- ***Too many unknowns?***
- **Semi-analytic**

$$Q_i(\vec{z}) = \sum_{\mu} \tilde{Q}_i^{\mu_1 \dots \mu_r} z_1^{\mu_1} \dots z_r^{\mu_r}$$

  - Keep a subset of variables analytic ⇒ easy to search
  - The integral set is the same ⇒ still very efficient
  - More than one block-triangular form is needed

$$Q_i(z_{1,0}, \dots z_{r-1,0}, z_r) = \sum_{\mu_r} \tilde{Q}_i^{\mu_0} z_r^{\mu_r}$$

$$Q_i(z_{1,0}, \dots z_{r-2,0}, z_{r-1}, z_r) = \sum_{\mu_{r-1},\mu_r} \tilde{Q}_i^{\mu_{r-1}\mu_r} z_{r-1}^{\mu_{r-1}} z_r^{\mu_r}$$

… …

- **Adaptive search**
  1. $n = 1$
  2. Search n-variable block-triangular form within time limit T
  3. If search succeed, $n++$ and go to step 2, otherwise go to step 4
  4. Perform reduction by solving the most efficient linear system(i-variable)
- **Full potential of block-triangular form**