A Tale of Two Packages

pfd-parallel and *NeatIBP*

for multi-loop computations

Yang Zhang

University of Science and Technology of China

Feb 14, 2023

Mathematical Structures in Feynman Integrals

Apologize

I could not attend this exciting conference …

Although the Covid travel restriction in China was lift last December, I could not get a German visa on time …

Apologize for giving an online talk,

and thank the conference organizers a lot for giving me this opportunity!

A tale of two packages

photo from USPS website

pfd-parallel

simplifies rational functions
in amplitudes computations,
by multivariate partial fraction

NeatIBP

generates small size
IBP systems based on
syzygy/module intersection

pfd-parallel

A package simplifies rational functions
in amplitudes computations
by multivariate partial fraction

Bendle, Boehm, Heymann, Ma, Rahn, Ristau, Wittmann, Wu, Xu and YZ

https://github.com/singular-gpispace/pfd-parallel/

arXiv: 2104.06866
package finished on Oct. 2022
to be submitted to Comp. Phys. Comm.

Why do we need multivariate partial fraction?

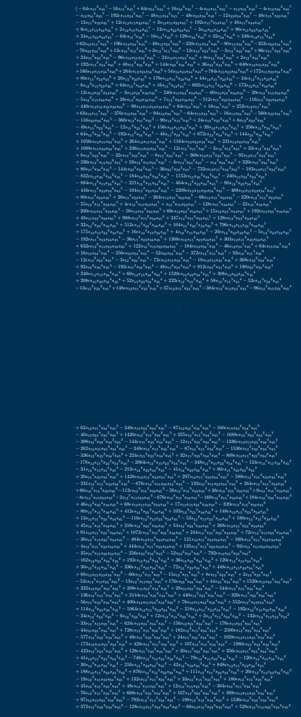
It is well known that sometimes in amplitudes computations,
multivariate partial fraction can greatly simplify rational functions.

simplify coefficients in Amplitudes

simplify IBP reduction coefficients

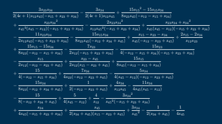
boost canonical DE construction

Meyer, CPC 222 (2018)
Abreu, Dormans, Febres Cordero, Ita, Page, Sotnikov, JHEP 05 (2019) 08
Agarwal, Buccioni, von Manteuffel, Tancred, PRL 127 (26) (2021) 26200
… …



5 page long expression

=



1/3 page expression

An example from IBP reduction coefficient in
Boehm, Wittmann, Wu, Xu, YZ,
JHEP 12 (2020) 054

pfd-parallel

Massively-Parallel
Partial Fraction Decomposition

a massively parallel framework for partial fraction decomposition of rational functions
based on the Singular/GPI-Space framework

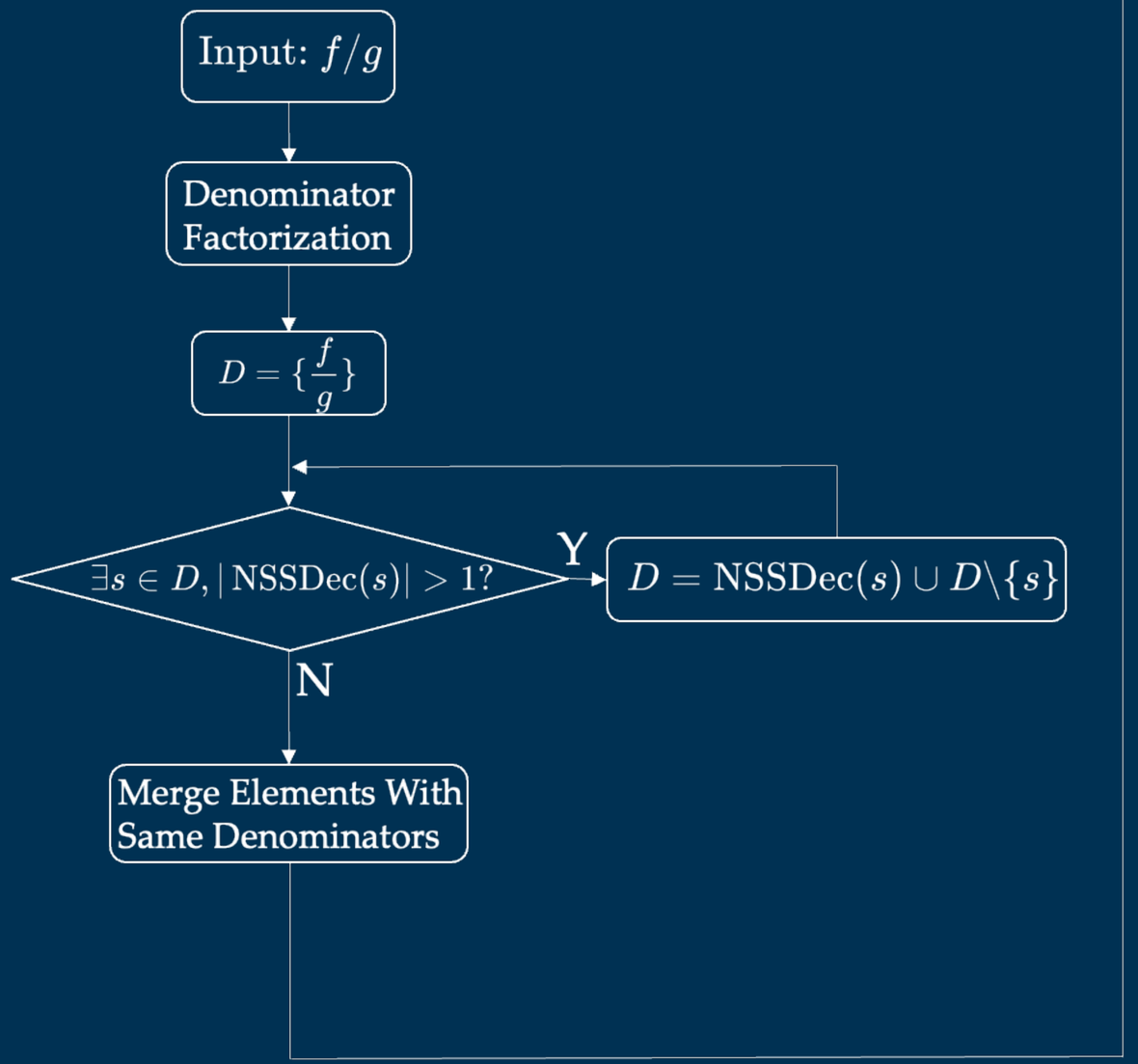
Improved Leintars algorithm

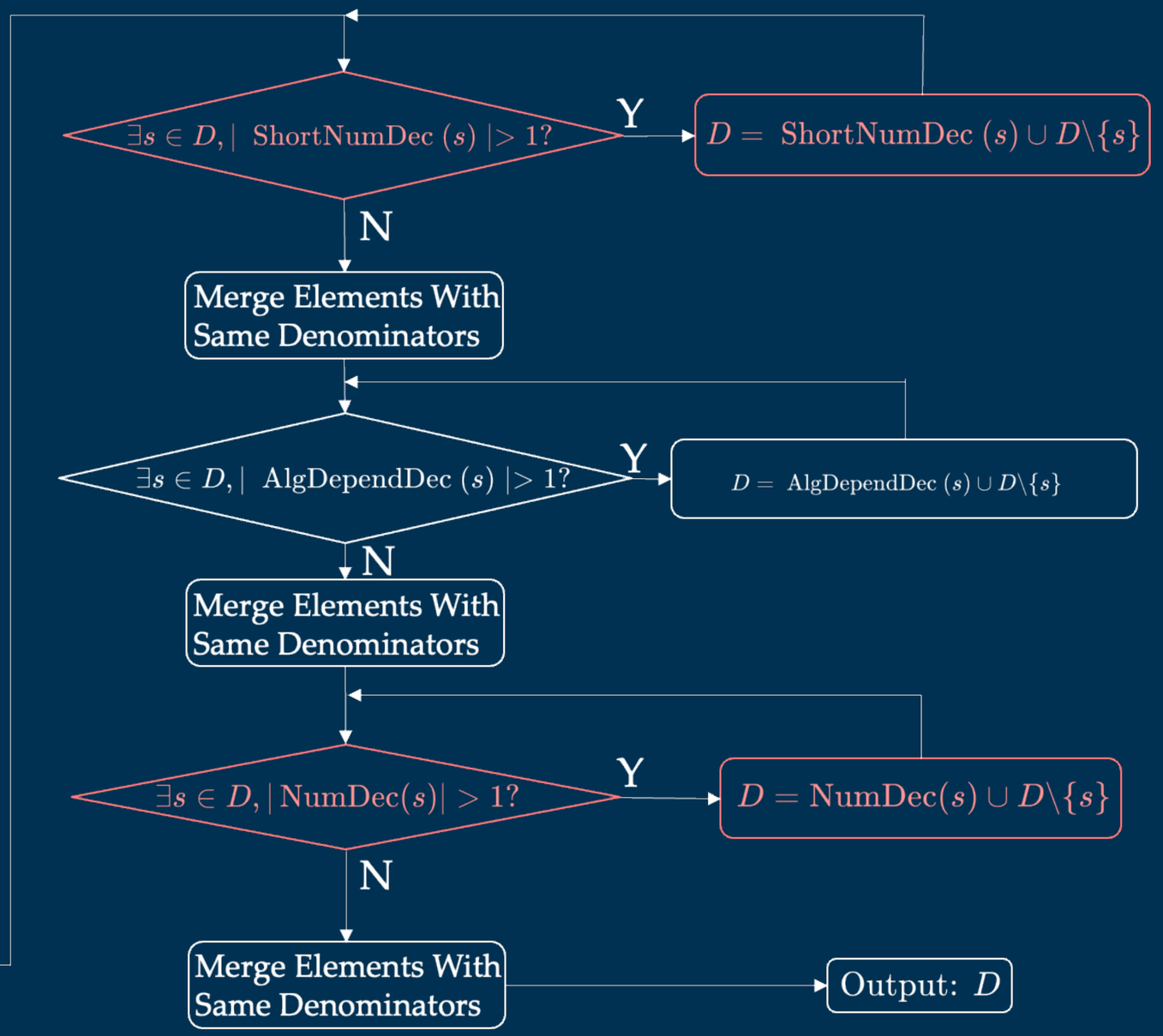
Boehm, Wittmann, Wu, Xu, and YZ, JHEP 12 (2020) 054

Heller and von Manteuffel: CPC. 271 (2022) 108174.

MultivariateApart algorithm

Improved Leintars' algorithm

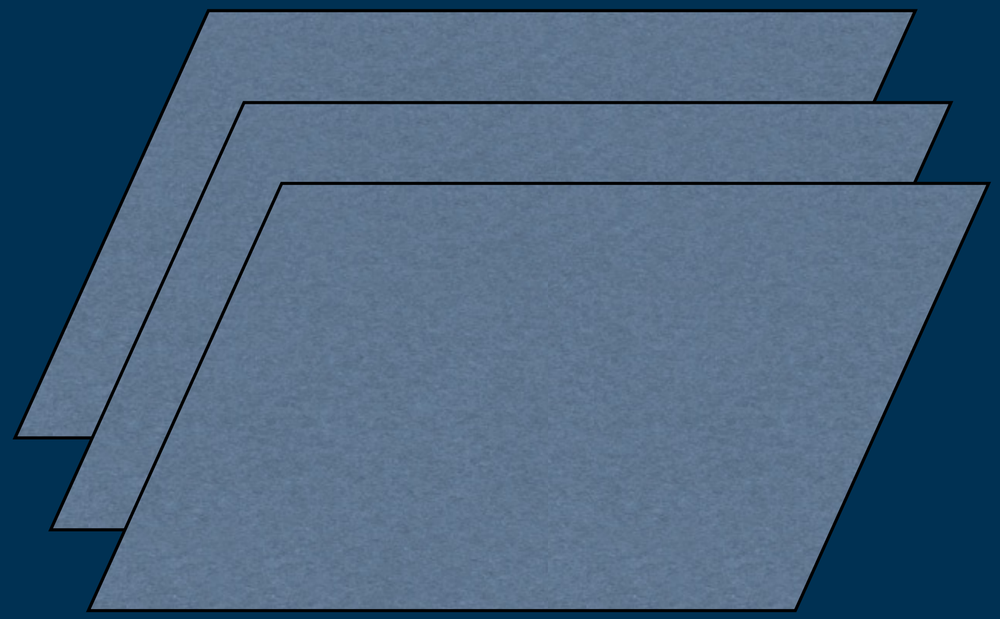




Boehm, Wittmann, Wu, Xu, and YZ, JHEP 12 (2020) 054

pfd-parallel

Massively-Parallel
Partial Fraction Decomposition



Partial Fraction Decomposition can be a heavy computation,
so parallelization is important

We implement several layers of parallelization

parallelization between several rational functions
parallelization between terms in one rational function

complicated workflow
so we implement the
workflow manager
GPI-space

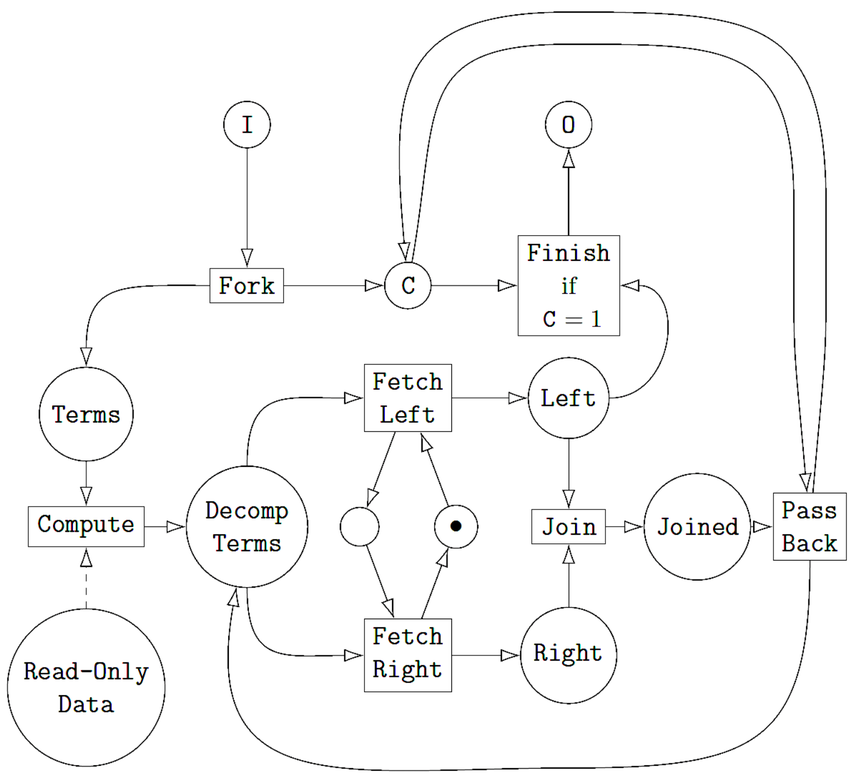
pfd-parallel

Massively-Parallel
Partial Fraction Decomposition

GPI-Space, a workflow management system,
created by the Competence Center High Performance Computing of Fraunhofer ITWM.

Combine Singular with GPI-Space
to create a way of performing massively parallel computations in computer algebra.

https://www.mathematik.uni-kl.de/~boehm/singulargpispace/



For details,
refer to Janko Boehm's talk and his website

Petri-Net for the parallization over
the decomposition of terms

pfd-parallel Example

Simplifying coefficients in multi-loop scattering amplitudes

Two-loop leading color $pp \to W^{\pm} + \gamma + j$ from Badger, Hartanto, Krys, Zoia
JHEP 05 (2022) 035

6 kinematic variables

pfd-parallel

1096 MB → 9 MB

with MultivariateApart algorithm
implemented in GPI-space/Singular

Running on a node with 48 CPUs, ~ 5.5 hours

NeatIBP

A package generates small size
IBP systems based on
syzygy/module intersection

Wu, Boehm, Ma, Xu and YZ

https://github.com/yzhphy/NeatIBP

arXiv: 2303. ****
to be submitted to Comp. Phys. Comm.

It is well know that IBP reduction could be
the slowest step for an amplitude computation

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{\partial}{\partial l_i^\mu} \frac{v_i^\mu}{D_1^{\alpha_1} \dots D_k^{\alpha_k}} = 0 \qquad \textit{Chetyrkin, Tkachov (1981)}$$

Standard Laporta algorithm may provide a system containing
too many integrals other than target or master integrals.

Gaussian elimination with symbolic parameter is
extremely difficult

Many way to improve the situation …

other than the standard Laporta algorithm

Non-commutative algebra to avoid Gaussian elimination by a reduction over the integral index operators

Smirnov, Smirnov , JHEP01(2006)001
…

Syzygy to decrease the number of integrals in an IBP system

Gluza, Kajda, Kosower, PRD. 83.045012
Schabinger, JHEP 01 (2012) 077
Larsen YZ, PRD 93 (2016) 4, 041701 …

For more details about syzygy IBP, refer to Bakul's talk

Intersection theory to map the space of Feynman integrals to a twisted cohomology group

Mastrolia, Mizera JHEP 02 (2019) 139
Frellesvig, Gasparotto, Laporta, Mandal, Mastrolia, Mizera JHEP 05 (2019) 153
Frellesvig, Gasparotto, Mandal, Mastrolia, Mattiazzi, Mizera PRL. 123.201602
…

Auxiliary mass flow to find integrals relations based on the eta expansion

Liu, Ma Phys. Rev. D 99 (2019), no 7 071501
Guan, Liu, Ma, Chin. Phys. C 44(2020) 9, 093106
…

other than the standard Laporta algorithm

Non-commutative algebra to avoid Gaussian elimination by a reduction over the integral index operators

Smirnov, Smirnov , JHEP01(2006)001
…

Syzygy to decrease the number of integrals in an IBP system

Gluza, Kajda, Kosower, PRD. 83.045012
Schabinger, JHEP 01 (2012) 077
Larsen YZ, PRD 93 (2016) 4, 041701 …

For more details about syzygy IBP, refer to Bakul's talk

Intersection theory to map the space of Feynman integrals to a twisted cohomology group

Mastrolia, Mizera JHEP 02 (2019) 139
Frellesvig, Gasparotto, Laporta, Mandal, Mastrolia, Mizera JHEP 05 (2019) 153
Frellesvig, Gasparotto, Mandal, Mastrolia, Mattiazzi, Mizera PRL. 123.201602
…

Auxiliary mass flow to find integrals relations based on the eta expansion

Liu, Ma Phys. Rev. D 99 (2019), no 7 071501
Guan, Liu, Ma, Chin. Phys. C 44(2020) 9, 093106
…

Syzygy for IBP reduction

smart vector choice

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{\partial}{\partial l_i^\mu} \frac{v_i^\mu}{D_1^{\alpha_1} \dots D_k^{\alpha_k}} = 0$$

Suppose that we want to forbid the increase of the propagator index α_i, we can require that,

$$\Big(\sum_{j=1}^{L} v_j^\mu \frac{\partial}{\partial l_j^\mu} D_i\Big) + g_i D_i = 0$$

Syzygy equation

Gluza, Kajda, Kosower, PhysRevD. 83.045012

where both v_j^μ and g_i contain polynomials in loop momenta.

ways to apply syzygy method

Syzygy from Linear Algebra, Schabinger, *JHEP 01 (2012) 077*

Module Intersection, Larsen,YZ, *Phys.Rev.D* 93 (2016) 4, 041701

Dual Conformal Symmetry, Bern, Enciso, Ita, Zeng, Phys. Rev. D 96, 096017 (2017)

… …

Syzygy for IBP reduction

smart vector choice

$$\int \frac{d^D l_1}{i\pi^{D/2}} \cdots \int \frac{d^D l_L}{i\pi^{D/2}} \frac{\partial}{\partial l_i^\mu} \frac{v_i^\mu}{D_1^{\alpha_1} \dots D_k^{\alpha_k}} = 0$$

Suppose that we want to forbid the increase of the propagator index α_i, we can require that,

$$\left(\sum_{j=1}^{L} v_j^\mu \frac{\partial}{\partial l_j^\mu} D_i\right) + g_i D_i = 0$$

Syzygy equation

Gluza, Kajda, Kosower, PhysRevD. 83.045012

where both v_j^μ and g_i contain polynomials in loop momenta.

ways to apply syzygy method

Syzygy from Linear Algebra, Schabinger, *JHEP 01 (2012) 077*

Module Intersection, Larsen,YZ, *Phys.Rev.D* 93 (2016) 4, 041701

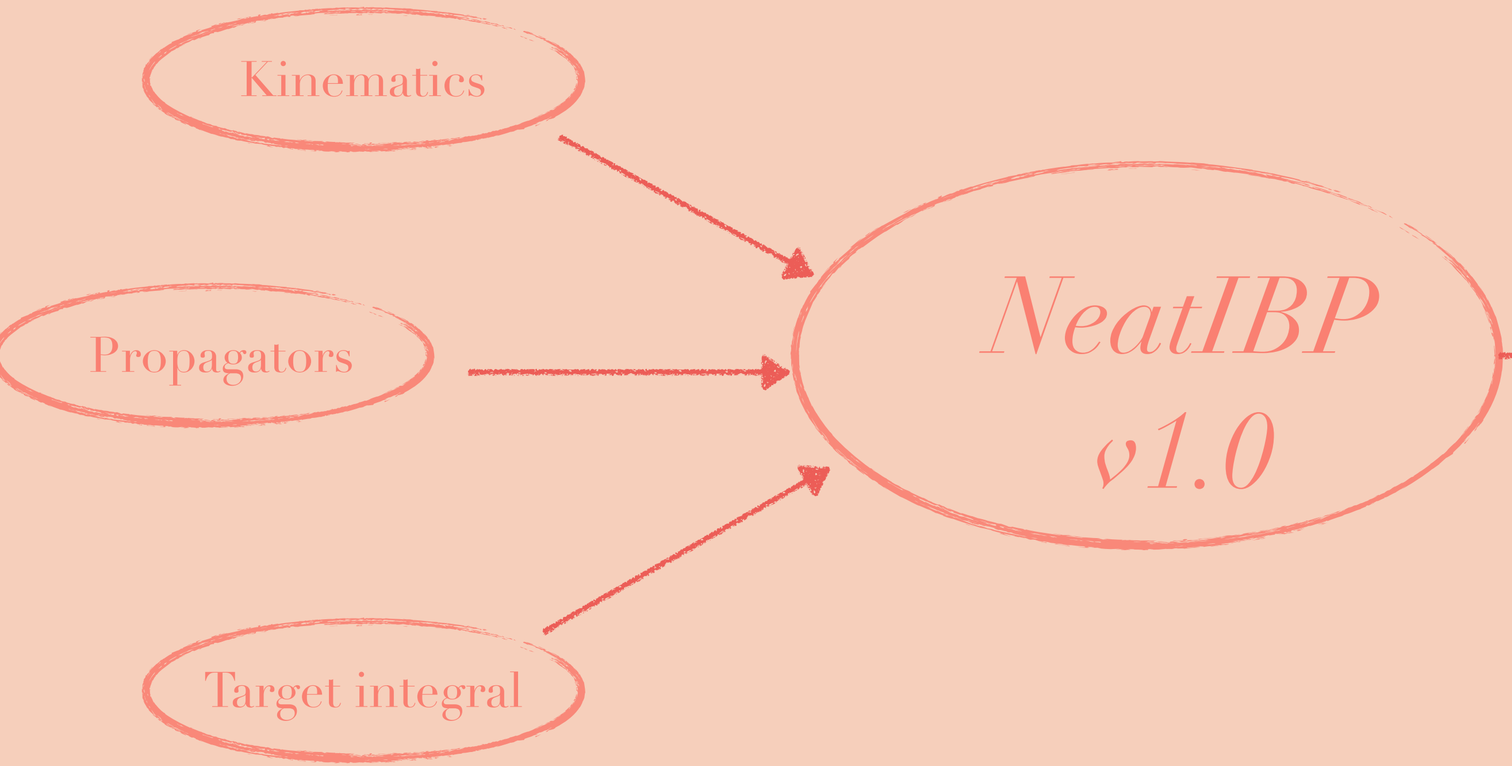
Dual Conformal Symmetry, Bern, Enciso, Ita, Zeng, Phys. Rev. D 96, 096017 (2017)

… …

NeatIBP v1.0

an automatic package to generate small-size IBP
based on syzygy/module intersection

Boehm, Ma, Wu, Xu and YZ



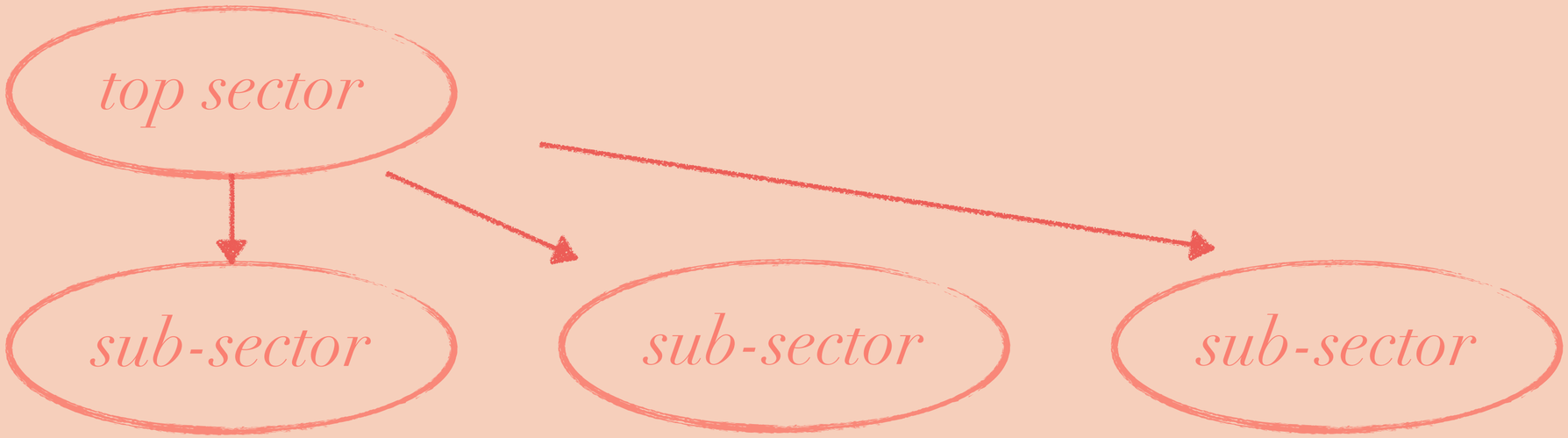
small size IBP relations with
restricted propagator indices
which is enough to reduce
target to master integrals

https://github.com/yzhphy/NeatIBP

development version
welcome to download and try

NeatIBP v1.0

an automatic package to generate small-size IBP
based on syzygy/module intersection



sector-by-sector
module intersection

generate IBPs with propagator indices restricted
finite-field method to select useful and independent IBP relations

Determine the master integrals and accumulate IBP
relations in the same time

NeatIBP v1.0 Implementation notes

Mathematica/Singular codes

module intersection is done
with the Schreyer's algorithm in Singular

Parallelization

is over different sectors

Finite Field Computations

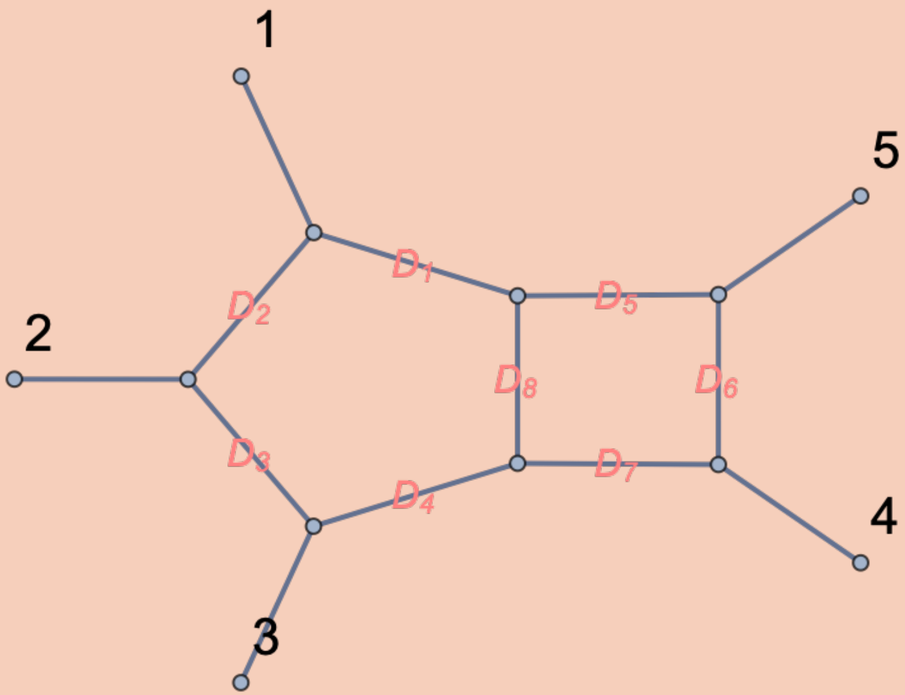
to select useful and independent IBPs
powered by the package SpaSM
https://github.com/cbouilla/spasm
by Charles Bouillaguet

Unitarity cut applicable

Symmetry relation implemented

NeatIBP v1.0

Example 1 Massless pentagon-box



5 Mandelstam variables

(hardest ones with deg-5 numerator)

2503 target integrals without double propagators

12694 IBP relations without double propagators enough

on a workstation with 10 cores
NeatIBP running time ~21 minutes

short and sparse IBP system
with integer-valued Mandelstam variables
Finite-field reduction < 1 second

NeatIBP v1.0 Example 1 Massless pentagon-box

Look at the input files

```
(*------------input files----------------*)
kinematicsFile = workingPath<>"kinematics.txt";(*workingPath will be set as the current working path*)
targetIntegralsFile = workingPath<>"targetIntegrals.txt";

(*-----------dependency settings---------*)
SingularApp = "/usr/bin/Singular";

(*-----------directory settings----------*)
ReductionOutputName="pb1";
outputPath=Automatic;(*recommended*)

(*outputPath=workingPath<>"test/";*)(*alternative*)


(*-----------math settings---------------*)
IntegralOrder = "Global"
OptionSimplification = 12;
NeedSymmetry=True;

(*-----------usage settings--------------*)
MemoryUsedLimit=100000;(*MB*)
ThreadUsedLimit=20;
```

config.txt

NeatIBP v1.0

Example 1 Massless pentagon-box

Look at the input files

```
LoopMomenta={l1,l2};
ExternalMomenta={k1,k2,k3,k4};
Propagators=#^2&/@{l1,l1-k1,l1-k1-k2,l1-k1-k2-k3,l2,l2+k1+k2+k3+k4,l2+k1+k2+k3,l1+l2,l1-k1-k2-k3-k4,l2+k1,l2+k2};
Kinematics={k1^2 -> 0, k2^2 -> 0, k3^2 -> 0, k4^2 -> 0, k1 k2 -> s12/2,
k1 k3 -> 1/2 (-s12 - s23 + s45), k1 k4 -> 1/2 (-s15 + s23 - s45),
k2 k3 -> s23/2, k2 k4 -> 1/2 (s15 - s23 - s34), k3 k4 -> s34/2};
GenericPoint={s12 -> 13, s23 -> 2, s34 -> 73, s45 -> 7, s15 -> 213};
```

kinematics.txt

```
{G[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, -5], G[1, 1, 1, 1, 1, 1, 1, 1, 0, -1, -2], G[1, 1, 1, 1, 1, 1, 1, 1, 0, -2, -1],
G[1, 1, 1, 1, 1, 1, 1, 1, 0, -5, 0], G[1, 1, 1, 1, 1, 1, 1, 1, -1, 0, -3], G[1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -4],
G[1, 1, 1, 1, 1, 1, 1, 1, -1, -2, -1], G[1, 1, 1, 1, 1, 1, 1, 1, -1, -3, 0], G[1, 1, 1, 1, 1, 1, 1, 1, -1, -4, 0],
G[1, 1, 1, 1, 1, 1, 1, 1, -2, 0, -1], G[1, 1, 1, 1, 1, 1, 1, 1, -2, -1, -1], G[1, 1, 1, 1, 1, 1, 1, 1, -3, 0, 0],
G[1, 1, 1, 1, 1, 1, 1, 1, -3, 0, -1], G[1, 1, 1, 1, 1, 1, 1, 1, -3, -2, 0], G[1, 1, 1, 1, 1, 1, 1, 1, -4, 0, -1],
G[1, 1, 1, 1, 1, 1, 1, 1, -4, -1, 0], G[1, 1, 1, 1, 1, 1, 1, 1, -5, 0, 0], G[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, -4],
G[1, 1, 1, 1, 1, 1, 0, 1, 0, 0, -3], G[1, 1, 1, 1, 1, 1, 0, 1, 0, -2, -1], G[1, 1, 1, 1, 1, 1, 0, 1, -1, -3, 0],
G[1, 1, 1, 1, 1, 1, 0, 1, -2, 0, -2], G[1, 1, 1, 1, 1, 1, 0, 1, -3, 0, -1], G[1, 1, 1, 1, 1, 1, 0, 1, -4, 0, -1],
G[1, 1, 1, 1, 1, 1, 0, 1, -4, -1, 0], G[1, 1, 1, 1, 1, 1, -1, 1, -2, 0, 0], G[1, 1, 1, 1, 1, 1, -2, 1, 0, -1, 0],
G[1, 1, 1, 1, 1, 1, -3, 1, -1, 0, -1], G[1, 1, 1, 1, 1, 0, 1, 1, 0, 0, -1], G[1, 1, 1, 1, 1, 0, 1, 1, 0, 0, -3],
G[1, 1, 1, 1, 1, 0, 1, 1, 0, 0, -4], G[1, 1, 1, 1, 1, 0, 1, 1, -1, -1, 0], G[1, 1, 1, 1, 1, 0, 1, 1, -1, -2, -2],
G[1, 1, 1, 1, 1, 0, 1, 1, -2, 0, 0], G[1, 1, 1, 1, 1, 0, 1, 1, -2, 0, -3], G[1, 1, 1, 1, 1, 0, 1, 1, -2, -2, 0],
G[1, 1, 1, 1, 1, 0, 1, 1, -3, 0, 0], G[1, 1, 1, 1, 1, 0, 1, 1, -3, 0, -2], … …
```

target.txt

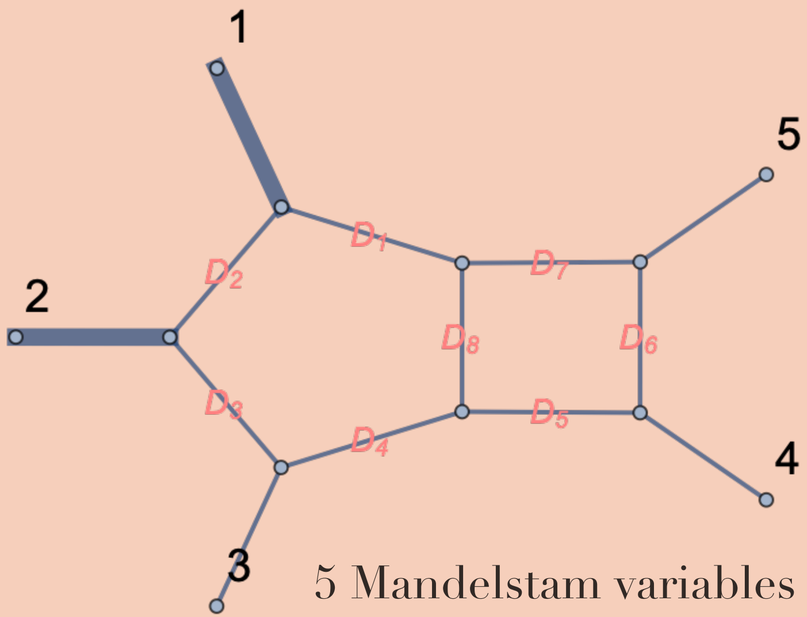
NeatIBP v1.0 Example 1

then in terminal enter

./run.sh

NeatIBP v1.0

Example 2　pentagon-box with two masses



5 Mandelstam variables
+ 2 Mass parameters

a difficult problem

Impose a spanning cut

$$D_2 = D_5 = D_7 = 0$$

top sector target integrals with numerator degree 5

↓

6142 IBP relations without double propagators

enough

on a workstation with 10 cores
NeatIBP running time ~30 minutes

NeatIBP v1.0

Example 2　pentagon-box with two masses

```
(*------------input files----------------*)
kinematicsFile = workingPath<>"kinematics.txt";(*workingPath will be set as the current working path*)
targetIntegralsFile = workingPath<>"targetIntegrals.txt";

(*-----------dependency settings---------*)
SingularApp = "/usr/bin/Singular";
SparseRREF`SpaSMLibrary = "/usr/local/lib/libspasm.so"

(*-----------directory settings----------*)
ReductionOutputName="pb2m257cut";
outputPath=Automatic;(*recommended*)

(*outputPath=workingPath<>"test/";*)(*alternative*)

(*-----------math settings---------------*)
IntegralOrder = "Global"
OptionSimplification = 12;
NeedSymmetry=False;
CutIndices={2,5,7};
GenericD = d -> 137/3;

(*-----------usage settings--------------*)
MemoryUsedLimit=100000;(*MB*)
ThreadUsedLimit=18;
```

Apply a spanning cut

$$D_2 = D_5 = D_7 = 0$$

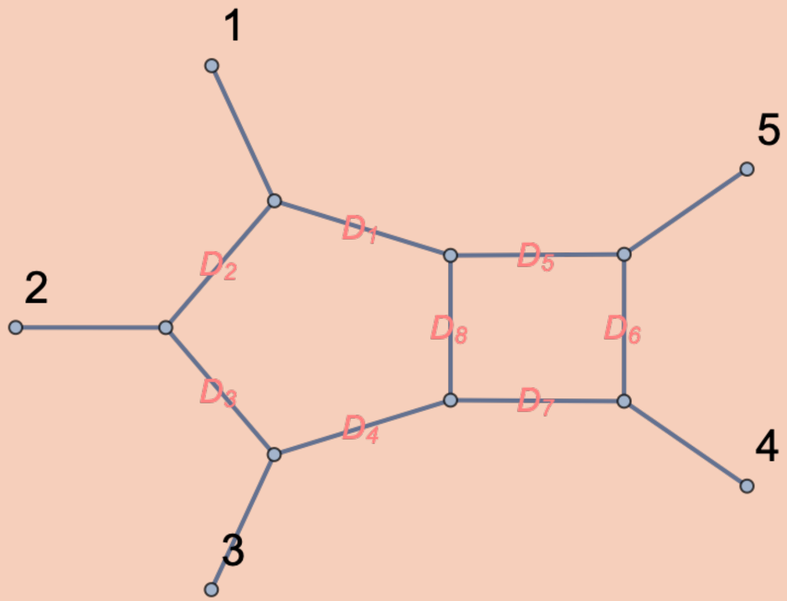
config.txt

NeatIBP v1.0 Example 3 with double propagators

Slogan said: “syzygy IBP method does not apply for integrals with double propagators”

This is wrong.

syzygy IBP method does apply for integrals with double propagators



on a workstation with 10 cores
NeatIBP running time ~51 minutes

61 master integrals’ derivatives in Mandelstam variables

880 target integrals; most of them with double propagators

↓

3374 IBP relations, enough

NeatIBP v1.0

Issues and outlook

- NeatIBP v1.0 uses Mathematica quite a lot. In future, we will focus on open-source Languages.
- NeatIBP v1.0 uses Singular's Schreyer algorithm. We have been testing linear algebra syzygy method for a long time, but not used in this version …

looking forward to Bakul's talk

- The IBP seeding for the syzygy method has many possibilities . It seems that there are a lot of room for tuning …
- We will apply the massive parallelization framework GPI-space for the future NeatIBP versions

Summary

pfd-parallel **and** *NeatIBP*

Based on algebraic geometry

aims at cutting-edge computation of Feynman integrals

various updates of the two packages will appear in this summer

Looking forward to meet you in Europe this summer!

IBP in Baikov representation with constraints

Require

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 propagator degree increase:

$$a_i(z) \in \langle z_i \rangle, \quad 1 \le i \le m$$

These $\big(a_1(z), \ldots a_k(z)\big)$ form a module $M_2 \subset R^k$.

polynomials

Both M_1 and M_2 are pretty simple ...

Larsen YZ
Phys.Rev.D 93 (2016) 4, 041701

$$M_1 \cap M_2$$

Intersection of two modules