

IBPs with KIRA: examples and challenges

(together with Philipp Maierhöfer and Peter Uwer)

PARTICLEFACE: Case studies of analytical and numerical methods of multiloop calculations for future e^+e^- colliders — Katowice

Johann Usovitsch



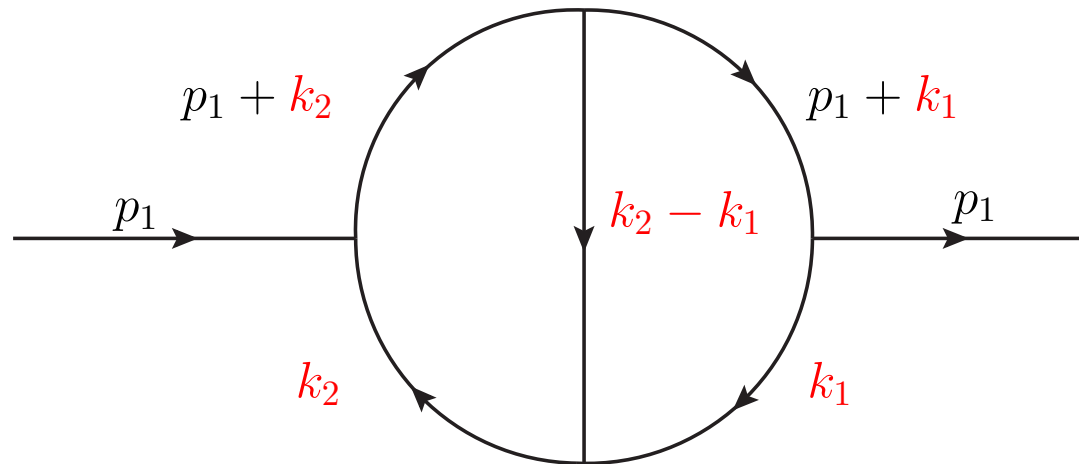
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

02. Oktober 2018

Outline

- 1 Introduction
- 2 Implementation - Kira
- 3 Examples and Challenges
- 4 New feature
- 5 Summary and Outlook

Scalar Integrals



$$I(a_1, \dots, a_5) = \int \frac{d^d k_1 d^d k_2}{[k_1^2]^{a_1} [(p_1 + k_1)^2]^{a_2} [k_2^2]^{a_3} [(p_1 + k_2)^2]^{a_4} [(k_2 - k_1)^2]^{a_5}}$$

- Integral depends explicitly on the exponents a_i
- Number of the loop momenta k_1, k_2 : $L = 2$
- Number of the propagators $[\dots]$: $N = 5$

IBP Identities

$$I(a_1, \dots, a_5) = \int \frac{d^d k_1 d^d k_2}{[k_1^2]^{a_1} [(p_1 + k_1)^2]^{a_2} [k_2^2]^{a_3} [(p_1 + k_2)^2]^{a_4} [(k_2 - k_1)^2]^{a_5}}$$

Integration by parts (IBP) identities [Tkachov,ChetyrkinTkachov,1981]:

$$\int d^d k_1 \dots d^d k_L \frac{\partial}{\partial (k_i)_\mu} \left((q_j)_\mu \frac{1}{[P_1]^{a_1} \dots [P_N]^{a_N}} \right) = 0, \quad q_j = p_1, \dots, p_E, k_1, \dots, k_L$$

$$c_1 I(a_1, \dots, a_N - 1) + \dots + c_m I(a_1 + 1, \dots, a_N) = 0, \quad c_i = \text{Coefficients}$$

- Gives relations between the scalar integrals with different exponents a_f
- Number of $L(E + L)$ IBP equations, $i = 1, \dots, L$ and $j = 1, \dots, E + L$
- $a_f = \text{symbols}$: Look for recursion relations, LiteRed [Lee,2012]
- $a_f = \text{integers}$: Sample a system of equations, Laporta algorithm [Laporta,2000]

Laporta Algorithm [Laporta,2000]

Sample system of IBP equations, Reduze [Studerus,Manteuffel,2012] language

- Seed integrals: $r \in [r_{\min}, r_{\max}]$, $s \in [s_{\min}, s_{\max}]$, $S = \sum_{i=1}^N \theta_j \times 2^{j-1}$
- $r = \sum_{f=1}^N a_f$ mit $a_f > 0$, $f = 1, \dots, N$
- $s = -\sum_{f=1}^N a_f$ mit $a_f < 0$, $f = 1, \dots, N$
- $\theta_j = 1$ for each $a_f > 0$ else $\theta_j = 0$
- T topology number

Fire [Smirnov,2008] language

- Avoid reductions of scalar integrals $\notin (r, s)$
- Gauß type algorithms: Write "difficult" seed integrals $\in (r, s, S, T)$ as a linear combination of "simpler" scalar integrals $\in (r, s, S, T)$ (master integrals)
- Different public implementations: Air [Lazopoulos,Anastasiou,2004], Fire [Smirnov,2008] and Reduze [Studerus,Manteuffel,2012] and Kira

Kira version 1.2

Kira is an implementation of the Laporta algorithm

Get open Kira beta version from gitlab (30.09.2018) at:
<https://gitlab.com/kira-pyred/kira.git>

- New equation generator which is $\sim 10^L$ faster than Kira v.1.1
- Improved parallelization - no openMP
- Compiles on your Mac / New build system: Meson
- Get relations from higher sectors – minimize the number of master integrals
- Start a reduction with a preferred list of master integrals
- Focus the reduction only to a subset of master integrals — set all other coefficients to zero, since Kira v. 1.0 and v. 1.1
- New flexible seed notation is introduced, while the old is preserved
- Choose between 8 different integral Laporta orderings
- Coefficient simplifications are based on heuristics — option `-algebra` is "removed"
- New feature: Algebraic reconstruction

gg→H at 3 loops: integralfamilies.yaml

```
integralfamilies:
```

```
- name: Xhiggs3l1_mmmmmmm00
```

```
  loop_momenta: [ l1, l2, l3 ]
```

```
  top_level_sectors: [511] # important option
```

```
  propagators:
```

```
    - [ "l1", "m^2" ]
```

```
    - [ "l2", "m^2" ]
```

```
    - [ "l3", "m^2" ]
```

```
    - [ "l1 - q1", "m^2" ]
```

```
    - [ "l2 - q1 - q2", "m^2" ]
```

```
    - [ "l1 - l2", 0 ]
```

```
    - [ "-l2 + l3 + q1 + q2", 0 ]
```

```
    - [ "l1 - l2 + l3", "m^2" ]
```

```
    - [ "l1 - l2 + l3 + q2", "m^2" ]
```

```
    - { bilinear: [ [ "l1", "l3" ], 0 ] }
```

```
    - { bilinear: [ [ "l2", "q1" ], 0 ] }
```

```
    - { bilinear: [ [ "l3", "q1" ], 0 ] }
```

gg→H at 3 loops: Old v.s. new jobs.yaml interface

jobs:

- reduce_sectors:

sector_selection: # Old

select_recursively: # Old

- [Xhiggs3l1_mmmmmm00,511] # Old

identities: # Old

ibp: # Old

- { r: [t, 10], s: [0, 4] } # Old

reduce: # New

- {r: 10, s: 4} # New

select_integrals: # important option

select_mandatory_recursively: # important option

- {r: 10, s: 4, d: 1} # important option

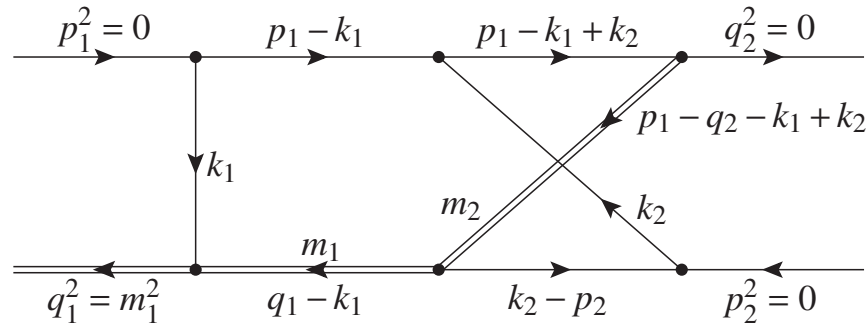
- Kira implicitly knows from integralfamilies.yaml that the user wants to reduce the topology name: Xhiggs3l1_mmmmmm00
- From top_level_sectors: [511] Kira assumes that the user wants to reduce the sector: 511

Reduction of a three loop non-planar topology

Algorithm	Kira v.1.1 (32 cores)	Kira v.1.2 (16 cores)
Generate system of equations	7.9 h	-
Reduce numerically with Pyred	3.6 h	-
Generate and reduce with Pyred	-	3.4 h
Build triangular form	26 h	4.8 h (thread pools)
Backward substitution	18.8 d	4.1 d (heuristics)

- Seed specification: $\{r: 10, s: 4, d: 1\}$
- Speedup comes from less calls to Fermat: $382.502.520 \times 5$ (Kira v.1.1) v.s. 981 (Kira v.1.2)
- After the reduction over the finite fields (modulo 64 Bit prime number) is finished, you know the master integrals

Algebraic coefficient simplification



Type	$T_{\text{Kira v.1.1}}^{m_2^2 = \frac{3}{14} m_1^2}$	$T_{\text{Kira v.1.2}}^{m_2^2 = \frac{3}{14} m_1^2}$	$T_{\text{Kira v.1.1}}$	$T_{\text{Kira v.1.2}}$	$T_{\text{Reduze 2}}^{m_2^2 = \frac{3}{14} m_1^2}$	$T_{\text{FIRE 5}}^{m_2^2 = \frac{3}{14} m_1^2}$
default	2.4 h	1 h	-	11.5 h	2.7 d	23.5 h
A	35.3 min	28.4 min	10 h	5.8 h	-	22.4 h

- default: select_mandatory_recursively: [{r: 7, s: 4}]
- A: select_mandatory_recursively: [{r: 7, s: 4, d: 0}]
- Mass ratio is used $m_2^2 = \frac{3}{14} m_1^2$.
- Reduze 2 A. von Manteuffel and C. Studerus (2012), FIRE 5
A. V. Smirnov (2014)

Kira bench mark

NNLO $pp \rightarrow H_j$ — IBP reduction with Kira v.1.0

- Full mass dependence
- 11 topologies, of which 3 are nonplanar
- Reduction was done for the amplitude and the DEQs at the same time
- Run time: 3 weeks on a single computer node (Dual Intel E5-2630, 2x10 cores 2.2 GHz)

Algebraic reconstruction

Backward substitution gives: $I(\{a_i\}) = \sum_j^M C_j M_j$, M_j master integral

- $C_j = \sum_{i=1}^N c_i$,
- $N \approx \mathcal{O}(10^2) - (10^5)$
- Naiv sum gives a snow ball effect: Intermediate sum grows to more complicated terms than the final result.
- One solution since Kira v. 1.0 is to constantly sort the terms c_i and the intermediate sums in their string length.

Second solution since Kira v. 1.2 is the algebraic reconstruction

- Sample $\sum_{i=1}^N c_i$ by setting at least one parameter $\left\{ \frac{s}{m_1^2}, \frac{t}{m_1^2}, \frac{m_{i \neq 1}^2}{m_1^2}, \dots \right\}$ to integer numbers
- Interpolate the final result from these samples

Implementation part 1

Dependence on at least 2 parameters, e.g.: $\{D, x\}$, $x = \frac{s}{m_1^2}$

- Sample once $C(D, x)$ for numeric value in D
- Get $C(x)$ rational function
- Get the degree of the polynomials (numerator and denominator) in x of $C(x)$: d_N and d_D
- Use $C(x)$ later as a reference point to eliminate sign and numeric prefactor ambiguities
- Original work in this field is based on, see arxiv: 1805.01873 1712.09737 1511.01071 by Yang Zhang and his collaborators

Implementation part 2

Sample $C(D, x)$ $\max(d_N + 2, d_D + 2)$ for numeric values x_j in x

- Get multiple functions $C(D, x) \rightarrow \{C(D, x_j)\}$
- Test that all numerators and denominators have the same number of terms, if not, resample

Interpolate the numerator and the denominator of $C(D, x)$ individually, by using the Newtonian interpolation formula

- $$C(D, x) = \sum_{i=1}^{d_N+1, d_D+1} a_i \prod_{j=1}^{i-1} (x - x_j)$$
- $a_1 = C(D, x_1)$
- $a_2 = \frac{C(D, x_2) - a_1}{x_2 - x_1}$
- $a_3 = \left(\frac{C(D, x_3) - a_1}{x_3 - x_1} - a_2 \right) \frac{1}{x_3 - x_2}$
- \dots
- $a_{d_N+1} = \left(\left(\frac{C(D, x_{d_N+1}) - a_1}{x_{d_N+1} - x_1} - a_2 \right) \frac{1}{x_{d_N+1} - x_2} - \dots - a_{d_N} \right) \frac{1}{x_{d_N+1} - x_{d_N}}$

Implementation part 3

- To activate the algebraic reconstruction use:
`algebraic_reconstruct: true`
- Kira decides based on heuristics to use the algebraic reconstruction algorithm or not
- Heuristics are: Number of terms in a sum, length of the biggest coefficients
- All implementation details are “hidden under the hood” — await improvements and more benchmarks
- Possible usage: Treat coefficients of the master integrals individually

Summary and Outlook

- Kira version 1.2 is available: <https://gitlab.com/kira-pyred/kira.git> and
- includes:
- Fast equation generator
- Improved parallelization
- New flexible seed notation, while the old is preserved
- New feature: Algebraic reconstruction
- Todo list: Provide an interface to reduce arbitrary linear system of equations
- Possible applications: 3-loop and 4-loop IBP-reductions, 5-point 2-loop IBP-reductions