Kira a Feynman Integral Reduction Program and Future Plans

(with Fabian Lange, Philipp Maierhöfer, Jonas Klappert)

CERN workshop for FCC studies

Johann Usovitsch

13. January 20

Outline

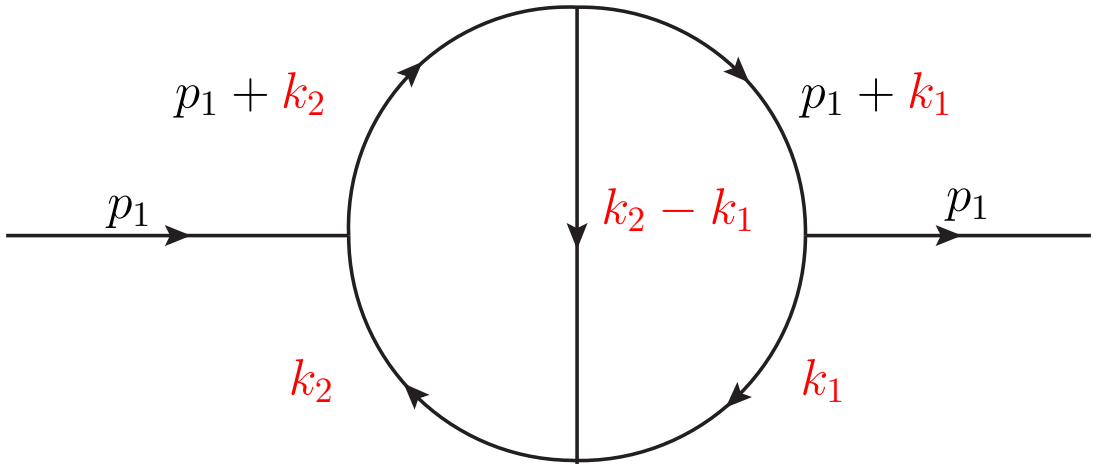
Introduction

- Kira is a linear solver for sparse linear system of equations with main application to **Feynman integral reduction**
- **The development of Kira** is dedicated to extend the range of feasible high precision calculations for present and our **Future Circular Collider**

Feynman integral reduction applications

- Integration-by-parts (IBP)[Chetyrkin, Tkachov, 1981] and Lorentz invariance [Gehrmann, Remiddi, 2000] identities for scalar Feynman integrals are very important in quantum field theoretical computations (multi-loop computations)
- **Reduce the number of Feynman integrals** to compute, which appear in scattering amplitude computations
- **Compute these integrals** analytically or numerically with the method of differential equations [Kotikov, 1991; Remiddi, 1997; Henn, 2013; Argeri et al., 2013; Lee, 2015; Meyer, 2016] or difference equations[Laporta, 2000; Lee, 2010]
- Use the method of sector decomposition [Heinrich,2008] (pySecDec [Borowka et al., 2018] and Fiesta4 [Smirnov, 2016])
- Use the linear reducibility of the integrals (HyperInt [Panzer, 2014]) to compute the Feynman integrals analytically or numerically

Scalar integrals



$$I(a_1, \ldots, a_5) = \int \frac{d^D k_1 d^D k_2}{[{k_1}^2 - m_1^2]^{a_1} [(p_1 + k_1)^2]^{a_2} [{k_2}^2]^{a_3} [(p_1 + k_2)^2]^{a_4} [(k_2 - k_1)^2]^{a_5}}$$

- Integral depends explicitly on the exponents a_f
- Loop momenta: k_1, k_2, $L = 2$
- Number of propagators: $N = 5$
- Scales: p_1^2, m_1^2
- Explicit dependence on the regularization parameter D

IBP identities

$$I(a_1,\ldots,a_5)=\int \frac{d^D k_1\, d^D k_2}{[{k_1}^2-m_1^2]^{a_1}[(p_1+k_1)^2]^{a_2}[{k_2}^2]^{a_3}[(p_1+k_2)^2]^{a_4}[(k_2-k_1)^2]^{a_5}}$$

Integration-by-parts (IBP) identities:

$$\int d^D \boldsymbol{k_1}\ldots d^D \boldsymbol{k_L}\frac{\partial}{\partial(\boldsymbol{k_i})_\mu}\left((q_j)_\mu \frac{1}{[P_1]^{a_1}\ldots[P_N]^{a_N}}\right) = 0$$

$$c_1(\{a_f\})I(a_1,\ldots,a_N-\mathbf{1})+\cdots+c_m(\{a_f\})I(a_1+\mathbf{1},\ldots,a_N) = 0$$

$q_j = p_1,\ldots,p_E,k_1,\ldots,k_L$
m number of terms generated by one IBP

Reduction: express all integrals with the same set of propagators but with different exponents a_f as a linear combination of some basis integrals (master integrals)

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

Laporta algorithm [Laporta, 2000]

Boundary conditions to sample the IBP equations

- Set a_f to integer values in the scalar integrals $I(a_1, \ldots, a_5)$
- Restrict all possible integrals: $r \in [r_{\min}, r_{\max}]$, $s \in [s_{\min}, s_{\max}]$
- $r = \sum_{f=1}^{N} a_f$ with $a_f > 0,\ f = 1, \ldots, N$
- $s = -\sum_{f=1}^{N} a_f$ with $a_f < 0,\ f = 1, \ldots, N$
- Two integrals carry the same sector number S if their inverse propagators are the same

- Reduce only a chosen set of integrals to a fixed number of basis integrals
- Public implementations: Air [Lazopoulos, Anastasiou, 2004], FIRE [A. V. Smirnov et al., 2008, 2013, 2014, 2019] and Reduze [Studerus, 2010] and Reduze 2 [Studerus, von Manteuffel, 2012] and Kira [Maierhöfer, Usovitsch, Uwer, 2017]

Laporta algorithm challenges

- The system of equations generated the Laporta way contains many redundant equations
- The coefficients are polynomials in the dimension D and all scales $\{s_{12}, s_{23}, m_1, m_2, ..\}$
- The number of equations may go up to billions and more
- Solving linear system of equations generated with the Laporta algorithm are CPU, disk and RAM expensive computations
- **Make trade offs to finish the reduction, e.g.: decrease the CPU costs but increase RAM or disk costs**
- **Explore algorithmic improvements!**

Algebraic Reconstruction

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

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

Second solution since **Kira 1.2** is the algebraic reconstruction

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

We use the program **FERMAT** to aid the simplification of rational functions

Finite Field Reconstruction: Kira + FireFly

- Reconstruction of multivariate rational functions from **samples over finite integer fields** [T. Peraro, 2016]
- Public implementations available: FireFly [J. Klappert and F. Lange, 2019], FIRE 6 [A. V. Smirnov and F. S. Chuharev, 2019] and FiniteFlow [T. Peraro, 2019]
- **FireFly has been combined with Kira's native finite field linear solver**

What is special about FireFly

- FireFly uses **Zippel algorithm** in the multivariate case instead of the nested Newton algorithm to interpolate the Polynomials and this is a **good choice**, because:
- Polynomials appearing in IBP reduction problems look like: $E_1 = 1 + x + x^2 + y + y^2 + xy$, people call it dense, but Zippel algorithm needs just 6 probes and Newton 9
- Nested Newton interpolation and the Zippel algorithm are of the same complexity for problems like: $(1+x+x^2)(1+y+y^2) = E_2 = 1+x+x^2+y+xy+x^2y+y^2+xy^2+x^2y^2$
- As we count the last example E_2 has 3 terms less than E_1.
- Zippel needs as many probes as the number of terms in an arbitrary multivariate polynomial

What is special about FireFly

- Main observation: IBP reductions involving 2 scales have the complexity of E_2, thus FIRE6 and FiniteFlow are as good as FireFly (probably)
- IBP reductions involving 3 or more scales have the complexity of E_1, thus FireFlow dominates over other public codes (probably)

Kira + FireFly, main features

- The run time for the sampling over finite field is dominated by the forward elimination of the reduction
- **Strategy**: Compute the rational functions appearing in the forward elimination phase
- Use this as an input to reconstruct the final rational functions appearing in the backward substitution
- Kira supports now MPI: do calculations across different computers on all cores available, this gives opportunities for a very big parallelization
- Kira reconstructs entire reduction of Feynman integrals from numerical samples
- Final public implementation of Kira + FireFly is in documentation phase, we wait for benchmarks to finish

Algorithm to factorize the denominators - **the setup**

$$I_j = \sum_k C_{jk} M_k$$

- With M_k being the master integrals
- C_{jk} the rational functions, depending on the space time dimension d and the kinematics $\{s_{12}, s_{23}, m_1, m_2, ...\}$
- I_j are the integrals we want to reduce

Algorithm to factorize the denominators - **mechanics**

$$I_j = \sum_k C_{jk} M_k \quad \rightarrow C_{jk} = \frac{n_{jk}}{d_{jk}}$$

$$\rightarrow d_{jk} = \prod_l d_{jkl}(d)^{a_l} \prod_m d_{jkm}(\{s_{12}, s_{23}, m_1, m_2, ...\})^{a_m}$$

- d_{jk}, d_{jkl}, n_{jk} are polynomials and a_l are integers
- We observe (who is we: Vladimir Smirnov made me anxious about it during the Amplitudes conference, also you can find it in FIRE6 paper [A. Smirnov, F.S. Chuharev, 2019]) that it is possible to find a master integral basis, where the d-dependence factorizes in the denominators from other scales. In addition the denominator factorizes in simple kinematic building blocks
- What is simple: kinematic building blocks are polynomials of lower degree compared to the total degree of the kinematic part of the denominators

Algorithm to factorize the denominators

- First find d factorizable form:

step 1: Complete a reduction with all integrals involving 2 dots and 2 scalar products up to the top topology with some arbitrary basis. Set all kinematics to numeric values but d.

step 2: Scan for all denominators involving d of degree higher than 1, let us call all denominators of higher degree bad.

- The crucial steps are the following.

step3: Since we have a very big reduction involving many integrals we can go from bottom to top and scan ALL integrals which involve the same bad denominator involving the d-dependence. The first sector S_1 which contains a MI(also possibly being in lower sector S_2), which has this bad denominator will be taken for the next step.

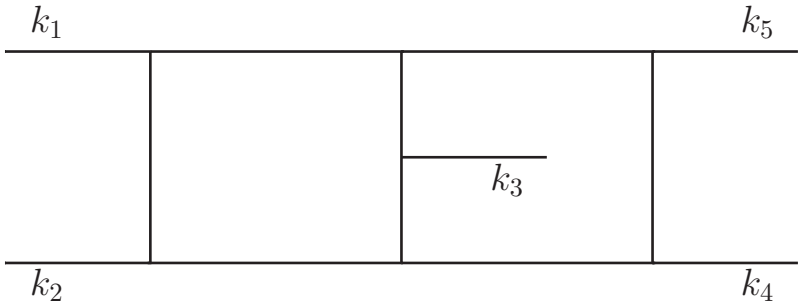
step4: Change basis in this sector S_1, not in S_2. And check whether the bad denominator disappeared and that no other is introduced.

step5: Repeat for all denominators. Now we repeat the reduction by setting kinematics to different values to see whether polynomials in d are independent of the kinematics.

Algorithm to factorize the denominators - **conclusion**

- The lesson we take is that the bad denominators in front of the MI in sector S_2 are not responsible ad hoc for the bad denominators but higher sectors like S_1 do.
- In my example I needed to change basis in sectors with more inverse propagators to remove the bad denominator in the sectors with less number of inverse propagators

Example: box-pentagon - **specs**



- Inverse propagators: l_1, $l_1 - k_1$, $l_1 - k_{1,2}$, l_2, $l_2 - k_{1,2,3}$, $l_2 - k_{1,2,3,4}$, $l_1 - l_2$, $l_1 - l_2 + k_3$, $l_1 - k_{1,2,3,4}$
- Auxiliary propagators: $l_2 - k_1$, $l_2 - k_{1,2}$
- Kinematics: $k_1 k_2 = \frac{s_{12}}{2}$, $k_1 k_3 = \frac{s_{45} - s_{12} - s_{23}}{2}$, $k_1 k_4 = \frac{s_{23} - s_{15} - s_{45}}{2}$, $k_2 k_3 = \frac{s_{23}}{2}$, $k_2 k_4 = \frac{s_{15} - s_{23} - s_{34}}{2}$, $k_3 k_4 = \frac{s_{34}}{2}$, $s_{12} = 1$, $k_1^2 = k_2^2 = k_3^2 = k_4^2 = k_5^2 = 0$
- Momenta conservation: $k_1 + k_2 + k_3 + k_4 + k_5 = 0$
- Total number of master integrals: 108

Example: box-pentagon - **change of basis**

- Initial preferred master integral basis: contains only master integrals with dots (no scalar products)
- Change the basis of 6 master integrals to:

$$(0,1,0,1,1,0,1,2,0,0,0) \to (0,1,0,1,1,0,1,1,0,-1,0)$$
$$(0,1,1,1,1,0,1,2,0,0,0) \to (0,1,1,1,1,0,2,1,0,0,0)$$
$$(1,0,1,0,1,1,1,2,0,0,0) \to (1,-1,1,0,1,1,1,1,0,0,0)$$
$$(1,0,1,0,1,1,2,0,0,0,0) \to (1,-1,1,0,1,1,1,0,0,0,0)$$
$$(0,1,1,1,0,1,1,2,0,0,0) \to (0,2,1,1,0,1,1,1,0,0,0)$$

Example: box-pentagon - **denominator building blocks**

$$I_j = \sum_k C_{jk} M_k \quad \rightarrow C_{jk} = \frac{n_{jk}}{d_{jk}}$$

$$\rightarrow d_{jk} = \prod_l d_{jkl}(d)^{a_l} \prod_m d_{jkm}(\{s_{12}, s_{23}, m_1, m_2, ...\})^{a_m}$$

d_{jkl}	d_{jkm}
$d-8$	$(s_{15} - s_{23} + s_{45}), (-1 - s_{15} + s_{34})$
$d-6$	$(-s_{15} + s_{23} - s_{45}), (-1 - s_{15} + s_{34}), (-1 + s_{34})$
$d-5$	$1 + s_{23} - s_{45}$
$d-4$	$(-s_{15} + s_{23} + s_{34}), (-1 + s_{34} + s_{45}), (s_{34} + s_{45})$
$d-3$	$(s_{15}^2 - 2s_{15}s_{23} + s_{23}^2 + 2s_{15}s_{23}s_{34} - 2s_{23}^2 s_{34} + s_{23}^2 s_{34}^2$ $-2s_{15}^2 s_{45} + 2s_{15}s_{23}s_{45} + 2s_{15}s_{34}s_{45} + 2s_{23}s_{34}s_{45}$ $+2s_{15}s_{23}s_{34}s_{45} - 2s_{23}s_{34}^2 s_{45} + s_{15}^2 s_{45}^2 - 2s_{15}s_{34}s_{45}^2 + s_{34}^2 s_{45}^2)$
$d-2$	$(s_{15}s_{34} - s_{23}s_{34} + s_{23}s_{34}^2 - s_{15}s_{45} + s_{23}s_{45} + 2s_{34}s_{45}$ $+s_{15}s_{34}s_{45} + s_{23}s_{34}s_{45} - s_{34}^2 s_{45} + s_{15}s_{45}^2 - s_{34}s_{45}^2)$
$d-1$	$(1 + s_{23}), (-s_{15} + s_{34}), (s_{23}), (s_{15})$
$2d-11$	$1 + s_{23} - s_{34} - 2s_{23}s_{34} - 2s_{45} - s_{23}s_{45} + s_{34}s_{45} + s_{45}^2$
$2d-9$	$(-1 + s_{34} + s_{45} + s_{34}s_{45}), (s_{15} - s_{23}), (s_{23} + s_{34})$
$2d-7$	$-s_{15} + s_{23} - s_{23}s_{34} - s_{45} + s_{15}s_{45} - 2s_{23}s_{45} + s_{45}^2$
$3d-10$	$(-s_{23} - s_{45} - s_{23}s_{45} + s_{45}^2), (s_{45}), (s_{34})$
$3d-8$	$(-1 + s_{15} - s_{23} + s_{45}), (s_{23} - s_{45}), (1 + s_{15} - s_{34} - s_{45})$
$3d-14$	$(-1 + s_{45}), (s_{15} - s_{23} + s_{23}s_{34} - s_{15}s_{45} + s_{34}s_{45})$

Example: box-pentagon - reduction of 21 integrals

- Suppose we perform the reduction of these 21 integrals:
  $(1,1,1,1,1,1,1,1,-5,0,0)$, $(1,1,1,1,1,1,1,1,0,-5,0)$,
  $(1,1,1,1,1,1,1,1,0,0,-5)$, $(1,1,1,1,1,1,1,1,-4,-1,0)$,
  $(1,1,1,1,1,1,1,1,-4,0,-1)$, $(1,1,1,1,1,1,1,1,-1,-4,0)$,
  $(1,1,1,1,1,1,1,1,0,-4,-1)$, $(1,1,1,1,1,1,1,1,-1,0,-4)$,
  $(1,1,1,1,1,1,1,1,0,-1,-4)$, $(1,1,1,1,1,1,1,1,-3,-2,0)$,
  $(1,1,1,1,1,1,1,1,-3,0,-2)$, $(1,1,1,1,1,1,1,1,-3,-1,-1)$,
  $(1,1,1,1,1,1,1,1,-2,-3,0)$, $(1,1,1,1,1,1,1,1,0,-3,-2)$,
  $(1,1,1,1,1,1,1,1,-1,-3,-1)$, $(1,1,1,1,1,1,1,1,-2,0,-3)$,
  $(1,1,1,1,1,1,1,1,0,-2,-3)$, $(1,1,1,1,1,1,1,1,-1,-1,-3)$,
  $(1,1,1,1,1,1,1,1,-2,-2,-1)$, $(1,1,1,1,1,1,1,1,-2,-1,-2)$,
  $(1,1,1,1,1,1,1,1,-1,-2,-2)$
- We get master integrals, where all denominators factorize into the building blocks from the above table d_{jkl} and d_{jkm}
- How do I know that this is true, without doing the complete reduction, yet?

Get full analytic structure for the denominators

- I used methods of algebraic reconstruction: set all variables to numerical values but one, lets keep s_{23} as a symbol
- Extract all the denominators appearing in the reduction and factorize, which is easy since all denominators are supposed to factorize due to the special master integral basis choice we did before
- And very fast because the denominators depend on just one variable; in this example just s_{23}
- Now vary the numerical values for all variables (generate samples) but leave s_{23} as a symbol
- Reconstruct from these samples all building blocks, e.g.:
  $-18 + s_{23} \to -s_{15} + s_{23} - s_{45}$
  $-10 + s_{23} \to 1 + s_{23} - s_{45}$
  $6 + s_{23} \to -s_{15} + s_{23} + s_{34}$
- Once the full dependence for the denominators is known, we can tell Kira to reconstruct just the numerators by canceling the now known denominators

Upcoming Features of Kira Version 1.3

Kira's, development release

Get Kira on gitlab: https://gitlab.com/kira-pyred/kira.git

- Master equations; the basis for a reduction consists of a linear combination of integrals
- Generation of differential equations
- Kira + FireFly: individual reconstruction of forward elimination and back substitution
- Possible benchmarks: some $2 \to 3$ processes from QCD
- General propagators
- Symbolic IBP-reduction

Summary and Outlook

- Many parallelization improvements
- Utilize more the finite field methods
- New benchmarks to come soon
- denominator reconstruction
- Kira is an all-rounder for multi-scale as well as for multi-loop computations
- More powerful than Mathematica's LinearSolve; which gives many applications outside of Feynman integrals