Integration-by-parts: Survey

Peter Marquard

Zeuthen October 2020

Outline

Disclaimer

- This talk represents a personal, thus biased, view of the problem
- I will concentrate on Laporta's approach to the problem
- It is neither exhaustive nor complete

Outline

Motivation

- In a typical Feynman-diagrammatic calculation many ($\mathcal{O}(10^3)$ – $\mathcal{O}(10^7)$) Feynman integrals can appear.
- In general, there is a trade-off between the number of integrals appearing and the number of their physical scales.
- The appearing integrals are elements of a vector space with few basis elements.
- The basis elements (aka master integrals) and the relation of the integrals to them can be obtained using integration-by-parts methods.
- The fact that Feynman integrals are not linearly independent is also very important in the context of differential or difference equations.

Notation

Consider a family of Feynman integrals with

- L loops (loop momenta k_i)
- E external legs (momenta q_i)
- N internal lines
- $\mathcal{I} = L(E-1) + L(L+1)/2$ invariants

$$\int \left(\prod_i^L d^d k_i \right) \prod_{j=1}^N \frac{1}{(P_j^2)}$$

where

$$P_j^2 = \left(\sum_{m=1}^L A_{jm} k_m + \sum_m^E B_{jm} q_m \right)^2 - m_j^2$$

Notation

Consider a family of Feynman integrals with

- L loops (loop momenta k_i)
- E external legs (momenta q_i)
- N internal lines
- $\mathcal{I} = L(E-1) + L(L+1)/2$ invariants

$$\int \left(\prod_i^L d^d k_i \right) \prod_{j=1}^N \frac{1}{(P_j^2)^{a_j}}$$

where

$$P_j^2 = \left(\sum_{m=1}^L A_{jm} k_m + \sum_m^E B_{jm} q_m \right)^2 - m_j^2$$

Notation

Consider a family of Feynman integrals with

- L loops (loop momenta k_i)
- E external legs (momenta q_i)
- N internal lines
- $\mathcal{I} = L(E-1) + L(L+1)/2$ invariants

$$\int \left(\prod_i^L d^d k_i \right) \prod_{j=1}^N \frac{1}{(P_j^2)^{a_j}} \prod_{j=N+1}^{\mathcal{I}} (P_j^2)^{b_j}$$

where

$$P_j^2 = \left(\sum_{m=1}^L A_{jm} k_m + \sum_m^E B_{jm} q_m \right)^2 - m_j^2$$

Notation

Consider a family of Feynman integrals with

- L loops (loop momenta k_i)
- E external legs (momenta q_i)
- N internal lines
- $\mathcal{I} = L(E-1) + L(L+1)/2$ invariants

$$\mathcal{J}(a_1, \ldots, a_N, -b_{N+1}, \ldots, -b_{\mathcal{I}}) = \int \left(\prod_i^L d^d k_i \right) \prod_{j=1}^N \frac{1}{(P_j^2)^{a_j}} \prod_{j=N+1}^{\mathcal{I}} (P_j^2)^{b_j}$$

where

$$P_j^2 = \left(\sum_{m=1}^L A_{jm} k_m + \sum_m^E B_{jm} q_m \right)^2 - m_j^2$$

Sectors

- We have families of integrals defined by the form of their propagators
- All integrals of a family belong to a sector, which is defined by the propagators with positive powers

$$S[J(k_1, \ldots, k_N)] = \{i \in \{1, \ldots, N\} | \, k_i > 0\}$$

e,g.

$$J(1, 0, 1, -2, 1) \in S_{1,3,5}$$

The sectors correspond to the lines present in the corresponding Feynman diagram.

Integration-By-parts

In dimensional regularization

$$0 = \int d^d k \frac{\partial}{\partial k^\mu} f(k)$$

and it follows

$$0 = \int \left(\prod_i^L d^d k_i \right) \frac{\partial}{\partial k^\mu} \prod_{j=1}^N \frac{1}{(P_j^2)^{a_j}} \prod_{j=N+1}^{\mathcal{I}} (P_j^2)^{b_j}$$

Integration-By-parts

In dimensional regularization

$$0 = \int d^d k \frac{\partial}{\partial k^\mu} f(k)$$

and it follows

$$0 = \int \left(\prod_i^L d^d k_i \right) \frac{\partial}{\partial k^\mu} p^\mu \prod_{j=1}^{N} \frac{1}{(P_j^2)^{a_j}} \prod_{j=N+1}^{\mathcal{I}} (P_j^2)^{b_j}$$

with $p \in \{k, q\}$
Denote this by

$$0 = \mathcal{O}_{\mathrm{IBP}}(k, p) \mathcal{J}(n_1, \ldots, n_{\mathcal{I}})$$

General structure

Define operators $\mathbf{i}^+, \mathbf{i}^-$ with properties

$$\begin{aligned}\mathbf{i}^+ \mathcal{J}(n_1, \ldots, n_\mathcal{I}) &= n_i \mathcal{J}(n_1, , \ldots, n_i + 1, \ldots, n_\mathcal{I}) \\ \mathbf{i}^- \mathcal{J}(n_1, \ldots, n_\mathcal{I}) &= \mathcal{J}(n_1, , \ldots, n_i - 1, \ldots, n_\mathcal{I})\end{aligned}$$

then the general form of an IBP relation is

$$\mathcal{O}_{\mathrm{IBP}}(k, p) = d\delta_{k\,p} + \sum C_{ij}\, \mathbf{i}^+ \mathbf{j}^- + \sum D_k(s_{ij}, m_i^2)\, \mathbf{k}^+$$

$s_{ij} = (q_i + q_j)^2$

$L \times (L + E - 1)$ relations

Laporta's algorithm

generate a system of linear equation using the IBP relations and solve it starting from the most complicated one

implemented in many public (and private) codes

- `FIRE` [Smirnov]
- `Reduze` [v. Manteuffel(, Studerus)]
- `Kira` [Maierhöfer, Usovitsch, Uwer]

Alternatives

solve relations in a symbolic way to obtain

- find explicit recursion relations, manually or in an automated way (LiteRed [Lee])
- rules for certain special configurations
  e.g. the triangle rule

  $\hookrightarrow$ talk by Jos Vermaseren

- IBPs without IBPs

  $\hookrightarrow$ talk by Hjalte Frellesvig

Outline

The Problem

It is big

Laporta's algorithm

- Generate a large system of linear for the integrals by applying the IBP operator to seed integrals.
- For the set of seed integrals one can e.g. choose all integrals

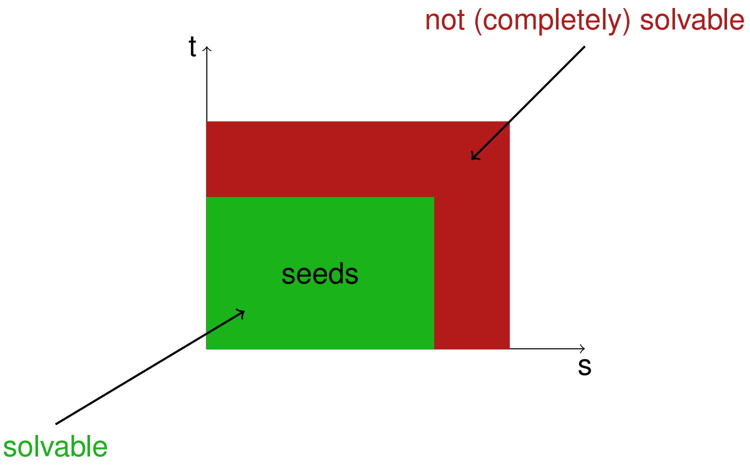
$$\mathcal{J}(n_1, \ldots, n_{\mathcal{I}}),$$

with

$$s = \sum_i n_i \theta(n_i), \qquad t = -\sum_i n_i \theta(-n_i)$$

$$s \leq S, \qquad t \leq T$$

Integral content of the system of equations



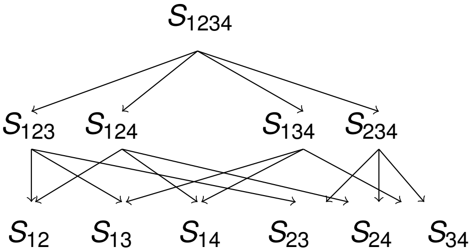
The System

$$
\begin{pmatrix}
* & * & * & * & \cdots & \cdots & * & * \\
* & * & * & * & \cdots & \cdots & * & * \\
* & * & * & * & \cdots & \cdots & * & * \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\
* & * & * & * & \cdots & \cdots & * & * \\
* & * & * & * & \cdots & \cdots & * & * \\
* & * & * & * & \cdots & \cdots & * & *
\end{pmatrix}
\begin{pmatrix}
J_1 \\ J_2 \\ J_3 \\ J_4 \\ \vdots \\ \vdots \\ J_{N-1} \\ J_N
\end{pmatrix}
=
\begin{pmatrix}
0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0
\end{pmatrix}
$$

$M \times N$ matrix X, with, in general, $M > N$, but $\mathrm{rank}\,(X) < N$.

Levels of sectors

- The system of equation arranges itself in layers given be the different sectors.



- The sectors within the same layer can be solved independently and the solutions.
- The solutions have to be fed into the layers above.
- The sectors do mostly not talk to each other, but there are rare cases where equations in higher sectors can introduce relations between masters in different lower sectors.

Symmetries

IBP does not know about symmetries of the integrals

Symmetries can most easily be seen by going back to the corresponding graph. At this level one can easily determine, if

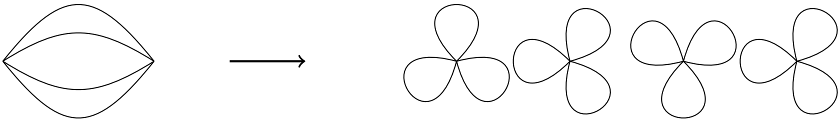
- two or more sectors are the same
- one can put e.g. dots on equivalent lines within the same sector

Symmetries

IBP does not know about symmetries of the integrals

Symmetries can most easily be seen by going back to the corresponding graph. At this level one can easily determine, if

- two or more sectors are the same
- one can put e.g. dots on equivalent lines within the same sector



Solving the system of equations

Order the integrals by their difficulty, e.g. number of lines, dots, irred. numerators

Transform the corresponding matrix to

- reduced row echelon form
- using a Gauss-Jordan elimination

In general this procedure scales like $\mathcal{O}(N)^3$ where we only counted the necessary number of operations and do not take their complexity into consideration.

N.B. Scaling strictly only true for dense systems.

Structure of the solved system

$$
\left(\begin{array}{ccc|ccccc|ccc}
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\hline
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\hline
0 & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
\vdots & & \vdots & \vdots & \vdots & \ddots & 1 & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & * & \cdots & *
\end{array}\right)
\begin{pmatrix} J_N \\ \vdots \\ J_{K+1} \\ J_K \\ \vdots \\ J_{M+1} \\ J_M \\ \vdots \\ J_1 \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

Structure of the solved system

$$
\left(\begin{array}{ccc|ccccc|ccc}
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\hline
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\hline
0 & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
\vdots & & \vdots & \vdots & \vdots & \ddots & 1 & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & * & \cdots & *
\end{array}\right)
\begin{pmatrix} J_N \\ \vdots \\ J_{K+1} \\ J_K \\ \vdots \\ J_{M+1} \\ J_M \\ \vdots \\ J_1 \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

Structure of the solved system

$$
\left(\begin{array}{ccc|ccccc|ccc}
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\hline
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\hline
0 & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
\vdots & & \vdots & \vdots & \vdots & \ddots & 1 & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & * & \cdots & *
\end{array}\right)
\begin{pmatrix} J_N \\ \vdots \\ J_{K+1} \\ J_K \\ \vdots \\ J_{M+1} \\ J_M \\ \vdots \\ J_1 \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

Structure of the solved system

$$
\left(\begin{array}{ccc|ccccc|ccc}
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\hline
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\vdots & & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & & \vdots \\
* & \cdots & * & 0 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
\hline
0 & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
\vdots & & \vdots & \vdots & \vdots & \ddots & 1 & 0 & * & \cdots & * \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & * & \cdots & *
\end{array}\right)
\begin{pmatrix} J_N \\ \vdots \\ J_{K+1} \\ J_K \\ \vdots \\ J_{M+1} \\ J_M \\ \vdots \\ J_1 \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

Problems

Redundant equations: To calculate a “0” takes time, too. They can make up 50% of the system.

Not fully reduced block: Takes a lot of time, especially during back-substitution.

Fully reduced block: Computation suffers from intermediate expression growth.

Outline

Reduce the size of the problem

- Reduce the size of the system equations
  - clever/proper choice of seed integrals

Reduce the size of the problem

- Reduce the size of the system equations
    - clever/proper choice of seed integrals
- Pre-conditioning
    - Run the problem for suitable (integer) values of the parameters
    - note the equations needed to reduce the searched after integrals
    - run the full problem using only these equations

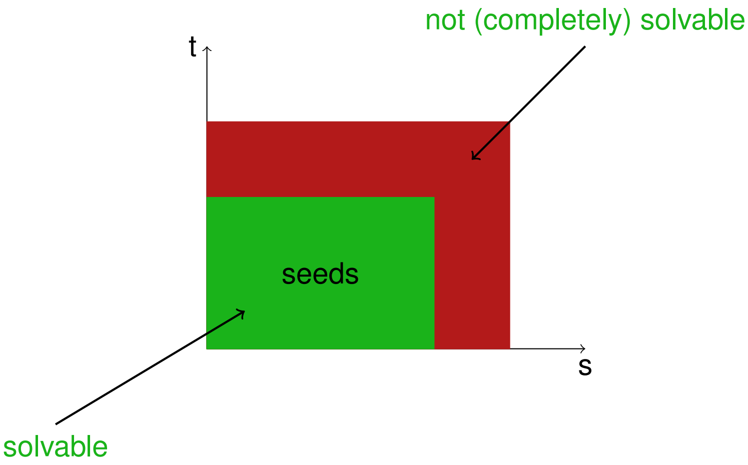
Reduce the size of the problem

- Reduce the size of the system equations
  - clever/proper choice of seed integrals
- Pre-conditioning
  - Run the problem for suitable (integer) values of the parameters
  - note the equations needed to reduce the searched after integrals
  - run the full problem using only these equations
- Pre-filtering
  - Use integer run to remove redundant equations

Reduce the size of the problem

- Reduce the size of the system equations
  - clever/proper choice of seed integrals
- Pre-conditioning
  - Run the problem for suitable (integer) values of the parameters
  - note the equations needed to reduce the searched after integrals
  - run the full problem using only these equations
- Pre-filtering
  - Use integer run to remove redundant equations
- Pre-Ordering
  - If (good) masters are already known, reducing to them can improve performance

Can the red part be avoided?



Dot and/or Numerator free IBPs: Syzygies

In a nutshell, use the freedom in choosing the vector p in the original definition of the IBP identity [Gluza,Kajda,Kosower '00]

$$0 = \int \left(\prod_{i}^{L} d^d k_i \right) \frac{\partial}{\partial k^\mu} p^\mu \prod_{j=1}^{N} \frac{1}{(P_j^2)^{a_j}} \prod_{j=N+1}^{\mathcal{I}} (P_j^2)^{b_j}$$

p^μ can be chosen such that

- no additional dots
- no additional numerators

appear in the relations.

Reduce the complexity of the operations

How to avoid the expensive rational algebra operations?

- Insert prime numbers for dimension d and all the appearing invariants m_i , s_{ij} and run the reduction.

Reduce the complexity of the operations

How to avoid the expensive rational algebra operations?

- Insert prime numbers for dimension d and all the appearing invariants m_i , s_{ij} and run the reduction.
- If one knows a function at sufficiently many points it can be reconstructed.

Reduce the complexity of the operations

How to avoid the expensive rational algebra operations?

- Insert prime numbers for dimension d and all the appearing invariants m_i , s_{ij} and run the reduction.
- If one knows a function at sufficiently many points it can be reconstructed.
- Ratios of (arbitrary precision) integers are no good, so use finite field methods on top

Reduce the complexity of the operations

How to avoid the expensive rational algebra operations?

- Insert prime numbers for dimension d and all the appearing invariants m_i , s_{ij} and run the reduction.
- If one knows a function at sufficiently many points it can be reconstructed.
- Ratios of (arbitrary precision) integers are no good, so use finite field methods on top
- Reconstruct the full rational dependence by Chinese Remainder Theorem and rational reconstruction

Finite field methods

- implemented in most public Laporta reduction codes
  - FIRE
  - KIRA [Klappert, Lange, Maierhöfer, Usovitsch] using FireFly [Klappert, Lange]
  - (Reduze) $\rightarrow$ (private) FinRed

Finite field methods

- implemented in most public Laporta reduction codes
  - FIRE
  - KIRA [Klappert, Lange, Maierhöfer, Usovitsch] using FireFly [Klappert, Lange]
  - (Reduze) $\rightarrow$ (private) FinRed
- probably best to do the whole calculation up to the end using finite fields

  $\hookrightarrow$ `FiniteFlow` [Peraro]

Outline

Conclusions

- The field of Integration-By-Parts reductions is generally in a good shape
- New ideas are being implemented in public codes and made available
- Personal view:
  - Ball is back in the field of calculating master integrals
  - For much more complicated problems one might need to find a completely different approach to the overall problem