FIRE6: Feynman Integral REduction with Modular Arithmetic

A.V. Smirnov

Research computing center, MSU

March 18, 2019

Feynman integrals

- One diagram leads to Feynman integrals of same structure
- $F(a_1, \ldots, a_n) = \int \cdots \int \frac{\mathrm{d}^d k_1 \ldots \mathrm{d}^d k_h}{E_1^{a_1} \ldots E_n^{a_n}}$.
- E are quadratic over k (loop momenta) and some external momenta q and can contain extra scalar variables, for example $(k_1 - q_2)^2 - m^2$
- a_i are integer indices.

Feynman integrals

- A classical strategy (Chetyrkin, Tkachov, 1981) is to write so-called integration by part relations and solve them representing all integrals as a linear combination of so-called master integrals;
- First performed by hand, then an algorithm was suggested (Laporta, Remiddi, 1996)
- First public algorithm AIR (Anastasiou, Lazopoulos, 2004)
- Many more algorithms
- It was shown that the number of master-integrals is always finite (Smirnov, Petukhov, 2010), but the proof is non-constructive.

Integration by part relations

- The formula for Feynman integrals

$$F(a_1, \ldots, a_n) = \int \cdots \int \frac{\mathrm{d}^d k_1 \ldots \mathrm{d}^d k_h}{E_1^{a_1} \ldots E_n^{a_n}}.$$

- leads to equations between them

$$\int \ldots \int \left[\frac{\partial}{\partial k_j} \cdot k_l \prod_{i=1}^{L} \frac{1}{E_i^{a_i}} \right] \mathrm{d}^d k_1 \ldots \mathrm{d}^d k_h = 0$$

An example

- $F(a) = \int \frac{\mathrm{d}^d k}{(k^2 - m^2)^a}$
  $F(a) = 0$ for integer $a \leq 0$.
  We need $F(a)$ for positive integer a.

An example

- $F(a) = \int \frac{\mathrm{d}^d k}{(k^2 - m^2)^a}$
  $F(a) = 0$ for integer $a \leq 0$.
  We need $F(a)$ for positive integer a.
- Apply IBP
  $\int \mathrm{d}^d k \frac{\partial}{\partial k} \cdot \left(k \frac{1}{(k^2 - m^2)^a} \right) = 0$

An example

- $F(a) = \int \frac{\mathrm{d}^d k}{(k^2 - m^2)^a}$
  $F(a) = 0$ for integer $a \leq 0$.
  We need $F(a)$ for positive integer a.
- Apply IBP
  $\int \mathrm{d}^d k \frac{\partial}{\partial k} \cdot \left(k \frac{1}{(k^2 - m^2)^a} \right) = 0$
- Taking derivatives:
  $\frac{\partial}{\partial k} \cdot k = \frac{\partial}{\partial k_\mu} \cdot k_\mu = d$

An example

- $k \cdot \frac{\partial}{\partial k} \frac{1}{(k^2-m^2)^a} = -a \frac{2k^2}{(k^2-m^2)^{a+1}}$
  $= -2a \left[\frac{1}{(k^2-m^2)^a} + \frac{m^2}{(k^2-m^2)^{a+1}} \right]$

An example

- $k \cdot \frac{\partial}{\partial k} \frac{1}{(k^2-m^2)^a} = -a\frac{2k^2}{(k^2-m^2)^{a+1}}$
  $= -2a\left[\frac{1}{(k^2-m^2)^a} + \frac{m^2}{(k^2-m^2)^{a+1}}\right]$
- IBP relation
  $(d-2a)F(a) - 2am^2F(a+1) = 0$

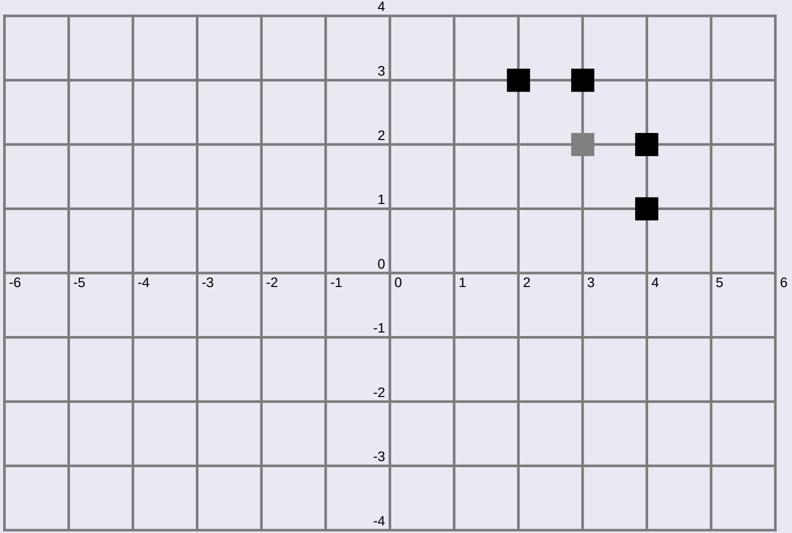
An example

- $k \cdot \frac{\partial}{\partial k} \frac{1}{(k^2-m^2)^a} = -a \frac{2k^2}{(k^2-m^2)^{a+1}}$
  $= -2a \left[\frac{1}{(k^2-m^2)^a} + \frac{m^2}{(k^2-m^2)^{a+1}} \right]$
- IBP relation
  $(d-2a)F(a) - 2am^2 F(a+1) = 0$
- Its solution
  $F(a) = \frac{d-2a+2}{2(a-1)m^2} F(a-1)$

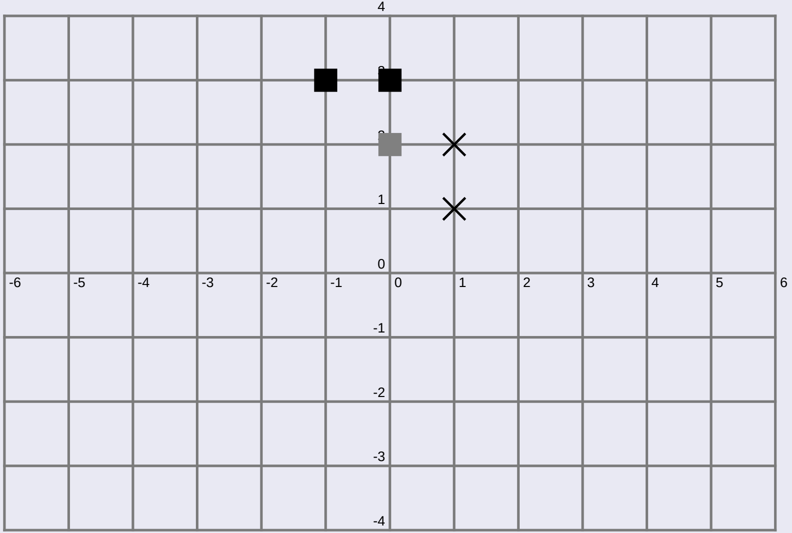
Integration by part relations

- $F(a_1, \ldots, a_n) = \int \cdots \int \frac{\mathrm{d}^d k_1 \ldots \mathrm{d}^d k_h}{E_1^{a_1} \ldots E_n^{a_n}}$.
- Differentiate the expression by an internal momenta and take the scalar product with any momenta, the result has to be zero
- We have index changes and constructions like $(dE_i/dk_j) \cdot v$. It is again quadratic. Re-decompose it by E_i and there are not enough propagators E_i add some more to get the full basis
- One gets $(int + 1) * int/2 + int * ext$ indices.
- There is a set of relations between integrals with different indices. Solve them?

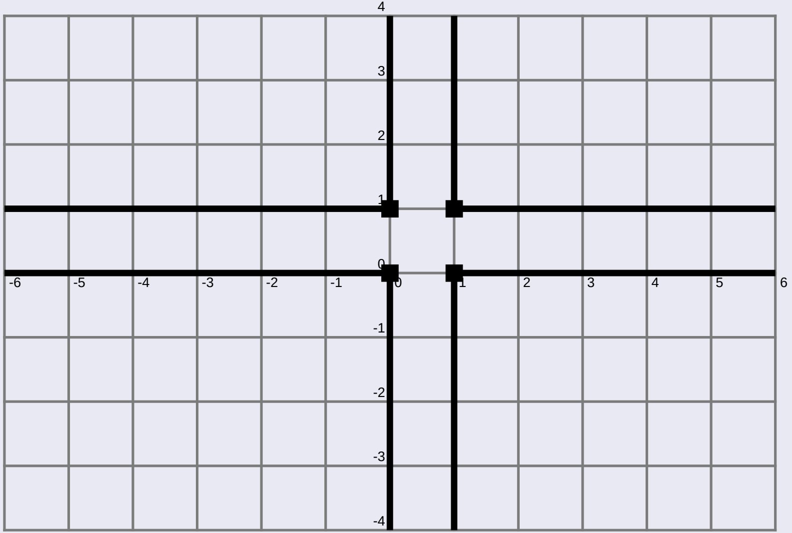
Integration by part relations (indices)



Integration by part relations (indices, no forward move)



Integration by part relations (sectors and corner points)



Reduction strategy

- Reduce sector-wise. We prefer going to lower sectors. This also has a physical meaning, due to boundary conditions many of sectors with multiple non-positive indices consist of zero-integrals.
- Level of a sector = number of positive indices.
- Sectors of same level can in principle be reduced in parallel.
- There are also symmetries that can map sectors of same level to each other (with sums).
- Each sector requires an ordering inside.

Reduction strategy in a sector

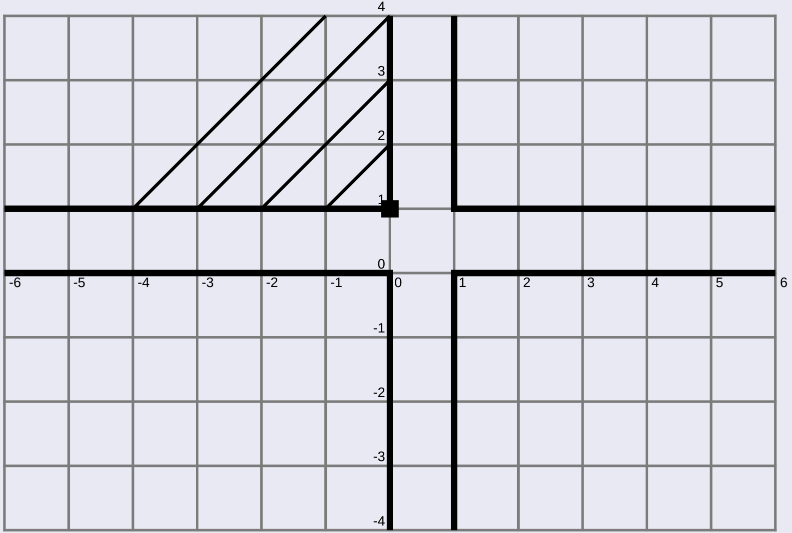
- Ordering inside a sector. We consider the shift from the corner ($a_i - 1$ for positive, $-a_i$ for negative), the shift consists of non-negative numbers)
- An ordering matrix (non-degenerate). To compare two shift vectors, multiple them by the matrix and compare from first to last.
- What to have as the first rows in the matrix?
- Normally first there is a row of 1, meaning we compare the total shift from the corner, complexity.
- Then there goes a row of 1 corresponding to negative (FIRE) or positive (?) indices.

Ordering matrix used by FIRE

Suppose we have 7 indices and a sector $\{-1, 1, 1, 1, 1, -1, -1\}$ then we get such a matrix.

$$
\begin{array}{l}
\{\{1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1\},\\
\ \{1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1\},\\
\ \{0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1\},\\
\ \{0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0\},\\
\ \{0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0\},\\
\ \{0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0\},\\
\ \{0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1\}\}
\end{array}
$$

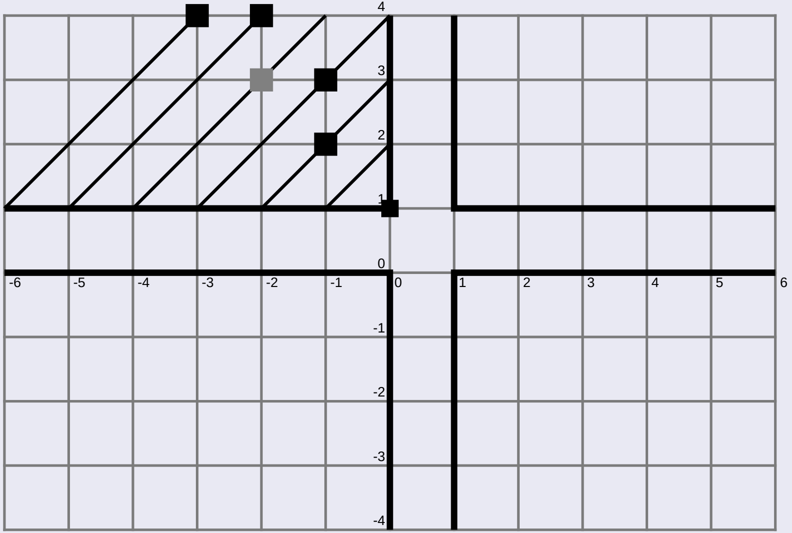
Integration by part relations (sectors and corner points)



Reduction strategy in a sector

- The complexity can be measured in two numbers, total shift for positive indices and total shift for negative indices.
- I will replace $(0, n)$ with $(1, n)$ and $(n, 0)$ with $(n, 1)$ since those IBPs will also have to be generated.
- First generate IBPs in points of complexity $(1, 1)$ (and lower), solve them.
- Go to $(2, 1)$ and $(1, 2)$ or only one of those
- Go to $(3, 1)$, $(2, 2)$ and $(1, 3)$ or a part of those
- Until you reduce all your integrals

Level dependency in a sector



Reduction strategy

- Point of same total complexity but differing “dot” and “numerators” complexity can be reduced in parallel
- Sometimes you need to choose other irreducible integrals - preferred integrals change the ordering
- If starting from higher sectors, expressions in lower sectors have to be masked
- Not all IBPs are needed, some can be thrown away... or even the smallest possible set left if there is already a run with substituted variables

FIRE6

- FIRE: one of the public available reduction programs
- No private version at the moment
- Current version: 6.2
- `https://arxiv.org/abs/1901.07808`, submitted to CPC
- Other public programs: AIR (Anastasiou, Lazopoulos, 2004), Litered (Lee, 2013), Reduze (Manteuffel, Studerus, 2009), Kira (Maierhöfer, Usovitsch, Uwer, 2017)

FIRE6 - get it

- Distributed via bitbucket
- Either binary packages
  `bitbucket.org/feynmanIntegrals/fire/downloads/`
- Or download the sources and compile (recommended)
- git clone https://bitbucket.org/feynmanIntegrals/fire.git

FIRE6 - build it

- ./configure (or with options)
- ./configure --enable_zlib --enable_snappy --enable_lthreads --enable_tcmalloc --enable_zstd
- make dep
- make
- make test

FIRE6 - basic approach

- Create a start file in Mathematica
- Build LiteRed diagram
- Convert LiteRed files to lbases
- Run the reduction in c++
- Use tables in Mathematica (can now be converted to rules by Tables2Rules)

FIRE6 - stability

- Thoroughly tested, should be much more stable than FIRE5.2
- Tests that are run for each pull request to dev or master
- Most crashes are related to no RAM left, tries to inform the user
- #clean to clean the semaphores on a machine where were killed jobs
- #storage to store intermediate result for the cases of hardware failure, reboots or changing thread options

FIRE6 - parallelization

- #threads and #sthreads for separate work in different sectors
- #fthreads for the number of fermat workers (consider the separate fermat mode)
- #lthreads to enable parallization inside sectors (only forward now)
- prime number approach (Manteuffel, Schabinger, 2016)
- multiple computers for one reduction
- master splitting (Chawdhry, Lim, Mitov, 2018)

FIRE6 - prime version

- Give values to variables by #variables d->100 in config or the variables option
- Set a prime number, call the FIRE6p binary
- Run multiple jobs with a script or the FIRE6_mpi launcher on a cluster
- Reconstruct tables in Mathematica
- Alternative way - use the information from the prime run but no reconstruction (#masters for the list of masters, #*hint* for the list of IBPs that should be used)

FIRE6 - reconstruction

- First there goes the reconstruction from integers modular p to rational numbers;
- Then a rational function has to be reconstructed. There are two basic ways:
- Newton formula
  $f(x) = c0 + (x - x_0)(c_1 + (x - x_1)(c_2 + \ldots)\ldots)$
- Thiele formula
  $f(x) = c0 + (x - x_0)/(c_1 + (x - x_1)/(c_2 + \ldots)\ldots)$
- A recursive combination for multiple variables

FIRE6 - reconstruction

- How to proceed?
- 1 variable - direct Thiele
- 2 variables - the denominators should be better split to use Newton-Newton after reconstructing them
- Fix one variable, do Thiele, find the worst possible denominator
- Do the same for the other
- Knowing both, take their product, multiple by it and run Newton-Newton
- more variables - no reconstruction procedure at the moment

FIRE6 - multiple computers

- specify a connection port with #port
- launch the FLAME6 binary with the same config, it starts waiting
- launch FIRE6 binary, it prints it IP in a file
- FLAME6 connects to specified IP and port
- FIRE5 starts giving tasks both to local and to remote (via FLAME6) threads

FIRE6 - split masters

- Get the list of masters in a file (perhaps a prime number pass), can use Tables2Masters
- Use the #masters |n-m|file in the config or the -masters n-m option
- Run separate reductions where only a subset of masters in non-zero
- Use CombineTables to join the tables into a single file

FIRE6 - RAM

- #memory mode normally suggested for memory databases for sectors
- Anyway sector databases are stored to disk when not used
- The bigger #threads (or #sthreads), the more RAM is used
- Split master mode should decrease RAM usage on substitutions a lot
- Choose a suitable compressor. Snappy or ZStandard for polynomials, lz4 with the small mode for primes (or no compressor)

FIRE6 - extra masters?

- Extra master integrals are a serious problem for the reduction
- Most extra masters are removed by LiteRed - FIRE6 uses internal symmetries also
- Recent discovery. Not all rules are useful! - do not mix equal masters with different sector nature!
- Some more relations if found can be added to .rules
- pos_pref option to use more LiteRed internal symmetries if needed

FIRE6 - master integral choice

- We prefer dots, not numerators, since more symmetries can be found
- Controlled by the pos_pref option (can be negative now too)
- Specify #preferred masters
- The goal is to have denominators split into a function of d and other variables
- The d part should better be a product of linear terms

Reduction - unanswered questions

- How should the propagators be chosen? Seems very important!
- How to find the best bases? Influences reduction a lot, important to estimate denominators for the prime approach
- When going a sector down, how much should the complexity (shift from the corner) of an integral increase? Ideal is 1, but sometimes it is 2. Can we avoid it?

FIRE6 - plans

- Keep on improving algorithms for modular reduction
- More parallelization
- More extensive use of LiteRed to use partial reduction rules


EZT A FÁT
RICHARD FEYNMAN
NOBEL-DÍJAS FIZIKUS
ÜLTETTE
1972. JÚNIUS 13-ÁN