FORM DEVELOPERS' WORKSHOP 2026

FORM
when things become difficult

Computing amplitudes and integrals with FORM, from a user's perspective

Lorenzo Tancredi (TUM)
Nikhef, 25 June 2026

STARTING POINT

My perspective

A FORM user, not a FORM developer.

Users with physics goal:

Priorities sometimes look different when seen from the point where a calculation is stuck.

I keep coming back to FORM at the boundary where the standard pipeline stopped being enough.

- Real cutting-edge problems, not “benchmarks”
- Flexibility for user interventions is better than full automation

THESIS

The central message

> **FORM is the program we want to turn to when things do not work.**

Reliable

runs for days, reproducible, robust

Fast

term-level algebra at serious scale

Flexible

the missing algorithm stays in the user's hands

without writing a private algebra system in C++/Python…

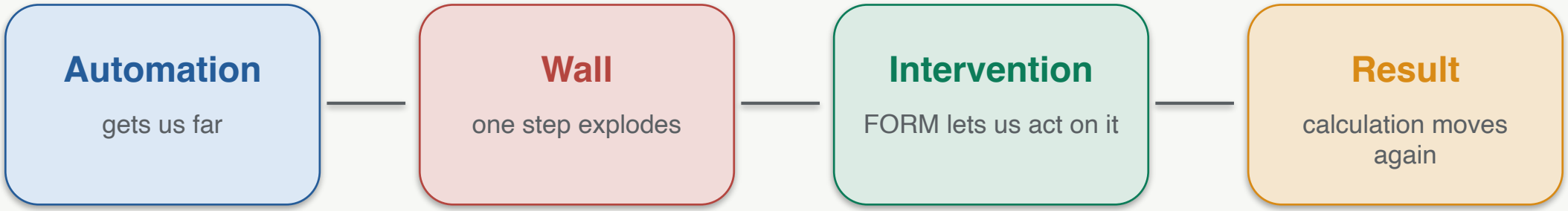
PIPELINE

Where FORM enters

The algebra-heavy middle of an amplitude calculation.

Full-blown brute force approach = expression swell out of control

The recurring pattern



The important point: the user must own the strategy!

TIMELINE

A decade of examples

Three moments where FORM mattered.

state of the art … ?

2013-2015

$pp \to ZZ/WW$

partial fractions many scales

2015-2016

$gg \to Hj$

difficult massive IBP sectors

2021-2023

massless $2 \to 3$

huge rational functions

2023-2026

massive $2 \to 3$

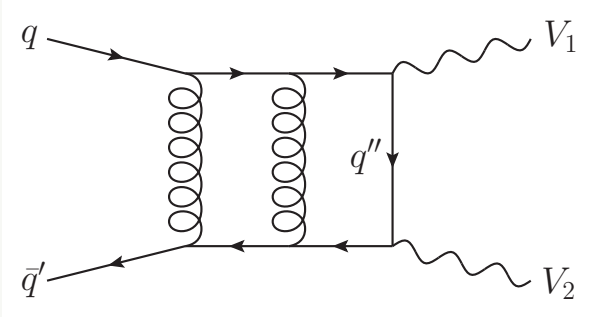
huge rational functions
roots
elliptic sectors and beyond …

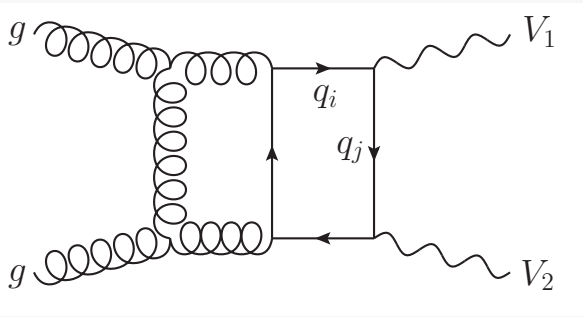
2013-2015

Case study 1: VV amplitudes

Two-loop QCD corrections to $q\bar{q} \to VV$ and $gg \to VV$

arXiv:1503.04812 | arXiv:1503.08835





Problem

A genuinely multiscale rational structure: many invariants, many denominators, hard in Mathematica at the time *(s,t,m1,m2)*

Need

Control the expression swell, simplify amplitude fast and produce stable numerical code: **vvamp**

What FORM made possible

Custom partial fractioning (PF) as a user-level algorithm.

Before

Huge multiscale rational expressions

FORM code

Compact expression in terms of GPLs

- Not generic all purpose simplification
- A targeted partial fractioning routines (iterated univariate!)
- *Choosing right order of variables, PF linear, quadratic and cubic denominators one variable at the time did the trick!*

Lesson from VV

FORM way of thinking exposed the right structure:

today everybody tries to do partial fractioning in very sophisticated ways

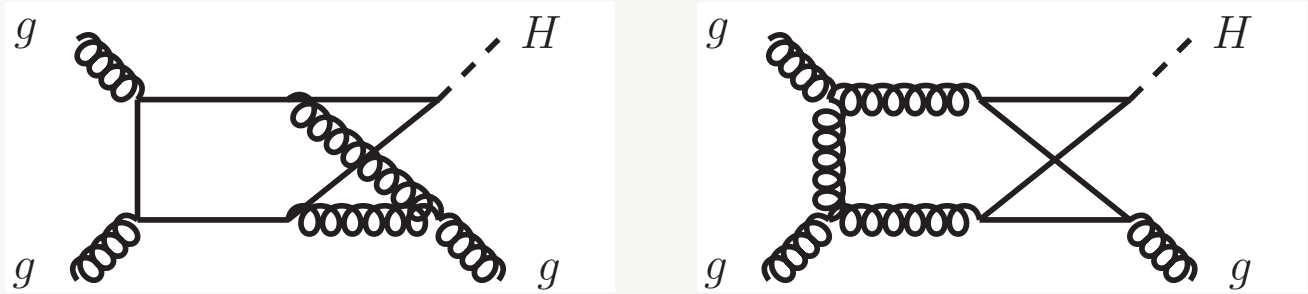
- **Programmable:** the missing manipulation could be implemented.
- **Deterministic:** the result was inspectable and reproducible (it does what you want)
- **Scalable:** the expressions could be pushed through *scaling it on larger machines*

2015-2016

Case study 2: gg -> H+j

Two-loop amplitude with bottom-mass effects.

arXiv:1610.03747



Physics target

Top-bottom interference effects in Higgs production at finite transverse momentum.

Computational wall

Some difficult sectors in the IBP reduction could not be completed automatically at the time.

CASE STUDY 2

FORM intervention

Finish the hard algebraic sectors without re-inventing the wheel

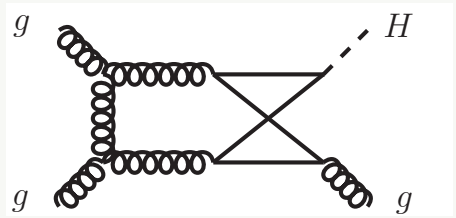
Automated IBP

mostly works

FIRE + Reduze2
(no Kira at the time :-))

Difficult sectors

NPL massive stuck



FORM surgery

controlled

- Solve IBPs for the top sector in FORM terms of *unreduced subsector*
- To achieve that, apply symmetries and shifts to subsector to minimize them in number
- Reduze+FIRE for subsectors, combined in Mathematica!

Reduction closed

usable

Not elegant, but controlled: often the only way out for cutting edge problems

Lesson from $gg \to Hj$

FORM's value is especially where *automation fails* (and things become interesting :-))

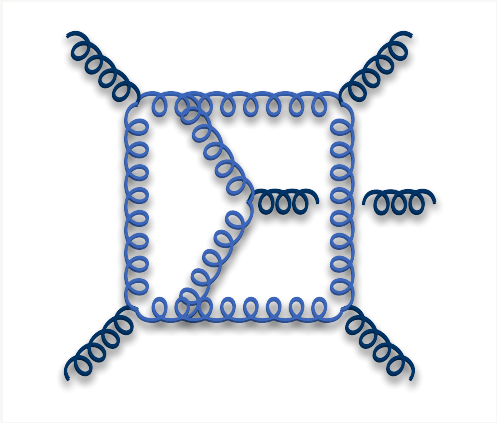
- When the standard pipeline fails, to implement fixes fast and flexibly
- A posteriori, the problem was easy: solve top-sector IBPs ONLY
- Standard codes (Reduze, FIRE) would not do this because programmed to solve the full problem at once (again, full automation is not always what we need!)

2021-2023

Case study 3: five-point amplitudes

Diphoton+jet and all five-parton scattering at two loops.

arXiv:2105.04585 | arXiv:2311.09870



New scale

Five-point, two-loop, full-color and helicity information.

New bottleneck

Huge rational functions coming from IBP reduction and amplitude assembly.

- **Finred** (Manteuffel) produced IBPs up to 2GB each
- Possible approach: **finite fields** to reconstruct only the finite remainder in terms of special functions
- Issue: **120 permutations** at once numerically, possible but **not for free!**

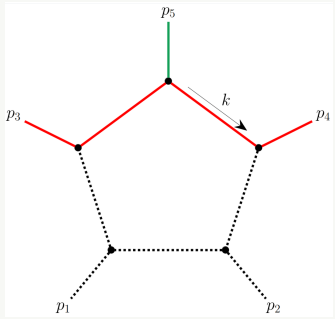
- We used **FORM** to **partial fraction** amplitudes and IBPs, then included one in the other and **partial fractioned** again
- In addition, used **PF decomposition** to find **relations among rational functions,** solving linear systems in FORM

2023-2026

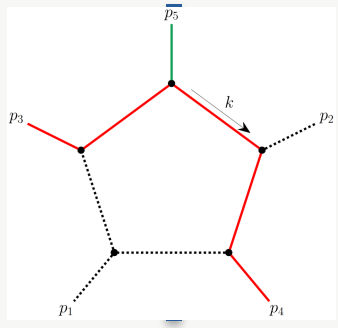
Case study 4: *massive* five-point amplitudes

top-antitop production with a massive (vector) boson

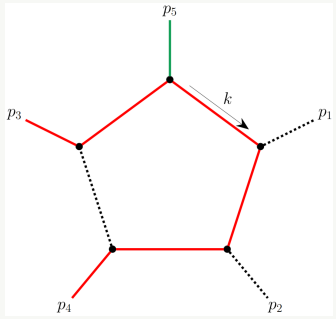
arXiv:2312.10015 | arXiv:2502.14952 | arXiv:2606.09503



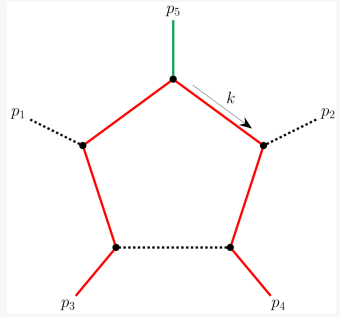
(a) Topology A



(b) Topology B



(c) Topology C



(d) Topology D

New bottleneck

Masses at 5 points generate a new level of complexity. Already at one loop, ***polynomials of huge degree in 7 variables***

- **Finite Field techniques** cannot handle them easily: *reconstruction algorithms scale very badly beyond 5 independent variables*
- **FORM** could handle some of the simplification of the rational functions, not all (maxtermsize nightmare :))
- **FERMAT** was actually the only code able to perform simplifications to the end. I know we have FLINT now, but can FORM learn something from FERMAT?

New lesson

Symbolic FORM4.3 could already compete with modern FF based techniques!

Status for massive five-point amplitudes

Complexity moves one step further

- **Immense rational prefactors:** even if they can be reconstructed, largely useless
- **“Obvious” patch:** polynomials are simpler if we reconstruct only coefficients of special functions.
- **Not enough:** expressions still very difficult to handle, it is easier to solve IBPs numerically point by point and assemble result numerically.

What about Partial Fractions? How to do this in general? Hard question…

The problem changed

The hard questions in the five-point era:

- Can we produce the rational coefficients in a form that is usable?
- Does it even make sense to produce them analytically?
- Is it better to have a reliable routine to evaluate them numerically at each point?
- Are finite-field based methods still competitive when the number of scales increases?

Can FORM help here?

STRENGTHS

What FORM is very good at

Huge expressions

scales well with more terms

Exact algebra

symbolic manipulations

User flexibility

surgical solutions

Speed

built for large symbolic jobs

Where users still feel pain

Affectionate users like me

Large rational functions

function arguments, factorization, polyratfun …
“Increase MaxTermSize” is my worse nightmare

Sometimes just reading in a KB-large factorised expression can kill FORM completely…

New users

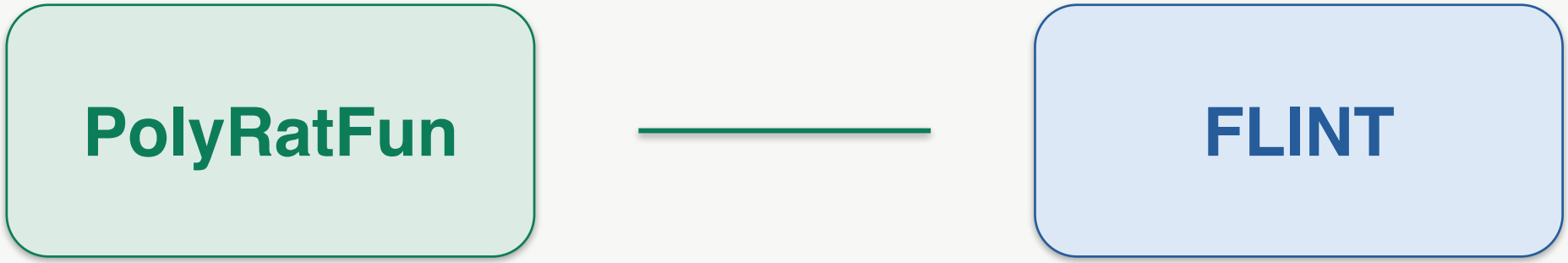
Documentation and syntax

it’s difficult to motivate young people to use FORM due to *syntax* and “strange” *limitations*

Someone used to mathematica or similar, finds limitations like MaxTermSize unappealing

PolyRatFun was a major step

And FORM 5 + FLINT is a great improvement, for what I can see



- Having lived in the pre-polyratfun era, this was truly a revolution

But the bottleneck remains

> **PolyRatFun still lives inside a term.**

Numerator / denominator too large ——— **MaxTermSize wall**

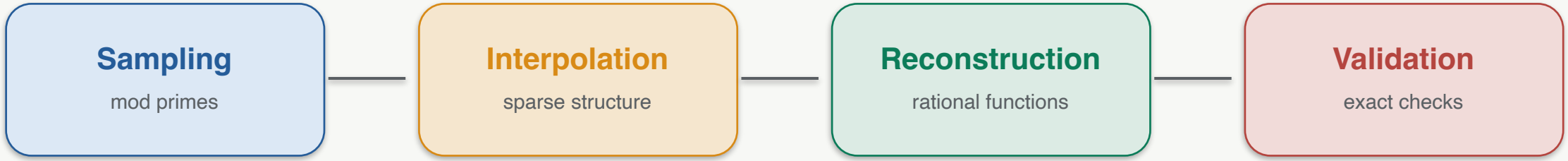
Rational objects often outgrow a single FORM term

If FORM cannot do something that Mathematica can do, this is a problem…

NEXT WORKFLOW

Finite fields and reconstruction

Modern amplitude calculations increasingly live here.



FORM has some important pieces. The missing part is an integrated, documented user workflow.

What is missing?

CLOSING

Final message

> **When everything works, we may get away without FORM**

> **When things get nasty, FORM is the tool we turn to!**

Thank you!