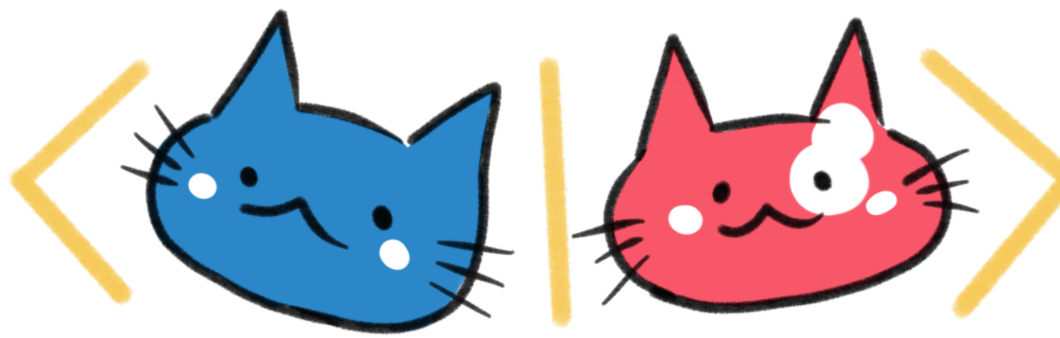
Computing intersection numbers with a rational algorithm

University of Turin, 22 March 2023



Gaia Fontana-University of Zürich

In collaboration with Tiziano Peraro

University of Zurich[UZH]

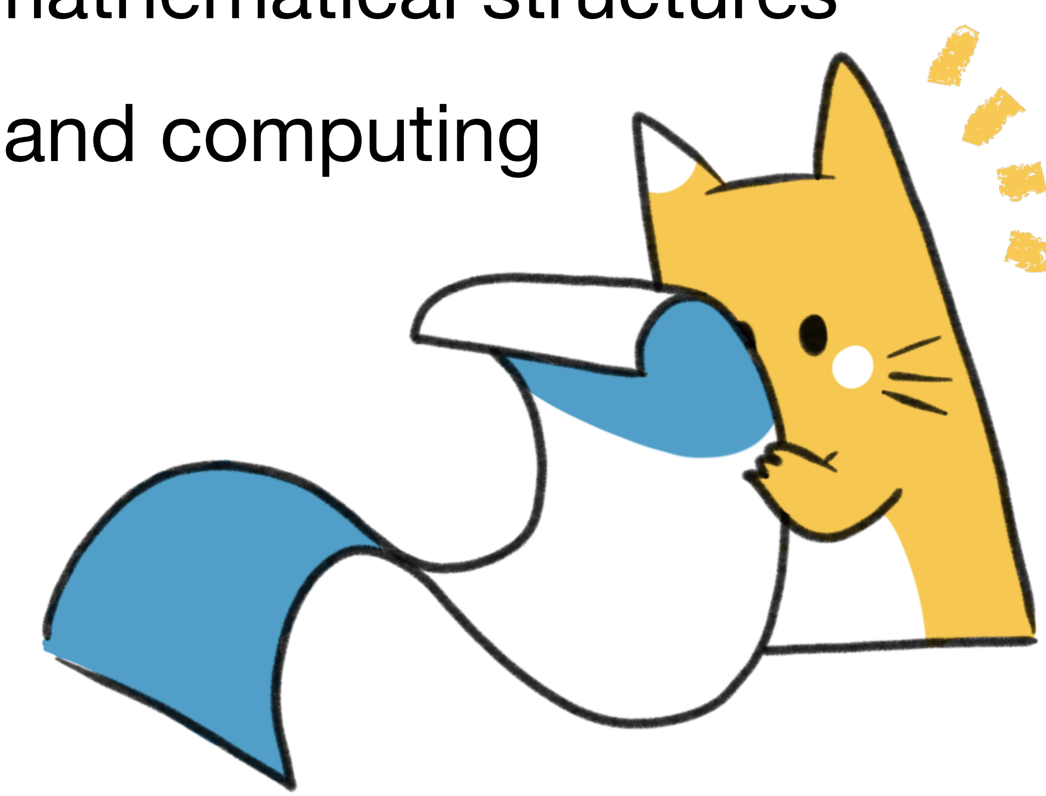
erc

Outline

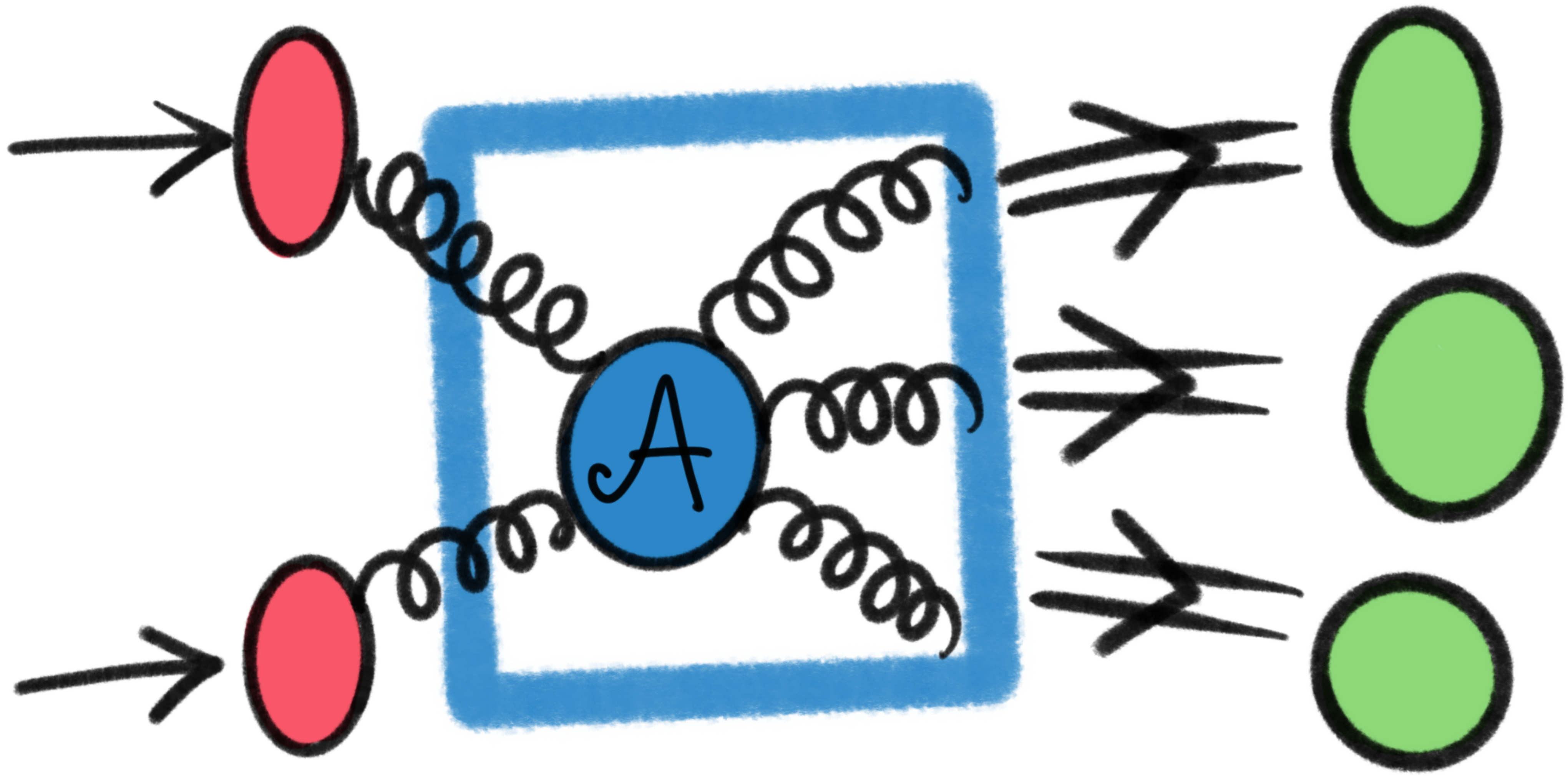
* why we do what we do (that is, decomposing Feynman integrals)
* new ways of doing that: a fast dive into intersection theory
* how can we make it better? rational algorithms!
* proof-of-concept implementation over finite-fields
* some examples that it actually works

Motivation

- LHC-HL upgrade: observables at %-level accuracy
  - new physics searches
  - testing SM
  - important: high-multiplicity and masses
- understanding of amplitudes and Feynman integrals
  - %-level ~ at least NNLO ~ $\geq$2 loops
  - Exploiting physical and mathematical structures
  - connections with maths and computing



Recipe for a theoretical prediction



many ingredients:

- PDFs to describe the proton structure
- hard scattering
- radiation and evolution to hadronic states

Precision calculation of perturbative scattering amplitudes

* at the core of theoretical predictions
* rich and interesting mathematical structures ;)
* a combination of

$$\mathscr{A} = \int_{-\infty}^{+\infty} \left(\prod_{i=1}^{\ell} d^d k_i \right) \frac{\mathscr{N}}{z_1 z_2 z_3 \cdots}$$

polynomial numerator

quadratic denominators

rational function in the components of loop momenta

⇒ calculation of integrals over the loop momenta

Computing loop amplitudes

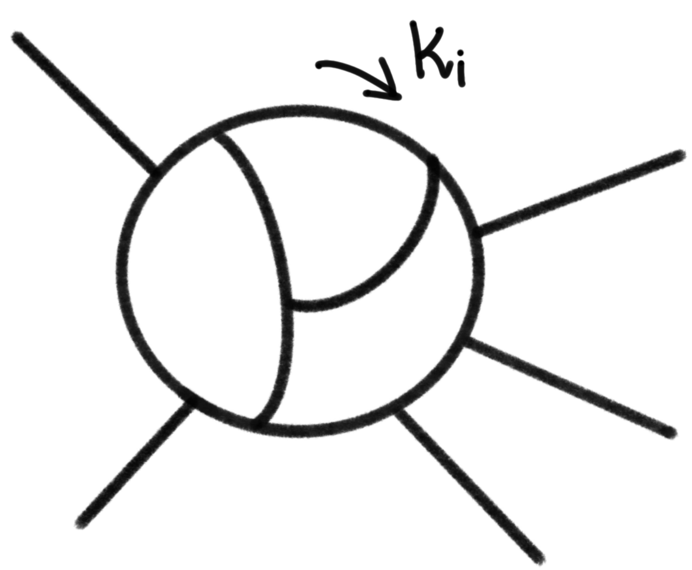
Amplitudes as a linear combination of Feynman integrals

$$\mathscr{A} = \sum_i a_j I_j$$

(rational coefficients with dependence on particle content)

(integrals in a "nice"/ standard form)

$$I_i[\alpha_1, \ldots, \alpha_n] = \int_{-\infty}^{+\infty} \left(\prod_{i=1}^{\ell} d^d k_i \right) \frac{1}{z_1^{\alpha_1} \ldots z_n^{\alpha_n}}$$



inverse propagators

$z_i = k_i^2 - m_i^2$

Not all are linearly independent!

Reduction to master integrals

why?

* extremely large number of integrals contributing to an amplitude
* properties/symmetries of an amplitude manifest only after the reduction
* important for the calculation of the integrals

Reduction into a basis of linearly independent master integrals $\{G_j\} \subset \{I_j\}$

$$I_j = \sum c_{jk} G_k$$

$\{G_j\}$ = minimal linearly independent set

*** this talk!**

Laporta algorithm

Feynman integrals in dimensional regularization obey linear relations,
e.g. Integration By Parts identities

IBPs

$$\int \Big(\prod_{i=1}^{\ell} d^d k_i\Big) \frac{\partial}{\partial k_i^\mu} \Big(\frac{v_j^\mu}{z_1^{\alpha_1} \ldots z_n^{\alpha_n}} \Big) = 0, \quad v^\mu = \begin{cases} p_i^\mu \text{ external} \\ k_i^\mu = \text{ loop} \end{cases}$$

\+ adding also Lorentz Invariance ids, symmetry relations, …

⇒ reduction as solution of a large
and sparse system of identities

Computation of MIs

* my PhD :)

Can be done

* analytically in terms of special functions (MPLs, elliptic functions, …)
* numerically (Sector decomposition, AMFlow)

most effective method is **Differential Equations (DE)**

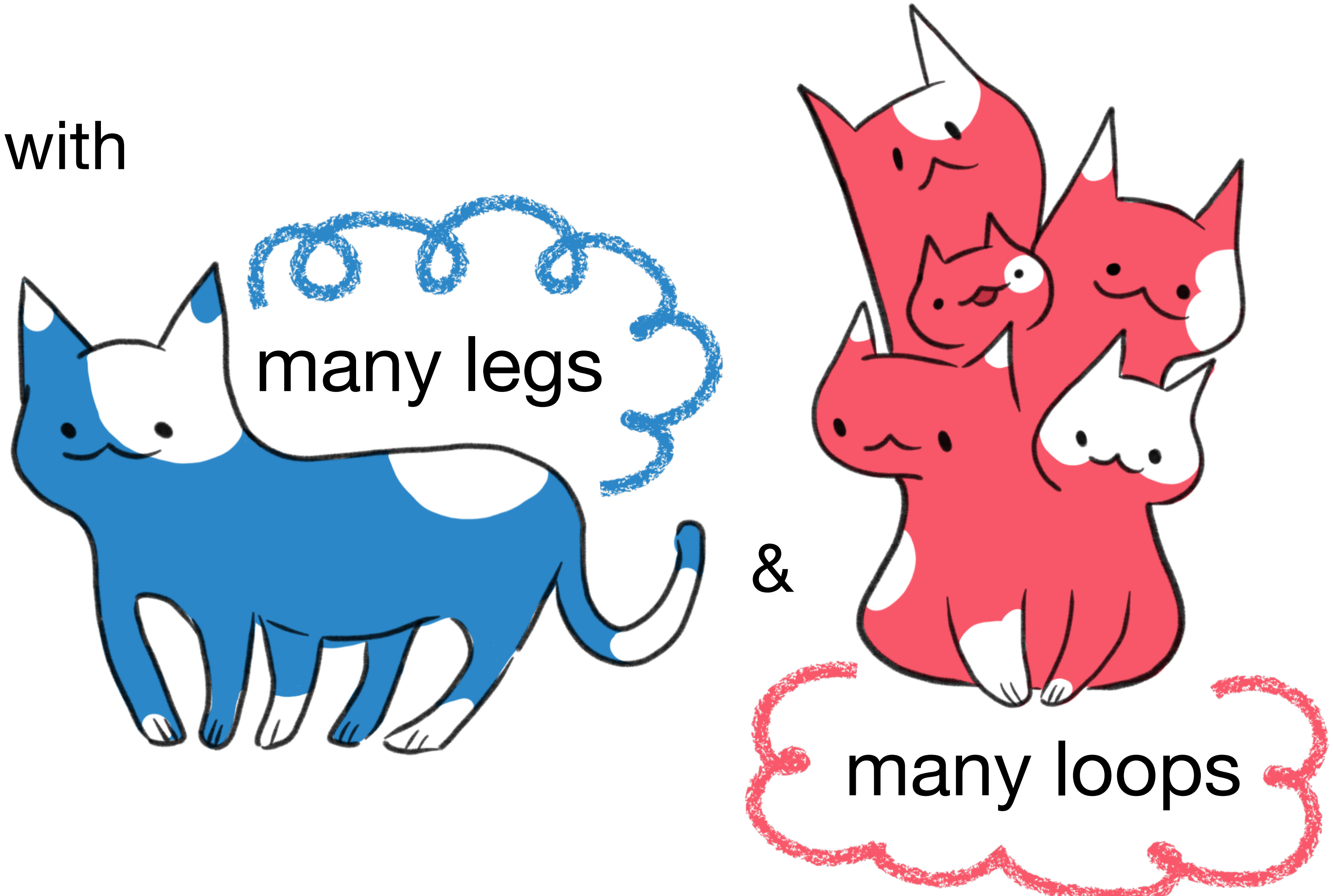
* derivative of MIs with respect to external invariants and/or masses
* reduce it again to MIs
* obtain a system of DEs for the MIs

$$\partial_x G_i = M_{ij} G_j, \;\; x = s_{ij}, m_i$$

* can be (often) put in a canonical form → Feynman integrals solved as iterated integrals over a fixed kernel

Algebraic complexity

processes with



give rise to HUGE intermediate expressions

drawbacks of Laporta procedure

* very large system → computational bottleneck
* algebraic structure of FI not manifest

Looking for other ways...

Intersection theory

- allows for a direct decomposition
- exploits the vector space structure obeyed by Feynman integrals

we consider n-folds integrals in $\mathbf{z} = (z_1, \ldots, z_n)$

integrals

$$|\varphi_R\rangle = \int dz_1 \ldots dz_n \frac{1}{u(\mathbf{z})} \varphi_R(\mathbf{z})$$

"dual" integrals

$$\langle \varphi_L | = \int dz_1 \ldots dz_n u(\mathbf{z}) \varphi_L(\mathbf{z})$$

with

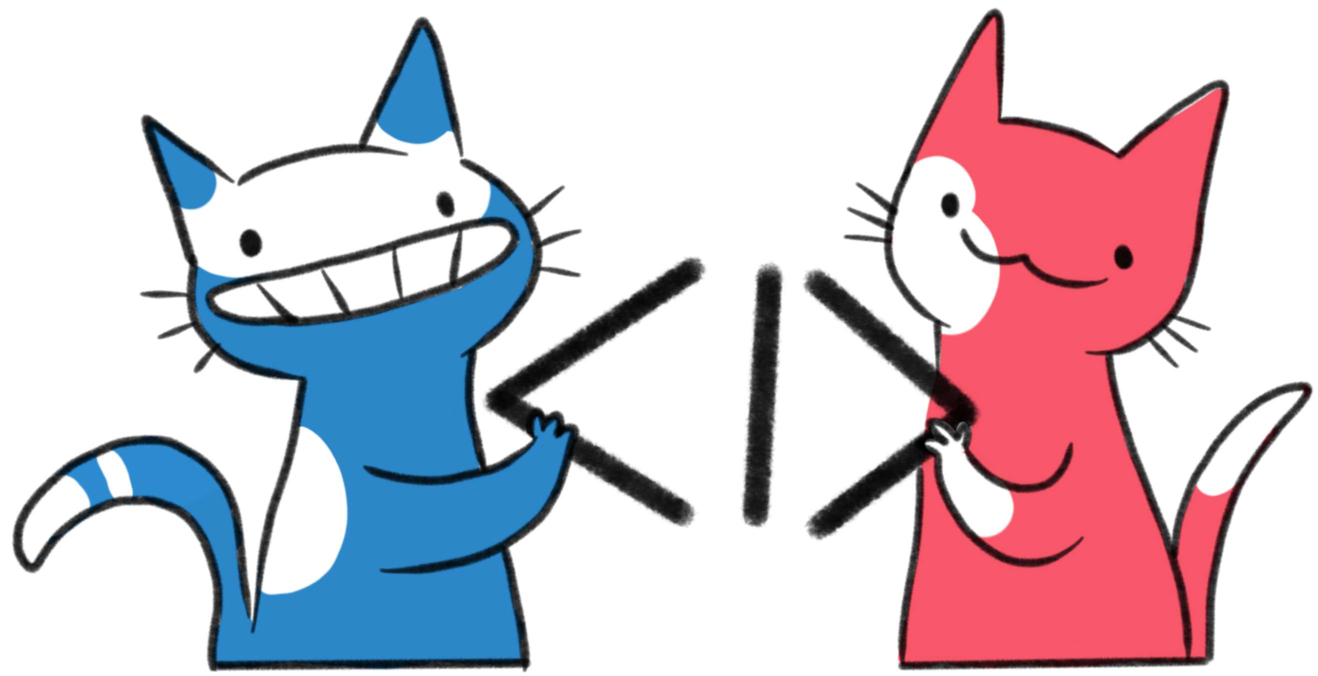
- φ_L / φ_R rational functions
- $u(\mathbf{z}) = \prod_j B(\mathbf{z})_j^{\gamma_j}$, $\begin{cases} \gamma_j \text{ generic exponents} \\ B_j \text{ polynomials} \end{cases}$

Intersection numbers

calculation of scalar products
between left and right integrals

Mastrolia, Mizera (2018)

$$\langle \varphi_L | \varphi_R \rangle$$



they're rational!

Vector space characterized by:

* Dimension ν
* Basis $| e_i^{(R)} \rangle$ and dual basis $\langle e_i^{(L)} |$
* Scalar product: **intersection number**

Change of representation

Baikov change of vars

$k_j \to z_j$

Baikov (1996)

$$I[\alpha_1, \ldots, \alpha_n] = \int \Big(\prod_{i=1}^{\ell} d^d k_i \Big) \prod_{j=1}^{n} \frac{1}{z_j^{\alpha_j}} \to \int dz_1 \ldots dz_n \, B^\gamma \prod_{j=1}^{n} \frac{1}{z_j^{\alpha_j}} = |\varphi_R\rangle$$

with

$$u(\mathbf{z}) = B^{-\gamma}, \;\; \gamma = \frac{d-E-L-1}{2}$$

$$|\varphi_R\rangle = \int dz_1 \ldots dz_n \frac{1}{u(\mathbf{z})} \varphi_R(\mathbf{z})$$

$$\varphi_R(\mathbf{z}) = \frac{1}{z_1^{\alpha_1} \ldots z_n^{\alpha_n}}$$

identifications

- $|\varphi\rangle$ generic vector
  → Feynman integral to reduce
- $\{|e_i^{(R)}\rangle\}_{i=1}^{\nu}$ basis vectors
  → master integrals

decomposition of integrals as

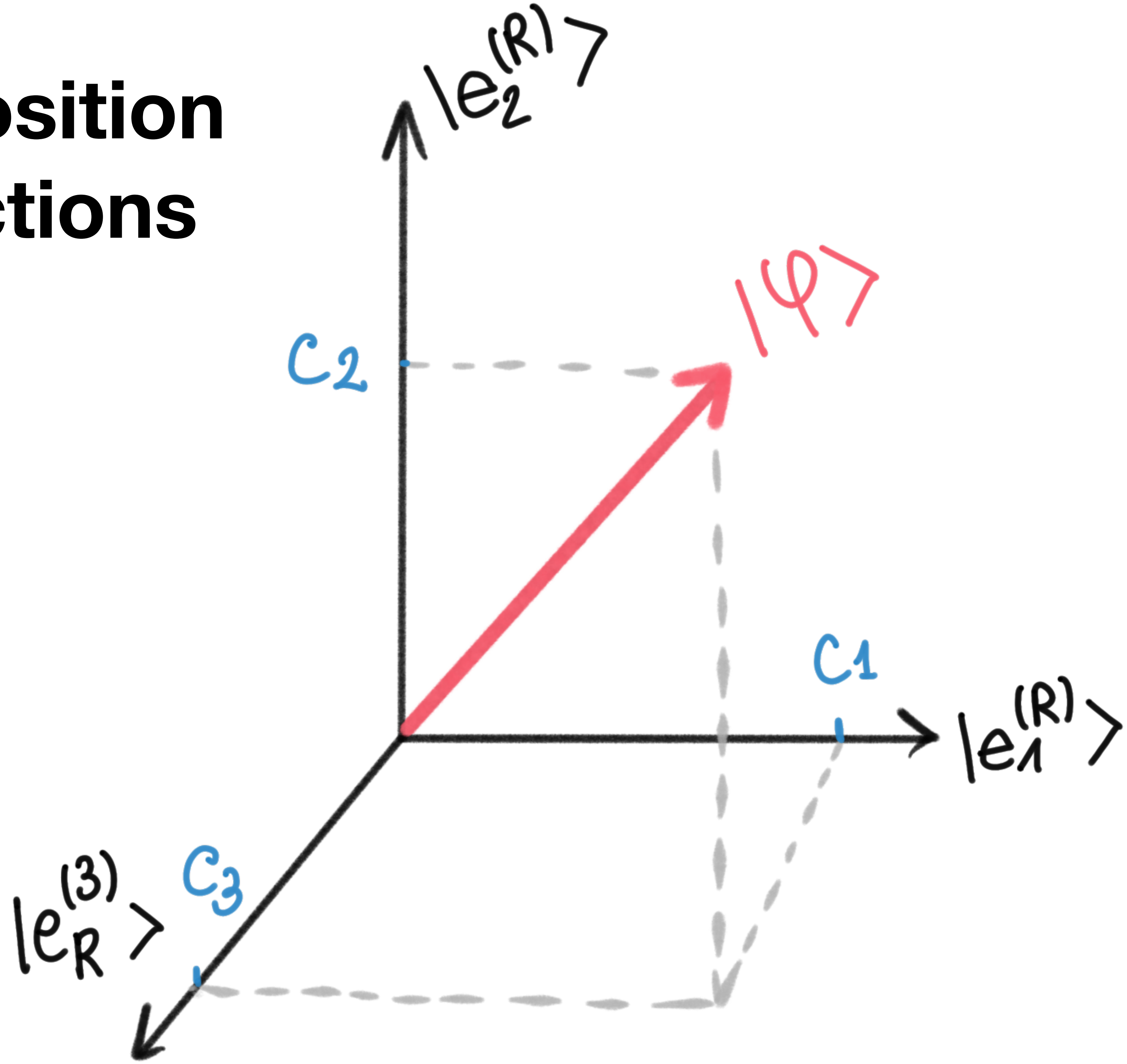
$$|\varphi_R\rangle = \sum_{i=1}^{\nu} c_i^{(R)} |e_i^{(R)}\rangle \qquad c_i^{(R)} = \sum_{j=1}^{\nu} (\mathbf{C}^{-1})_{ij} \langle e_j^{(L)} | \varphi_R \rangle$$

- similar formulae for dual integrals

where we introduced the metric

$$\mathbf{C}_{ij} = \langle e_i^{(L)} | e_j^{(R)} \rangle$$

decomposition as projections



- $|\varphi\rangle$ generic vector → Feynman integral to reduce
- $\{\,|e_i^{(R)}\rangle\}_{i=1}^{\nu}$ basis vectors → master integrals

reduction to master integrals

=

calculating the scalar products $\langle \varphi_L | \varphi_R \rangle$

computation of intersection numbers

Univariate algorithm

we have 1-fold integrals in the variable z

Frellesvig et al. (2019)

$$|\varphi_R\rangle = \int dz \frac{1}{u(z)} \varphi_R(z)$$

univariate intersection numbers

$$\langle \varphi_L | \varphi_R \rangle = \sum_{p \in \mathscr{P}_\omega} \text{Res}_{z=p}(\psi \varphi_R)$$

where ψ is the local solution of

$$(\partial_z + \omega)\psi = \varphi_L \qquad \omega \equiv \frac{\partial_z u}{u}$$

around each $p \in \mathscr{P}_\omega$

$$\mathscr{P}_\omega = \{z \,|\, z \text{ is a pole of } \omega\} \bigcup \{\infty\}$$

Ansatz around p

$$\psi = \sum_{i=\text{min}}^{\text{max}} c_i (z-p)^i + O\Big((z-p)^{\text{max}+1}\Big)$$

- plug the ansatz in the differential equation $(\partial_z + \omega)\psi = \varphi_L$
- solve for the c_i

Intersection numbers are always rational functions of the kinematic invariants and of the dimensional regulator

Multivariate algorithm

- we have n-folds integrals in the variables $\mathbf{z} = (z_1, \ldots, z_n)$

$$|\varphi_R\rangle = \int dz_n \, |\varphi_R\rangle_{n-1}$$

- $|\varphi_R\rangle_{n-1}$ is an $(n-1)$-fold integral in $z_1, \ldots z_{n-1}$

$$|\varphi_R\rangle_{n-1} = \sum_{j=1}^{\nu_{(n-1)}} \varphi_{R,j} \, |e_j^{(R)}\rangle_{n-1}$$

basis of master integrals for the $(n-1)^{th}$ layer

we decompose $n-$fold integrals into a basis of $(n-1)-$fold master integrals

$$\langle \varphi_L | \varphi_R \rangle = \sum_{p \in \mathscr{P}_{\mathbf{\Omega}}} \mathrm{Res}_{z_n = p} \left(\psi_j \langle e_j^{(L)} | \varphi_R \rangle_{n-1} \right)$$

- where ψ_j is the local solution of

$$\partial_{z_n} \psi_j + \psi_i \mathbf{\Omega}_{ji} = \varphi_{L,j}, \quad (j = 1,\ldots,n)$$

- around each $p \in \mathscr{P}_{\mathbf{\Omega}}$

$$\mathscr{P}_{\mathbf{\Omega}} = \{ z_n \,|\, z_n \text{is a pole of } \mathbf{\Omega} \} \bigcup \{\infty\}$$

- and $\mathbf{\Omega}$ is defined via

$$\partial_{z_n} \langle e_i^{(L)} |_{n-1} = \sum_j \mathbf{\Omega}_{ij} \langle e_j^{(L)} |_{n-1}$$

- solved locally with the **ansatz**

$$\psi_i = \sum_{j=\text{min}}^{\text{max}} c_{ij} (z_n - p)^j + O\left((z_n - p)^{\text{max}+1} \right)$$

* integrands are rational
* intersection numbers are rational

BUT

* non-rational contributions in intermediate stages
* after taking the sum over all residues we see cancellations

non-rational terms in the poles of ω and $\mathbf{\Omega}$

$\mathscr{P}_\omega$ $\mathscr{P}_{\mathbf{\Omega}}$

* **computational bottleneck**
* **non-suitable for applications with finite-fields**

$p(z)$-adic series expansion

$p-$adic numbers

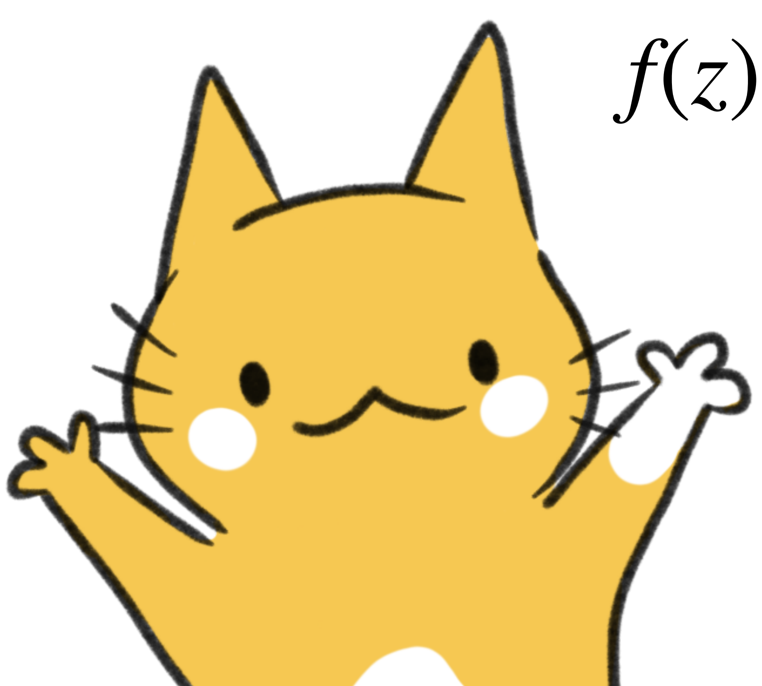
expansion of a rational number as series expansion of a prime number p with coefficients given by remainder of integer division

$p(z)-$adic functions

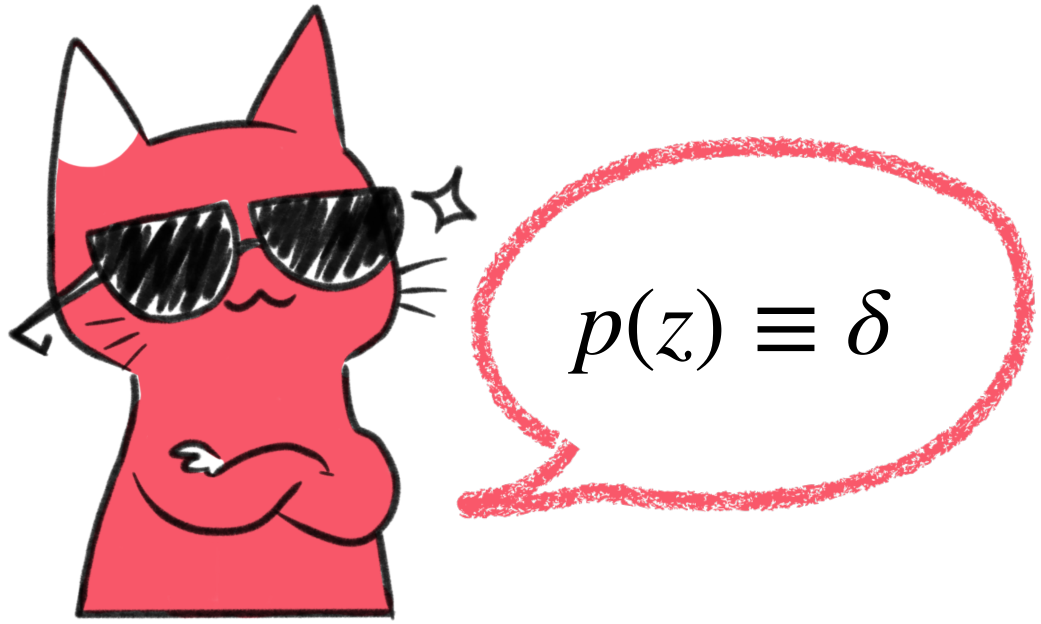
expansion of a rational function as series expansion of a prime polynomial $p(z)$ with coefficients given by remainder of polynomial division

def

$$f(z) = \sum_{i=\min}^{\infty} c_i(z)\, p^i(z) \qquad \begin{cases} c_i(z) = \sum_{j=0}^{\deg p-1} c_{ij}\, z^j \\ c_{ij} \in \mathbb{Q} \end{cases}$$



shortcut



- polynomial remainder w.r.t. $p(z) - \delta$, i.e. substituting $p(z)$ with δ

$$\lfloor f(z) \rfloor_{p(z)-\delta} \equiv f(z) \bmod p(z) - \delta$$

- expansion for $\delta \to 0$

$$\lfloor f(z) \rfloor_{p(z)-\delta} \Big|_{\delta \to 0} = \sum_{i=\text{min}}^{\text{max}} \sum_{j=0}^{\text{deg p}-1} c_{ij} z^j \delta^i + O\left(\delta^{\text{max}+1}\right)$$

Example: univariate algorithm

GF, Peraro (2022)

$$\langle \varphi_L | \varphi_R \rangle = \sum_{p(z) \in \mathscr{P}_\omega[z]} \langle \varphi_L | \varphi_R \rangle_{p(z)}$$

(similar for multivariate case)

summing over all $p(z) \in \mathscr{P}_\omega[z]$

$\mathscr{P}_\omega[z] =$ { factors of the denominator of ω } $\bigcup$ { ∞ }

- Each addend of the form $\langle \varphi_L | \varphi_R \rangle_{p(z)}$ is the sum of all contributions to the intersection number coming from the roots of $p(z)$

- $\langle \varphi_L | \varphi_R \rangle_\infty$ is computed as the contribution at $p = \infty$ with the "standard" algorithm to solve $(\partial_z + \omega)\psi = \varphi_L$

- we make an ansatz of the form

$$\psi = \sum_{i=\text{min}}^{\text{max}} \sum_{j=0}^{\text{deg p}-1} c_{ij} z^j p(z)^i + O\left(p(z)^{\text{max}+1}\right)$$

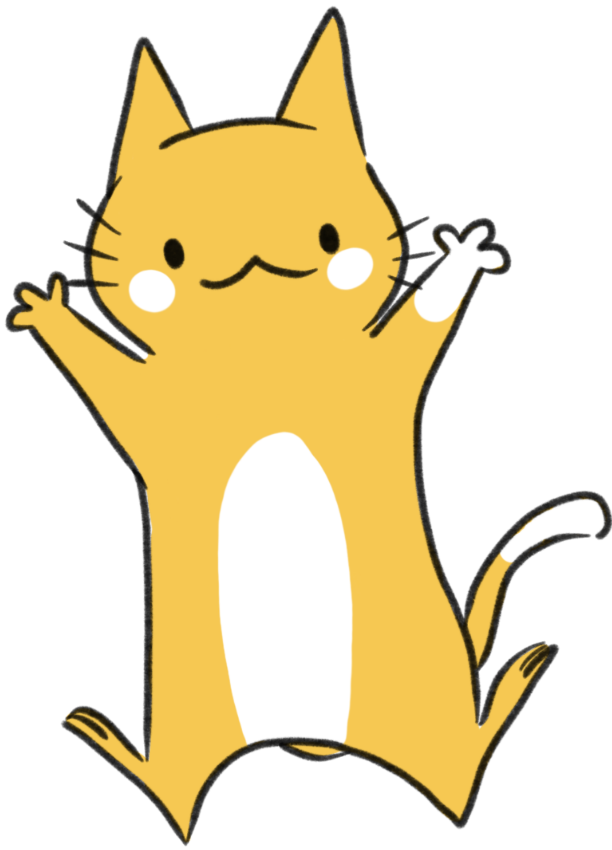
- we multiply the solution by φ_R

$$\psi\varphi_R = \sum_{i}^{-1} \sum_{j=0}^{\text{deg p}-1} \tilde{c}_{ij} z^j p(z)^i + O\left(p(z)^0\right)$$

- by the univariate global residue theorem

Weinzierl (2021)

$$\langle \varphi_L | \varphi_R \rangle = \frac{\tilde{c}_{-1,\text{deg p}-1}}{l_c}$$



Finite-fields and rational reconstruction

Kant (2014)
Von Manteuffel, Schabinger (2014)

* Dealing with algebraic complexity
  * large intermediate expressions
  * intermediate steps more complicated than final result

idea: reconstruct analytic results from numerical evaluations

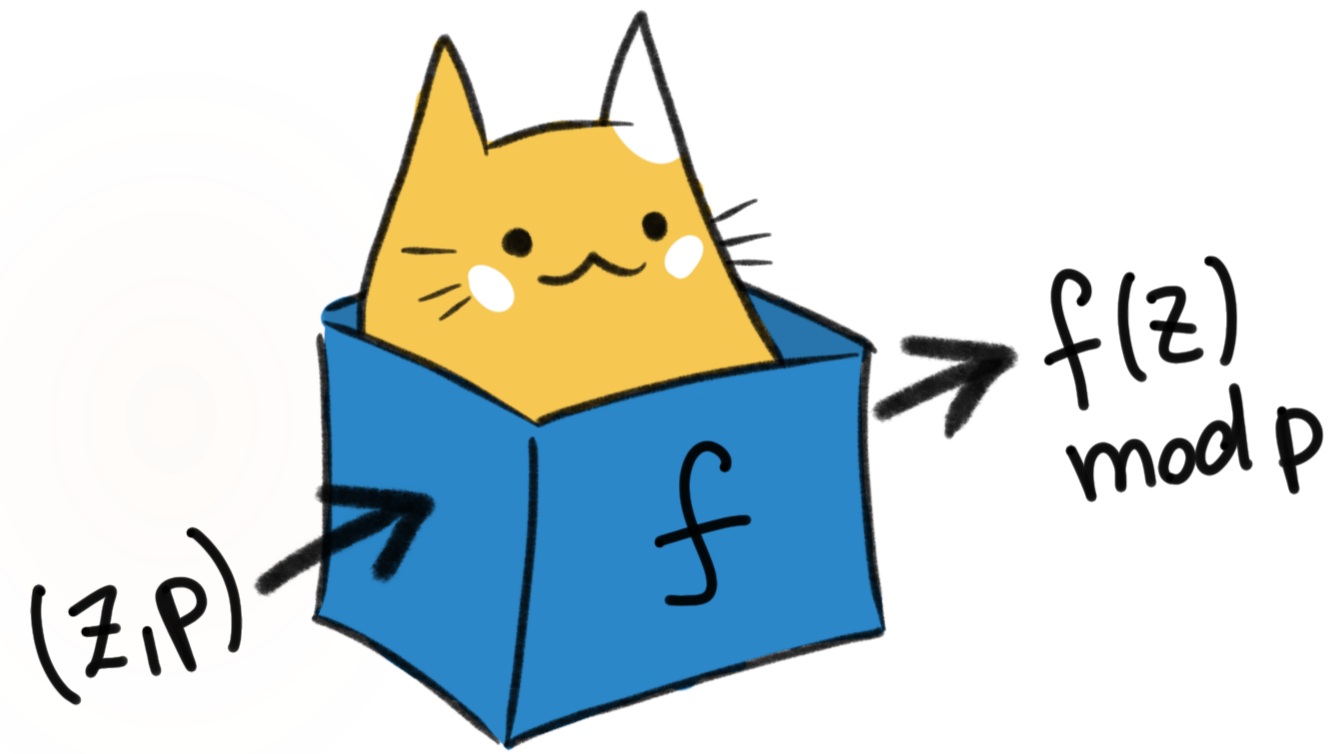
* evaluations over finite-fields $\mathscr{Z}_p$ (computing modulo a prime p)
* use machine-size integers $p < 2^{64}$ (fast and exact)
* collect numerical evaluations and infer analytic results

* rational algorithms
* parallelizable
* examples: *FinRed*, *FiniteFlow*, *FireFly+Kira*, *Caravel*

how do they work

We have a rational function $f(z)$ whose analytic form is not known

* black box interpolation: numerical procedure to evaluate $f(z)$ at $\mathbf{z}$ over the field $\mathscr{Z}_p = (0,\ldots,p-1)$

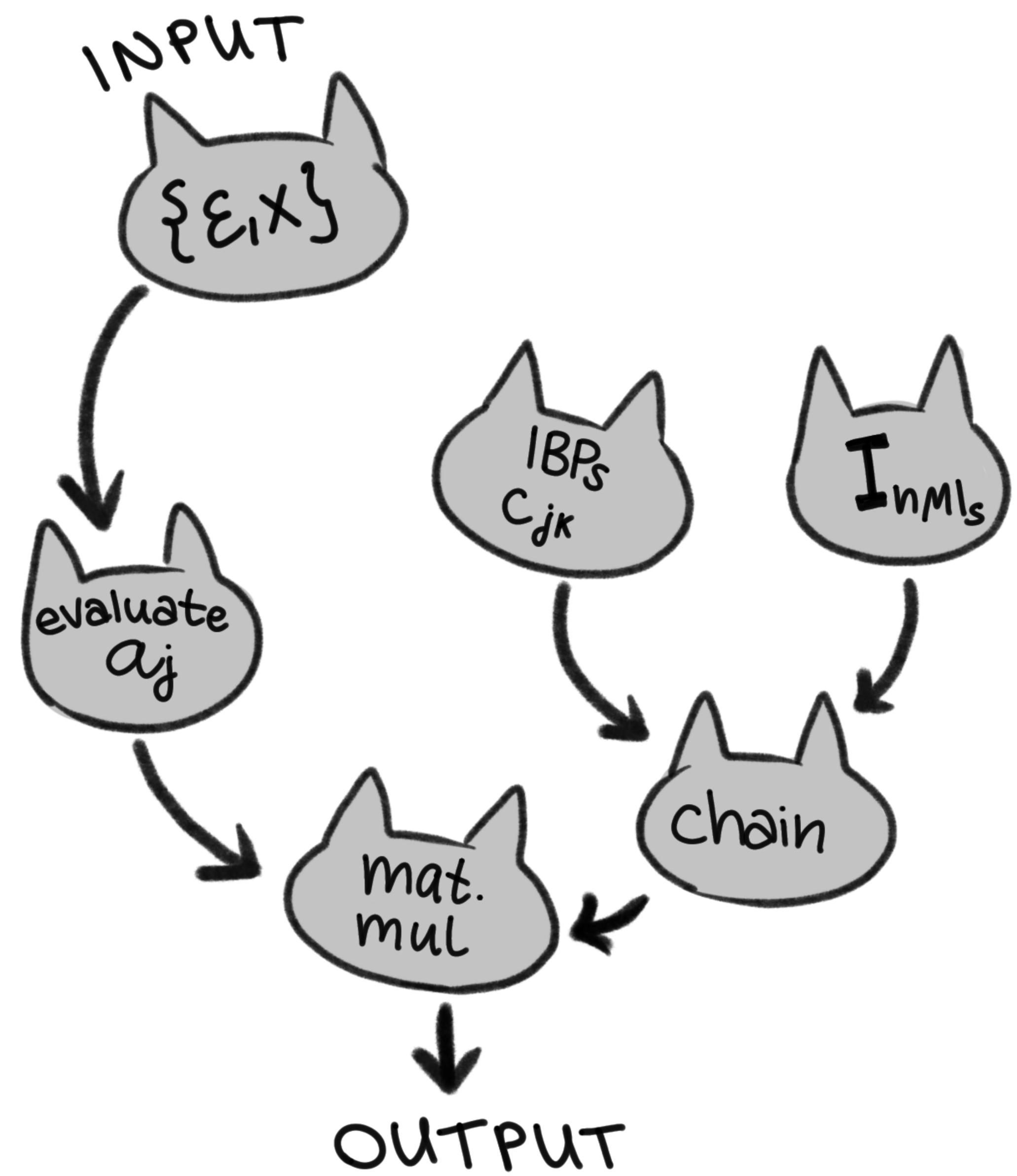


* evaluate $f(z)$ numerically for several $\mathbf{z}$ and p
* multivariate reconstruction algorithms → analytic form of $f(z)$ mod p
* upgrade analytic $f(z)$ over $\mathbb{Q}$ using rational reconstruction algorithm and Chinese remainder theorem

FiniteFlow Peraro (2019)

Framework to build numerical algorithms with

* high-level interface → *Mathematica* package
* computational graphs



idea:

* set of core algorithms implemented in C++
* custom algorithms made by linking the core ones in a high-level interface
* graph evaluation implemented in C++

finite-fields implementation

GF, Peraro (to appear)

implementation on FiniteFlow of the **multivariate recursive** algorithm

method based on solution of linear systems and series expansions

$\Rightarrow$ **rational operations**

input

list of n-variate intersection numbers we want to reduce, e.g.

$$\{\langle e_j^{(L)} | \varphi_R \rangle, \langle e_j^{(L)} | e_i^{(R)} \rangle\}$$

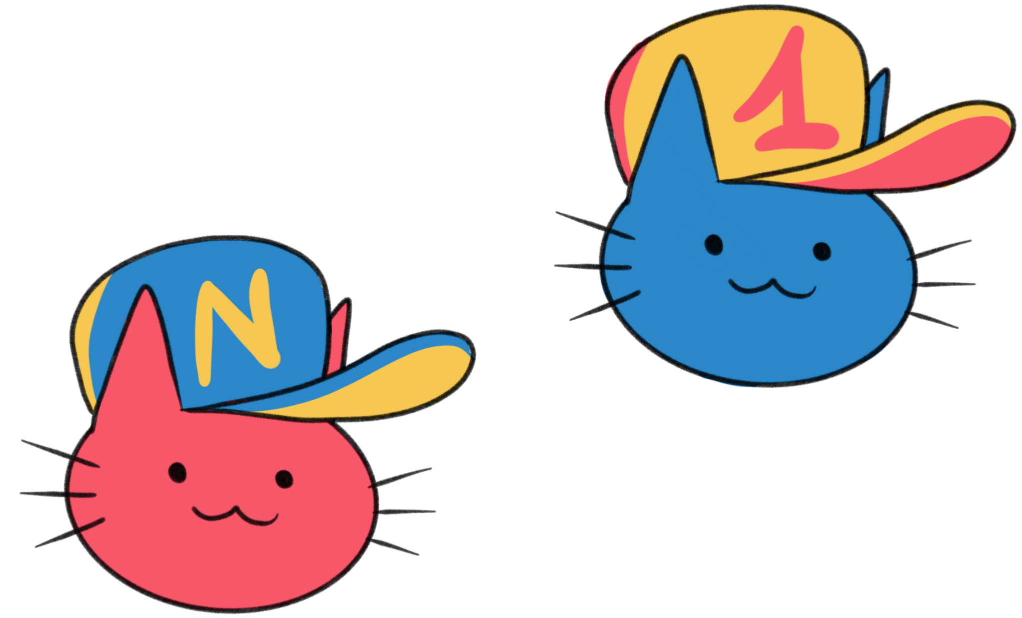
preliminary step

we can deduce the intersection numbers needed for each step

* $\langle \varphi_L | e_j^{(R)} \rangle_{n-1}$
* $\langle e_i^{(L)} | e_j^{(R)} \rangle_{n-1}$
* $\left(\partial_{z_n} \langle e_j^{(L)} |_{n-1}\right) | e_j^{(R)} \rangle_{n-1}$
* $\langle e_j^{(L)} | \varphi_R \rangle$

univariate algorithm

analytic input: $u(\mathbf{z})$



multivariate algorithm

needs as inputs

- denominator factors $p_i(z_n)$
- $(n-1)$-variate intersection numbers **reconstructed in z_n only**

dealing with poles

* $p = 0,\infty \rightarrow$ **Laurent expansion**
* all other factors $\rightarrow p(z)$-adic expansion

our implementation is an **iteration:**
starting from 1-forms, we compute all the necessary intersection numbers to get the n-forms given as input

input for the n^{th}- step

- $\mathbf{\Omega}_{ij}$
- $\langle \varphi_L | = \sum_{i=1}^{\nu_{n-1}} \varphi_{L,j} \langle e_j^{(L)} |$
- $\langle e_j^{(R)} | \varphi_R \rangle$

$\Big\} \; \chi_n$

between two steps:

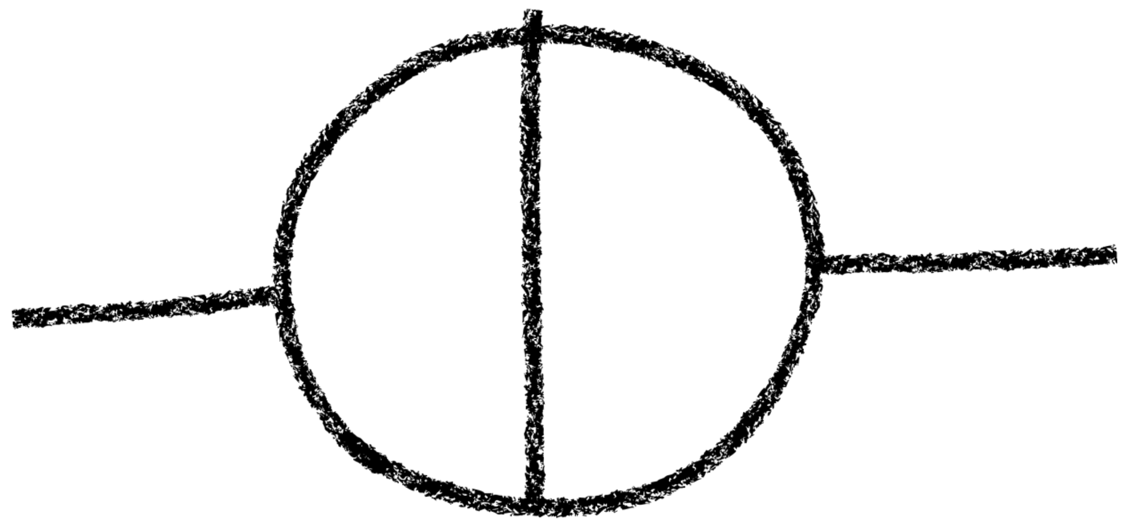
- rational reconstruction of $\mathcal{X}_n$ **only** in z_n, with everything else set to a number mod p
- identify denominator factors of $\mathcal{X}_n$ in z_n, **fully** reconstruct a simple subset of them

avoid reconstructing large intermediate expressions in all variables

can be done in a small number of evaluations

example: two-loops bubble

* definition of the family



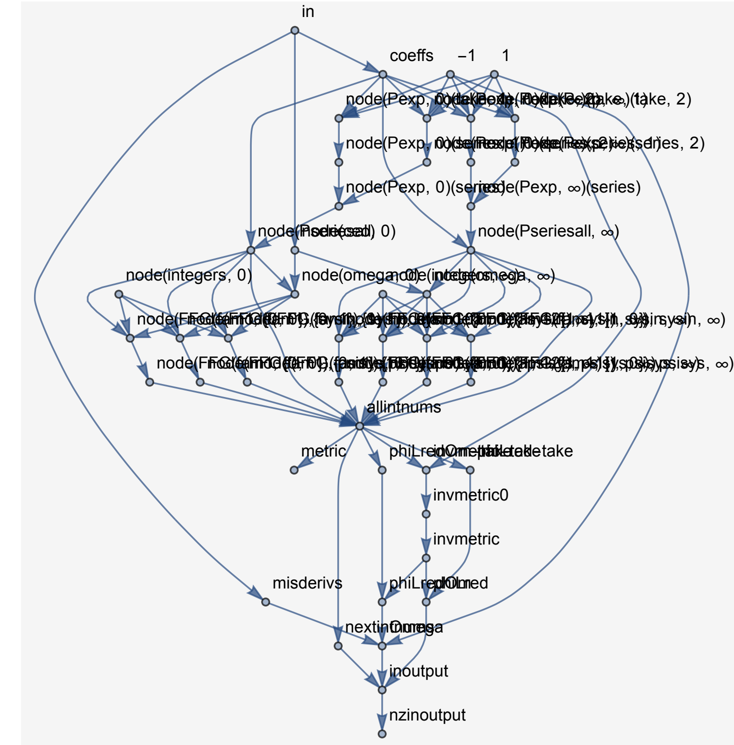
propagators:

* $z_1 = k_1^2$
* $z_2 = (k_1 - p_1)^2$
* $z_3 = (k_1 - k_2)^2$
* $z_4 = k_2^2$
* $z_5 = (k_2 - p_1)^2$

* list of int. num.s we want to calculate

we proceed with the ordering $z_1 \to z_2 \to z_3 \to z_4 \to z_5$

$1-$forms in z_1



* basis

$$\{\,|e_j^{(R)}\rangle\} = \{\langle e_j^{(L)}|\,\} = \left\{1, \frac{1}{z_1}\right\}$$

* $\{z_2, z_3, z_4, z_5, d, s\}$ are fixed parameters
* compute int nums and Ω_{ij} for next step
* evaluation over FF and reconstruction only in z_2

denominator factors:

$$\left\{s^2 z_3 + s z_3^2 - s z_3 z_4 - s z_3 z_5 + s z_4 z_5 + z_2\left(-s z_4 - s z_3 + z_4^2 - z_3 z_4 - z_5 z_4 + z_3 z_5\right) + z_2^2 z_4,\right.$$
$$\left. z_2^2 + (-2 z_3 - 2 z_5) z_2 + z_3^2 + z_5^2 - 2 z_3 z_5\right\}$$

```
univ rec...
- done
Generated 28 sample points
Approximate time per evaluation (single core): 6.4e-05 sec.
```

$2-$forms in z_2

- basis

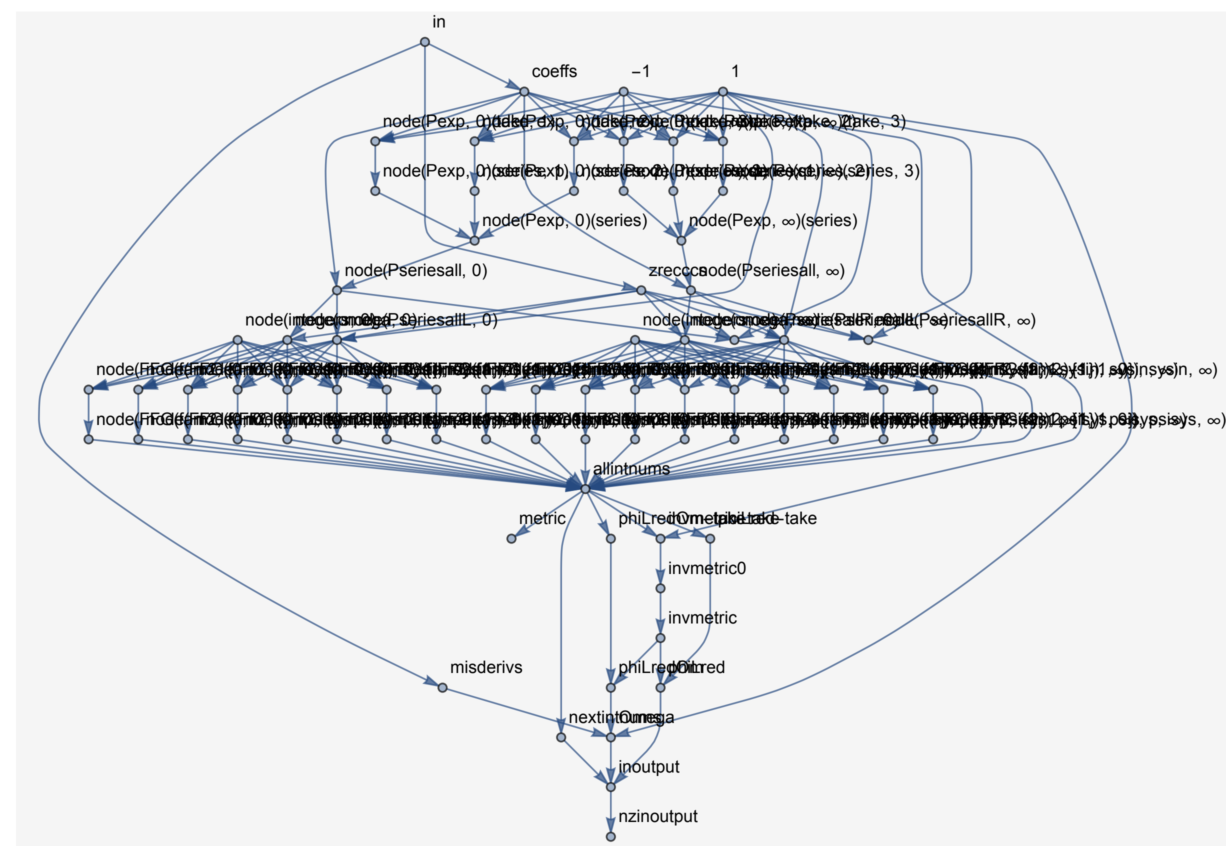
$$\{\,|e_j^{(R)}\rangle\} = \{\langle e_j^{(L)}|\,\} = \left\{1, \frac{1}{z_1}, \frac{1}{z_2}, \frac{1}{z_1 z_2}\right\}$$

- $\{z_3, z_4, z_5, d, s\}$ are fixed parameters
- evaluation over FF and reconstruction only in z_3

denominator factors:

$$\left\{z_3 - z_4, z_3 - z_5, z_3(s - z_4 - z_5) + z_3^2 + z_4 z_5\right\}$$

and so on …



until…

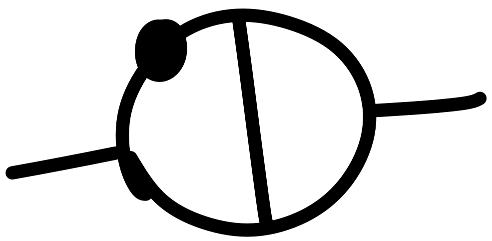
last step: $5-$forms in z_5

- basis

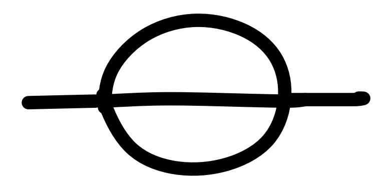
$$\{\,|e_j^{(R)}\rangle\} = \{\langle e_j^{(L)}|\,\} = \left\{\frac{1}{z_1z_2z_4z_5},\frac{1}{z_1z_3z_5},\frac{1}{z_2z_3z_4}\right\}$$

- $\{d,s\}$ are fixed parameters
- compute int. nums. and project into MIs

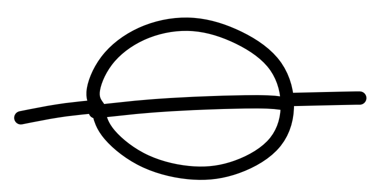
output:



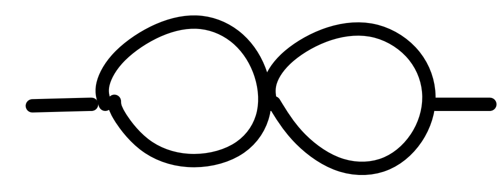
$$= \frac{3(3d-10)(3d-8)}{(d-6)(d-4)s^3}$$



$$+ \; + \frac{3(3d-10)(3d-8)}{(d-6)\,s^3}$$

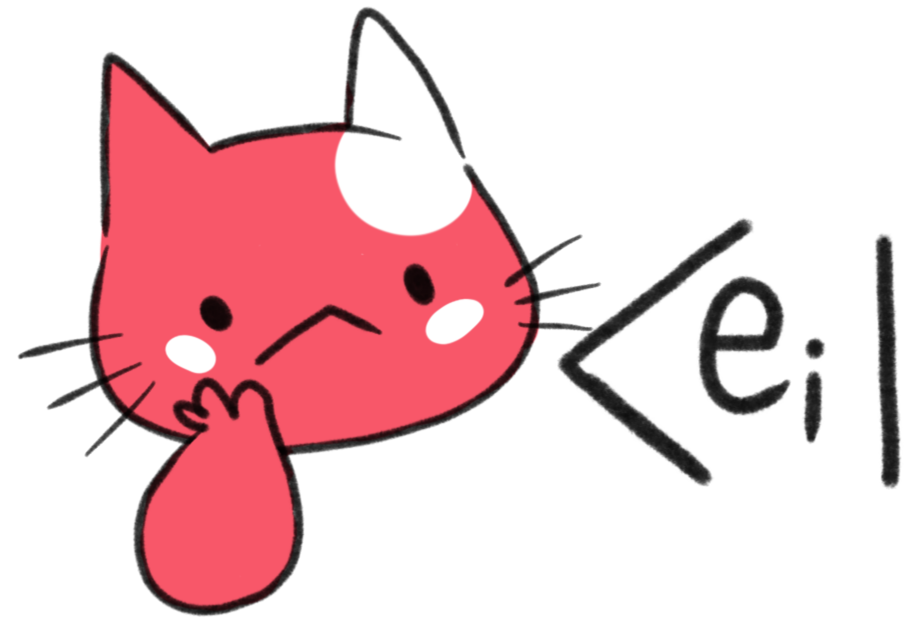


$$+\,2\frac{(d-3)}{s^2}$$



Basis choice

We assumed the basis for each layer to be known….
how do we find them?



2 strategies

- master-monomials analysis
- over-complete basis approach

master-monomials analysis

- number of MI related to the number of solutions of some polynomial eq.s
  - finding the “independent monomials”

Lee,Pomeransky (2013)

- use this poly eq.s to guess the number of MI for each layer
  - # MI = # of independent monomials
  - choice of MI ↔ choice of “master monomials”

Over-complete basis approach

start from a list of integrals that form a over-complete basis

$$\tilde{\mathscr{E}}_R = \{\,|\tilde{e}_j^{(R)}\rangle\}, \quad \tilde{\mathscr{E}}_L = \{\langle\tilde{e}_j^{(L)}|\,\}$$

compute an over-complete metric → not invertible!

$$\tilde{\mathbf{C}}_{ij} = \langle\tilde{e}_i^{(L)}\,|\,\tilde{e}_j^{(R)}\rangle$$

dual basis: column-reduction of $\tilde{\mathbf{C}}_{ij}$

$$\mathscr{E}_L = \{\langle e_j^{(L)}|\,\} = \{\langle\tilde{e}_j^{(L)}|\,\}_{j\in\text{indep. columns}} \subset \tilde{\mathscr{E}}_L$$

basis: row-reduction of independent columns

$$\mathscr{E}_R = \{\,|e_j^{(R)}\rangle\} = \{\,|\tilde{e}_j^{(R)}\rangle\}_{j\in\text{indep. rows}} \subset \tilde{\mathscr{E}}_R$$

Conclusions

* important step in th. predictions: reduction to master integrals
* new approach: intersection theory
* $p(z)$–adic expansion of rational functions
* FF implementation of rational algorithm for intersection numbers
* test on many one- and two-loops examples

& outlook

* work directly at the amplitude level
* simplify/optimize implementation
* finish the paper :)

Thank you for your attention!