Scattering amplitudes over finite fields and multivariate functional reconstruction

Tiziano Peraro

Higgs Centre for Theoretical Physics
University of Edinburgh

Milan Christmas meeting 2016
Università degli Studi di Milano
22 December 2016

based on: T.P., JHEP **1612** (2016) 030 [arXiv:1608.01902 [hep-ph]].

Outline

1. Introduction and motivation
2. Finite fields and multivariate reconstruction
3. Applications to scattering amplitudes
4. Summary & Outlook

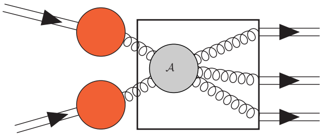
Introduction and motivation

Introduction and motivation

- Experiments at the Large Hadron Collider (LHC)
  - high-accuracy experimental data (up to % level)
  - high c.o.m. energy ⇒ multi-particle final states
  - large SM background (could hide new/interesting physics)

We need scattering amplitudes for theoretical predictions with

- high accuracy
- multi-particle interactions



Scattering amplitudes

- LO not reliable ⇒ need at least NLO
- NNLO needed for high precision

⇒ need to compute loop integrals

Perturbative calculation of scattering amplitudes

- Tree-level
  - solved for virtually any process/theory
    - Feynman diagrams, Berends-Giele/BCFW recursion
  - highly automated
  - focus on performance (especially for real emissions)
- One loop
  - essentially solved for any process/theory
    - tensor/integrand reduction, generalized unitarity, known MIs
  - automated, even for multi-leg processes
  - focus on performance and numerical stability
- Two loops
  - many calculations in recent years
    - diagrams, IBPs, differential equations, etc...
  - it's gradually becoming the new standard...
  - ...but still essentially restricted to $2 \to 2$ processes

Computation of scattering amplitudes

- Decomposed as

$$\mathcal{A} = \sum_k c_k I_k$$

  - I_k integrals (special functions)
  - c_k rational functions of kinematic invariants ($\Leftarrow$ focus of this talk)
  - c_k are a bottleneck at high multiplicity
- One loop $\rightarrow$ mostly numerical
  - many automated codes and toolchains
- Higher loops $\rightarrow$ mostly analytic
  - faster/more stable numerical evaluation
  - some techniques become more convenient (e.g. IBPs)
  - allows many checks/manipulations/studies (singularities, limits, ...)
  - can be used to infer general analytic/algebraic properties

  $\Rightarrow$ more control

Analytic calculation of scattering amplitudes

- Main bottleneck: large intermediate expressions
  - they can be orders of magnitude larger than the final result
  - not constrained by properties and symmetries of the result
- Tools for mitigating the problem
  - computer algebra systems specialised in handling large expressions (e.g. FORM [Vermaseren et al.])
  - generalized unitarity $\Rightarrow$ intermediate steps are gauge invariant

The main idea of this talk

- reconstruct analytic result from “numerical” evaluations
- no large intermediate expression (just numbers!)

Reconstruction of rational functions

- Which kind of "numerical" evaluation is good?
  - floating-point evaluation
    - ✓ very fast
    - ✗ affected by numerical instabilities
  - evaluation over the rational field $\mathcal{Q}$
    - ✓ exact
    - ✗ intermediate results have large numerators/denominators
      $\Rightarrow$ requires slow multi-precision arithmetic
  - evaluation over finite-fields
    - ✓ a finite-number of elements, which can be represented by machine-size integers
    - ✓ fast
    - ✓ exact
    - ✗ some information is lost and must be recovered by repeating the reconstruction over several finite fields

Functional reconstruction over finite fields

- Finite fields
  - used under-the-hood by computer algebra systems (e.g. in polynomial factorization/GCD)
  - proposed for IBPs (univariate applications)
    [von Manteuffel, Schabinger (2014–2016)]

- Efficient algorithm for functional reconstruction [T.P. (2016)]
  - works on (dense) multivariate polynomials and rational functions
  - implemented in C++ code (proof of concept)
  - the input is a numerical procedure computing a function
  - the output is its analytic expression
- Applications
  - spinor-helicity
  - tree-level recursion
  - multi-loop integrand reduction and generalized unitarity

Finite fields and multivariate reconstruction

Finite fields

- In this talk we consider finite fields $\mathcal{Z}_p$, with p prime
- We define

$$\mathcal{Z}_n = \{0, \dots, n-1\}$$

  - addition, subtraction, and multiplication via modular arithmetic

$$4+5\Big|_{\mathcal{Z}_7} = (4+5) \text{ mod } 7 = 2$$

  - if $a \in \mathcal{Z}_n$ and a, n are coprime, we can define an inverse

$$b = a^{-1} \in \mathcal{Z}_n, \qquad a \times b \text{ mod } n = 1$$

  - if $n = p$ prime, an inverse exists for every $a \in \mathcal{Z}_p \Rightarrow \mathcal{Z}_p$ is a field
  - every rational operation is well defined in $\mathcal{Z}_p$

Polynomials and rational functions

- multi-index notation: variables $\mathbf{z} = (z_1, \ldots, z_n)$ and integer list $\alpha = (\alpha_1, \ldots, \alpha_n)$

$$\mathbf{z}^\alpha \equiv \prod_{i=1}^{n} z_i^{\alpha_i}, \qquad |a| = \sum_i \alpha_i$$

- Given a generic field $\mathcal{F}$
  - $\mathcal{F}[\mathbf{z}]$ is the ring of polynomials in $\mathbf{z}$ with coefficients in $\mathcal{F}$

$$f(\mathbf{z}) = \sum_\alpha c_\alpha \, \mathbf{z}^\alpha.$$

  - $\mathcal{F}(\mathbf{z})$ is the field of rational functions in $\mathbf{z}$ with coefficients in $\mathcal{F}$

$$f(\mathbf{z}) = \frac{p(\mathbf{z})}{q(\mathbf{z})} = \frac{\sum_\alpha n_\alpha \, \mathbf{z}^\alpha}{\sum_\beta d_\beta \, \mathbf{z}^\beta},$$

    - technicality: set $d_{\text{min}\beta} = 1$ to make the representation unique.

Rational reconstruction

Functional reconstruction

Reconstruct the monomials z^α and their coefficients from numerical evaluations of the function (over finite fields)

- from $\mathcal{Q}$ to $\mathcal{Z}_p$

$$q = a/b \in \mathcal{Q} \quad \longrightarrow \quad q \bmod p \equiv a \times (b^{-1} \bmod p) \bmod p$$

- how to go back from $\mathcal{Z}_p$ to $\mathcal{Q}$?
- rational reconstruction algorithm: given $c \in \mathcal{Z}_n$ find its pre-image $q = a/b \in \mathcal{Q}$ with "small" a, b [Wang (1981)]
  - it's correct when $a, b \lesssim \sqrt{n}$
- make n large enough using Chinese reminder theorem
  - solution in $\mathcal{Z}_{p_1}, \mathcal{Z}_{p_2} \ldots \Rightarrow$ solution in $\mathcal{Z}_{p_1 p_2 \ldots}$

The black-box interpolation problem

Given a polynomial or rational function f in the variables $\mathbf{z} = (z_1, \ldots, z_n)$

- reconstruct analytic form of f, given a numerical procedure

$$\mathbf{z} \longrightarrow \boxed{\; f \;} \longrightarrow f(\mathbf{z}),$$

- modified black-box interpolation problem, for usage with finite fields

$$(\mathbf{z}, p) \longrightarrow \boxed{\; f \;} \longrightarrow f(\mathbf{z}) \bmod p.$$

  - the two are equivalent because of Chinese reminder theorem
- no further assumptions on f

Multivariate functional reconstruction (summary)

T.P. (2016)

- Univariate polynomials
  - based on Newton's interpolation formula
- Univariate rational functions
  - based on Thiele's (1838–1910) interpolation formula
- Multivariate polynomials
  - recursive application of Newton's interpolation
- Multivariate rational functions
  - use ideas proposed for sparse interpolation [A. Cuyt, W. Lee (2011)]
  - combined with Newton and Thiele's interpolation for dense case

- Notes:
  - all implemented in C++
  - results automatically come out GCD-simplified

Applications to scattering amplitudes

Four-dimensional spinor-helicity formalism

Mangano, Parke

- tree-level amplitudes are rational functions of spinor variables $|p\rangle$ and $|p]$
- satisfying the Dirac equation (in Weyl components)

$$p^\mu \, \sigma_\mu |p\rangle = p^\mu \, \sigma_\mu |p] = 0$$

- momenta and polarization vectors

$$p^\mu = \frac{1}{2} \langle p| \, \sigma^\mu \, |p], \quad \epsilon_+^\mu(p) = \frac{\langle \eta|\sigma^\mu|p]}{\sqrt{2}\,\langle \eta \, p\rangle}, \quad \epsilon_-^\mu(p) = \frac{\langle p|\sigma^\mu|\eta]}{\sqrt{2}\,[p\,\eta]}$$

- helicity amplitudes are combinations of spinor products, e.g.

$$\mathcal{A}_{5g}(1^+, 2^-, 3^+, 4^-, 5^+) = i\, g_s^3 \, \frac{\langle 2\,4\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\langle 3\,4\rangle\langle 4\,5\rangle\langle 5\,1\rangle}$$

- redundancy: spinor components are not all independent

A brief digression on spinor phases

- under a little group tranformation (complex redefinition of phase)

$$|i\rangle \to t_i\, |i\rangle, \qquad |i] \to \frac{1}{t_i}\, |i],$$

  an n-point amplitude $\mathcal{A}(1,\ldots,n)$ transforms as

$$\mathcal{A}(1,\ldots,n) \to \left(\prod_{i=1}^{n} t_i^{-2\,h_i}\right) \mathcal{A}(1,\ldots,n),$$

  where h_i is the helicity of the i-th particle (e.g. $\pm 1/2$ for fermions and ± 1 for gluons)

- extract from the amplitude an overall factor $\mathcal{A}^{(\text{phase})}(1,\ldots,n)$ which transform as the amplitude
- consider $\tilde{A}$ such that

$$\mathcal{A} = \underbrace{\mathcal{A}^{(\text{phase})}}_{\text{only depends on helicities}} \times \underbrace{\tilde{\mathcal{A}}}_{\text{phase-free}}$$

Choice of kinematic variables

Hodges (2009), Badger, Frellesvig, Zhang (2013), Badger (2016)

- phase-free rational parametrization of the n-point phase-space and the spinor components using $3n-10$ momentum-twistor variables
- 5-point example → 5 variables $\{x_1,\ldots,x_5\}$

$$|1\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \qquad |1] = \begin{pmatrix}1\\ \frac{x_4-x_5}{x_4}\end{pmatrix}, \qquad x_k = x_k(s_{ij}, \mathrm{tr}(\sigma_5\, 1\,2\,3\,4))$$

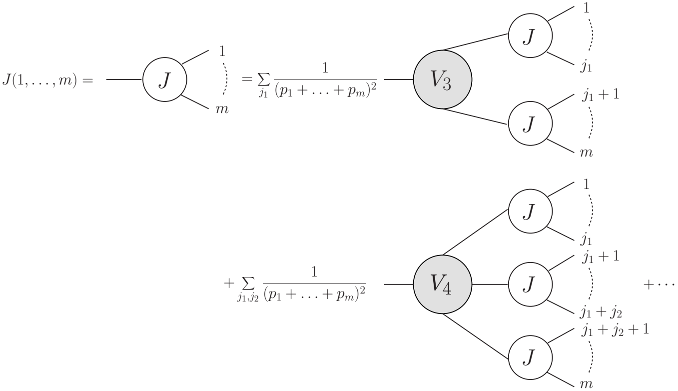
$$|2\rangle = \begin{pmatrix}0\\1\end{pmatrix}, \qquad |2] = \begin{pmatrix}0\\ x_1\end{pmatrix}, \qquad p_i^\mu = \frac{\langle i|\,\sigma^\mu\,|i]}{2}$$

$$|3\rangle = \begin{pmatrix}\frac{1}{x_1}\\1\end{pmatrix}, \qquad |3] = \begin{pmatrix}x_1\,x_4\\ -x_1\end{pmatrix},$$

$$|4\rangle = \begin{pmatrix}\frac{1}{x_1}+\frac{1}{x_1\,x_2}\\1\end{pmatrix}, \qquad |4] = \begin{pmatrix}x_1\,(x_2\,x_3 - x_3\,x_4 - x_4)\\ -\frac{x_1\,x_2\,x_3\,x_5}{x_4}\end{pmatrix},$$

$$|5\rangle = \begin{pmatrix}\frac{1}{x_1}+\frac{1}{x_1\,x_2}+\frac{1}{x_1\,x_2\,x_3}\\1\end{pmatrix}, \qquad |5] = \begin{pmatrix}x_1\,x_3\,(x_4-x_2)\\ \frac{x_1\,x_2\,x_3\,x_5}{x_4}\end{pmatrix}.$$

Tree-level amplitudes via Berends-Giele recursion



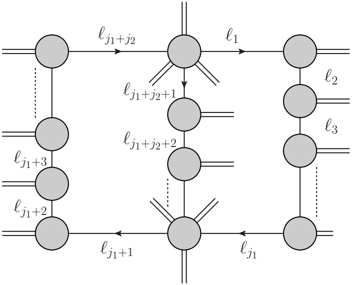
- very efficient for numerical calculations
- functional reconstruction techniques can exploit this for obtaining analytic results

Integrand reduction and generalized unitarity

Britto, Cachazo, Feng (2004), Giele, Kunszt, Melnikov (2008), Bern, Dixon, Kosower et al. (2008)

- Generalized unitarity
  - build irreducible integrands from multiple cuts
  - multiple-cuts $\Rightarrow$ loop propagators go on-shell, $\ell_i^2 = 0$
  - integrand factorizes as product of trees (summed over internal helicities)
  - multiple cuts $\Rightarrow$ unitarity cuts

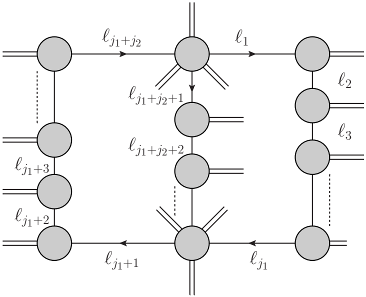
- # unitarity cuts $\ll$ # diagrams
  - lower complexity
- Every intermediate step is gauge invariant
  - no ghosts
  - more compact expressions



Generalized unitarity over finite fields

T.P. (2016)

- Amplitudes over finite fields
  - momentum-twistor variables
  - loop states: embed in 6-dim.
  - spinor-helicity in 4 and 6 dim.
  - tree-level recursion
  - two-loop d-dim. unitarity cuts

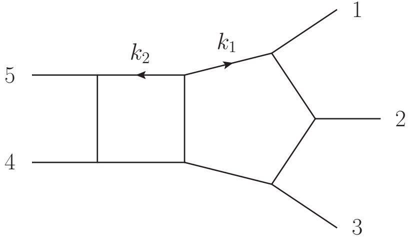


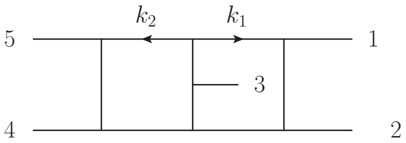
Finite-field implementation

- explicit six-dim. representation of loop states
- efficient numerical techniques for analytic calculations
- two-loop unitarity cuts from Berends-Giele off-shell currents

Finite fields and functional reconstruction: examples

- five-gluon on-shell integrands of maximal cuts ($\equiv$ top-level topology) for





(for a complete set of helicities)

Finite fields and functional reconstruction

- penta-box

Helicity	Non-vanishing coeff.	Max. terms	Max. degree	Avg. non-zero terms
$(1^+,2^+,3^+,4^+,5^+)$	14	19	8	15.00
$(1^-,2^+,3^+,4^+,5^+)$	27	443	19	152.96
$(1^+,2^-,3^+,4^+,5^+)$	37	1977	24	674.97
$(1^+,2^+,3^+,4^-,5^+)$	61	474	18	184.05
$(1^-,2^-,3^+,4^+,5^+)$	35	1511	24	278.77
$(1^-,2^+,3^+,4^+,5^-)$	79	7027	34	1112.82
$(1^+,2^+,3^+,4^-,5^-)$	18	19	8	15.00
$(1^-,2^+,3^-,4^+,5^+)$	41	2412	22	368.41
$(1^+,2^-,3^+,4^-,5^+)$	85	18960	42	3934.96
$(1^-,2^+,3^+,4^-,5^+)$	85	10386	37	1803.52

- double-pentagon

Helicity	Non-vanishing coeff.	Max. terms	Max. degree	Avg. non-zero terms
$(1^+,2^+,3^+,4^+,5^+)$	104	1937	26	626.39
$(1^-,2^+,3^+,4^+,5^+)$	104	1449	27	601.43
$(1^+,2^+,3^-,4^+,5^+)$	104	1554	23	642.90
$(1^-,2^-,3^+,4^+,5^+)$	99	1751	26	739.05
$(1^+,2^-,3^-,4^+,5^+)$	104	2524	24	923.71
$(1^-,2^+,3^+,4^+,5^-)$	104	1838	27	823.00
$(1^-,2^+,3^+,4^-,5^+)$	104	1307	24	630.48

Summary & Outlook

Summary

- Finite-fields and functional reconstruction techniques
  - can be use to solve complex algebraic problems
  - any function which can be implemented as a sequence of rational operations is suited for these algorithms
- Applications to scattering amplitudes
  - spinor-helicity in four and six dimensions
  - tree-level calculations
  - integrand reduction via generalized unitarity

Outlook

- complete five-point two-loop calculations
- apply the algorithm to other techniques (e.g. diagrammatic techniques, tensor reduction, IBPs,. . .)

THANKS!

BACKUP SLIDES

Extended euclidean algorithm

- given integers a, b, find s, t such that

$$a\, s + b\, t = \gcd(a, b).$$

- algorithm: generate sequences of integers $\{r_i\}$, $\{s_i\}$, $\{t_i\}$ and the integer quotients $\{q_i\}$ as follows

$$\begin{aligned} r_0 &= a & \cdots &= \cdots \\ s_0 &= 1 & q_i &= \lfloor r_{i-2}/r_{i-1} \rfloor \\ t_0 &= 0 & r_i &= r_{i-2} - q_i\, r_{i-1} \\ r_1 &= b & s_i &= s_{i-2} - q_i\, s_{i-1} \\ s_1 &= 0 & t_i &= t_{i-2} - q_i\, t_{i-1} \\ t_1 &= 1 & & \end{aligned}$$

- stop when $r_k = 1 \Rightarrow t = t_{k-1}$, $s = s_{k-1}$, $\gcd(a, b) = r_{k-1}$
- multiplicative inverse: if $b = n$ and $\gcd(a, n) = 1 \Rightarrow s = a^{-1}$.

Chinese reminder theorem

- given $a_1 \in \mathcal{Z}_{n_1}$, $a_2 \in \mathcal{Z}_{n_2}$ (n_1, n_2 co-prime) find $a \in \mathcal{Z}_{n_1 n_2}$ such that

$$a \bmod n_1 = a_1, \qquad a \bmod n_2 = a_2.$$

- rational reconstruction over $\mathcal{Q}$
  - reconstruct a function f over several finite fields $\mathcal{Z}_{p_1}, \mathcal{Z}_{p_2}, \ldots$
  - recursively combine the result in $\mathcal{Z}_{p_1 p_2 \cdots}$ using the Chinese reminder
  - use the rational reconstruction algorithm on the combined result over $\mathcal{Z}_{p_1 p_2 \cdots}$ to obtain a guess over $\mathcal{Q}$
  - when $\prod_i p_i$ is large enough the reconstruction is successful
  - the termination criterion is consistency over several finite fields
- we can choose the primes p_i small enough to use machine-size integers
- multi-precision arithmetic only required for Chinese reminder
- 1 or 2 primes are usually sufficient

Rational reconstruction: example

- Reconstruct $q = -611520/341$ from its images over finite fields
- $\mathcal{Z}_{p_1}$, with $p_1 = 897473$

$$a_1 = q \text{ mod } p_1 = 13998,$$
$$\text{first guess: } a_1 \xrightarrow{\text{rational rec. over } \mathcal{Z}_{p_1}} g_1 = -411/577$$

- $\mathcal{Z}_{p_2}$, with $p_2 = 909683$

$$a_2 = q \text{ mod } p_2 = 835862$$
$$g_1 \text{ mod } p_2 = 807205 \quad \Rightarrow \quad \text{guess } g_1 \text{ is wrong}$$

- Chinese reminder: $a_1, a_2 \longrightarrow a_{12} \in \mathcal{Z}_{p_1 p_2}$, with $p_1 p_2 = 816415931059$

$$a_{12} \equiv q \text{ mod } p_1 p_2 = 629669763217 \xrightarrow{\text{rational rec. over } \mathcal{Z}_{p_1 p_2}} g_2 = -611520/341$$

- calculation over other fields $\mathcal{Z}_{p_3}, \ldots$ confirm the guess g_2

Univariate polynomials

- Newton' interpolation formula, form a sequence $\{y_0, y_1, \ldots\}$

$$f(z) = \sum_{r=0}^{R} a_r \prod_{i=0}^{r-1} (z - y_i)$$

$$= a_0 + (z - y_0)\Big(a_1 + (z - y_1)\Big(a_2 + (z - y_2)\big(\cdots + (z - y_{r-1})\, a_r\big)\Big)\Big)$$

- each coefficient a_i can be determined by evaluations $f(y_j)$ with $j \leq i$
  - good when degree is not known
- conversion into canonical form

$$f(z) = \sum_{r=0}^{R} c_r\, z^r .$$

  - addition of univariate polynomials,
  - multiplication of a univ. polynomial by a linear univ. polynomial

Univariate rational functions

- Thiele's (1838–1910) interpolation formula

$$f(z) = a_0 + \cfrac{z - y_0}{a_1 + \cfrac{z - y_1}{a_2 + \cfrac{z - y_3}{\cdots + \cfrac{z - y_{r-1}}{a_N}}}}$$

$$= a_0 + (z - y_0) \left(a_1 + (z - y_1) \left(a_2 + (z - y_2) \left(\cdots + \frac{z - y_{N-1}}{a_N} \right)^{-1} \right)^{-1} \right)^{-1},$$

- analogous to Newton's for rational functions
  - good when degrees of numerator/denominator are not known
- if degrees are known and $d_0 = 1$ (see later), just solve the system

$$f(z) = \frac{\sum_{r=0}^{R} n_r\, z^r}{\sum_{r'=0}^{R'} d_{r'}\, z^{r'}} \quad \Rightarrow \quad \sum_{r=0}^{R} n_r\, y_i^r - \sum_{r'=1}^{R'} d_{r'}\, y_i^{r'} f(y_i) = f(y_i)$$

Multivariate polynomials

- recursive Newton's formula

$$f(z_1,\ldots,z_n) = \sum_{r=0}^{R} a_r(z_2,\ldots,z_n) \prod_{i=0}^{r-1}(z_1 - y_i),$$

- like univariate with

$$f(y_j) \longrightarrow f(y_j, z_2, \ldots, z_n), \qquad a_j \longrightarrow a_j(z_2,\ldots,z_n).$$

- convert it back to canonical representation using
  - addition of multivariate polynomials,
  - multiplication of a multiv. polynomial by a linear univ. polynomial.
- very efficient, even for large polynomials

Multivariate rational functions

- dense algorithm, adapted from sparse one by A. Cuyt, W. Lee (2011)
- overall normalization
  - assume non-vanishing constant term in denominator ($d_{(0,\ldots,0)} = 1$)
  - if not the case, shift args. by appropriate vector $\mathbf{s}$, using $f_{\mathbf{s}} = f(\mathbf{z} + \mathbf{s})$
- define new function $h \in \mathcal{F}(t, \mathbf{z})$ as

$$h(t, \mathbf{z}) \equiv f(t\,\mathbf{z}) = f(t\,z_1, \ldots, t\,z_n) = \frac{\sum_{r=0}^{R} p_r(\mathbf{z})\, t^r}{1 + \sum_{r'=1}^{R'} q_{r'}(\mathbf{z})\, t^{r'}}$$

where

$$p_r(\mathbf{z}) \equiv \sum_{|\alpha|=r} n_\alpha\, \mathbf{z}^\alpha, \qquad q_{r'}(\mathbf{z}) \equiv \sum_{|\beta|=r'} d_\beta\, \mathbf{z}^\beta.$$

$\Rightarrow$ univ. rational fun. in t with (homogeneous) multiv. polynomial coefficients