

***p*-adic reconstruction of rational functions in multiloop amplitudes**Herschel A. Chawdhry^{*}*Department of Physics, University of Oxford, Parks Road, Oxford, OX1 3PU, United Kingdom, and Department of Physics, Florida State University, 77 Chieftan Way, Tallahassee, Florida 32306, USA*

(Received 28 December 2023; accepted 20 August 2024; published 13 September 2024)

Numerical reconstruction techniques are widely employed in the calculation of multiloop scattering amplitudes. In recent years, it has been observed that the rational functions in multiloop calculations greatly simplify under partial fractioning. In this article, we present a technique to reconstruct rational functions directly in partial-fractioned form, by evaluating the functions at special integer points chosen for their properties under a p -adic metric. As an application, we apply this technique to reconstruct the largest rational function in the integration-by-parts reduction of one of the rank-5 integrals appearing in two-loop five-point full-color massless amplitude calculations in quantum chromodynamics. The number of required numerical probes (per prime field) is found to be around 25 times smaller than in conventional techniques, and the obtained result is 130 times smaller. The reconstructed result displays signs of additional structure that could be used to further reduce its size and the number of required probes.

DOI: [10.1103/PhysRevD.110.056028](https://doi.org/10.1103/PhysRevD.110.056028)**I. INTRODUCTION**

Multiloop scattering amplitudes are a cornerstone of high-precision predictions in particle physics. These amplitudes are challenging to calculate and one of the key bottlenecks in these computations is the symbolic calculation of complicated multivariable rational functions, which appear in the final expression for the amplitudes as well as related results such as tables of integration-by-parts identities (IBPs) [1,2].

In principle, calculating these rational functions only requires elementary polynomial arithmetic: addition, subtraction, multiplication, and division. But the rational functions become large at intermediate stages of the calculation (as compared to the initial or final stages) and so the arithmetical operations at intermediate stages become very slow. This “intermediate-expression swell” phenomenon is well known in computer algebra, and in that field it has been common since the 1960s [3–7] to instead perform such calculations numerically at sample points in a finite field \mathbb{F}_p or p -adic field \mathbb{Q}_p and then construct the analytic form of the final result by interpolation techniques. The computational time of this approach is largely determined by the number of required numerical samples, which depends only on the complexity of the final expression and

thereby bypasses the complexity of intermediate stages. During the past decade, finite-field methods have been directly adopted in multiloop amplitude calculations with much success [8–12].

The rational functions reconstructed by finite-field methods are typically interpolated in common-denominator form, i.e., as a ratio of two polynomials. In multiloop calculations in recent years, it has been observed that these rational functions, once reconstructed, can be simplified by up to 2 orders of magnitude by partial fractioning them [13–20]. In principle, it would be desirable and advantageous to exploit this simplification earlier, i.e., *during* the reconstruction, so as to reduce the number of probes required by up to 2 orders of magnitude. But as we will show in Sec. II, the simplification under partial fractioning is not a generic feature of rational functions, but instead seems to be a special property of the specific rational functions appearing in IBPs and amplitudes. Exploiting this simplification therefore requires developing specialized techniques that go beyond those used in generic computer algebra calculations. This is in contrast to the use of finite-field methods, which are the computer algebraist’s standard solution to the widespread computer algebra phenomenon of intermediate-expression swell.

To date, some work has been performed with the aim of optimizing numerical reconstruction methods for high-energy physics use cases. As will be explained in Sec. II, it is possible to guess [18,21] the common denominator of the rational functions in multiloop calculations, thereby reducing the number of required numerical probes by a factor of 2. A partial-fractioned reconstruction technique based on very high-precision floating-point evaluations was

^{*}Contact author: hchawdhry@fsu.edu

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI. Funded by SCOAP³.

presented in Refs. [22,23]. Within a finite-field context, some benefits may also be obtained by reconstructing in one variable at a time and performing single-variable partial fractioning at some intermediate stages [6,24–28], possibly in conjunction with expanding in ϵ , where $D = 4 - 2\epsilon$ is the spacetime dimension variable. Techniques based on algebraic geometry and evaluations in \mathbb{Q}_p have been proposed [29–31] for eliciting information about the numerator of a rational function prior to performing a finite-field reconstruction, and Ref. [32] mentions combining these with the methods of Ref. [22].

In this work, we present a new technique to reconstruct rational functions directly in partial-fractioned form. Our technique uses p -adic probes to reconstruct the rational functions one partial-fractioned term at a time, exploiting the simplification under partial fractioning and exposing hints of further patterns and structure. We will apply this technique to reconstruct one of the largest rational functions appearing in the table of two-loop five-point massless nonplanar IBPs, a highly complicated example which is at the edge of the capabilities of current tools and methods. It will be shown that, for this example function, our technique requires 25 times fewer numeric (\mathbb{Q}_p) probes than conventional (\mathbb{F}_p -based) reconstruction, and leads to a 130-fold reduction in the size of the final result.

The rest of this article is organized as follows. In Sec. II, we give preliminary remarks to motivate our strategy. In Sec. III we briefly describe some properties of \mathbb{Q}_p that we use. In Sec. IV we present our reconstruction method. In Sec. V we analyze the results and performance. A summary and concluding remarks are presented in Sec. VI.

II. PRELIMINARY REMARKS

The focus of this work is on reducing the number of numerical probes required to reconstruct rational functions, since this is the dominant computational cost. This number is independent of the choice of method for performing the numerical probes. We therefore develop and test our techniques by taking a large rational function for which we have an analytic expression, and performing so-called “black-box” probes on it. Specifically, we choose to work with the largest rational function appearing in the largest IBP expression used in Ref. [16] for calculating the complete set of full-color two-loop amplitudes for $pp \rightarrow \gamma\gamma j$ in massless quantum chromodynamics (QCD). This is also the second-largest IBP expression needed for the corresponding three-jet amplitudes of Ref. [33]. We will call this rational function R_* .

The inspiration for our method is the observation that, as shown in Table I, R_* becomes ~ 100 times smaller after multivariate partial fractioning, compared to its size in common-denominator form. Thus, if everything were known about R_* except the integer coefficients in the numerators of the partial-fractioned expression, only

TABLE I. Simplification of R_* under partial fractioning. Common-denominator form has the numerator fully expanded and the denominator fully factorized. Partial-fractioned form is obtained using `MultivariateApart` [18] with option `UseFormProgram > True`. (See Table II for results obtained in this work.) Sizes are as reported using `ByteCount` in *Mathematica*. The number of free parameters is obtained by counting the number of terms in the fully expanded numerator(s).

Form of expression	Size (MB)	Parameters to fit
Common denominator	605	1,369,559
Partial fractioned	4	14,558

14,558 free parameters would need to be fitted and so 94 times fewer numerical evaluations would be required compared to conventional reconstruction techniques. The aim of this work is to achieve a speedup of this nature *without* requiring any prior knowledge about R_* .

The denominator of the common-denominator form of the rational function of interest is usually straightforward to obtain because it is a product of simple factors, whose powers can be found by performing a numerical probe at an integer kinematic point where some of those factors are prime numbers [18]. The factors themselves can be guessed in a variety of ways: either by examining the symbol (if known) of the differential equations describing the master integrals [21], or by examining the denominators of the rational functions in the solution to a low-numerator-rank IBP system, or simply by trial and error.

We believe, however, that the simplification in Table I does not follow from the mere fact that the denominator of R_* factorizes. To see this, we considered several rational functions from the solutions to the two-loop five-point massless IBP equations. For each rational function R we constructed a second rational function, \tilde{R} , obtained by taking R in common-denominator form and replacing with random numbers all coefficients in its fully expanded numerator, while leaving the denominator unchanged. We observed that each R gets simplified upon partial fractioning, and the simplification factor is largest for the largest rational functions. Yet if we partial-fraction \tilde{R} , no simplification occurs; indeed the partial-fractioned form of \tilde{R} is typically slightly larger than its common-denominator form, regardless of whether it is measured using `ByteCount` or the number of free numerator parameters. We conclude that the above-mentioned simplification of R_* upon partial fractioning does not occur for generic rational functions, but is instead a special property of R_* , which we conjecture will generalize to many IBP and amplitude expressions.¹ Therefore, it should be expected that

¹We emphasize that the selection of R_* as a working example was not on the basis of any such properties, but was on the contrary because it is an exceedingly complicated expression that is on the boundary of current computational techniques.

exploiting this simplification will require custom techniques beyond those developed for generic computer algebra problems.

To understand the reason for the simplification in Table I and guide a strategy for exploiting it, we applied `MultivariateApart` [18]² to several examples of R and \tilde{R} . In each case we compared the resulting expressions³

$$R = \sum_i \frac{n_i}{d_i}, \tag{1}$$

$$\tilde{R} = \sum_j \frac{\tilde{n}_j}{d_j}. \tag{2}$$

For all the examples studied, we observed that the sum in Eq. (1) contains fewer terms than the sum in Eq. (2). Furthermore, all of the terms in Eq. (1) also appear in Eq. (2), albeit with different numerators—in other words, $\{d_i\}$ is a subset of $\{d_j\}$. We noted that if the partial-fractioned terms that are present in \tilde{R} but vanish in R could be identified in advance, it would give a large simplification. In the case of R_* , we estimated this simplification would be a factor of 28 compared to the common-denominator form, reducing the number of free parameters from 1,369,559 to 48,512. For this reason, the core aim of our strategy presented in Sec. IV is to identify, as cheaply as possible, which partial-fractioned terms vanish. The remaining factor of $\frac{48,512}{14,558} \approx 3.3$ between this and the figure in Table I arises because many of the partial-fractioned terms in Eq. (1) have numerators containing fewer terms than the most generic polynomial that could be expected; we will leave to future work the exploitation of this further simplification.⁴ In addition, our results (see Sec. V) suggest that further patterns and structure are present, which could be exploited in future work to obtain significant further speedups.

III. \mathbb{Q}_p AND ITS IMPLEMENTATION

In the field of computer algebra, it is common to obtain rational numbers not from floating-point real evaluations, which are prone to rounding errors, but instead from evaluations in a finite field \mathbb{F}_p or *p*-adic field \mathbb{Q}_p . The

²`MultivariateApart` implements the Leinartas algorithm [34], which was also implemented in Refs. [13,35–37].

³In this work, we find it helpful to adopt a convention of allowing no overall integer factors in the denominator of a partial-fractioned form, instead preferring to put such factors into the numerator, e.g., preferring $\frac{y^2z + 8yz^2}{3(x+y)(x-z)}$ instead of $\frac{y^2z + 8yz^2}{3(x+y)(x-z)}$.

⁴In contrast to the $\mathcal{O}(30\text{--}100)$ simplification factors targeted here, we note that performing single-variable partial fractioning on these examples of R produces simplifications by only a factor of between 2 and 12 relative to common-denominator form. The precise factor depends on the choice of variable with respect to which such a single-variable partial fractioning is performed.

p-adic numbers \mathbb{Q}_p are an extension of the rational numbers \mathbb{Q} but are distinct from the real numbers \mathbb{R} . Reference [29] points out that *p*-adic fields are well suited to studying the singular limits of polynomials and rational functions, which is helpful for studying the partial fractioning of rational functions as desired in this work. Specifically, we will use the *p*-adic numbers to evaluate rational functions at special points that make chosen denominators “small” under a *p*-adic absolute value $|\cdot|_p$ (defined below), allowing the corresponding numerators to be isolated and examined, but not small under the standard absolute value $|\cdot|$, thereby ensuring that the evaluations are numerically stable. Introductory material on *p*-adic numbers and their use in computer algebra can be found in Ref. [7], but here we will only mention that in this work we rely on the following facts:

- (1) Given a prime number *p*, any rational number can be uniquely expanded as a *p*-adic series, i.e., a series $\sum_m c_m p^m$ with $c_m \in \{0, \dots, p - 1\}$. Such series are analogous to power series in a small parameter, so large powers of *p* are considered to be small according to the so-called *p*-adic absolute value $|\cdot|_p$,

$$\left| \frac{a}{b} p^n \right|_p = \frac{1}{p^n}, \tag{3}$$

where *a*, *b*, and *n* are integers, and *a* and *b* are indivisible by *p*. An example of such a series expansion is

$$\frac{1}{2} = 4 + 3 * 7 + 3 * 7^2 + 3 * 7^3 + \mathcal{O}(7^4), \tag{4}$$

where we have chosen to expand with *p* = 7. Equation (4) can easily be verified by multiplying both sides of the equation by 2 and performing carries whenever $c_m \geq p$. While the terms represented by the notation “ $\mathcal{O}(7^4)$ ” on the rhs of Eq. (4) are clearly large under the usual absolute value, they are small under the so-called 7-adic absolute value defined by Eq. (3) with *p* = 7:

$$\left| \frac{1}{2} - 4 - 3 * 7 - 3 * 7^2 - 3 * 7^3 \right|_7 = \left| -\frac{2401}{2} \right|_7 = \frac{1}{7^4}. \tag{5}$$

- (2) Elementary arithmetical operations commute with performing *p*-adic expansions.
- (3) Any rational number is congruent (mod *p*) to the leading term of its *p*-adic expansion, assuming the denominator of that rational number is not a multiple of *p*. Taking the example in Eq. (4), we have

$$\frac{1}{2} \equiv 4 \pmod{7}, \quad (6)$$

which can be verified by multiplying both sides by 2.

There are many choices of practical methods to implement p -adic numbers on a computer. For example, just as with the real numbers, two common options are fixed-point and floating-point representations. While in this work we focus on reducing the number of required probes, which is independent of the choice of representation, let us mention that in practice we have here chosen to represent each p -adic number by an integer that is p -adically close to it, simply by truncating the p -adic series. For instance, the p -adic series on the rhs of Eq. (4) can be represented by the integer value 1201:

$$4 + 3 * 7 + 3 * 7^2 + 3 * 7^3 = 1201. \quad (7)$$

Every p -adic evaluation in this work is performed as a rational evaluation at an integer point chosen in this way. Unlike floating-point (real or p -adic) calculations, rational evaluations are exact and immune from rounding errors and loss of precision, regardless of whether they are performed as black-box evaluations on an analytic expression for R , as we do here, or as step-by-step solutions of a set of IBP equations, which is one of the applications envisaged. In principle, rational evaluations should suffer from intermediate-expression swell, but we avoid this: in this work we surprisingly find that it is possible to use small primes $p \approx 101$ and series truncated to small powers of p , so that we typically perform evaluations at ten-digit integer values for the kinematic points and space-time dimension.⁵ Numerically solving IBPs and amplitudes at such integer points is well within the capabilities of standard publicly available tools that internally use finite fields or exact integer arithmetic.

IV. METHOD

In order to exploit the observations from Sec. II, we have designed a method to reconstruct rational functions directly in partial-fractioned form Eq. (1), one partial-fractioned term $\frac{n_i}{d_i}$ at a time. A set of all possible denominators $\{d_i\}$ is straightforward to determine by examining the easily obtainable (see Sec. II) denominator of the common-denominator-form expression. As explained in Sec. II, the speedup in this paper will arise because, for many of the possible denominators d_i , the corresponding n_i is zero. Reconstructing one partial-fractioned term at a time ensures that, if a partial-fractioned term vanishes, we can notice this

⁵The number of digits varies between 4 and 20. In future work we intend to adopt a more sophisticated approach to reconstructing the spacetime-dimension variable, which we estimate would reduce the number of digits in all our larger integer evaluation points to around 10 digits.

cheaply and avoid reconstructing its numerator. A key further advantage of reconstructing one partial-fractioned term at a time is that our method will scale well for even larger rational functions than R_* , for reasons explained in Sec. IVA 5.

Reconstructing one partial-fractioned term at a time also has other benefits, which we foresee but will leave to further work: for instance, noting that the bottleneck in cutting-edge calculations is sometimes a very small number of particularly large rational functions, we believe reconstructing one partial-fractioned term at a time would give maximum scope for on-the-fly observation of patterns that can be exploited in the remaining partial-fractioned terms. Examples of this might be the optimal choice of numerator variables for particular combinations of denominator factors, or the appearance of commonly occurring integer or polynomial prefactors in the numerators of some partial-fractioned terms, or even (as we observe *post hoc* in Sec. V) the appearance of identical numerators in several partial-fractioned terms. Additionally, we believe our method of reconstructing one partial-fractioned term at a time will provide a powerful tool to better analytically understand—and eventually further exploit—the simplification that partial fractioning produces for rational functions in amplitudes and IBP expressions.

Let R denote the rational function we wish to reconstruct and let N be the number of variables it contains. We start by observing that if we can find a special p -adic point \bar{x} at which the denominator d_k of one partial-fractioned term n_k/d_k becomes smaller than all the others, i.e., if for some $\bar{x} \in \mathbb{Q}_p^N$,

$$\exists k: \quad \forall i \neq k, \quad |d_k(\bar{x})|_p < |d_i(\bar{x})|_p, \quad (8)$$

then evaluating the complete rational function R at that p -adic point \bar{x} will give a series

$$R(\bar{x}) = \frac{n_k(\bar{x})}{d_k(\bar{x})} + \mathcal{O}(p^{-m+1}), \quad (9)$$

where $m = -\log_p(|d_k(\bar{x})|_p)$.⁶ In general, this series is $\mathcal{O}(p^{-m})$ and the coefficient of p^{-m} gives useful information about $n_k(\bar{x})$.⁷ In particular, if $n_k = 0$, the $\mathcal{O}(p^{-m})$ term will vanish and so the leading term of the series $R(\bar{x})$ will be $\mathcal{O}(p^{-m+1})$ instead. Furthermore, even when $n_k \neq 0$, we can use Eq. (9) to obtain the leading p -adic digit of $n_k(\bar{x})$, in effect obtaining a finite-field evaluation of n_k . By repeating for other values of \bar{x} that still satisfy Eq. (8) for the same k , we can gather sufficient information to reconstruct the analytic form of n_k , as we will explain below. We have

⁶In this work, \log_p does not denote the p -adic logarithm sometimes seen in the mathematical literature, but instead just an ordinary logarithm with base p .

⁷The handling of rare exceptions to this is described in Secs. IVA 4 and IVA 5.

therefore devised a reconstruction strategy comprising several steps, which we will summarize here and then discuss in detail below.

- (1) Find the common-denominator-form denominator of R .
- (2) Enumerate a complete set of candidates for the denominators $\{d_i\}$ appearing in Eq. (1). At this stage, they are merely candidates, most of which will later turn out to have a vanishing numerator $n_k = 0$.
- (3) Choose the p -adic evaluation points.
- (4) Filter the candidate denominators $\{d_i\}$ by using probes at the chosen points.
- (5) Reconstruct the numerator of one candidate by performing additional probes.
- (6) Repeat steps 4 and 5 to reconstruct the other terms.

From this terse summary the reader might see a superficial resemblance with the reconstruction procedure based on floating-point numbers described in Ref. [22]. While both methods seek to produce compact results, we believe the techniques that we describe in our work offer a practical new capability to study and maximally exploit the simplification offered by partial fractioning, particularly in cases where the rational functions are very large or where floating-point rounding errors need to be avoided.

A. Details of the reconstruction strategy

1. Find the common denominator

Step 1 is straightforward, as explained in Sec. II above. The obtained common denominator $\Delta(x)$ is typically a product of polynomials $f_a(x)$, each raised to some power $\nu_a \in \mathbb{N}$:

$$\Delta(x) = \prod_a [f_a(x)]^{\nu_a}. \quad (10)$$

In our example case, since R_* is a function of five-point massless kinematics, each polynomial $f_a(x)$ has degree 1 in the kinematic variables and space-time dimension. The code developed in this work therefore assumes linear factors, and we leave the extension to higher-order polynomial factors to future work. Also, although not essential to what follows, let us mention that since R_* comes from an IBP reduction into a pure basis of master integrals, each irreducible factor in the common denominator is a polynomial in either the kinematic variables or the space-time dimension D but not both. This feature helps us streamline some of the steps in our calculation.

2. Enumerate candidates for $\{d_i\}$

Having found the common denominator $\Delta(x)$, we now wish to enumerate a set of candidates for the denominators $\{d_i\}$ appearing in Eq. (1). Each denominator d_i will be the product of the elements of a subset G_i of the factors $\{f_a\}$ in Eq. (10), each raised to some power $\mu_{i,a} \in \mathbb{N}$, i.e.,

$$d_i(x) = \prod_b [f_b(x)]^{\mu_{i,b}}, \quad (11)$$

where the index b is understood to run only over the factors present in d_i . It is convenient to implicitly define $\mu_{i,a} = 0$ for the remaining factors f_a not present in G_i . Because of identities such as

$$\frac{1}{yz(y+z)} + \frac{1}{y^2(y+z)} - \frac{1}{y^2z} \equiv 0, \quad (12)$$

where y and z are generic variables, choices will need to be made about which partial-fractioned terms to use as a basis. The `MultivariateApart` package uses a basis determined solely by the common denominator and a specified variable ordering. Such a basis is helpful in analytic calculations because it aids vectorized addition of partial-fractioned expressions, but has a disadvantage in a numerical context because the individual partial fractioned terms n_i/d_i in its output sometimes have higher powers $\mu_{i,b}$ of a denominator factor f_b compared to the corresponding power ν_b in the common-denominator expression. In this work we find it preferable⁸ to use a basis in which

$$\mu_{i,b} \leq \nu_b \quad \forall i, b. \quad (13)$$

Therefore, here in step 2 we create an overcomplete basis by enumerating a complete set $\{d_i\}$ of candidate denominators consistent with Eq. (13), without paying regard to identities such as Eq. (12). Since in subsequent steps we perform the reconstruction one partial-fractioned term at a time, elements of the overcomplete basis that are redundant due to equations like (12) will automatically get filtered out after sufficiently many other basis terms have been reconstructed. To create our overcomplete basis, we use Eq. (11) to tabulate every possible $d_i(x)$, requiring only that $G_i \subset \{f_a\}$ and that Eq. (13) holds. Note that it is permissible for two distinct denominators $d_i(x)$, $d_{i'}(x)$ to share the same factors (i.e., $G_i = G_{i'}$) and differ only in one or more of the powers $\mu_{i,b}, \mu_{i',b}$.

For reasons of efficiency, during step 2 we discard any denominator d_i if its factors $f_b \in G_i$ cannot all simultaneously vanish without making all dimensionful variables in R vanish. Such denominators are not required in our basis since they can either be further partial fractioned, or would get filtered out by a later step. For the same reason, we only allow denominators to contain at most one factor containing the space-time dimension variable.

⁸While this basis is helpful for reconstructing rational functions in partial-fractioned form from numerical evaluations, we do not claim it to necessarily be better for other purposes. Indeed, the size of R_* as reconstructed in our basis is a few percent larger than in the basis of `MultivariateApart`.

3. Choose p -adic evaluation points

Having generated a set of candidate denominators $\{d_i\}$, we next wish to perform p -adic probes of the black-box function R in order to identify any denominator d_i whose corresponding numerator n_i vanishes. Any p -adic evaluation point \bar{x} will induce a weights vector

$$w = \{-\log_p(|f_1(\bar{x})|_p), -\log_p(|f_2(\bar{x})|_p), \dots\}. \quad (14)$$

This weights vector w largely determines which partial-fractioned term (or terms) will be picked out by a p -adic evaluation $R(\bar{x})$, since in general the contribution of a partial-fractioned term n_i/d_i to the series (9) will be of p -adic order

$$-\sum_a \mu_{i,a} w_a, \quad (15)$$

with some caveats to be described shortly. Here w_a is the component of w corresponding to the factor f_a , i.e., $w_a = -\log_p(|f_a(\bar{x})|_p)$.

In this work we generate each p -adic evaluation point \bar{x} by first selecting a desired weights vector w , in a manner described below, and then finding a point \bar{x} satisfying Eq. (14) for that particular w . In practice, Eq. (9) may fail to hold if the partial-fractioned form of R contains explicit factors of p , either in a denominator or in a term in one of the numerators. It may also fail if by chance $n_i(\bar{x}) = 0 + \mathcal{O}(p)$ for some otherwise nonzero numerator n_i . To avoid this, having selected w we probe R at three independent points $\bar{x}_1, \bar{x}_2, \bar{x}_3$, each satisfying Eq. (14) in a distinct p -adic field $\mathbb{Q}_{p_1}, \mathbb{Q}_{p_2}, \mathbb{Q}_{p_3}$ respectively. In most cases, these three probes produce identical valuations $-\log_{p_i}(|R(\bar{x}_i)|_{p_i})$, and we will denote this valuation as $R[w]$:

$$\begin{aligned} R[w] &= -\log_{p_1}(|R(\bar{x}_1)|_{p_1}) \\ &= -\log_{p_2}(|R(\bar{x}_2)|_{p_2}) \\ &= -\log_{p_3}(|R(\bar{x}_3)|_{p_3}). \end{aligned} \quad (16)$$

If the three valuations are not identical, we adopt a heuristic procedure whereby we perform a few additional probes (each in a different p -adic field) until one valuation has been obtained in three more probes than all other valuations and in at least twice as many probes as all other valuations. In this work, we choose p_n to be the n th prime larger than 100, so $p_1 = 101, p_2 = 103, \dots$

It remains to be described how the weights are chosen. We generate a weights vector w by first choosing an algebraically independent⁹ subset $\{h_b\}$ of the factors $\{f_a\}$ in Eq. (10) and then assigning a weight \tilde{w}_b to each

factor h_b . The complete weights vector w in Eq. (14) is inferred from $\{\tilde{w}_b\}$. For example, if $\{f_a\} = \{y, z, y+z\}$ and if we have assigned weights $\{\tilde{w}_b\} = \{5, 2\}$ to the factors $\{h_b\} = \{y, y+z\}$, i.e., $y = \mathcal{O}(p^5), y+z = \mathcal{O}(p^2)$, then it follows that $z = \mathcal{O}(p^2)$ and so $w = \{5, 2, 2\}$.

For comprehensiveness, we tabulated every possible subset of the factors $\{f_a\}$, although for efficiency we ignore factors containing (only) the space-time dimension variable. For each subset $\{h_b\}$, we generate weights in several ways which we have heuristically found to be useful:

- (i) $\tilde{w}_b = 1 \quad \forall b$
- (ii) for some i , set $\tilde{w}_b = \begin{cases} 4 & \text{if } b = i \\ 1 & \text{if } b \neq i \end{cases}$
- (iii) for some i , set $\tilde{w}_b = \begin{cases} 1 & \text{if } b = i \\ 4 & \text{if } b \neq i \end{cases}$
- (iv) $\{\tilde{w}_b\}$ is a permutation of $\{1, 2, \dots, |\{\tilde{w}_b\}|\}$.

Note that some choices of $\{h_b\}$ and $\{\tilde{w}_b\}$ are equivalent to others. For example, the choice $\{y, z\} = \{\mathcal{O}(p), \mathcal{O}(p^4)\}$ leads to the same w as the choice $\{y+z, z\} = \{\mathcal{O}(p), \mathcal{O}(p^4)\}$. We highlight that the above-generated weight vectors, despite only containing small entries $\tilde{w}_b \leq 4$, are sufficient to uniquely pick out individual partial-fractioned terms in the manner that we will describe in Sec. IVA 5.

In total we obtain 6,062 distinct weight vectors, which we will use in the remaining steps to generate p -adic probe points for both filtering and reconstruction purposes.

4. Filter candidate denominators

We now wish to filter the candidates generated in Sec. IVA 2. To do so, we perform a total of 6,062 p -adic probes per prime field, each probe corresponding to one of the weight vectors w generated in the previous step. We expect it to be possible to reduce the number of probes by being more selective about the choices of $\{h_b\}$ and $\{w_b\}$, but since this number is already relatively small we leave that to future work and here simply state that these probes are sufficient. Note that although Sec. IVA 6 requires us to repeat the filtering step many times, these probes do not need to be repeated if the results are cached and the probes are performed with exact integer arithmetic (as in our code) or with sufficiently high p -adic precision in a floating-point-style p -adic evaluation code.

Having performed the probes in the manner described above, we can now filter the candidate denominators $\{d_i\}$. We do this by discarding any candidate d_i if there exists a w for which both of the following relations hold:

$$R[w] > -\sum_b \mu_{i,b} w_b \quad (17)$$

$$\langle f_b : \mu_{i,b} \neq 0 \rangle = \langle f_{b'} : (\mu_{i,b'} \neq 0 | w_{b'} \neq 0) \rangle, \quad (18)$$

where the notation $\langle \rangle$ indicates the polynomial ideal (see, e.g., Ref. [7] for a definition) and $|$ denotes the Boolean

⁹For the massless five-point example R_* in this paper, algebraic independence here is equivalent to linear independence, since all the factors f_a are linear.

OR operation. Here Eq. (17) indicates that the divergence of R at points with p -adic weight w is milder than it would have been if a term with denominator d_i had been present, assuming n_i does not vanish at the probed points. Equation (18) is necessary to ensure that indeed n_i does not vanish at the probed points; for instance, it is important that a partial-fractioned term like $\frac{x^2}{y^4z^5}$ does not get filtered out by probing at $\{x, y, z\} = \{\mathcal{O}(p), \mathcal{O}(p), \mathcal{O}(p)\}$ and obtaining a series $\mathcal{O}(p^{-7})$, which is less divergent than the $\mathcal{O}(p^{-9})$ series that one might have expected based on the behavior of the denominator y^4z^5 alone.

5. Reconstruct a partial-fractioned term

Having performed the filtering, we are now ready to reconstruct a partial-fractioned term. We remind the reader that in this work we seek to reconstruct rational functions one partial-fractioned term n_k/d_k at a time, to obtain the benefits described at the start of this section. This requires finding a probe point \bar{x} satisfying Eq. (8). At first sight, it might be expected that this could potentially require exponentially large values for the weights w_b that generate \bar{x} , which in turn would lead to a need for p -adic evaluations at very high p -adic precision. Remarkably, however, we find in this work (albeit without proof) that it is always possible to generate a suitable weight vector w whose weights w_b are small and for which any point \bar{x} satisfying Eq. (14) will also satisfy Eq. (8). In fact, preliminary work indicates that in future calculations we will continue to be able to pick out one partial-fractioned term at a time while keeping the weights w_b modest in size.

The weight vectors we use for reconstruction are essentially based on those generated in Sec. IVA 3, which we highlight are small: w_b never exceeds 4. We remind the reader that for R_* the factors of the common denominator Δ contain either D or the kinematic factors but not both, and that the weight vectors in Sec. IVA 3 had ignored factors containing D . We find that there always exists a weight vector amongst those generated in Sec. IVA 3 that selects either a unique partial-fractioned term, in accordance with Eq. (8), or several partial-fractioned terms whose denominators differ only by a D -dependent factor. To ensure we reconstruct a single partial-fractioned term at a time, having identified such a weight vector we multiply all its entries w_b by a heuristically chosen factor of 2 or 3 as necessary, and then set one of the D -related entries to 1.¹⁰ The weights vector w obtained in this way uniquely picks out a single partial-fractioned term as desired.

Next, having obtained one such w and the denominator d_k that it uniquely picks out, we construct a simple fully

¹⁰A simpler alternative would be to treat the D -dependent factors on an equal footing to the kinematic factors in all filtering and reconstruction steps. This may lead to a smaller maximum size for w_b , and hence simpler p -adic evaluation points.

generic polynomial ansatz for n_k , constrained only by its mass dimension and the number of variables.¹¹ The mass dimension of n_k can be deduced from the mass dimensions of R and d_k . We determine the number of variables by observing that in general when performing partial fractioning, the numerator n_k of a partial-fractioned term is uniquely defined only up to the addition of arbitrary polynomial multiples of one or more of the factors $f_b \in G_k$ of d_k . Since in our case we have only linear polynomials as factors, the redundancy in n_k can be removed by using each linearly independent factor in d_k to eliminate one variable from n_k . The ansatz that we construct for n_k therefore contains only the residual variables.

Now given w , d_k , and an ansatz for n_k , we generate as many points \bar{x} as there are free parameters in the ansatz, which in this work turns out to be up to 56. Each of these points \bar{x} is chosen to satisfy Eq. (14), for a fixed prime $p_1 = 101$, and at each point we evaluate $R(\bar{x})$ to obtain a sample $n_k(\bar{x})$ modulo p_1 according to Eq. (9). By matrix inversion we obtain an analytic form for n_k modulo p_1 , which we then verify by generating a few more points while keeping p_1 fixed. Note that standard polynomial interpolation methods [7] scale linearly or quadratically with the number of unknowns whereas the ansatz-based interpolation techniques often seen in particle physics scale cubically due to the cost of matrix inversion. We emphasize that reconstructing one partial-fractioned term at a time offers the key advantage that the matrix to be inverted is small (no larger than 56 by 56 in this work), which means our approach can scale well in future calculations.

Having obtained the analytic form of n_k modulo p_1 , we then repeat for a few more primes $p_2 = 103, p_3 = 107, \dots$ until we have sufficient information to reconstruct the exact rational form of n_k using the Chinese remainder theorem and rational number reconstruction [38]. Note that in contrast to finite-field calculations where machine-word-sized primes $p \sim 4 * 10^9$ or $p \sim 2 * 10^{19}$ are standard, p -adic calculations can benefit from faster arithmetic using small primes such as $p = 101$ (or even smaller), regardless of whether the probes are performed with exact integers, floating point, fixed point, or other representations. Since the coefficients in partial-fractioned form are simpler than in common-denominator form, we typically require only three or four primes despite the small size of the primes, plus one more prime for performing checks. The number of required prime fields required to reconstruct a free parameter is proportional to the complexity of that free

¹¹Note that in this work, we do not follow the common practice of “setting a variable to 1,” as this would remove our capability to make that variable p -adically small when performing probes. Instead, we always work with homogeneous polynomials of well-defined mass dimension. We emphasize that this does not add any inefficiency, since a homogeneous polynomial of fixed degree has the same number of free parameters as an inhomogeneous polynomial with one fewer variable but the same maximum degree.

parameter¹² and so in a few cases we used up to nine prime fields (plus one more for checks).

6. Repeat to reconstruct more terms

Having reconstructed a term $R_{\text{rec}} = n_k/d_k$, we next wish to reconstruct the rest of the terms. To do so, we repeat the filtering (Sec. IVA 4) and reconstruction (Sec. IVA 5) steps, but now seek to reconstruct $R - R_{\text{rec}}$. All probes are hence performed on $R - R_{\text{rec}}$ instead of R , and in practice we do this by separately probing R and R_{rec} and then subtracting. Whenever we obtain a new reconstructed term, we then add it onto R_{rec} and then repeat the filtering and reconstruction steps again.

As a technical point, we should highlight the importance of lifting each numerator n_k from \mathbb{F}_p to the rationals in the previous step *before* attempting to reconstruct further terms. If n_k/d_k is added onto R_{rec} after merely reconstructing in \mathbb{F}_p , subleading p -adic contributions from n_k/d_k will continue to contribute to evaluations of $R - R_{\text{rec}}$ and so prevent the reconstruction of other terms with denominators similar to d_k .

Regarding efficiency, we note that R_{rec} is cheap to probe, since we always have an analytic expression for it. Probes of R are therefore the main cost that we seek to minimize. Note that as mentioned in Sec. IVA 4, the probe points used in the filtering step are fixed and so if the probes are exact and the value of R at each of those points is saved once, there is no need to probe R again when repeating the filtering step.¹³ The reconstruction step, conversely, requires probing R (and R_{rec}) at points specially chosen to select a specific denominator. In principle, large efficiency gains could be obtained in the reconstruction step by recycling probes performed in previous iterations of the reconstruction step for similar denominators, but this would require careful further study and so we leave it to future work.

V. RESULTS AND DISCUSSION

By employing the technique presented in Sec. IV, we reconstructed R_* in full. We did so using no knowledge of R_* other than its mass dimension, its common denominator,

TABLE II. Comparison of original and reconstructed form of R_* . The original expression is in common-denominator form, with numerator fully expanded and denominator fully factorized. Sizes are as reported using ByteCount in *Mathematica*. The number of free parameters is obtained by counting the number of terms in the fully expanded numerator(s).

Expression	Size (MB)	Parameters to fit
Original	605	1,369,559
Reconstructed	4.5	52,527 (of which 15,403 nonzero)

and the results of “black-box” p -adic probes. As explained above, the p -adic probes were used for two purposes: filtering (see Sec. IVA 4) and reconstruction (see Sec. IVA 5). For filtering we performed 6,023 p -adic probes per prime field, while for reconstruction we performed a total of 6×10^4 probes per prime field. As mentioned above, we use very small-sized primes $p = \mathcal{O}(100)$ and yet we mostly require only three or four prime fields (plus one more for checks) thanks to the relative simplicity of numerical coefficients in partial-fractioned form compared to common-denominator form. We verified that the reconstructed expression is exactly equal to the original expression. As shown in Table. II, the reconstructed result is 134 times smaller than the common-denominator form that would be produced by conventional finite-field methods. The total number of free parameters fitted was 52,527. Each free parameter accounts for 1 of the 6×10^4 probes (per prime field) performed for reconstruction, the remaining probes being used for checking purposes out of prudence, although these checks never failed. We remind the reader that we never fit more than 56 free parameters at a time since, as discussed extensively in the previous sections, our result was reconstructed one partial-fractioned term at a time, bringing the many advantages already described.

The reconstructed result exhibits further structure which in future work it would be beneficial to study and exploit. This is best seen and discussed with an example. We will consider the following reconstructed terms:

$$\begin{aligned}
& \frac{\frac{45}{1024} s_{45}^6 s_{12}^3}{(D-3)s_{34}^4 s_{51}(-s_{23} + s_{45} + s_{51})^3} + \frac{\frac{9}{5120} s_{45}^6 s_{12}^3}{(D-1)s_{34}^4 s_{51}(-s_{23} + s_{45} + s_{51})^3} - \frac{\frac{693}{5120} s_{45}^6 s_{12}^3}{(2D-7)s_{34}^4 s_{51}(-s_{23} + s_{45} + s_{51})^3} \\
& - \frac{\frac{3}{1024} s_{45}^6 s_{12}^3}{s_{34}^4 s_{51}(-s_{23} + s_{45} + s_{51})^3} + \frac{-\frac{45s_{45}^6 s_{51}^2}{1024} - \frac{135s_{45}^6 s_{51} s_{12}}{1024} - \frac{135s_{45}^6 s_{12}^2}{1024}}{(D-3)s_{34}^4 (s_{23} - s_{45} - s_{51})^3} + \frac{-\frac{9s_{45}^6 s_{51}^2}{5120} - \frac{27s_{45}^6 s_{51} s_{12}}{5120} - \frac{27s_{45}^6 s_{12}^2}{5120}}{(D-1)s_{34}^4 (s_{23} - s_{45} - s_{51})^3} \\
& + \frac{\frac{693s_{45}^6 s_{51}^2}{5120} + \frac{2079s_{45}^6 s_{51} s_{12}}{5120} + \frac{2079s_{45}^6 s_{12}^2}{5120}}{(2D-7)s_{34}^4 (s_{23} - s_{45} - s_{51})^3} + \frac{-\frac{3s_{45}^6 s_{51}^2}{1024} - \frac{9s_{45}^6 s_{51} s_{12}}{1024} - \frac{9s_{45}^6 s_{12}^2}{1024}}{s_{34}^4 (-s_{23} + s_{45} + s_{51})^3}, \tag{19}
\end{aligned}$$

¹²For simplicity, in this work we used the same number of prime fields for all the free parameters in a given numerator n_i , but optimizing by removing this redundancy would be straightforward.

¹³A cache of this sort should be straightforward to implement in software, although for simplicity in this proof-of-concept work we have not done so.

which form a small part of our full reconstructed result. Here s_{ij} are the five kinematic variables of R .

First, let us note that 70% of the free parameters that we fitted turn out to be zero, as anticipated from the discussion at the end of Sec. II. This can be seen in expression (19) in the following way. Expression (19) contains 16 numerator terms and therefore accounts for 16 of the 15,403 non-zero free parameters mentioned in Table II. Considering the first term in (19), we note that *a priori* there was no reason for the numerator to only contain a term $\sim s_{45}^6 s_{12}^3$; it could equally well have contained other mass-squared-dimension-9 combinations of s_{45} and s_{12} , such as $s_{45}^2 s_{12}^7$. To obtain (19) we therefore actually fitted a total of 220 free parameters, of which 204 turned out to be zero. These terms thus account for 220 of the 52,527 free parameters mentioned in Table II. Identifying the vanishing parameters in advance would reduce the number of parameters to be fitted from 52,527 to 15,403, and reduce the number of probes correspondingly.

Second, some of the numerators in our reconstructed result are linearly related to each other by a simple integer multiple. Looking at two of the numerators in the expression (19)

$$n_1 = -\frac{45s_{45}^6 s_{51}^2}{1024} - \frac{135s_{45}^6 s_{51} s_{12}}{1024} - \frac{135s_{45}^6 s_{12}^2}{1024}, \quad (20)$$

$$n_2 = -\frac{9s_{45}^6 s_{51}^2}{5120} - \frac{27s_{45}^6 s_{51} s_{12}}{5120} - \frac{27s_{45}^6 s_{12}^2}{5120}, \quad (21)$$

we notice $n_1 = 25n_2$. If this can economically be discovered prior to reconstruction, it would further reduce the number of free parameters to be fitted and thus the number of probes required.

Third, we notice that in some cases it is possible to combine several of our reconstructed terms and obtain a simpler expression. For example, if we combine together all the terms in expression (19), we obtain the following simple term:

$$-\frac{\frac{3}{512} D(D^2 - 4) s_{45}^6 (s_{51} + s_{12})^3}{(D - 3)(D - 1)(2D - 7) s_{34}^4 s_{51} (-s_{23} + s_{45} + s_{51})^3}. \quad (22)$$

Note, however, that the first two properties do not necessarily imply the third, and we observe from examining other reconstructed terms that combining them in this manner does not always simplify them. The results in Table II do not employ any such recombination of terms, and further study is required to understand which cases are amenable to such simplification and to devise a manner to exploit it during the reconstruction itself, rather than afterwards. This is an interesting direction for exploration, with the potential to yield a further order-of-magnitude reduction in the number of free parameters to be fitted, the

number of probes required per prime field, and the size of the final result. Additionally, since the numerical coefficient $\frac{3}{512}$ in (22) is somewhat simpler than coefficients like $\frac{2079}{5020}$ in (19), fewer prime fields would be required to fit this coefficient.

It is worthwhile to note that, while the patterns and structure exploited in this work—as well as those left for future work—could be studied *post hoc* by partial fractioning an expression obtained by conventional means, our method of reconstructing one partial-fractioned term at a time provides the capability to study and exploit these structures *during* the reconstruction process. For cutting-edge calculations where obtaining any analytic expression in the first place is the principal challenge and goal, we believe this new capability will be a valuable asset.

Going further, we emphasize the desirability of analytically studying the simplifications explored in this work, possibly in conjunction with the observations in Refs. [22,29,32]. We hope our techniques will prove to be useful tools in this regard, with benefits for our theoretical understanding as well as the speed of calculations.

VI. CONCLUSION AND OUTLOOK

In this work we presented a technique to reconstruct rational functions in partial-fractioned form by using numerical evaluations in p -adic fields. Using the example of R_* , a highly nontrivial rational function at the edge of the reach of current calculational techniques, we showed that our technique can reconstruct such functions using 25 times fewer numerical probes than conventional techniques based on finite-field probes, and yields a 100-fold simplification in the size of the reconstructed result.

The rational functions appearing in the final results for amplitudes are typically simpler than the IBP reductions used to compute them, and so having demonstrated our techniques on this large expression from the two-loop five-point massless nonplanar IBP table, we are confident that our techniques can efficiently reconstruct any two-loop five-point massless full-color amplitude. The natural next step would be to apply our techniques to two-loop five-point processes with masses, most of which currently remain unknown. We expect our partial-fractioned reconstruction technique will continue to be beneficial in the massive case, although some technical changes will be required in our code since we can no longer rely on the denominator factors f_a to be linear. More generally, experience shows that partial-fractioning produces simplifications in amplitudes whenever several kinematic scales are present, and so we expect our techniques will be applicable to a wide range of higher-point or higher-loop amplitudes.

Since rational function reconstruction can be used in a wide variety of multiloop contexts, ranging from integrand construction to IBPs to differential equations, we focused in

this work on reducing the number of probes without regard to the choice of method for performing the probes themselves, for which many options exist. One option is a floating-point-like representation of p -adic numbers, which we believe can be achieved with relatively little overhead compared to finite field probes particularly if small primes $p \sim \mathcal{O}(100)$ are used, but may be susceptible to loss of p -adic precision depending on the specific framework used for performing the probes. Another option is to use exact integer probes at points p -adically close to the desired p -adic probe points—see Eq. (7). Such probes completely avoid precision loss and would benefit from a speedup by using small primes but at present it may still be prudent to assume each probe to be slower than a conventional finite-field probe. Since integer probes are exact, we expect them to be well suited to reconstruction probe recycling: using the same probes to reconstruct numerators for multiple terms that have similar denominators. This would further reduce the number of required probes beyond the improvements already reported here and could partly or fully mitigate against any additional cost of p -adic probes compared to finite-field probes.

Finally, we observed that our reconstructed result for R_* displays further patterns and structures which would be worthwhile to study, understand, and exploit in future

work. These observations provide hints of the potential to obtain even further improvements in the speed and reach of this calculational method, as well as potential avenues for starting to seek further understanding of the physical origin of these simplifications and of the structure of the rational functions appearing in scattering amplitudes and IBPs.

ACKNOWLEDGMENTS

We are grateful to Federico Buccioni, Fabrizio Caola, Fernando Febres Cordero, Stephen Jones, Stefano Laporta, Andrew M^cClement, and Alex Mitov for helpful discussions. We also thank the authors of Ref. [16] for providing analytic expressions for the IBP coefficients used as examples in this work. This work has been funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 804394 hipQCD). The work was also supported in part by the U.S. Department of Energy under Grant No. DE-SC0010102. We are also grateful to the Galileo Galilei Institute for hospitality and support during the scientific program on “Theory Challenges in the Precision Era of the Large Hadron Collider,” where part of this article was written.

-
- [1] F. V. Tkachov, A theorem on analytical calculability of 4-loop renormalization group functions, *Phys. Lett.* **100B**, 65 (1981).
 - [2] K. G. Chetyrkin and F. V. Tkachov, Integration by parts: The algorithm to calculate β -functions in 4 loops, *Nucl. Phys.* **B192**, 159 (1981).
 - [3] B. I. Borosh and A. S. Fraenkel, Exact solutions of linear equations with rational coefficients by congruence techniques, *Math. Comput.* **20**, 107 (1966).
 - [4] G. E. Collins, Computing time analyses for some arithmetic and algebraic algorithms, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 1968.
 - [5] M. T. McClellan, The exact solution of systems of linear equations with polynomial coefficients, *J. ACM* **20**, 563 (1973).
 - [6] P. S. Wang, A p -adic algorithm for univariate partial fractions, in *Proceedings of the Fourth ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC ’81 (Association for Computing Machinery, New York, 1981), pp. 212–217.
 - [7] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 1st ed. (Cambridge University Press, Cambridge, England, 1999).
 - [8] A. von Manteuffel and R. M. Schabinger, A novel approach to integration by parts reduction, *Phys. Lett. B* **744**, 101 (2015).
 - [9] T. Peraro, Scattering amplitudes over finite fields and multivariate functional reconstruction, *J. High Energy Phys.* **12** (2016) 030.
 - [10] J. Klappert and F. Lange, Reconstructing rational functions with FireFly, *Comput. Phys. Commun.* **247**, 106951 (2020).
 - [11] T. Peraro, FiniteFlow: Multivariate functional reconstruction using finite fields and dataflow graphs, *J. High Energy Phys.* **07** (2019) 031.
 - [12] J. Klappert, S. Y. Klein, and F. Lange, Interpolation of dense and sparse rational functions and other improvements in FireFly, *Comput. Phys. Commun.* **264**, 107968 (2021).
 - [13] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, B. Page, and V. Sotnikov, Analytic form of the planar two-loop five-parton scattering amplitudes in QCD, *J. High Energy Phys.* **05** (2019) 084.
 - [14] J. Boehm, M. Wittmann, Z. Wu, Y. Xu, and Y. Zhang, IBP reduction coefficients made simple, *J. High Energy Phys.* **12** (2020) 054.
 - [15] B. Agarwal, F. Buccioni, A. von Manteuffel, and L. Tancredi, Two-loop leading colour QCD corrections to $q\bar{q} \rightarrow \gamma\gamma g$ and $qg \rightarrow \gamma\gamma q$, *J. High Energy Phys.* **04** (2021) 201.
 - [16] B. Agarwal, F. Buccioni, A. von Manteuffel, and L. Tancredi, Two-loop helicity amplitudes for diphoton plus jet production in full color, *Phys. Rev. Lett.* **127**, 262001 (2021).

- [17] D. Bendle, J. Boehm, M. Heymann, R. Ma, M. Rahn, L. Ristau, M. Wittmann, Z. Wu, H. Xu, and Y. Zhang, pfd-parallel, a Singular/GPI-Space package for massively parallel multivariate partial fractioning, *Comput. Phys. Commun.* **294**, 108942 (2024).
- [18] M. Heller and A. von Manteuffel, MultivariateApart: Generalized partial fractions, *Comput. Phys. Commun.* **271**, 108174 (2022).
- [19] H. A. Chawdhry, M. Czakon, A. Mitov, and R. Poncelet, Two-loop leading-colour QCD helicity amplitudes for two-photon plus jet production at the LHC, *J. High Energy Phys.* **07** (2021) 164.
- [20] S. Badger, M. Becchetti, E. Chaubey, R. Marzucca, and F. Sarandrea, One-loop QCD helicity amplitudes for $pp \rightarrow t\bar{t}j$ to $\mathcal{O}(\epsilon^2)$, *J. High Energy Phys.* **06** (2022) 066.
- [21] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, and B. Page, Analytic form of planar two-loop five-gluon scattering amplitudes in QCD, *Phys. Rev. Lett.* **122**, 082002 (2019).
- [22] G. Laurentis and D. Maître, Extracting analytical one-loop amplitudes from numerical evaluations, *J. High Energy Phys.* **07** (2019) 123.
- [23] G. De Laurentis and D. Maître, Two-loop five-parton leading-colour finite remainders in the spinor-helicity formalism, *J. High Energy Phys.* **02** (2021) 016.
- [24] D. Bendle, J. Böhm, W. Decker, A. Georgoudis, F.-J. Pfrendt, M. Rahn, P. Wasser, and Y. Zhang, Integration-by-parts reductions of Feynman integrals using singular and GPI-space, *J. High Energy Phys.* **02** (2020) 079.
- [25] S. Badger, C. Brønnum-Hansen, D. Chicherin, T. Gehrmann, H. B. Hartanto, J. Henn, M. Marcoli, R. Moodie, T. Peraro, and S. Zoia, Virtual QCD corrections to gluon-initiated diphoton plus jet production at hadron colliders, *J. High Energy Phys.* **11** (2021) 083.
- [26] S. Badger, H. B. Hartanto, and S. Zoia, Two-loop QCD corrections to $Wb\bar{b}$ production at hadron colliders, *Phys. Rev. Lett.* **127**, 012001 (2021).
- [27] S. Badger, H. B. Hartanto, J. Kryś, and S. Zoia, Two-loop leading-colour QCD helicity amplitudes for Higgs boson production in association with a bottom-quark pair at the LHC, *J. High Energy Phys.* **11** (2021) 012.
- [28] S. Abreu, F. Febres Cordero, H. Ita, M. Klinkert, B. Page, and V. Sotnikov, Leading-color two-loop amplitudes for four partons and a W boson in QCD, *J. High Energy Phys.* **04** (2022) 042.
- [29] G. De Laurentis and B. Page, Ansätze for scattering amplitudes from p-adic numbers and algebraic geometry, *J. High Energy Phys.* **12** (2022) 140.
- [30] G. De Laurentis, H. Ita, M. Klinkert, and V. Sotnikov, Double-virtual NNLO QCD corrections for five-parton scattering: The gluon channel, *Phys. Rev. D* **109**, 094023 (2024).
- [31] G. De Laurentis, H. Ita, and V. Sotnikov, Double-virtual NNLO QCD corrections for five-parton scattering: The quark channels, *Phys. Rev. D* **109**, 094024 (2024).
- [32] J. M. Campbell, G. De Laurentis, and R. K. Ellis, Vector boson pair production at one loop: Analytic results for the process $q\bar{q}\ell\bar{\ell}\ell'\bar{\ell}'g$, *J. High Energy Phys.* **07** (2022) 096.
- [33] B. Agarwal, F. Buccioni, F. Devoto, G. Gambuti, A. von Manteuffel, and L. Tancredi, Five-parton scattering in QCD at two loops, *Phys. Rev. D* **109**, 094025 (2024).
- [34] E. K. Leinartas, Factorization of rational functions of several variables into partial fractions, *Izvestiia vysshikh uchebnykh zavedenii Matematika / Ministerstvo vysshego obrazovaniia SSSR* **22**, 35 (1978), <https://scholar.google.com/scholar?cluster=1418867281511936050>.
- [35] A. Raichev, Leinartas's partial fraction decomposition, [arXiv:1206.4740](https://arxiv.org/abs/1206.4740).
- [36] C. Meyer, Transforming differential equations of multi-loop Feynman integrals into canonical form, *J. High Energy Phys.* **04** (2017) 006.
- [37] C. Meyer, Algorithmic transformation of multi-loop master integrals to a canonical basis with CANONICA, *Comput. Phys. Commun.* **222**, 295 (2018).
- [38] D. E. Knuth, *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms* (Addison-Wesley Longman Publishing Co., Inc., Reading, MA, 1997).