Scaling up to Multivariate Rational Function Reconstruction

Andreas Maier

Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, 15738 Zeuthen, Germany

I present an algorithm for the reconstruction of multivariate rational functions from black-box probes. The arguably most important application in high-energy physics is the calculation of multi-loop and multi-leg amplitudes, where rational functions appear as coefficients in the integration-by-parts reduction to basis integrals. I show that for a dense coefficient the algorithm is nearly optimal, in the sense that the number of required probes is close to the number of unknowns.

1 Introduction

The reduction of a large number of scalar Feynman integrals to a smaller set of basis (or master) integrals is an almost universal step in precision calculations in quantum field theories. In many cases, it is also among the most challenging parts of the computation, and has therefore seen lots of attention and development over the years.

The current standard approach is to derive integration-by-parts identities [1, 2] for a set of seed integrals with fixed powers of propagators and irreducible scalar products and solve the resulting system of linear relations via Gauss elimination [3]. The result is a subset of the seed integrals, each expressed in terms of a linear combination of basis integrals. The coefficients are rational functions of kinematic invariants and the space-time dimension.

It is often advantageous to insert numerical values for the dimension and the invariants and solve the system over a finite field [4]. This strategy was initially used to quickly eliminate redundant relations [5]. However, solving the system with sufficiently many different probes, i.e. different numeric values for the variables, it is possible to reconstruct the full result [6, 7]. In this way, one avoids large intermediate expressions and can restrict the reconstruction to those coefficients that are actually needed for the final result, for example a scattering amplitude. Further advantages include ease of parallelisation, lower memory usage, and the possibility to optimise the system further after a computationally cheap pilot run.

This general strategy has been further developed along several directions. Right from the start, one can attempt to find combinations of relations that lead to better behaved systems or even an explicit recursive solution to the reduction problem [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Since in a finite-field reduction the very same system of linear equations has to be solved many times, it can be worthwhile to record the solution steps once, optimise the recorded sequence, and replay it to rapidly obtain further probes [20]. Other efforts target the reconstruction itself. The complexity of the rational function coefficients depends on the chosen basis integrals, and a judicious choice results in a factorisation between the dimension and the kinematic invariants [21, 22]. Identifying common factors in the coefficients can further reduce the number of probes required for the reconstruction [23, 24, 25, 26]. Finally, the number of required probes depends on the chosen reconstruction method, and various algorithms with different strengths have been explored [27, 28, 29, 30].

In the following, I present an algorithm for "scaling up" from rational function reconstruction in a single variable to the multivariate case. In section 2, I review reconstruction in a single variable using Thiele interpolation [31]. I then discuss a way to generalise the method to the multivariate case in section 3. The intended application is the reconstruction of coefficients in the reduction to basis integrals. In section 4, I apply the algorithm to complex reduction coefficients in a massive four-loop propagator example and in the two-loop amplitude for diphoton plus jet production [32]. I find that the number of required probes is close to optimal when the fraction of vanishing polynomial coefficients in the numerator and the denominator of the rational function is small.

2 Univariate Rational Function Reconstruction

Excellent introductions into the reconstruction of polynomials and rational functions are given in [7, 27]. Let us briefly review the case of a univariate rational function.

We are given a rational function f in a single variable for which we want to find an explicit form

$$f(x) = \frac{p_n(x)}{p_d(x)}, \tag{1}$$

where p_n, p_d are unknown polynomials with no common roots. f is a "black box", meaning that our only piece of information is an algorithm for computing $f(t)$ for any t in the domain of f. One strategy is to construct rational interpolations f_N for N probes $(t_i, f(t_i))$ with $i = 1, \ldots, N$. If N is large enough, we then find $f_N = f$ with high probability.

We start with a single probe $(t_1, f(t_1))$ and a constant interpolation

$$f_1(x) = a_1. \tag{2}$$

Requiring $f_1(t_1) = f(t_1)$ we immediately find $a_1 = f(t_1)$. We then add a second probe $(t_2, f(t_2))$. If $f_1(t_2) = f(t_2)$ we note that we found agreement and continue with the next probe. Otherwise, we introduce the interpolation

$$f_2(x) = a_1 + \frac{x - t_1}{a_2} \tag{3}$$

with $a_1 = f(t_1)$ as before and $a_2 = \frac{t_2 - t_1}{f(t_2) - a_1}$. In general, after N independent probes the interpolation has the form [31]

$$f_N(x) = a_1 + \cfrac{x - t_1}{a_2 + \cfrac{x - t_2}{a_3 + \cfrac{x - t_3}{\cdots + \cfrac{x - t_{N-1}}{a_N}}}}. \tag{4}$$

For adding the next probe $(t_{N+1}, f(t_{N+1}))$ we compute

$$c_1 = f(t_{N+1}), \qquad c_{i+1} = \frac{t_{N+1} - t_i}{c_i - a_i}, \tag{5}$$

for all $i \leq N$ and construct f_{N+1} with $a_{N+1} = c_{N+1}$ and $a_1, \ldots, a_N$ taken from f_N. If the denominator in equation (5) vanishes, the probe adds no new information. We then check if our present interpolation already agrees for this point, i.e. $f_N(t_{N+1}) = f(t_{N+1})$, and terminate the reconstruction as soon as some chosen number of probes are predicted correctly. Otherwise we continue with the next probe.

In many cases, the computational cost of the reconstruction is dominated by either the evaluation of the probes or by the divisions in the calculation of the auxiliary constants c_j in equation (5). In the latter case, the alternative division-free recursion

$$n_1 = f(t_{N+1}), \qquad n_{i+1} = (t_{N+1} - t_i)d_i, \tag{6}$$
$$d_1 = 1, \qquad d_{i+1} = n_i - a_i d_i, \tag{7}$$

leading to $a_{N+1} = \frac{n_{N+1}}{d_{N+1}}$ can be much more efficient.

Note that the reconstruction is optimal if two conditions are fulfilled. First, the degrees of the numerator and the denominator in f should be equal or the degree of the numerator should be larger by one. Second, the numerator and denominator polynomials should be perfectly dense, with all coefficients non-zero. In this case the number of required probes is equal to the number of unknown coefficients plus the chosen number of probes used for confirmation. Empirically, the rational functions encountered in integration-by-parts reduction without any kinematic invariants are close to ideal with a typical overhead of about 10% in the number of required probes.

3 Scaling up to Multiple Variables

In general, the rational function to be reconstructed has the form

$$f(x_1, \ldots, x_n) = \frac{\sum C_{p_1,\ldots,p_n} x_1^{p_1} \cdots x_n^{p_n}}{\sum D_{q_1,\ldots,q_n} x_1^{q_1} \cdots x_n^{q_n}}, \tag{8}$$

where the sums run over all powers $0 \leq p_1 \leq P_1, \ldots, 0 \leq p_n \leq P_n$ in the numerator and $0 \leq q_1 \leq Q_1, \ldots, 0 \leq q_n \leq Q_n$ in the denominator. The degrees $P_1, \ldots, P_n$ and $Q_1, \ldots, Q_n$ are a priori unknown. To reconstruct a pair (P_i, Q_i) of degrees we can set all other variables to some fixed value, i.e. $x_j = t_j$ for all $j \neq i$, and perform a univariate rational function reconstruction in the remaining free variable x_i [7, 27, 33].

Once we know the degrees, we can in principle determine the unknown coefficients C, D by simply solving a linear system of equations. Knowing the value of $f(t_1, \ldots, t_n)$, we obtain the linear equation

$$\sum C_{p_1,\ldots,p_n} t_1^{p_1} \cdots t_n^{p_n} = f(t_1, \ldots, t_n) \sum D_{q_1,\ldots,q_n} t_1^{q_1} \cdots t_n^{q_n} \tag{9}$$

directly from the definition in equation (8). For $N+1$ coefficients $C_{p_1,\ldots,p_n}, D_{q_1,\ldots,q_n}$, we require N probes to express them in terms of a single coefficient which we can set to an arbitrary non-zero value to fix the overall normalisation. If the numerator and denominator polynomials are dense, the reconstruction is optimal in the sense of needing the lowest possible number of probes. However, assuming constant-time arithmetic for the arguments and coefficients, the time complexity for solving the dense linear system is $\mathcal{O}(N^3)$ with a space complexity of $\mathcal{O}(N^2)$. In practice it is usually better to use a method that requires more probes but has better scaling behaviour.

The univariate reconstruction based on Thiele interpolation we discussed in section 2 only requires $\mathcal{O}(N)$ space to store the arguments $t_1, \dots, t_N$ and the coefficients $a_1, \dots, a_N$. To determine these coefficients, one needs to calculate $N(N+1)/2 = \mathcal{O}(N^2)$ auxiliary coefficients (cf. equation (5)), each of which can be computed in constant time. Can we generalise that method to the multivariate case while preserving the superior scaling behaviour?

The main idea is to set all of the variables $x_1, \dots, x_n$ to a single variable x, scaled to distinct powers so that we can recover the full dependence on $x_1, \dots, x_n$ after the reconstruction. Concretely, we consider the auxiliary function $g(x) = f(x^{\alpha_1}, \dots, x^{\alpha_n})$ with

$$\alpha_1 = 1, \qquad \alpha_{i+1} = [1 + \max(P_i, Q_i)]\alpha_i. \tag{10}$$

We then use univariate reconstruction in x to find an explicit form for $g(x)$. For each term of the form $C_i x^i$ we can formally interpret the power i as a number in a mixed radix numeral system, where the individual digits correspond to the powers $p_1, \dots, p_n$ of $x_1, \dots, x_n$.

Let us consider a simple example for a black-box function $f(x_1, x_2)$. Setting x_2 to a fixed value t_2 and using univariate reconstruction in x_1 we find

$$f(x_1, t_2) = \frac{C_0(t_2) + C_1(t_2)x_1}{D_0(t_2) + D_2(t_2)x_1^2}. \tag{11}$$

We ignore the coefficients depending on t_2; our only goal was to learn that the largest power of x_1 is 2. This tells us to set $\alpha_2 = 2+1$, so we introduce the auxiliary function $g(x) = f(x, x^3)$. From univariate reconstruction we obtain

$$g(x) = \frac{1 + x + x^4}{1 + x^2 + x^3}. \tag{12}$$

The last step is to read off the corresponding powers of the original variables x_1, x_2. For the exponents in g we have mixed radix notation $1 = 0_? 1_3, 2 = 0_? 2_3, 3 = 1_? 0_3, 4 = 1_? 1_3$, where the subscript indicates the numeral base of the corresponding position and the base of the leading digit is irrelevant. This tells us that the original function is

$$f(x_1, x_2) = \frac{1 + x_1 + x_1 x_2}{1 + x_1^2 + x_2}. \tag{13}$$

The method described so far can suffer from accidental cancellations between numerator and denominator. For example, for $f(x_1, x_2) = \frac{x_2}{x_1}$ we would obtain the auxillary function $g(x) = x$, which would lead us to believe the original function was $f(x_1, x_2) = x_1$. To prevent this, we additionally shift the rescaled argument by a randomly chosen number.[1] Spurious cancellations have to involve two or more different variables. We therefore expect to avoid them by having at least one shifted variable in each possible combination, i.e. we shift each variable except one.

Let us summarise the algorithm. Given a rational black-box function f in n variables $x_1, \dots, x_n$

1. For each variable x_i with $i < n$, find the largest powers P_i and Q_i in the numerator and denominator. To do this, set all other variables to randomly chosen values, $x_j = t_j$ for all $j \neq i$, and use univariate reconstruction in x_i.
2. Compute the scaling powers $\alpha_1, \dots, \alpha_n$ using equation (10) and choose random shifts $s_1, \dots, s_n$. One of the shifts can be set to zero, e.g. $s_1 = 0$.
3. Use univariate reconstruction to find $g(x)$ from probes $(t_i, f(t_i^{\alpha_1} + s_1, \dots, t_i^{\alpha_n} + s_n))$.
4. For each term $C_i x^i$ in $g(x)$, recover the powers $p_1, \dots, p_n$ of the original variables $x_1, \dots, x_n$ from the mixed-radix digits of i. Then, replace $x^i \to (x_1 - s_1)^{p_1} \cdots (x_n - s_n)^{p_n}$.

For the main application we have in mind, namely the reconstruction of coefficients in the reduction to basis integrals, one aims to recover many rational functions from probes with the same arguments. One way to use the same arguments $t_i^{\alpha_1} + s_1, \dots, t_i^{\alpha_n} + s_n$ for different functions $f, h, \dots$ is to choose the exponents $\alpha_1, \dots, \alpha_n$ according to the maximum powers of the respective variables in *any* of the numerators and denominators of $f, h, \dots$. Often, the highest powers of all variables will be determined by a single function, such that the maximum number of required probes remains unaffected. The price to pay is that for the simple functions many vanishing coefficients will be reconstructed, increasing the computing time required for the reconstruction itself. Alternatively, different sets of probe arguments can be used for functions of widely disparate complexity.

[1] This random shift is also used with a slightly different purpose in the reconstruction algorithm by Cuyt and Lee [7, 27, 33]. There, the goal is to ensure a unique structure and uniform coefficient normalisation of the reconstructed function.

One of the main advantages of numerical reduction is ease of parallelisation. This requires that subsequent probes can be chosen without having to wait for the outcome of feeding earlier probes into the reconstruction algorithm. In this respect, algorithms for the reconstruction of dense rational functions tend to perform better than methods aimed at sparse rational functions with many vanishing coefficients, as for the latter the probe selection typically has to be adjusted dynamically. The presented algorithm mostly decouples the seed choice from the reconstruction progress. For n variables, the selection strategy has to be updated n times — after determining the powers of each of the first $n-1$ variables and once more to use the final rescaled arguments.

4 Application to the Reduction to Basis Integrals

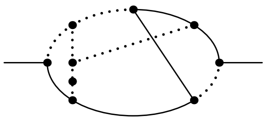
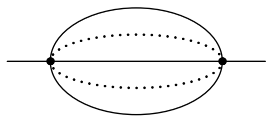
Let us now assess the efficiency of the algorithm presented in section 3 in practical applications. For brevity, we will refer to the new method as "scaling" reconstruction. We compare it to an algorithm proposed by Cuyt and Lee [33]. This algorithm is described in detail in [7, 27]. We briefly recall the main steps. First, the variables are rescaled with a common factor t and shifted, leading to $(x_1, \dots, x_n) = (ty_1 + s_1, \dots, ty_n + s_n)$ with $y_n = 1$. One then performs a univariate rational reconstruction in t. Starting from the highest powers, the coefficients of t^i are reconstructed as polynomials in $y_1, \dots, y_n$ and transformed back to the original variables. For the comparison we use state-of-the-art implementations in the public codes `FireFly` [27, 29, 34] and `FiniteFlow` [28, 35]. The comparison code and the example rational functions in computer-readable form are available from https://github.com/a-maier/scaling-rec.

For the scaling algorithm, the reconstruction is first performed over a number of prime fields $\mathbb{Z}_P$, using arithmetic algorithms taken from NTL [36]. We start with $P = 1\,152\,921\,504\,606\,846\,883$ and move to the next smaller prime numbers as needed. The resulting coefficients are lifted to a higher characteristic with the Chinese remainder theorem, specifically Bézout's identity. The actual rational coefficients are then reconstructed from the finite-field integers via an algorithm by Wang [37]. Again, details are given in [7, 27].

In principle, the reconstruction can be simplified tremendously exploiting the structure of the result [23, 24, 25]. Both `FireFly` and `FiniteFlow` can factorise the numerator and denominator to a certain degree. `FiniteFlow` determines the minimal degree in each variable to automatically factor out common monomials. `FireFly` optionally performs univariate factorisation to identify any factors depending on a single variable. Since we are mainly interested in assessing the underlying reconstruction algorithms, we disable `FireFly`'s factorisation in the following comparisons. As there is no option to switch off factorisation in `FiniteFlow`, we additionally compare the reconstruction after removing all monomial factors from the function to be reconstructed.

4.1 Massive Four-Loop Propagator

Our first benchmark point is a coefficient in the differential equation [38, 39] for a four-loop massive propagator. Setting the mass $m = 1$, the propagator is a function of $z = p^2$, where p is the external four-momentum. One obtains

$$z\frac{d}{dz}$$  $$= q(z,d)$$  $$+ \ldots, \qquad (14)$$

with the ellipsis indicating a linear combination of further basis integrals with less complex coefficients. $q(z,d)$ is a rational function in z and the space-time dimension d, where the numerator degrees in z and d are $P_z = P_d = 81$ and the denominator degrees are $Q_z = 80$ and $Q_d = 78$. The numbers of probes required for reconstructing the function over the first characteristic are shown in table 1 for the scaling algorithm and `FireFly` together with a hypothetical optimal algorithm that can determine one unknown coefficient per probe.

For a full rational function reconstruction probes are needed in several additional prime fields. Since different implementations vary vastly in the amount of reused information we refrain from a quantitative comparison.

Method	Number of Probes
This work	13 594
`FireFly`	16 373
Optimal	12 721

Table 1: Number of probes required to reconstruct a specific coefficient in the differential equation for a four-loop propagator over the first characteristic.

Comparing the "Optimal" entry of table 1 to the total number of 13 123 monomials in the ansatz given by equation (8) we observe that $q(z,d)$ is dense in the sense that about 97% of the coefficients in the ansatz are non-zero. We see that the scaling algorithm performs close to optimal, with an overhead of about 7% additional probes. In comparison, `FireFly` requires approximately 29% more probes.

The denominator of the reconstructed function contains an overall factor of d^1z^3. The improvement gained by identifying and removing this factor is illustrated in table 2, where we now also include `FiniteFlow`. The `FiniteFlow` entry does not include a few hundred probes used to determine the overall degree of the rational function and the degrees with respect to the individual variables from Thiele interpolation, c.f. section 2 [40]. The other entries count the total number of function evaluations.

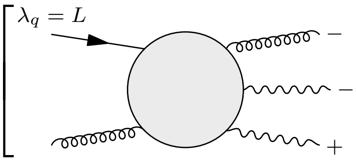
Method	Number of Probes
This work	13 594
`FireFly`	16 020
`FiniteFlow`	$\gtrsim$ 18 205
Optimal	12 721

Table 2: Number of probes required to reconstruct a specific coefficient in the differential equation for a four-loop propagator over the first characteristic after removing an overall monomial factor. The `FiniteFlow` entry does not include a few hundred probes used for degree determinations.

For the algorithm presented in section 3, the scaling powers in equation (10) are completely determined by the numerator in the present example. Thus, removing factors from the denominator does not affect the number of probes needed. However, we do observe a slight reduction in the number of required evaluations with `FireFly`, reducing the overhead to 26% compared to the optimum. The number of evaluations needed with `FiniteFlow` exceeds the number of non-vanishing coefficients to be determined by about 43%.

4.2 Diphoton Plus Jet Production at Two Loops

Next, let us consider the two-loop amplitude for diphoton plus jet production, taken from [32]. Specifically, we choose the parity-even contribution with a left-handed quark and a gluon in the initial state, a negative gluon helicity, opposite-sign photon helicities, and no closed fermion loops. Denoting the quark helicity by λ_q, the number of active flavours by n_f, the number of colours by N_C, and the parity transformation operator by $\hat{P}$, the reduction has the structure

$$\frac{1+\hat{P}}{2}\Big[$$


$$\Big]_{n_f=0} = c_{-2}N_C^{-2} + c_0 N_C^0 + c_2 N_C^2, \qquad (15)$$

where c_{-2}, c_0, c_2 are linear combinations of pentagon functions [41] with rational coefficients. From c_0 we select the largest of these coefficients by Mathematica's `ByteCount`. We write this coefficient as a rational function in $x_{23}, x_{34}, x_{45}, x_{51}$, where $x_{ij} = \frac{s_{ij}}{s_{12}}$, and

$$s_{12} = (p_1 + p_2)^2, s_{23} = (p_2 - p_3)^2, s_{34} = (p_3 + p_4)^2, s_{45} = (p_4 + p_5)^2, s_{51} = (p_1 - p_5)^2$$

are Mandelstam invariants. After determining the numerator and denominator degrees our ansatz according to equation (8) contains 136 934 unknown coefficients. However, the actual rational function is much sparser than in the example in section 4.1 and only approximately 22% of these coefficients are non-zero. In this example, the full coefficient can be reconstructed using a single prime field. We collect the number of required probes in table 3.

Method	Number of Probes
This work	169 132
`FireFly`	163 094
Optimal	30 490

Table 3: Number of probes required to reconstruct the coefficient in the reduction of the two-loop diphoton plus jet amplitude.

The scaling algorithm introduced in section 3 performs slightly worse than `FireFly`'s reconstruction. Both algorithms are far from optimal for this scenario, requiring more than five probes for each unknown coefficient.

The number of required reconstruction probes after removing an overall monomial factor $x_{23}^2 x_{34}^2 x_{45}^2 x_{51}^2$ is shown in table 4. As in section 4.1, we see no improvement for the implementation of the algorithm presented in this work. In contrast, `FireFly` needs approximately 20% fewer function evaluations than before. Most strikingly, `FiniteFlow` is much closer to optimal than both `FireFly` and the scaling reconstruction implementation, especially when using the `FFPolyVandermonde` alternative polynomial reconstruction method. Even when enabling its identification of univariate factors, `FireFly` still requires 87 485 probes, substantially more than `FiniteFlow`. This difference between `FiniteFlow` and `FireFly` is unexpected and deserves closer inspection. However, since the focus of the present work is on the scaling algorithm and dense reconstruction, we leave further investigation to future work.

Method	Number of Probes
This work	169 132
`FireFly`	129 894
`FiniteFlow`	$\gtrsim$ 49 216
`FiniteFlow` with `FFPolyVandermonde`	$\gtrsim$ 47 381
Optimal	30 490

Table 4: Number of required probes after removing the overall monomial prefactor. The `FiniteFlow` entries do not include a few hundred probes used for degree determinations.

5 Conclusion

I have presented an algorithm for the reconstruction of dense multivariate rational functions. Multiple variables are mapped onto a single variable, using scaling powers and shifts chosen such that the mapping can be inverted. In this way, the problem is reduced to well-known univariate rational reconstruction.

The algorithm is tested on two examples taken from complex reductions to basis integrals, a massive four-loop propagator and a two-loop five-point amplitude. For the dense rational function encountered in the four-loop problem, the required number of probes exceeds the number of unknown coefficients by only about 7%. This compares favourably with the current state-of-the-art programs `FireFly` [27, 29] and `FiniteFlow` [28].

In the sparse two-loop example, the number of probes needed is about 4% above the `FireFly` result when disabling factorisation. However, a comparison to `FiniteFlow` reveals that in this case there is substantial room for improvements for both `FireFly` and the presented algorithm. A further promising avenue for future research would be to combine the univariate mapping with sparse rational reconstruction in a single variable, see e.g. [42].

Acknowledgements

I thank P. Marquard for enlightening discussions and helpful comments on the manuscript. I further thank F. Lange and T. Peraro for communication on `FireFly` and `FiniteFlow` and for contributing code to the comparison in section 4.

References

[1] K. G. Chetyrkin and F. V. Tkachov, *Integration by parts: The algorithm to calculate β-functions in 4 loops*, Nucl. Phys. B **192** (1981) 159.

[2] F. V. Tkachov, *A theorem on analytical calculability of 4-loop renormalization group functions*, Phys. Lett. B **100** (1981) 65.

[3] S. Laporta, *High precision calculation of multiloop Feynman integrals by difference equations*, Int. J. Mod. Phys. **A15** (2000) 5087 [hep-ph/0102033].

[4] M. Kauers, *Fast solvers for dense linear systems*, Nucl. Phys. B Proc. Suppl. **183** (2008) 245.

[5] P. Kant, *Finding Linear Dependencies in Integration-By-Parts Equations: A Monte Carlo Approach*, Comput. Phys. Commun. **185** (2014) 1473 [1309.7287].

[6] A. von Manteuffel and R. M. Schabinger, *A novel approach to integration by parts reduction*, Phys. Lett. B **744** (2015) 101 [1406.4513].

[7] T. Peraro, *Scattering amplitudes over finite fields and multivariate functional reconstruction*, JHEP **12** (2016) 030 [1608.01902].

[8] S. G. Gorishnii, S. A. Larin, L. R. Surguladze and F. V. Tkachov, *Mincer: Program for Multiloop Calculations in Quantum Field Theory for the Schoonschip System*, Comput. Phys. Commun. **55** (1989) 381.

[9] M. Steinhauser, *MATAD: A Program package for the computation of MAssive TADpoles*, Comput. Phys. Commun. **134** (2001) 335 [hep-ph/0009029].

[10] J. Gluza, K. Kajda and D. A. Kosower, *Towards a Basis for Planar Two-Loop Integrals*, Phys. Rev. D **83** (2011) 045012 [1009.0472].

[11] R. M. Schabinger, *A New Algorithm For The Generation Of Unitarity-Compatible Integration By Parts Relations*, JHEP **01** (2012) 077 [1111.4220].

[12] R. N. Lee, *Presenting LiteRed: a tool for the Loop InTEgrals REDuction*, 1212.2685.

[13] R. N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, J. Phys. Conf. Ser. **523** (2014) 012059 [1310.1145].

[14] H. Ita, *Two-loop Integrand Decomposition into Master Integrals and Surface Terms*, Phys. Rev. D **94** (2016) 116015 [1510.05626].

[15] K. J. Larsen and Y. Zhang, *Integration-by-parts reductions from unitarity cuts and algebraic geometry*, Phys. Rev. D **93** (2016) 041701 [1511.01071].

[16] B. Ruijl, T. Ueda and J. A. M. Vermaseren, *Forcer, a FORM program for the parametric reduction of four-loop massless propagator diagrams*, Comput. Phys. Commun. **253** (2020) 107198 [1704.06650].

[17] A. Pikelner, *FMFT: Fully Massive Four-loop Tadpoles*, Comput. Phys. Commun. **224** (2018) 282 [1707.01710].

[18] Z. Wu, J. Boehm, R. Ma, H. Xu and Y. Zhang, *NeatIBP 1.0, a package generating small-size integration-by-parts relations for Feynman integrals*, Comput. Phys. Commun. **295** (2024) 108999 [2305.08783].

[19] X. Guan, X. Liu, Y.-Q. Ma and W.-H. Wu, *Blade: A package for block-triangular form improved Feynman integrals decomposition*, Comput. Phys. Commun. **310** (2025) 109538 [2405.14621].

[20] V. Magerya, *Rational Tracer: a Tool for Faster Rational Function Reconstruction*, 2211.03572.

[21] A. V. Smirnov and V. A. Smirnov, *How to choose master integrals*, Nucl. Phys. B **960** (2020) 115213 [2002.08042].

[22] J. Usovitsch, *Factorization of denominators in integration-by-parts reductions*, 2002.08173.

[23] 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** (2019) 082002 [1812.04586].

[24] G. De Laurentis and B. Page, *Ansätze for scattering amplitudes from p-adic numbers and algebraic geometry*, JHEP **12** (2022) 140 [2203.04269].

[25] H. A. Chawdhry, *p-adic reconstruction of rational functions in multiloop amplitudes*, Phys. Rev. D **110** (2024) 056028 [2312.03672].

[26] X. Liu, *Reconstruction of rational functions made simple*, Phys. Lett. B **850** (2024) 138491 [2306.12262].

[27] J. Klappert and F. Lange, *Reconstructing rational functions with FireFly*, Comput. Phys. Commun. **247** (2020) 106951 [1904.00009].

[28] T. Peraro, *FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs*, JHEP **07** (2019) 031 [1905.08019].

[29] J. Klappert, S. Y. Klein and F. Lange, *Interpolation of dense and sparse rational functions and other improvements in FireFly*, *Comput. Phys. Commun.* **264** (2021) 107968 [2004.01463].

[30] A. V. Belitsky, A. V. Smirnov and R. V. Yakovlev, *Balancing act: Multivariate rational reconstruction for IBP*, *Nucl. Phys. B* **993** (2023) 116253 [2303.02511].

[31] T. N. Thiele, *Interpolationsrechnung*. B. G. Teubner, 1909.

[32] 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** (2021) 262001 [2105.04585].

[33] A. A. M. Cuyt and W. Lee, *Sparse interpolation of multivariate rational functions*, *Theor. Comput. Sci.* **412** (2011) 1445.

[34] J. Klappert, S. Y. Klein and F. Lange, FireFly version 2.0.3, `https://gitlab.com/firefly-library/firefly`.

[35] T. Peraro, FiniteFlow revision 7754b861168ae5aea4c4d555a37e2e88d18b7725, `https://github.com/peraro/finiteflow`.

[36] V. Shoup, *NTL: A Library for doing Number Theory*. `https://libntl.org/`

[37] 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, (New York, NY, USA), p. 212–217, Association for Computing Machinery, 1981, DOI.

[38] A. V. Kotikov, *Differential equations method: New technique for massive Feynman diagrams calculation*, *Phys. Lett. B* **254** (1991) 158.

[39] E. Remiddi, *Differential equations for Feynman graph amplitudes*, *Nuovo Cim. A* **110** (1997) 1435 [hep-th/9711188].

[40] T. Peraro, Private communication.

[41] T. Gehrmann, J. M. Henn and N. A. Lo Presti, *Pentagon functions for massless planar scattering amplitudes*, *JHEP* **10** (2018) 103 [1807.09812].

[42] Q. Huang and X. Gao, *Sparse rational function interpolation with finitely many values for the coefficients*, *CoRR* **abs/1706.00914** (2017) [1706.00914].