Advances in computational methods for Quantum Field Theory calculations
Ruijl, B.J.G.

Author: Ruijl, B.J.G.

Title: Advances in computational methods for Quantum Field Theory calculations

Issue Date: 2017-11-02

7 CONCLUSIONS

In this chapter we start by answering the three research questions in sec. 7.1. Using these answers, we address the problem statement in 7.2. Finally, we provide an outlook in sec. 7.3.

7.1 ANSWERS TO THE RESEARCH QUESTIONS

> **RQ1:** *To what extent can the number of arithmetic operations of large multivariate polynomials be reduced?*

In chapter 2 we consider various algorithms to improve expression simplification. We find that the state space of Horner schemes is flat, which makes it a good candidate for Stochastic Hill Climbing [11]. We show that for relevant polynomials derived from scattering experiments there is a speed gain of about a factor ten compared to MCTS methods to find a near-optimal solution. Since evaluations are slow, this means less time has to be spent in creating an expression suitable for Monte Carlo integration. The quality of the solution is often more than an order of magnitude better than the input.

> **RQ2:** *How can we construct a program that can compute four-loop massless propagator integrals more efficiently?*

In chapter 3 we constructed the FORCER program, which uses parametric integration-by-parts (IBP) reductions to reduce four-loop massless propagator integrals. We have demonstrated that FORCER is much faster than its competitors, and is able to compute the four-loop beta function in only 3 minutes [1, 9]. In chapter 4 we have computed physically relevant processes at four loops using FORCER [5, 8, 115]. The three most important calculations are (1) the finite pieces of the propagators and three-vertices with one vanishing momentum [2], (2) the computation of the non-singlet splitting function to $N = 16$ [115], and (3) the reconstruction of the large-n_c leading to a new term in the cusp anomalous dimension [115, 116].

> **RQ3:** *To what extent can we compute the poles of five-loop diagrams using only four-loop diagrams more efficiently?*

In chapter 5 we have generalised the R^*-operation to be applicable to Feynman diagrams with arbitrary numerator structure [3]. This allows for the computation of the poles of a much broader class of integrals. The R^*-method will generate many counterterms, but we describe how to exploit symmetries between them to reduce their number. After more optimisations described in 6, we have computed the five-loop beta function for Yang-Mills theory with fermions [4]. This confirms

7

the QCD result from [40] and is an important ingredient for future calculations. Our calculation took six days on one 32-core machine. The computation in [40] took 1.5 years on 20 machines with 8 cores.

7.2 ANSWER TO THE PROBLEM STATEMENT

Now that we have addressed the research questions, we are able to answer the problem statement.

> **Problem statement:** *In what way can we improve the performance of QFT calculations with respect to obtaining more precise predictions?*

In answering **RQ1**, we have sped up Monte Carlo integration by improving the input expressions. Since more samples can be made, the precision of the result is increased. Our code is used in pySecDec [222], and by the GRACE collaboration [223].

In answering **RQ2**, and **RQ3** we have developed methods to drastically improve calculations of physical observables at four and five loops. Especially the computation of splitting functions and the five-loop beta function, are valuable basic ingredients in many other calculations.

Based on our findings, we may conclude that we have improved the performance of QFT calculations in three different regions. Since all our methods can be applied in practice to compare theory to experiment in colliders, we may conclude that we have improved the precision of predictions.

7.3 FUTURE RESEARCH

Below we provide four areas for future research, viz. (A) expression simplification, (B) IBP reductions, (C) Mellin moment computations, and (D) Higgs decay calculations.

(A) EXPRESSION SIMPLIFICATION We have shown that applying Horner's rule and removal of common subexpressions leads to much smaller polynomials. Polynomials could be simplified even further if algebraic structures are recognised. An example is identifying squares:

$$2ab + b^2 + 2ac + 2bc + c^2 \to (a + b + c)^2 - a^2 \,. \tag{306}$$

Recognising which terms to combine into a square in order to maximally reduce the expression is difficult (especially if numerical stability has to be taken into account as well). A first option is to see if Monte Carlo Tree Search [67] can be applied to find the best way to complete the squares. The action in each state could be the selection of a monomial that should be included in the square. A challenge is to find heuristics to guide the random playout, such that fewer samples are required.

A second option is to train a neural network to identify which monomials should be used to complete a square. The input of the network could be the exponent array of the polynomial. The output layer could yield a binary value for each monomial that determines whether it is included in the square or not. One of the challenges of a neural network is to keep the number of weights down, so that the network can be trained faster. For images, convolutional neural networks are successful, since they exploit the idea that parts of images can form a pattern by themselves [224, 225]. Presumably, something similar can be realised for expressions, but it is not obvious which monomials of the expression form a substructure that is analogous to a subrectangle in an image.

(B) IBP REDUCTIONS We have shown that parametric integration-by-parts reduction rules can provide faster reductions than Laporta methods. At the moment the reduction rules require some manual intervention. If an algorithm could be devised that automatically finds high-quality reduction rules, it would mean a revolution in the field.

Currently, we are working on studying and implementing some ideas from Boolean Satisfiability problems, by defining constraints on terms that should be removed from the system. Our latest effort can reduce some hard systems, but it may require more than 500 gigabytes of disk space before a solution is found.

Additionally, it is worthwhile to study which IBP equations actually contribute to the final reduction rule. Since most equations drop out in our experience, skipping these equations from the start may save a large amount of time.

(C) MELLIN MOMENTS We have computed four-loop Mellin moments of splitting functions. A major challenge is that the complexity of the integrals scales linearly with the Mellin moment N. This makes it very time consuming to compute higher Mellin moments. The OPE method yields better scaling and may allow us to compute more Mellin moments. The hard part is that operators have to be constructed, which is especially difficult for the gluon. We expect new results soon [115].

Using the R^*-operation, splitting functions may be computed at five loops as well. One difficult point for the optical theorem method is that the harmonic projection creates many terms. If this operation could be postponed until after expensive operations in the R^*-routines, similar to the delayed Feynman rules, the computation could be performed much faster.

(D) HIGGS DECAY Using the R^*-operation and FORCER, we may be able to compute the Higgs decay to gluons, $H \to gg$, to five loops. The challenge is that the process consists of quartically divergent diagrams. As a result, the diagrams have to be Taylor expanded to the fourth order, which creates many terms and high-tensor subgraphs. One way to speed up the program is to choose a convenient infrared rearrangement (IRR) that places the line with the worst IR-divergence between the external lines. As a result, fewer subdiagrams will be created. Alternatively, we could add a mass to that line, which reduces the number of counterterms even further. Since the mass is only on one line, the massive part can be factorised out as a one-loop bubble.

We hope the computation will be completed within a few months of this writing.