Gröbner Bases in Perturbative Calculations

Vladimir P.Gerdt[a]

[a]Laboratory of Information Technologies, Joint Institute for Nuclear Research, 141980 Dubna, Russia

In this paper we outline the most general and universal algorithmic approach to reduction of loop integrals to basic integrals. The approach is based on computation of Gröbner bases for recurrence relations derived from the integration by parts method. In doing so we consider generic recurrence relations when propagators have arbitrary integer powers treated as symbolic variables (indices) for the relations.

1. Introduction

Since its invention in [1,2], the method of integration by parts has become the commonly used platform for reduction of loop integrals to basic (master) integrals. The distinguishing feature of this method is rapidly increasing volume of symbolic algebraic computation needed for manipulation with recurrence relations (RR) when the number of internal and external lines or/and loops increases, especially in the presence of such physical parameters as masses, external momenta and the space-time dimension. Over the last years a number of algorithmic ideas, approaches and software packages has been reported [3,4,5,6] to reduce, given a set of RR, loop integrals to a minimal set of master integrals.

There are two main aspects of an algorithmic approach to this problem: (i) universality, i.e. applicability to the most general form of RR and (ii) computational efficiency. Most of practically done computations are based on the approach of Laporta [4] since it deals directly with RR and provides a rather efficient reduction procedure based on Gaussian elimination. However, this elimination is restricted to RR with specified values of indices. As specification takes place, one extracts only partial information from RR. Moreover, in typical multiloop calculations it may lead to an enormous number of intermediate RR [6] that can annihilate the computational efficiency.

Another approach suggested by Tarasov [3,5] appeals to potentially much more universal algorithmic techniques based on computation of *Gröbner bases* (GB). But for all that, RR are transformed into partial differential equations. Such a transformation can only be done for internal lines endowed with different masses which are independent variables for the differential equations. Insertion of these extra variables may increase substantially the running time and storage space required for computing GB. Furthermore, if one considers indices of RR as (discrete) symbolic variables for the differential system, then in the aggregate with extra independent variables (masses) this tends to increase the volume of computation so much that becomes hardly possible for problems of interest in modern multiloop calculations. Thus, in practice, one can treat differential systems for loop integrals only with fixed numerical powers of propagators.

In this paper we consider RR generated by the integration by parts method as linear finite-difference equations for loop integrals whose coefficients can (polynomially) depend on indices and all relevant physical parameters such as masses, scalar products of external momenta and the space-time dimension. We present basic concepts of the GB technique (Sect.2) as the most universal algorithmic tool for RR with symbolic indices. GB allow one to analyze algorithmically the complete set of algebraic restrictions imposed on loop integrals by RR. Specifically, we apply (Sect.3) the GB technique to find a minimal set of basic (master) integrals which are not fixed by RR, and to reduce any other loop integral to these basic ones. We illustrate this technique by a simple example of one-loop propagator type integral taken

from [3]. We also briefly discuss computational issues of the GB approach applied to RR (Sect.4).

The concept of GB was invented by Buchberger in 1965 for systems of multivariate commutative polynomials and then extended to systems of linear partial differential equations, noncommutative polynomials and other algebraic structures (see, for example, book [7]). Due to their algorithmic universality, GB have found numerous fruitful applications. Special program modules for their computing in the case of commutative polynomials built-in all modern computer algebra systems such as Maple, Mathematica and others. In addition, Maple has special librarian functions for computing GB for differential equations, which were used in [3,5], and for noncommutative Ore algebras [8].

2. Gröbner Bases for Recurrence Relations

Since any tensor multi-loop Feynman integral by shifting space-time dimension can be reduced to scalar integrals [9], we shall, without the loss of generality, consider the following scalar integrals

$$\mathcal{I}_\nu = \int d^d k_1 \cdots d^d k_L \frac{1}{\prod_{i=1}^n P_i^{\nu_i}} \qquad (1)$$

with n internal lines. Here propagators P_i have symbolic exponents ν_i which we collect together into the *multi index* $\nu = \{\nu_1, \nu_2, \ldots, \nu_n\}$. RR derived from integration by parts form a homogeneous system of linear finite-difference equations in the integral $\mathcal{I}_\nu$ as function of its indices. Let $\nu_i - \lambda_i$ be the minimal value of the i-th index entering in the RR system. Here $\lambda_i \geq 0$ are explicitly known integers. The set of all multi indices with nonnegative components will be denoted by $N_{\geq 0}^n$, and, hence, $\lambda \in N_{\geq 0}^n$.

Let $\mu = \{\nu_1 - \lambda_1, \nu_2 - \lambda_2, \ldots, \nu_n - \lambda_n\}$ and the integral $\mathcal{I}_\mu \neq 0$ be a function of indices in ν. Then the RR are a set of equalities $f_j = 0$ including finite sums of the form

$$f_j = \sum_\alpha b_\alpha^j D^\alpha \mathcal{I}_\mu, \qquad j = 1, \ldots, p \qquad (2)$$

Here $D^\alpha = D_1^{\alpha_1} \cdots D_n^{\alpha_n}$ with $\alpha = \{\alpha_1, \ldots, \alpha_n\} \in N_{\geq 0}^n$, and each D_i is the right-shift operator for the i-th index, i.e., $D_i \mathcal{I}_\mu = \mathcal{I}_{\mu_1, \ldots, \mu_i+1, \ldots, \mu_n}$.

Coefficients b_α^j depend polynomially on indices $\{\nu_1, \ldots, \nu_n\}$ and physical parameters, e.g., masses, scalar products of external momenta, space-time dimension d. Hereafter we shall call integrals $D^\alpha \mathcal{I}_\mu$ and sums f_j in (2) by (difference) *terms* and (linear difference) *polynomials*, respectively. Coefficients b_α^j of polynomials f_j will be considered as elements in the field $Q(\nu)$ of rational functions in indices whose coefficients in turn are parametric rational functions.

Consider now the set of all linear polynomials

$$< F >= \{ \sum_\beta b_\beta \, D^\beta(f) \mid f \in F, \ b_\beta \in Q(\nu) \} \ (3)$$

generated by the polynomial set $F = \{f_1, \ldots, f_p\}$ defined in (2).

Any element $g \in < F >$ yields the finite-difference equation $g = 0$ which is a consequence of the initial RR, and $< F >$ accumulates all the consequences. The set F of difference polynomials is called *a basis* of $< F >$.

Let $\succ$ be a total order on terms $D^\alpha \mathcal{I}_\mu$ such that for any $\alpha, \beta, \gamma \in N_{\geq 0}^n$ the following holds

(i) $D^\alpha \mathcal{I}_\mu \succ \mathcal{I}_\mu \Longleftrightarrow \alpha$ contains nonzero indices,

(ii) $D^\gamma D^\alpha \mathcal{I}_\mu \succ D^\gamma D^\beta \mathcal{I}_\mu \Longleftrightarrow D^\alpha \mathcal{I}_\mu \succ D^\beta \mathcal{I}_\mu$.

Such term orders are nothing else than *admissible term orders* in commutative polynomial algebra [10], if one compares multi indices of the terms. As typical examples of admissible orders we indicate "lexicographical" order $\succ_{\text{lex}}$ with $\alpha \succ_{\text{lex}} \beta$ when $\alpha_1 > \beta_1$ or when $\alpha_1 = \beta_1$ and $\alpha_2 > \beta_2$, etc., and "total degree" order $\alpha \succ_{\text{tdeg}} \beta$ when $\sum \alpha_i > \sum \beta_i$ or $\sum \alpha_i = \sum \beta_i$ and $\alpha \succ_{\text{lex}} \beta$.

Given a term order $\succ$, one can extract the *leading term* $\text{lt}(f)$ from every difference polynomial f. Its coefficient is called *the leading coefficient* and will be denoted by $\text{lc}(f)$. Now we can define GB for (3) as a finite polynomial set G satisfying

1. $< G >=< F >$.
2. For every $f \in < F >$ there is $g \in G$ and $\gamma \in N_{\geq 0}^n$ such that $\text{lt}(f) = D^\gamma(\text{lt}(g))$.

Note that conditions (i) and (ii) of admissibility for $\succ$ provide existence of GB and termination of an algorithm for its construction exactly as they do in commutative polynomial algebra [10,11]. If

the term order $\succ$ is considered as a certain "simplicity relation" between integrals, then condition (i) means that integral $\mathcal{I}_\mu$ is "simpler" than that obtained from $\mathcal{I}_\mu$ by shifting up some indices. Condition (ii) shows that such a simplicity relation between two integrals is stable under identical shifting of their corresponding indices.

For two multi indices α and β, α is called *a divisor* of β and β is called *a multiple* of α, if $\beta - \alpha \in N_{\geq 0}^n$. The *least common multiple* of α and β will be denoted by $\mathrm{lcm}(\alpha, \beta)$. Thus, property 2 means that the leading multi index of any consequence of RR as element in $< F >$ has a divisor among multi indices of the leading terms in GB. Property 1 implies that both sets of RR defined by F and G are fully equivalent since any consequence of one set is a consequence of another set and vice versa. A term whose multi index divides (is multiple of) another term is said to be a divisor (multiple) of this term.

An important notion of the GB techniques is *reduction*. Polynomial f is said to be *reducible modulo polynomial* g if f has a term u with a coefficient $b \in Q(\mu)$ such that u is multiple of $\mathrm{lt}(g)$, i.e., $u = D^\gamma \mathrm{lt}(g)$ for some $\gamma \in N_{\geq 0}^n$. In this case *an elementary reduction step* is given by

$$f \xrightarrow{g} f^\star = \frac{1}{b} f - D^\gamma \left(\frac{1}{\mathrm{lc}(g)} g \right) . \qquad (4)$$

If $f^\star$ in (4) is still reducible modulo g, then the second reduction step can be performed, etc., until an irreducible polynomial obtained. By property (i) of term order $\succ$, the number of elementary reductions is finite [11]. Similarly, one can *reduce a polynomial f modulo a finite polynomial set F* by doing elementary reductions of f modulo individual polynomials in F. The reduction process is terminated with polynomial $\bar{f}$ irreducible modulo F. In this case we say that $\bar{f}$ is in the *normal form modulo F* and write $\bar{f} = NF(f, F)$. The above described procedure gives an algorithm for computing the normal form.

From the above definition of GB G for set F, it follows that $NF(f, G) = 0$ for any polynomial f in $< F >$. Therefore, if G is known, one can algorithmically detect the dependency of an extra recurrence relation r on set F. It suffices to verify whether $NF(r, G)$ vanishes or not.

It is remarkable, that there is an algorithm for construction of GB for any finite set of linear difference polynomials and an admissible term order. This algorithm was discovered by Buchberger in 1965 for commutative algebraic polynomials. Below we present the Buchberger algorithm in its simplest form [10,11] adapted to linear finite-difference polynomials. First, we define the *S-polynomial* $S(f, g)$ for a pair of (nonzero) difference polynomials f and g as follows. Let $lt(f)$ and $lt(g)$ have multi indices α and β, respectively, and let $\gamma = \mathrm{lcm}(\alpha, \beta)$. Then

$$S(f, g) = D^{\gamma - \alpha} \left(\frac{1}{\mathrm{lc}(f)} f \right) - D^{\gamma - \beta} \left(\frac{1}{\mathrm{lc}(g)} g \right) .$$

Buchberger algorithm:

Start with $G := F$.
For a pair of polynomials $f_1, f_2 \in G$:
    Compute $S(f_1, f_2)$.
    Compute $h := NF(S(f_1, f_2), G)$.
    If $h = 0$, consider the next pair.
    If $h \neq 0$, add h to G and iterate.

Inter-reduction of the output GB, that is, reduction of every polynomial $g \in G$ modulo $G \setminus \{g\}$, gives a *reduced* GB. By normalizing (dividing by) the leading coefficients, we obtain the *monic* reduced GB that is uniquely defined [10,11] by the input polynomial set F and term order $\succ$.

3. Generic Master Integrals

Having computed GB for the set (3) generated by the initial set (2), one can algorithmically, just as in commutative algebra [11], find the maximal set of difference monomials (integrals) which are independent modulo set (3). In other words, there is no nonzero polynomial in (3) composed from those monomials, and any other monomial (integral) can be reduced to (expressed by means of) these monomials. Thus, the maximal independent set of monomials (integrals) is exactly the collection of basic integrals. They may be called *generic master integrals* to emphasize their relevance to RR with symbolic indices. The reduction of any integral to "masters" can also be done algorithmically (Sect.2) by means of GB.

By keeping track of all intermediate elementary reductions (4), one can obtain an explicit expres-

sion of any loop integral in terms of (generic) master integrals and their multiples. As we defined in Sect.3, a multiple of the integral is obtained by applying D^γ with $\gamma \in N_{\geq 0}^n$. It is clear, however, that $D^\gamma \mathcal{I}_\mu$ can be evaluated only if the dependence of $\mathcal{I}_\mu$ on indices is explicitly known.

But how to obtain the set of master integrals? It can be easily detected from the leading monomials of GB. Namely, all those monomials that are not multiple of any leading monomial in GB are independent (masters). This fact is very well known in theory of commutative Gröbner bases [11] and apparently applicable to linear finite-difference polynomials. Clearly, any multiple of a leading term in GB is reducible to (expressible in terms of) a linear combination of monomials that are not multiple of any leading term in the GB and with coefficients from $Q(\nu)$.

Though the particular form of master integrals depends on the choice of term order $\succ$, their number is invariant on $\succ$ and fully defined by initial RR. In fact, if one accepts the interpretation of $\succ$ as simplicity relation (Sect.2), the set of master integrals detected from GB contains the "simplest" integrals[1]. Usually a "total degree" order is preferable (cf. [4,6]) from the computational point of view since total degree GB are typically computed much faster then lexicographical ones.

For pure illustrative purposes consider RR for the one-loop propagator type integral $\mathcal{I}_{\nu_1 \nu_2}$ taken from paper [3] and put $m^2 = q^2 = 0$ in those relations, thus, keeping space-time dimension d as the only parameter

$$\nu_1 \mathcal{I}_{\nu_1 - 1\, \nu_2 + 1} - (d - \nu_1 - 2\nu_2)\mathcal{I}_{\nu_1 \nu_2} = 0,$$
$$\nu_1 \mathcal{I}_{\nu_1 + 1\, \nu_2 - 1} - \nu_2 \mathcal{I}_{\nu_1 - 1\, \nu_2 + 1} + (\nu_2 - \nu_1)\mathcal{I}_{\nu_1 \nu_2} = 0.$$

Their form (2) for $\mathcal{I}_{\nu_1 - 1, \nu_2 - 1} \neq 0$ is given by

$$[\nu_1 D_2^2 - (d - \nu_1 - 2\nu_2)]\,\mathcal{I}_{\nu_1 - 1, \nu_2 - 1} = 0,$$
$$[\nu_1 D_1^2 - \nu_2 D_2^2 + (\nu_2 - \nu_1)]\,\mathcal{I}_{\nu_1 - 1\, \nu_2 - 1} = 0,$$

where D_1, D_2 are the right-shift operators for indices ν_1 and ν_2, respectively. A total degree Gröbner basis at $D_1 \succ D_2$ calculated by Maple [2]

---

[1]For numerically fixed values of indices and particular integrals sometimes one uses other simplicity relations which do not satisfy (i) and (ii) [6].

[2]Algebra of shift operators is a particular case of the Ore algebra [8], and one can use it for computing GB for RR as already mentioned in [5]. However, this way of compu-

gives the following GB form of RR

$$[(d - \nu_1 - 2\nu_2)D_1 D_2 - \nu_1 D_2^2]\,\mathcal{I}_{\nu_1 - 1\, \nu_2 - 1} = 0,$$
$$[\nu_1(d - \nu_1 - 2\nu_2)D_1^2 +$$
$$(\nu_1(2\nu_2 - \nu_1) + \nu_2(2\nu_2 - d))D_2^2]\,\mathcal{I}_{\nu_1 - 1\, \nu_2 - 1} = 0,$$
$$D_2^3\,\mathcal{I}_{\nu_1 - 1\, \nu_2 - 1} = 0.$$

An immediate consequence of this GB is the equality $\mathcal{I}_{\nu_1 - 1\, \nu_2 + 2} = 0$ that follows from the last element of GB. This implies $D^\gamma \mathcal{I}_{\nu_1 - 1\, \nu_2 + 2} = 0$ for $\gamma \in N_{\geq 0}^n$. Similarly, the order $D_2 \succ D_1$ gives $\mathcal{I}_{\nu_1 + 2\, \nu_2 - 1} = 0$ and $D^\gamma \mathcal{I}_{\nu_1 + 1\, \nu_2 - 1} = 0$ for $\gamma \in N_{\geq 0}^n$ that can also be verified by calculating the normal form $NF(D_2^3 \mathcal{I}_{\nu_1 - 1\, \nu_2 - 1}, GB) = 0$.

The leading terms in the above GB are represented by the operators $D_1 D_2$, D_1^2, D_2^3 which generate four master integrals $\mathcal{I}_{\nu_1 - 1\, \nu_2 - 1}$, $\mathcal{I}_{\nu_1 \nu_2 - 1}$, $\mathcal{I}_{\nu_1 - 1\, \nu_2}$, $\mathcal{I}_{\nu_1 - 1\, \nu_2 + 1}$.

4. Computational Aspects

First of all, it should be emphasized that the above analysis is based only on RR themselves without any use of the "integral structure" of $\mathcal{I}_\mu$. If there are extra restrictions which follow from this structure, they should be incorporated into RR. For example, if a loop integral admits symmetry under some permutation of indices, one has to use RR in the properly symmetrized form. Thus, in example of Sect.4 $\mathcal{I}_{\nu_1 \nu_2} = \mathcal{I}_{\nu_2 \nu_1}$ [3], and if one adds RR with permuted indices, then GB becomes $D_1^2 \mathcal{I}_{\nu_1 - 1\, \nu_2 - 1}$, $D_1 D_2 \mathcal{I}_{\nu_1 - 1\, \nu_2 - 1}$, $D_2^2 \mathcal{I}_{\nu_1 - 1\, \nu_2 - 1}$.

It should be also noted that the GB technique is applicable to RR with specified indices. Provided with an appropriate ordering on the integrals (priority criteria [4,6]), the GB algorithm is just Gaussian elimination [11]. Hence, it is conceptually identical to the Laporta algorithm.

In the above illustrative and very simple example (Sect.3) we performed the GB calculation over the field. Generally, however, one has to be careful in manipulation with arbitrary indices occurring in the coefficients of difference polynomials. In calculating GB over a coefficient field, the leading coefficients of difference polynomials are treated generically, and division by them can

---

tation is very expensive and can be applied only to small problems (Sect.4).

be done at the intermediate steps of calculation. GB computed this way may loose its GB properties under some specifications of parameters. This happens when, in the course of reduction, the division by a leading coefficient was done, and this coefficient vanishes under the specification. To avoid this trouble, one can compute GB over a coefficient ring rather than over a field (e.g. without division). But computation over the ring may lead to growth of intermediate coefficients and make the computation much more tedious.

As it is well-known (see [11] and references therein), the running time and storage space required by the GB algorithms the GB tend to be exponential and superexponential in number of variables (indices of RR). Besides, the presence of parameters and symbolic indices in coefficients increases substantially the volume of computation. Therefore, to be practically applicable to modern multiloop calculations, one needs efficient implementation of the GB algorithms provided by special tuning to finite-difference polynomials of type (2).

To our knowledge, there are no GB packages available for multivariate finite-difference polynomials, except those designed for the (noncommutative) skew-polynomial algebras or Ore algebras [8]. Our experimentation with the Maple package implementing the Buchberger algorithm for skew polynomials, revealed its very low efficiency for RR with parameters. Thus, Maple on a 2 Ghz PC with 512 Mb RAM was not able to construct (in reasonable time) GB for the full (but still rather small) one-loop example from [3] when both m^2 and q^2 are nonzero and present in RR. To our opinion, this is because of extra computational costs caused by internal noncommutative settings of underlying built-in operations.

We find more perspective to adapt our involutive algorithms and software packages [12,13] to linear multivariate finite-difference polynomials. The involutive algorithms also compute GB, but they are based on the other course of intermediate computation different from that in the Buchberger algorithm. Our implementation of involutive algorithms already shown their higher efficiency [13] for conventional commutative polynomials. In addition, involutive algorithms admit

effective and natural parallelism, that may be crucial to manage with multiloop calculations needed in modern higher energy physics.

5. Acknowledgements

REFERENCES

1. K.G.Chetyrkin and F.V.Tkachov, Nucl.Phys. **B 192** (1981) 159.
2. F.V.Tkachov, Phys.Lett. **B 100** (1981) 65.
3. O.V.Tarasov, Nucl.Phys. **B502** (1997) 6479.
4. S.Laporta, Int.J.Mod.Phys. A **15** (2000) 5087.
5. O.V.Tarasov, arXiv:hep-ph/0403253.
6. C.Anastasiou and A.Lazopoulos, arXiv:hep-ph/0404258.
7. B.Buchberger and F.Winkler (eds.) Gröbner Bases and Applications, Cambridge University Press, 1998.
8. F.Chizak, *Gröbner Bases, Symbolic Summation and Symbolic Integration*, in [7], pp.32.
9. O.V.Tarasov. Phys.Rev. **D54** (1996) 6479.
10. B.Buchberger, *Introduction to Gröbner Bases*, in [7], pp.3.
11. D.Cox,J.Little and O'Shea. Ideals, Varieties and Algorithms, Springer-Verlag, New York, 1992.
12. V.P.Gerdt and Yu.A.Blinkov, MATCOM **45** (1998) 519; ibid., 540.
13. V.P.Gerdt, Yu.A.Blinkov and D.A.Yanovich, *Construction of Janet bases. II. Polynomial bases*, in: Computer Algebra in Scientific Computing / CASC'01, Springer-Verlag Berlin (2001) 249.