Computer Physics Communications 183 (2012) 2158–2164

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

\$Apart: A generalized MATHEMATICA Apart function ☆

Feng Feng

Center for High Energy Physics, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 20 February 2012
Accepted 30 March 2012
Available online 25 April 2012

Keywords:
CPC classification: 11.1

ABSTRACT

We have generalized the MATHEMATICA function `Apart` from 1 to N dimensions: the generalized function `$Apart` can decompose any linear dependent elements in $\mathcal{V}_x^*$ to irreducible ones. The elements in $\mathcal{V}_x^*$ can be viewed as the corresponding propagators which involve loop momenta, and the decomposition will be useful when one tries to perform loop calculations using packages such as FIRE and REDUZE, which have implemented the integration by parts (IBP) identities and Lorentz invariance (LI) identities. A description on how to use this package, combined with FIRE, FEYNARTS, and FEYNCALC packages, to do one-loop calculations in double quarkonium production in e^+e^- colliders is given, and the full source code for a specific process ($e^+e^- \to J/\psi + \eta_c$) is also available.

Program summary

Program title: \$Apart
Catalogue identifier: AEMK_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEMK_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence,
http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 451006
No. of bytes in distributed program, including test data, etc.: 4598053
Distribution format: tar.gz
Programming language: Mathematica.
Computer: Any computer where Mathematica is running.
Operating system: Any capable of running Mathematica.
Classification: 11.1.
External routines: FeynCalc, FeynArts, Fire (all included in the distribution file).
Nature of problem:
The traditional method of computing cross sections for a physical process in perturbative quantum field theory involves generating the amplitudes via Feynman diagrams and performing the dimensionally regularized loop integrals [1]. Simplifications of the expressions are performed at the analytical level; there, an essential part is the reduction of these loop integrals to a small number of standard integrals. This step can be performed at the amplitude level for tensor integrals or, after contraction of Lorentz indices, at the level of interferences for scalar integrals. Considering the case of scalar integrals, integration by parts (IBP) identities [2,3] and Lorentz invariance (LI) identities [4] may be used for a systematic reduction to a set of independent integrals, called master integrals (MI). The standard reduction algorithm by Laporta [5] defines an ordering for Feynman integrals, generates identities, and solves the resulting system of linear equations. Alternative methods to exploit IBP and LI identities for reductions have been proposed [6–9]; see also [10,11] and references therein. Public implementations of different reduction algorithms are available with the computer programs AIR [12], FIRE [13], and Reduze [14]. The usage of Fire [13], Reduze [14], etc. requires that the propagators must be decomposed to independent ones: for one dimension, there is a Mathematica function **Apart** to do this, while for N dimensions there is no such package yet, so we want to generalize the Mathematica function **Apart** to \$**Apart** in N dimensions.
Solution method:
We first prove that any linear dependent elements in $\mathcal{V}_x^*$ can be decomposed into the summation of linear independent ones: the procedure used in the proof gives us a method to perform the decomposition.

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

E-mail addresses: fengf@ihep.ac.cn, hepmail@163.com, physicsff@yahoo.com.

0010-4655/\$ – see front matter
doi:10.1016/j.cpc.2012.03.025

$Apart is such an Mathematica package that implements this method and generalizes the Mathematica **Apart** function from 1 to N dimensions.
Running time: Depends on the complexity of the system.

1. Introduction and notation

A polynomial is a mathematical expression involving a sum of powers in one or more variables multiplied by coefficients. A polynomial in one variable (i.e., a univariate polynomial) with constant coefficients is given by

$$a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0. \tag{1}$$

The highest power in a univariate polynomial is called its order, or sometimes its degree, and for a polynomial with more than one variable, i.e. a multivariate polynomial, one needs to find the degree of each term by adding the exponents of each variable in the term, and the largest such degree is the degree of the multivariate polynomial. For a rational expression, i.e. an expression that is the ratio of two polynomials, we can work out its degree by taking the degree of the top (numerator) and subtracting the degree of the bottom (denominator), for example,

$$\begin{aligned}\mathrm{Deg}\left[\frac{x^3+4x+9}{x^5+2x+1}\right] &\equiv \mathrm{Deg}\left[x^3+4x+9\right] - \mathrm{Deg}\left[x^5+2x+1\right] \\ &= -2. \end{aligned} \tag{2}$$

It is well known that there is a function in MATHEMATICA named `Apart` which rewrites a rational expression of a univariate polynomial as a sum of terms with minimal denominators and gives the partial fraction decomposition of the rational expression, for example,

$$\mathtt{Apart}\left[\frac{1}{(x-a)(x-b)}\right] = \frac{1}{(b-a)x-b} + \frac{1}{(a-b)(x-a)}, \tag{3a}$$

$$\begin{aligned}&\mathtt{Apart}\left[\frac{1}{(x-a)(x-b)(x-c)}\right] \\ &= -\frac{1}{(a-b)(b-c)(x-b)} + \frac{1}{(a-c)(b-c)(x-c)} \\ &\quad + \frac{1}{(a-b)(a-c)(x-a)}. \end{aligned} \tag{3b}$$

There is no such similar function for bivariate polynomials yet; for example, `Apart` does not change the following form at all:

$$\mathtt{Apart}\left[\frac{1}{x(x+a)(x+y+b)}\right] = \frac{1}{x(x+a)(x+y+b)}. \tag{4}$$

We want to generalize this function to some specific rational expressions of n-variate polynomials; for example, we expect

$$\begin{aligned}&\mathtt{Apart}\left[\frac{1}{x(x+a)(x+y+b)}, \{x, y\}\right] \\ &= \frac{1}{ax(x+y+b)} - \frac{1}{a(x+a)(x+y+b)}. \end{aligned} \tag{5}$$

First we introduce the notation. We consider the n-variate polynomial $\mathcal{V}_x^*$ with degree less than or equal to 1. The n variables are denoted as $\{x_i\}_{1\le i\le n}$, and the linear space which is spanned by n independent vectors $\{x_i\}$ over the coefficient field $\mathcal{F}$ is denoted as $\mathcal{V}_x$. It is clear that

$$\mathcal{V}_x^* = \mathcal{V}_x \oplus \mathcal{F}. \tag{6}$$

We call the k elements $\{e_i = v_i + f_i\}_{(1\le i\le k)} \subset \mathcal{V}_x^*$ linear independent if and only if their projective parts in $\mathcal{V}_x$, i.e. $\{v_i\}_{(1\le i\le k)} \subset \mathcal{V}_x$, are linear independent.

Now we consider the following special terms generated from the rational operations on the polynomial $\mathcal{V}_x^*$:

$$\prod_{i=1}^{N} e_i^{n_i}, \quad e_i \in \mathcal{V}_x^* \wedge n_i \in \mathbb{Z} \wedge N \ge 1, \tag{7}$$

where $\mathbb{Z}$ is the integer set, and $\{e_i\}_{1\le i\le N}$ are generally not linear independent, i.e. reducible. We want to decompose them into a summation of linear independent, i.e. irreducible, ones:

$$\prod_{i=1}^{N} e_i^{n_i} = \sum_j f_j \prod_{i=1}^{N_j} e_{k_{ji}}^{n_{ji}}, \tag{8}$$

where $1 \le k_{ji} \le N$, $1 \le N_j \le N$ and, for any fixed j, the N_j elements $\left\{e_{k_{ji}}\right\}_{1\le i\le N_j}$ are linear independent, i.e. some elements in $\{e_i\}_{1\le i\le N}$ have been eliminated such that the remaining elements become linear independent.

To give a proof of the decomposition in Eq. (8), let us consider a special case:

$$F(n_1, \ldots, n_N) \equiv \prod_{i=1}^{N} e_i^{n_i}, \tag{9}$$

where any $(N-1)$ elements from $\{e_i\}_{1\le i\le N}$ are linear independent, but the N elements $\{e_i\}_{1\le i\le N}$ are not, so there exists $\{f_i\}_{1\le i\le N}$, with all $f_i \ne 0$, such that

$$\sum_{i=1}^{N} f_i e_i = f. \tag{10}$$

Note that the linear independency in $\mathcal{V}_x^*$ is up to some constant f in the field $\mathcal{F}$.

We first look at Eq. (9) with all $n_i = -1$, i.e.,

$$F(-1, \ldots, -1) = \frac{1}{e_1}\frac{1}{e_2}\cdots\frac{1}{e_{N-1}}\frac{1}{e_N}. \tag{11}$$

- If $f \ne 0$, we can write Eq. (11) as

$$\begin{aligned}\prod_{i=1}^{N}\frac{1}{e_i} &= \frac{f}{f}\prod_{i=1}^{N}\frac{1}{e_i} = \frac{1}{f}\left(\sum_{j=1}^{N} f_j e_j\right)\prod_{i=1}^{N}\frac{1}{e_i} \\ &= \sum_{j=1}^{N}\frac{f_j}{f}\prod_{i=1, i\ne j}^{N}\frac{1}{e_i}; \end{aligned} \tag{12}$$

  since any $(N-1)$ elements from $\{e_i\}_{1\le i\le N}$ are linear independent, the final expression is irreducible, and is the desired decomposition.
- If $f = 0$, since all $f_i \ne 0$, without loss of generality, we take f_1 as an example:

$$\begin{aligned}\prod_{i=1}^{N}\frac{1}{e_i} &= \frac{e_1}{e_1}\prod_{i=1}^{N}\frac{1}{e_i} = \frac{1}{e_1}\left(-\frac{1}{f_1}\sum_{j=2}^{N} f_j e_j\right)\prod_{i=1}^{N}\frac{1}{e_i} \\ &= -\sum_{j=2}^{N}\frac{f_j}{f_1}\frac{1}{e_1^2}\prod_{i=2, i\ne j}^{N}\frac{1}{e_i}. \end{aligned} \tag{13}$$

We know that any $(N-2)$ elements from $\{e_i\}_{2\le i\le N}$ combined with e_1 are linear independent, so the final expression is also irreducible.

So we get the decomposition for Eq. (11). Now, considering the expression of Eq. (9) with all exponents of $\{e_i\}$ negative,

$$G(n_1,\ldots,n_N) \equiv F(-n_1,\ldots,-n_N) = \prod_{i=1}^{N} e_i^{-n_i} = \prod_{i=1}^{N} \frac{1}{e_i^{n_i}}, \quad n_i > 0, \tag{14}$$

we can factorize out a term $\prod_{i=0}^{N}\frac{1}{e_i}$ and perform the decomposition on it as follows.

- For the case when $f \neq 0$, we have

$$\prod_{i=1}^{N}\frac{1}{e_i^{n_i}} = \prod_{k=1}^{N}\frac{1}{e_k^{n_k-1}}\prod_{i=1}^{N}\frac{1}{e_i} = \prod_{k=1}^{N}\frac{1}{e_k^{n_k-1}}\left(\sum_{j=1}^{N}\frac{f_j}{f}\prod_{i=1,i\neq j}^{N}\frac{1}{e_i}\right) = \sum_{j=1}^{N}\frac{f_j}{f}\frac{1}{e_j^{n_j-1}}\prod_{k=1,k\neq j}^{N}\frac{1}{e_k^{n_k}}, \tag{15}$$

where it is clear that we have decomposed the original term into N terms, and furthermore that these terms have the same form as the original one except that one of the exponents n_i decreases by 1 in each term, i.e., we get the following recursive relation:

$$G(n_1,\ldots,n_N) = \sum_{j=1}^{N}\frac{f_j}{f}G\left(n_1,\ldots,n_j-1,\ldots,n_N\right), \tag{16}$$

and we can repeat the decomposition until one of the n_i decreases to 0.

- When $f = 0$,

$$\prod_{i=1}^{N}\frac{1}{e_i^{n_i}} = \prod_{k=1}^{N}\frac{1}{e_k^{n_k-1}}\prod_{i=1}^{N}\frac{1}{e_i} = \prod_{k=1}^{N}\frac{1}{e_k^{n_k-1}}\left(-\sum_{j=2}^{N}\frac{f_j}{f_1}\frac{1}{e_1^2}\prod_{i=2,i\neq j}^{N}\frac{1}{e_i}\right) = -\sum_{j=2}^{N}\frac{f_j}{f_1 e_1}\frac{1}{e_j^{n_j-1}}\prod_{k=1,i\neq j}^{N}\frac{1}{e_k^{n_k}}, \tag{17}$$

and this is similar to the case $f \neq 0$: the terms after decomposition have the same form as the original one, and one of the exponents n_i decreases by 1 in each term except e_1 whose exponent will increase by 1; the recursive relation is

$$G(n_1,\ldots,n_N) = -\sum_{i=2}^{N}\frac{f_i}{f_1}G\left(n_1+1,n_2,\ldots,n_i-1,\ldots,n_N\right), \tag{18}$$

and we can repeat the decomposition until one of the n_i $(i \geq 2)$ decreases to 0.

So, in each one of the two cases above, $G(n_1,\ldots,n_N)$ can be reduced to the summation of $G(n_1,n_2,\ldots,n_i=0,\ldots,n_N)$, which cannot be decomposed any more, i.e. it is irreducible, and is the desired result.

If at least one exponent $n_{i_0} > 0$ in Eq. (9), without loss of generality, taking $i_0 = 1$ and $n_1 > 0$, then the element e_1 can be written as

$$e_1 = \frac{1}{f_1}\left(f - \sum_{i=2}^{N} f_i e_i\right); \tag{19}$$

we can substitute Eq. (19) into Eq. (9) to eliminate e_1:

$$\prod_{i=1}^{N} e_i^{n_i} = \left[\frac{1}{f_1}\left(f - \sum_{j=2}^{N} f_j e_j\right)\right]^{n_1}\prod_{i=2}^{N} e_i^{n_i} = \sum_k f'_k \prod_{i=2}^{N} e_i^{n_{ki}}, \tag{20}$$

and now the final expression only involves $\{e_i\}_{(2\le i\le n)}$ which are linear independent, and it is irreducible.

To complete the proof, we will make the induction on N, i.e. the number of elements in $\{e_i\}$. It is trivial that this is valid for $N = 1$, and now, assuming that it is also valid for $N = 1, 2, 3, \ldots, K$, we want to prove that it is also valid for $N = K + 1$.

If $\{e_i\}_{1\le i\le K+1}$ are linear independent, i.e. irreducible, then there is no need for the decomposition; otherwise, there will be $M + 1$ $(M \le K)$ elements from $\{e_i\}$ which are not linear independent, but any M elements are linear independent. Without loss of generality, we take these elements as $\{e_i\}_{1\le i\le M+1}$:

$$\prod_{i=1}^{K+1} e_i^{n_i} = \prod_{i=1}^{M+1} e_i^{n_i}\prod_{j=M+2}^{K+1} e_j^{n_j}. \tag{21}$$

Then, according to the special case we have considered in Eq. (9), we have

$$\prod_{i=1}^{M+1} e_i^{n_i} = \sum_j f_j \prod_{i=1}^{N_j} e_{k_{ji}}^{n_{ji}}, \tag{22}$$

where all $N_j \le M$, so

$$\prod_{i=1}^{K+1} e_i^{n_i} = \left(\sum_j f_j \prod_{i=1}^{N_j} e_{k_{ji}}^{n_{ji}}\right)\prod_{m=M+2}^{K+1} e_m^{n_m} = \sum_j f_j\left(\prod_{i=1}^{N_j} e_{k_{ji}}^{n_{ji}}\prod_{m=M+2}^{K+1} e_m^{n_m}\right). \tag{23}$$

Since $N_j + (K - M) \le K$, i.e. the number of elements in $\{e_{k_{ji}}\}_{1\le i\le N_j} \cup \{e_m\}_{M+2\le m\le K+1}$ in the right-hand side (rhs) is less than $N = K + 1$, according to the assumptions, we have the following decomposition:

$$\prod_{i=1}^{N_j} e_{k_{ji}}^{n_{ji}}\prod_{m=M+2}^{K+1} e_m^{n_m} = \sum_k f'_k \prod_{i=1}^{N'_k} e_{k'_{ki}}^{n'_{ki}}, \tag{24}$$

with each term in the rhs irreducible, so we get the decomposition for $N = K + 1$:

$$\prod_{i=1}^{K+1} e_i^{n_i} = \sum_{k,j} f_j f'_k \prod_{i=1}^{N'_k} e_{k'_{ki}}^{n'_{ki}}. \tag{25}$$

Since each term in the rhs is irreducible, the proof is done.

The procedure also gives us a method to perform the decomposition. We will give an implementation in MATHEMATICA, i.e. the generalized `Apart` function: `$Apart`.

2. An implementation in MATHEMATICA

The basic functions in the package are:

- `$Apart[expr, {x, y, z, ...}]`

expr can be any form in Eq. (7), {x, y, z, ...} are the corresponding n-variate polynomial variables, and $Apart[expr, {x, y, z, ...}] will perform the decomposition on expr to give the irreducible forms, which are expressed with the function $ApartIR.

- $ApartIR[expr, {x, y, z, ...}, {$\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{N}$}, {$\mathrm{n}_1, \mathrm{n}_2, \ldots, \mathrm{n}_\mathrm{N}$}],
where expr is actually the product of $\mathrm{e}_i^{\mathrm{n}_i}$, i.e.

$$\mathtt{expr} = \prod_{i=1}^{\mathrm{N}} \mathrm{e}_i^{\mathrm{n}_i}. \tag{26}$$

We preserve the $\{\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{N}\}$ and $\{\mathrm{n}_1, \mathrm{n}_2, \ldots, \mathrm{n}_\mathrm{N}\}$ for later use, because they will be used as the input parameters for FIRE [13]. The irreducible form will be displayed as $\|\cdots\|$, and $RemoveApart can be used to remove | in $\|\cdots\|$.

- $RemoveApart[expr]
$RemoveApart is used to remove the HEAD in $Apart or $ApartIR, and is defined as

$RemoveApart[expr_]
:= expr/.${Apart[x_, _] :> x, $ApartIR[x_, ___] :> x}.

We can take Eq. (5) as the first concrete example:

$$\mathtt{\$Apart}\left[\frac{1}{x(x+a)(x+y+b)}, \{\mathrm{x}, \mathrm{y}\}\right]$$
$$\Rightarrow \frac{\left\|\frac{1}{x(b+x+y)}\right\| - \left\|\frac{1}{(a+x)(b+x+y)}\right\|}{a}. \tag{27}$$

As another relatively complicated case, we take

$$\mathtt{expr} = \frac{a}{(3a+b+c)(a+2b+d)^3(a+4b+9e)}. \tag{28}$$

If we take a and b as the only polynomial variables, there are only two elements which can appear in the same irreducible expression, and we get the output as

$Apart[expr, {a, b}]

$$\begin{aligned}
\Rightarrow & -\frac{(2c-d)\left\|\frac{1}{(3a+b+c)(a+2b+d)^3}\right\|}{2c-11d+45e} \\
& + \frac{5(4c-9e)\left\|\frac{1}{(3a+b+c)(a+2b+d)^2}\right\|}{(2c-11d+45e)^2} \\
& - \frac{55(4c-9e)\left\|\frac{1}{(3a+b+c)(a+2b+d)}\right\|}{(2c-11d+45e)^3} \\
& + \frac{121(4c-9e)\left\|\frac{1}{(3a+b+c)(a+4b+9e)}\right\|}{(2c-11d+45e)^3} \\
& + \frac{2(2d-9e)\left\|\frac{1}{(a+2b+d)^3(a+4b+9e)}\right\|}{-2c+11d-45e} \\
& + \frac{2(4c-9e)\left\|\frac{1}{(a+2b+d)^2(a+4b+9e)}\right\|}{(2c-11d+45e)^2} \\
& - \frac{22(4c-9e)\left\|\frac{1}{(a+2b+d)(a+4b+9e)}\right\|}{(2c-11d+45e)^3}.
\end{aligned} \tag{29}$$

We can check the output with the original Eq. (28) using the code

dexpr = $Apart[expr, {a, b}]
comp = expr − (dexpr//$RemoveApart)//Simplify. (30)

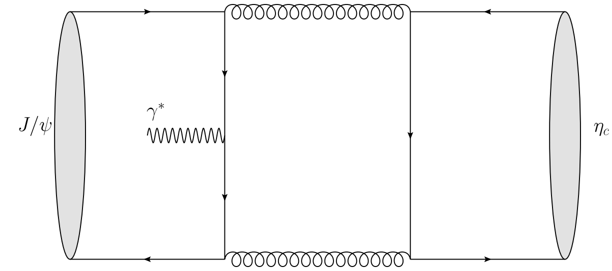

Fig. 1. A pentagon Feynman diagram for $e^+e^- \to \gamma^* \to J/\psi + \eta_c$.

The fact that comp gives zero indicates that the output dexpr is indeed identical with the original expr.

If take c as a polynomial variable as well, we have

$Apart[expr, {a, b, c}]

$$\begin{aligned}
\Rightarrow & -\left\|\frac{1}{(3a+b+c)(a+2b+d)^3}\right\| \\
& +(9e-2d)\left\|\frac{1}{(3a+b+c)(a+2b+d)^3(a+4b+9e)}\right\| \\
& +2\left\|\frac{1}{(3a+b+c)(a+2b+d)^2(a+4b+9e)}\right\|.
\end{aligned} \tag{31}$$

More complicated examples can be found in Example/Examples.nb in the source code.

Now, let us look at how to apply $Apart to a specific Feynman diagram from the process $e^+e^- \to \gamma^* \to J/\psi + \eta_c$, which is shown in the Fig. 1. After projecting the spin singlet and triplet for the charmonium η_c and J/ψ respectively with the spin projectors [15] and performing the DiracTrace on the fermion chains, we get the amplitude for this diagram as Eq. (32), given in Box I, where p_3 and p_4 are the momenta of J/ψ and η_c, respectively, γ and ψ represent the polarizations of γ^* and J/ψ, respectively, k is the loop momentum, m_c is the mass of the charm quark, and s is defined by

$$s \equiv \frac{Q^2}{4m_c^2} = \frac{(p_3+p_4)^2}{4m_c^2}. \tag{33}$$

If we take k^2, $k \cdot p_3$ and $k \cdot p_4$ as the 3-variate polynomial variables, then the amplitude $\mathcal{A}$ has the same form as Eq. (7) after expanding the numerator, so we can perform the $Apart operation on it:

$$\mathtt{\$Apart}[\mathcal{A}, \{k^2, k\cdot p_3, k\cdot p_4\}]. \tag{34}$$

The output is Eq. (35), given in Box II.

We can see that the 5-point integrals have been reduced to 3-point or 2-point integrals, so the scalar integrals have been greatly simplified. Details of this example can be found in Process/FC-43.nb in the source code, where 43 is the sequence number of the corresponding diagram which has been generated by FEYNARTS [16,17].

3. Application to physical loop calculations

The traditional method to compute cross sections for a physical process in perturbative quantum field theory involves generating the amplitudes via Feynman diagrams and performing the dimensionally regularized loop integrals [1]. Simplifications of the expressions are performed at the analytical level; there, an essential part is the reduction of these loop integrals to a small number of standard integrals. This step can be performed at the amplitude level for tensor integrals or, after contraction of Lorentz indices, at the level of interferences for scalar integrals.

$$\mathcal{A} = -\frac{16iC_AC_Feg_s^4\epsilon^{\gamma\psi p_3p_4}\left((s-4)sk^2m_c^2+k\cdot p_3^2+k\cdot p_4^2-(s-2)k\cdot p_3k\cdot p_4\right)}{3(D-2)m_c(s-4)sk^2\left(k^2+k\cdot p_3\right)\left(4m_c^2+k^2-2k\cdot p_4\right)\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)\left(k^2-k\cdot p_4\right)} \tag{32}$$

Box I.

$$\begin{aligned}
&-\frac{4iC_AC_Feg_s^4\epsilon^{\gamma\psi p_3p_4}}{3(D-2)m_c^3(s-4)(s-2)s}\left[-4\left\|\frac{1}{k^2\left(k^2+k\cdot p_3\right)\left(4m_c^2+k^2-2k\cdot p_4\right)}\right\|m_c^2\right.\\
&+2s\left\|\frac{1}{k^2\left(k^2+k\cdot p_3\right)\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)}\right\|m_c^2-4\left\|\frac{1}{k^2\left(4m_c^2+k^2-2k\cdot p_4\right)\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)}\right\|m_c^2\\
&+2s\left\|\frac{1}{\left(k^2+k\cdot p_3\right)\left(4m_c^2+k^2-2k\cdot p_4\right)\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)}\right\|m_c^2\\
&+2(s-2)s\left\|\frac{1}{\left(k^2+k\cdot p_3\right)\left(4m_c^2+k^2-2k\cdot p_4\right)\left(k^2-k\cdot p_4\right)}\right\|m_c^2\\
&+2(s-2)s\left\|\frac{1}{k^2\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)\left(k^2-k\cdot p_4\right)}\right\|m_c^2\\
&+2(s-2)\left\|\frac{1}{k^2\left(4m_c^2+k^2-2k\cdot p_4\right)}\right\|+4(s-2)\left\|\frac{1}{\left(k^2+k\cdot p_3\right)\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)}\right\|+(2-s)\left\|\frac{1}{k^2\left(k^2-k\cdot p_4\right)}\right\|\\
&+(4-2s)\left\|\frac{1}{\left(k^2+k\cdot p_3\right)\left(k^2-k\cdot p_4\right)}\right\|+(2-s)\left\|\frac{1}{\left(4m_c^2+k^2-2k\cdot p_4\right)\left(k^2-k\cdot p_4\right)}\right\|\\
&\left.+(4-2s)\left\|\frac{1}{\left(2sm_c^2+k^2-k\cdot p_3-2k\cdot p_4\right)\left(k^2-k\cdot p_4\right)}\right\|\right].
\end{aligned} \tag{35}$$

Box II.

Considering the case of scalar integrals, integration by parts (IBP) identities [2,3] and Lorentz invariance (LI) identities [4] may be used for a systematic reduction to a set of independent integrals, called master integrals (MI). The standard reduction algorithm by Laporta [5] defines an ordering for Feynman integrals, generates identities, and solves the resulting system of linear equations. Alternative methods to exploit IBP and LI identities for reductions have been proposed [6–9]; see also [10,11] and references therein. Public implementations of different reduction algorithms are available with the computer programs AIR [12], FIRE [13], and REDUZE [14].

As for the one-loop calculations, there are many automatic tools available to achieve the general one-loop amplitude, such as FEYNCALC [18] and FORMCALC [19], which are based on the traditional Passarino–Veltman [20–23] reduction of Feynman graphs, which can be generated automatically (FeynArts [16,17] or QGRAF [24]). In order to produce numerical results, tensor coefficient functions are calculated using LOOPTOOLS [19]. See also Refs. [25,26] and the references therein.

In the last few years, several groups have been working on the problem of constructing efficient and automatized methods for the computation of one-loop corrections for multi-particle processes. Many different interesting techniques have been proposed: these contain numerical and semi-numerical methods [27–30], as well as analytic approaches [31–34] that make use of unitarity cuts to build next-to-leading-order amplitudes by gluing on-shell tree amplitudes [35,36]. For a recent review of existing methods, see Refs. [37,38].

In this section, we want to use the `$Apart` and the FIRE [13] package combined with FEYNARTS [16,17] and FEYNCALC [19] to perform one-loop calculations; here, FEYNARTS [16,17] and FEYNCALC [19] are used to generate the Feynman diagrams, and to perform the DiracTrace respectively, and the rest, such as tensor or scalar integral reductions, will be handled by the `$Apart` and FIRE [13] packages. We will concentrate on the next-to-leading-order corrections in α_s to double quarkonium production in e^+e^- colliders. The basic procedure can be summarized as follows.

1. Use the FEYNARTS [16,17] package to generate all Feynman diagrams for the partonic process $e^+e^- \to \gamma^* \to c\bar{c}+c\bar{c}$.
2. Use FEYNCALC [19] to perform the `DiracTrace` and `SU(N)` color matrix trace.
3. Make an expansion in the relative momentum of the quark and the anti-quark in the corresponding quarkonium and project out `S−`, `P−`, `D−`, ... Waves.
4. Use `$Apart` to decompose the linear dependent propagators to independent ones.
5. Use the FIRE [13] package to reduce the general loop integrals to master integrals (MI).
6. Process the final results, e.g. to asymptotically expand the amplitudes or calculate the cross section.

Taking the process $e^+e^- \to J/\psi + \eta_c$ as an example, for which there are large discrepancies between the NRQCD leading-order predictions and experimental data, the important key step to resolve the discrepancy is that a large K factor of about 1.96 has been found in the next-to-leading-order corrections in α_s [39,40].

The calculations at leading order can be found in the directory `Process/Tree`, and the next-to-leading-order calculations can be separated into several parts.

- `Process/Tree/FC-RN.nb` is used to calculate corrections from the counter-terms where multiple renormalization is used.
- `Process/FC.nb` is used to calculate the general loop corrections.

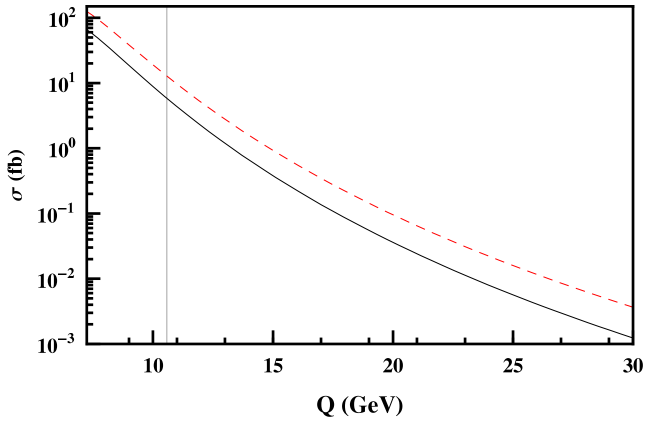

Fig. 2. Cross section for $e^+e^- \rightarrow J/\psi + \eta_c$ as a function of the center-of-mass energy. The renormalization scale μ is set at half of the center-of-mass energy and $m_c = 1.5$ GeV; the vertical line corresponds to $Q = 10.6$ GeV, the solid line to σ_{LO}, and the dashed line to σ_{NLO}.

Table 1
Cross sections with different charm quark mass m_c and renormalization scale μ; the input parameters are the same as in Ref. [40], and $Q = 10.6$ GeV.

m_c (GeV)	μ	α_s (μ)	σ_{LO}(fb)	σ_{NLO}(fb)	$\sigma_{\mathrm{NLO}}/\sigma_{\mathrm{LO}}$
1.5	m_c	0.369	16.09	27.51	1.710
1.5	$2m_c$	0.259	7.94	15.68	1.975
1.5	$Q/2$	0.211	5.27	11.14	2.113
1.4	m_c	0.386	19.28	34.92	1.811
1.4	$2m_c$	0.267	9.19	18.84	2.050
1.4	$Q/2$	0.211	5.76	12.61	2.189

- `Process/FC-Nf.nb` is used to calculate the corrections from the light quarks which are proportional to $(N_f - 1)$.
- `Process/Total.nb` will process the results generated from the above to give the numerical predictions or plots.

To compare with the results which are already present in other references, let us list some results which can be found in `Process/Total.nb`.

The asymptotically expanded amplitude at $s \gg 1$ is

$$\mathcal{A} = \mathcal{A}^{(0)} + \frac{\alpha_s}{\pi}\mathcal{A}^{(1)} + \mathcal{O}\left(\alpha_s^2\right),$$

$$\frac{\mathcal{A}^{(1)}}{\mathcal{A}^{(0)}} = \frac{1}{72}\left[39\ln^2 s - 9(3 + 10\ln 2)\ln s + 300\ln\frac{\mu}{m_c} + 3(195 - 53\ln 2)\ln 2 - 2\pi^2 - 92\right] + \frac{i\pi}{24}(-26\ln s + 30\ln 2 + 9) + \mathcal{O}\left(\frac{1}{s}\right),$$

$$s = \frac{Q^2}{4m_c^2}. \tag{36}$$

This result agrees with Eq. (6.4) of Ref. [41], in which only the real part of the asymptotic expansion is given, while our result also includes the imaginary part.

We use the same input parameters as in Ref. [40] to give the numerical results which are shown in Table 1 and Fig. 2; these are consistent with Refs. [39,40].

The method can be also used for calculations involving P-waves [42].

4. Summary

We have introduced a generalized MATHEMATICA `Apart` function, which will perform the decomposition on any linear dependent elements in $\mathcal{V}_x^*$ to reduce them to the irreducible form. The elements in $\mathcal{V}_x^*$ can be viewed as the corresponding propagators which involve loop momenta, and the decomposition will be useful when one tries to perform loop calculations using packages such as FIRE and REDUZE, which have implemented the integration by parts (IBP) identities and Lorentz invariance (LI) identities. A description of how to use this package, combined with FIRE, FEYNARTS, and FEYNCALC packages, to do one-loop calculations in double quarkonium production in e^+e^- colliders is given, and the full source code for a specific process ($e^+e^- \rightarrow J/\psi + \eta_c$), from generating Feynman diagrams to the asymptotic expansion of amplitudes and numerical results, is also available.

Acknowledgments

The author thanks Hai-Rong Dong, Wen-Long Sang, and Prof. Yu Jia for many useful discussions. The research was partially supported by the China Postdoctoral Science Foundation. Finally, I would like to commemorate my beloved mother.

References

[1] G. 't Hooft, M.J.G. Veltman, Regularization and renormalization of gauge fields, Nuclear Phys. B 44 (1972) 189.
[2] F.V. Tkachov, A theorem on analytical calculability of four loop renormalization group functions, Phys. Lett. B 100 (1981) 65.
[3] K.G. Chetyrkin, F.V. Tkachov, Integration by parts: the algorithm to calculate beta functions in 4 loops, Nuclear Phys. B 192 (1981) 159.
[4] T. Gehrmann, E. Remiddi, Differential equations for two-loop four-point functions, Nuclear Phys. B 580 (2000) 485. arXiv:hep-ph/9912329.
[5] S. Laporta, High-precision calculation of multi-loop Feynman integrals by difference equations, Internat. J. Modern Phys. A 15 (2000) 5087. arXiv:hep-ph/0102033.
[6] A.V. Smirnov, V.A. Smirnov, Applying Gröbner bases to solve reduction problems for Feynman integrals, J. High Energy Phys. 0601 (2006) 001. arXiv:hep-lat/0509187.
[7] A.V. Smirnov, An algorithm to construct Gröbner bases for solving integration by parts relations, J. High Energy Phys. 0604 (2006) 026. arXiv:hep-ph/0602078.
[8] J. Gluza, K. Kajda, D.A. Kosower, Towards a basis for planar two-loop integrals, Phys. Rev. D 83 (2011) 045012. arXiv:1009.0472 [hep-th].
[9] R.M. Schabinger, A new algorithm for the generation of unitarity-compatible integration by parts relations, arXiv:1111.4220 [hep-ph].
[10] R.N. Lee, Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals, J. High Energy Phys. 0807 (2008) 031. arXiv:0804.3008 [hep-ph].
[11] A.G. Grozin, Integration by parts: an introduction, Internat. J. Modern Phys. A 26 (2011) 2807. arXiv:1104.3993 [hep-ph].
[12] C. Anastasiou, A. Lazopoulos, Automatic integral reduction for higher order perturbative calculations, J. High Energy Phys. 0407 (2004) 046. arXiv:hep-ph/0404258.
[13] A.V. Smirnov, Algorithm FIRE—Feynman integral reduction, J. High Energy Phys. 0810 (2008) 107. arXiv:0807.3243 [hep-ph].
[14] C. Studerus, Reduze—Feynman integral reduction in C + +, Comput. Phys. Commun. 181 (2010) 1293. arXiv:0912.2546 [physics.comp-ph].
[15] E. Braaten, J. Lee, Exclusive double charmonium production from e^+e^- annihilation into a virtual photon, Phys. Rev. D 67 (2003) 054007;
E. Braaten, J. Lee, Exclusive double charmonium production from e^+e^- annihilation into a virtual photon, Phys. Rev. D 72 (2005) 099901. (erratum) hep-ph/0211085.
[16] J. Kublbeck, M. Bohm, A. Denner, FeynArts: computer algebraic generation of Feynman graphs and amplitudes, Comput. Phys. Commun. 60 (1990) 165.
[17] T. Hahn, Generating Feynman diagrams and amplitudes with FeynArts 3, Comput. Phys. Commun. 140 (2001) 418. hep-ph/0012260.
[18] R. Mertig, M. Bohm, A. Denner, FEYNCALC: computer algebraic calculation of Feynman amplitudes, Comput. Phys. Commun. 64 (1991) 345.
[19] T. Hahn, M. Perez-Victoria, Automatized one loop calculations in four-dimensions and D-dimensions, Comput. Phys. Commun. 118 (1999) 153. hep-ph/9807565.
[20] G. Passarino, M.J.G. Veltman, One loop corrections for e^+e^- annihilation into $\mu^+\mu^-$ in the Weinberg model, Nuclear Phys. B 160 (1979) 151.
[21] A. Denner, Techniques for calculation of electroweak radiative corrections at the one loop level and results for W physics at LEP-200, Fortschr. Phys. 41 (1993) 307. arXiv:0709.1075 [hep-ph].
[22] A. Denner, S. Dittmaier, Reduction of one-loop tensor 5-point integrals, Nuclear Phys. B 658 (2003) 175. arXiv:hep-ph/0212259.

[23] A. Denner, S. Dittmaier, Reduction schemes for one-loop tensor integrals, Nuclear Phys. B 734 (2006) 62. arXiv:hep-ph/0509141.
[24] P. Nogueira, Automatic Feynman graph generation, J. Comput. Phys. 105 (1993) 279.
[25] Z. Bern, L.J. Dixon, D.A. Kosower, Progress in one loop QCD computations, Annu. Rev. Nucl. Part. Sci. 46 (1996) 109. hep-ph/9602280.
[26] A. van Hameren, C.G. Papadopoulos, R. Pittau, Automated one-loop calculations: a proof of concept, J. High Energy Phys. 0909 (2009) 106. arXiv:0903.4665 [hep-ph].
[27] W.T. Giele, E.W.N. Glover, A calculational formalism for one loop integrals, J. High Energy Phys. 0404 (2004) 029. hep-ph/0402152.
[28] R.K. Ellis, W.T. Giele, G. Zanderighi, Semi-numerical evaluation of one-loop corrections, Phys. Rev. D 73 (2006) 014027. hep-ph/0508308.
[29] D. Forde, Direct extraction of one-loop integral coefficients, Phys. Rev. D 75 (2007) 125019. arXiv:0704.1835 [hep-ph].
[30] R.K. Ellis, W.T. Giele, Z. Kunszt, A numerical unitarity formalism for evaluating one-loop amplitudes, J. High Energy Phys. 0803 (2008) 003. arXiv:0708.2398 [hep-ph].
[31] Z. Bern, L.J. Dixon, D.C. Dunbar, D.A. Kosower, Fusing gauge theory tree amplitudes into loop amplitudes, Nuclear Phys. B 435 (1995) 59. hep-ph/9409265.
[32] Z. Bern, L.J. Dixon, D.C. Dunbar, D.A. Kosower, One loop n-point gauge theory amplitudes, unitarity and collinear limits, Nuclear Phys. B 425 (1994) 217. hep-ph/9403226.
[33] F. Cachazo, P. Svrcek, E. Witten, MHV vertices and tree amplitudes in gauge theory, J. High Energy Phys. 0409 (2004) 006. hep-th/0403047.
[34] F. Cachazo, P. Svrcek, E. Witten, Twistor space structure of one-loop amplitudes in gauge theory, J. High Energy Phys. 0410 (2004) 074. hep-th/0406177.
[35] R. Britto, B. Feng, P. Mastrolia, The cut-constructible part of QCD amplitudes, Phys. Rev. D 73 (2006) 105004. hep-ph/0602178.
[36] P. Mastrolia, On triple-cut of scattering amplitudes, Phys. Lett. B 644 (2007) 272. hep-th/0611091.
[37] S. Weinzierl, Automated calculations for multi-leg processes, PoS ACAT (2007) 005. arXiv:0707.3342 [hep-ph].
[38] Z. Bern, L.J. Dixon, D.A. Kosower, On-shell methods in perturbative QCD, Ann. Physics 322 (2007) 1587. arXiv:0704.2798 [hep-ph].
[39] Y.-J. Zhang, Y.-J. Gao, K.-T. Chao, Next-to-leading order QCD correction to $e^+e^- \to J/\psi + \eta_c$ at $\sqrt{s} = 10.6\text{GeV}$, Phys. Rev. Lett. 96 (2006) 092001. hep-ph/0506076.
[40] B. Gong, J.X. Wang, QCD corrections to $J/\psi + \eta_c$ production in e^+e^- annihilation at $\sqrt{s} = 10.6\text{GeV}$, Phys. Rev. D 77 (2008) 054028. arXiv:0712.4220 [hep-ph].
[41] Y. Jia, J.-X. Wang, D. Yang, Bridging light-cone and NRQCD approaches: asymptotic behavior of B_c electromagnetic form factor, J. High Energy Phys. 1110 (2011) 105. arXiv:1012.6007 [hep-ph].
[42] H.-R. Dong, F. Feng, Y. Jia, $O(\alpha_s)$ corrections to $J/\psi + \chi_{cJ}$ production at B factories, J. High Energy Phys. 1110 (2011) 141. arXiv:1107.4351 [hep-ph].