

OPEN ACCESS

## LiteRed 1.4: a powerful tool for reduction of multiloop integrals

To cite this article: Roman N Lee 2014 *J. Phys.: Conf. Ser.* **523** 012059

View the [article online](#) for updates and enhancements.

### You may also like

- [Lattice theta constants vs Riemann theta constants and NSR superstring measures](#)  
P. Dunin-Barkowski, A. Morozov and A. Sleptsov
- [Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals](#)  
R.N. Lee
- [Multiloop superstring amplitudes from non-minimal pure spinor formalism](#)  
Nathan Berkovits and Nikita Nekrasov



**ECS**  
The  
Electrochemical  
Society  
Advancing solid state &  
electrochemical science & technology

**DISCOVER**  
how sustainability  
intersects with  
electrochemistry & solid  
state science research

# LiteRed 1.4: a powerful tool for reduction of multiloop integrals

**Roman N. Lee**

Budker Institute of Nuclear Physics, 630090, Novosibirsk, Russia

E-mail: R.N.Lee@inp.nsk.su

**Abstract.** We review the *Mathematica* package LiteRed, version 1.4.

## 1. Introduction

Multiloop integrals are the building blocks of many calculations of radiative corrections in QFT. One of the key approaches to the calculation of the multiloop integrals is the IBP reduction, based on the integration-by-parts identities between integrals [1, 2]. The IBP reduction almost necessarily should be done with the aid of various computer programs. One of the most successful methods of the IBP reduction is the Laporta algorithm [3]. The algorithm is easy to implement and to use, and allows for a number of programming improvements. These advantages explain why many modern most powerful reduction programs heavily rely on this algorithm, in particular, AIR [4], FIRE [5, 6], Reduze [7, 8], and many private versions.

However, the Laporta algorithm has some weak points, which may put restrictions on its application. In particular, being intrinsically a brute-force search, this algorithm is both time- and memory-consuming. Another approach to the reduction is a derivation of the symbolic reduction rules. Its advantages are obvious: nothing is being solved in the process of reduction, therefore, the reduction is very fast. Symbolic rules are small in size, so, they can be easily saved for future calculations. The bottleneck of this approach is the search of these symbolic rules. Much effort has been devoted to the development of the approach connected with the notion of the Groebner basis [9, 10, 11, 12], but, for now, this approach is far from being satisfactory. Probably, the only, partly successful, attempt to implement this approach has been made in FIRE, where the notion of s-bases [11, 12] has been used.

Recently, the new IBP reduction package LiteRed has been presented in Ref. [13]. This package uses completely different approach to the reduction. At first stage it tries to find symbolic reduction rules using heuristics. Then it applies the rules to the specific reduction. Found rules are very lightweight and can be easily stored for the reuse. In this contribution we describe the version 1.4 of the LiteRed package. The package can be downloaded from <http://www.inp.nsk.su/~lee/programs/LiteRed/>.

## 2. Multiloop integral

To fix notation, let us assume that we are interested in the calculation of the  $L$ -loop integral depending on the  $E$  external momenta  $p_1, \dots, p_E$ . There are  $N = L(L+1)/2 + LE$  scalar



products depending on the loop momenta  $l_i$ :

$$s_{ij} = l_i \cdot q_j, \quad 1 \leq i \leq L, j \leq L + E, \quad (1)$$

where  $q_1, \dots, L = l_1, \dots, L$ ,  $q_{L+1}, \dots, L+E = p_1, \dots, E$ .

The general form of the integral is the following

$$J(\mathbf{n}) = J(n_1, n_2, \dots, n_N) = \int \frac{d^d l_1 \dots d^d l_L}{D_1^{n_1} D_2^{n_2} \dots D_N^{n_N}}, \quad (2)$$

$$D_\alpha = a_\alpha^{ij} l_i \cdot l_j + 2b_\alpha^{ik} l_i \cdot p_k + c_\alpha.$$

Here  $a_\alpha$ ,  $b_\alpha$ , and  $c_\alpha$  are  $L \times L$  matrices,  $L \times E$  matrices, and numbers, respectively. As usual, we assume that  $D_1, \dots, D_N$  form a complete basis in the sense that any  $s_{ik}$  can be uniquely expressed in terms of  $D_\alpha$ . The multiindex  $\mathbf{n} = (n_1, \dots, n_N)$  can be thought of as a point in  $\mathbb{Z}^N$ . Some of  $D_\alpha$  correspond to the denominators of the propagators, others may correspond to the irreducible numerators. E.g., the  $K$ -legged  $L$ -loop diagram with generic external momenta corresponds to  $E = K - 1$  and the maximal number of denominators is  $M = E + 3L - 2$ , so that the rest  $N - M = (L - 1)(L + 2E - 4)/2$  functions correspond to irreducible numerators.

### 2.1. Differential equations

The differential equations can be used for finding the master integrals. The simplest type of such equations is the differential equation with respect to the mass. Probably, the first example of their application is presented in Refs. [14, 15, 16]. The differential equations with respect to the invariant constructed of the external momenta have been introduced and applied in Refs. [17, 18, 19]. In general case, when there are  $E > 2$  external vectors, we have the following formulas:

$$\begin{aligned} \frac{\partial}{\partial(p_1 \cdot p_2)} J(\mathbf{n}) &= \sum [\mathbb{G}^{-1}]_{i2} p_i \cdot \partial_{p_1} J(\mathbf{n}) = \sum [\mathbb{G}^{-1}]_{i1} p_i \cdot \partial_{p_2} J(\mathbf{n}), \\ \frac{\partial}{\partial(p_1^2)} J(\mathbf{n}) &= \frac{1}{2} \sum [\mathbb{G}^{-1}]_{i1} p_i \cdot \partial_{p_1} J(\mathbf{n}). \end{aligned} \quad (3)$$

where  $\mathbb{G} = \mathbb{G}(p_1, \dots, p_E) = \begin{pmatrix} p_1^2 & \dots & p_1 \cdot p_E \\ \vdots & \ddots & \vdots \\ p_1 \cdot p_E & \dots & p_E^2 \end{pmatrix}$  is a Gram matrix.

Acting by the operator on the right-hand side on the integrand and performing the IBP reduction, one obtains the differential equation for  $J(\mathbf{n})$ .

### 2.2. Dimensional recurrences

Probably, the first appearance of the dimension shifting relations is in Ref. [20], where they have been derived for certain three-loop integrals in the momentum representation. Later, in Ref. [21] Tarasov derived dimensional relations using the parametric representation. It is interesting that the first approach led to the lowering recurrence, while the latter one led to the raising recurrence. The lowering (raising) recurrence relates one integral in  $d + 2$  ( $d - 2$ ) dimensions to several integrals in  $d$  dimensions.

As it was shown in Ref. [22], the two recurrences can be represented as

$$J^{(d-2)}(\mathbf{n}) = (\mu/2)^L \det \left\{ 2^{\delta_{ij}} \frac{\partial D_k}{\partial s_{ij}} A_k |_{i,j=1 \dots L} \right\} J^{(d)}(\mathbf{n}). \quad (4)$$

$$J^{(d+2)}(\mathbf{n}) = \frac{(2\mu)^L [V(p_1, \dots, p_E)]^{-1}}{(d-E-L+1)_L} P(B_1, \dots, B_N) J^{(d)}(\mathbf{n}), \quad (5)$$

where  $\mu = \pm 1$  for Euclidean/Minkovskian metric,  $\alpha_L = \alpha(\alpha+1)\dots(\alpha+L-1)$  is the Pochhammer symbol,  $V(v_1, \dots, v_k) = \det \mathbb{G}(v_1, \dots, v_k)$  is the Gram determinant, and  $P(D_1, \dots, D_N) = V(q_1, \dots, q_{L+E})$ .

The operators  $A_\alpha$  and  $B_\alpha$  are defined as follows

$$\begin{aligned} (A_i J^{(\mathcal{D})})(n_1, \dots, n_N) &= n_i J^{(\mathcal{D})}(n_1, \dots, n_i + 1, \dots, n_N), \\ (B_i J^{(\mathcal{D})})(n_1, \dots, n_N) &= J^{(\mathcal{D})}(n_1, \dots, n_i - 1, \dots, n_N). \end{aligned} \quad (6)$$

### 3. Parametric representation

The parametric representation (or Feynman parametrization), is, of no doubt, one of the most useful tools for the multiloop calculations. It is important for both analytical (in particular, with subsequent Mellin-Barnes representation) and numerical (in particular, together with sector decomposition) calculations of the multiloop integrals. Moreover, it may serve as a fundamental definition of the multiloop integrals for the case of non-integer dimensionality  $d$ .

But it is also important that the parametric representation is very useful for revealing relations between the integrals. One example is Tarasov's derivation of the raising dimensional recurrence relation [21]. In Ref. [23] the algorithm, based on the use of parametric representation, for the identification of the master integrals has been introduced. In recent paper [24] it was shown that parametric representation allows for a simple determination of the number of master integrals in the given sector.

LiteRed uses parametric representation for two purposes. First, it finds equivalent simple sectors by comparing their parametric representation. To account for the possible permutation of the parameters, it uses an approach combining the ideas from Ref. [23] and Ref. [25]. Second, it uses parametric representation to determine zero sectors. In this section we describe shortly this new algorithm.

The parametric representation of the integral  $J(\mathbf{n})$  has the form

$$J(\mathbf{n}) = \frac{\Gamma(\Sigma n - Ld/2)}{\prod_\alpha \Gamma(n_\alpha)} \int \prod_\alpha dz_\alpha z_\alpha^{n_\alpha-1} \delta(1 - \Sigma z) \frac{F^{Ld/2-\Sigma n}}{U^{(L+1)d/2-\Sigma n}}, \quad (7)$$

where  $\Sigma n = \sum_\alpha n_\alpha$ ,  $\Sigma z = \sum_\alpha z_\alpha$ ,  $U$  and  $F$  are the homogeneous polynomials of degrees  $L$  and  $L+1$ , respectively. These polynomials can be expressed in terms of quantities

$$a^{ij} = \sum_\alpha z_\alpha a_\alpha^{ij}, \quad b^i = \sum_\alpha z_\alpha b_\alpha^{ij} p_j, \quad c = \sum_\alpha z_\alpha c_\alpha \quad (8)$$

as follows

$$U = \det(a), \quad F = c \det(a) - (a^{\text{Adj}})^{ij} b^i \cdot b^j, \quad (9)$$

where  $a^{\text{Adj}} = \det(a) a^{-1}$  is the adjoint matrix. The representation (7) does not make sense when some of  $n_\alpha$  are nonpositive integers. In this case one has to replace the corresponding integration with the derivative at zero point. The resulting formula can be written as

$$J(\mathbf{n}) = \Gamma(\Sigma n - Ld/2) \prod_\alpha \hat{n}_\alpha \delta(1 - \Sigma_+ z) \frac{F^{Ld/2-\Sigma n}}{U^{(L+1)d/2-\Sigma n}}, \quad (10)$$

where the functional  $\hat{n}_\alpha$  is determined as

$$\hat{n}_\alpha[\phi(z_\alpha)] = \begin{cases} \int_0^\infty \frac{dz_\alpha z_\alpha^{n_\alpha-1}}{\Gamma(n_\alpha)} \phi(z_\alpha) & n_\alpha > 0 \\ (-1)^{n_\alpha} \phi^{(-n_\alpha)}(0) & n_\alpha \leq 0 \end{cases} \quad (11)$$

and the sum  $\Sigma_+ z = \sum_\alpha \theta_\alpha z_\alpha$  ( $\theta_\alpha = \Theta(n_\alpha - 1/2)$ ) goes over the variables, corresponding to the denominators. Remarkably, it is possible to rewrite Eq. (10) in the form, which contains  $U$  and  $F$  only in the combination  $F + U$ . Similar to Ref. [24], we have

$$J(\mathbf{n}) = \frac{\Gamma[d/2]}{\Gamma[(L+1)d/2 - \Sigma n]} \prod_\alpha \hat{n}_\alpha G^{-d/2}, \quad G = F + U. \quad (12)$$

The scaleless integral can be defined as the one which gains additional non-unity factor under some linear transformation of the loop momenta. In dimensional regularization scaleless integrals are set to zero. If  $j(\theta_1, \dots, \theta_N)$  is scaleless, then all integrals of the sector  $(\theta_1, \dots, \theta_N)$  are zero. We will call such a sector a *zero sector*.

A simple criterion of zero sectors has been formulated in Ref. [26]. According to this criterion, the sector is zero if the solution of the IBP equations in the corner point  $(\theta_1, \dots, \theta_N)$  result in the identity  $j(\theta_1, \dots, \theta_N) = 0$ . Note that this criterion may miss some scaleless sectors. Let us explain on a simple example why this happens. Consider the massless one-loop onshell propagator integral

$$J(n_1, n_2) = \int \frac{d^d l}{[l^2]^{n_1} [(l-k)^2]^{n_2}}, \quad k^2 = 0.$$

Obviously, this integral is zero for any  $n_1$  and  $n_2$ . However, it can be explicitly checked that the solution of the IBP identities in the corner point of the sector  $(1, 1)$  does not result directly to  $J(1, 1) = 0$ . In order to prove that the integral  $J(1, 1)$  is scaleless, let us consider instead the following operator

$$O = \partial_l \cdot \left( l + (l \cdot k) \tilde{k} - (l \cdot \tilde{k}) k \right),$$

where  $\tilde{k}$  is an auxiliary vector chosen to satisfy the conditions  $\tilde{k}^2 = 0$  and  $\tilde{k} \cdot k = 1$ . It is easy to check that  $Oj(1, 1) = (d-4)j(1, 1)$ . Since the operator  $O$  is a generator of the linear transformation  $l \rightarrow l + \epsilon \left( l + (l \cdot k) \tilde{k} - (l \cdot \tilde{k}) k \right)$ , the integral  $j(1, 1)$  is scaleless. The reason why the IBP identities failed to lead to the identity  $J(1, 1) = 0$  is that the construction of this identity required introduction of the auxiliary vector  $\tilde{k}$ .

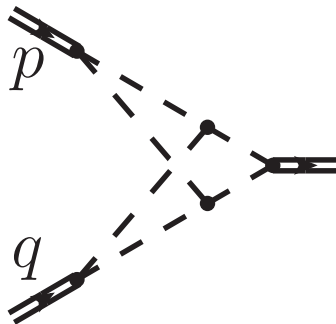
In some cases the number of zero sectors overlooked by the criterion of Ref. [26] is rather big. So, we formulate below another criterion, based on parametric representation, which detects virtually all zero sectors.

As we said above, for the detection of zero sectors it is sufficient to consider only the integral in the corner point of the sector. In particular, we may set all  $z_\beta$ , corresponding to numerators, to zero. In what follows we assume this is done and  $z_\alpha$  denotes a parameter, corresponding to the denominator of the sector. Consider an infinitesimal scaling of these parameters

$$z_\alpha \rightarrow \tilde{z}_\alpha = (1 + k_\alpha \omega) z_\alpha. \quad (13)$$

Here  $\omega$  is the infinitesimal parameter, and  $k_\alpha$  are some finite coefficients. Suppose that we are able to find such  $k_\alpha$  that the function  $G$  scales as follows:

$$G(\tilde{z}) = (1 + \omega) G(z). \quad (14)$$



**Figure 1.** Two-loop massless onshell vertex.

Then, making the change (13) in Eq. (12), we get

$$J(\boldsymbol{\theta}) = \left[ 1 + \omega \left( \sum_{\alpha} k_{\alpha} - d/2 \right) \right] J(\boldsymbol{\theta}). \quad (15)$$

The equation (14) does not depend on  $d$ , therefore, suitable  $k_{\alpha}$ , if they exist at all, can be chosen also independent of  $d$ . Therefore, the coefficient in Eq. (15) in front of  $\omega$  is not zero, and the integral  $J(\boldsymbol{\theta})$  is scaleless.

Note that Eq. (14) can be cast as

$$\sum_{\alpha} k_{\alpha} z_{\alpha} \frac{\partial G(z)}{\partial z_{\alpha}} = G(z), \quad (16)$$

which should be understood as equality of the two polynomials of  $z_{\alpha}$ . Collecting the coefficients in front of distinct monomials, we obtain a linear system of equation with respect to  $k_{\alpha}$ . The existence of the solution of this system can be established by ordinary algebraic means.

Therefore, we get the following

**Criterion of zero sector:** For a given sector, construct  $G = F + U$ . The sector is zero if Eq.(16) has a  $z$ -independent solution with respect to  $k_{\alpha}$ .

It is just this criterion which is implemented in LiteRed1.4.

#### 4. How LiteRed finds reduction rules

When trying to find the complete set of the reduction rules, LiteRed performs, roughly speaking, the same steps as a person would do. Let us explain this on the example of the two-loop massless onshell vertex shown in Fig. 1. We choose the basis

$$\{D_1, \dots, D_7\} = \{l^2, r^2, (p-l)^2, (q-r)^2, (-l+p+r)^2, (l+q-r)^2, (l-r)^2\}. \quad (17)$$

The function  $D_7 = (l-r)^2$  corresponds to the irreducible numerator. The diagram in Fig. 1 corresponds to the sector  $(1, 1, 1, 1, 1, 1, 0)$  (see the definition of sectors in Ref. [13]).

- First, LiteRed solves the IBP and LI identities in the general point  $\mathbf{n} = (n_1, n_2, n_3, n_4, n_5, n_6, n_7)$  with respect to the most complex integrals. Then it shifts the indices in the rules found so that they all have the form  $J(n_1, n_2, n_3, n_4, n_5, n_6, n_7) \rightarrow \dots$
- After that it analyzes the right-hand side to determine when each rule is applicable. The inapplicability may come from zeros in the denominators, or from the positive shifts in the indices, corresponding to the numerator, e.g., if the right-hand side contains the

integral  $J(n_1, n_2, n_3, n_4, n_5, n_6, n_7 + 1)$ . Note that it often happens that the latter integral appears with the coefficient  $n_7$  and then does not result in the applicability condition  $n_7 \neq 0$ . Acting in this way, LiteRed finds 9 rules with the following applicability conditions:  $\{c_1, \dots, c_7\} = \{n_6 \neq 1, n_5 \neq 1, \neg(n_3 = 1 \vee n_7 = 0), \neg(n_4 = 1 \vee n_7 = 0), n_6 \neq 1, \neg(n_1 = 1 \vee n_7 = 0), \neg(n_6 = 1 \vee n_7 = 0), \neg(n_2 = 1 \vee n_7 = 0), \neg(n_5 = 1 \vee n_7 = 0)\}$ .

- None of the found rules is applicable when the condition  $\neg(c_1 \vee \dots \vee c_7)$  is fulfilled. LiteRed reduces this condition to disjunctive normal form:  $(n_5 = 1 \wedge n_6 = 1 \wedge n_7 = 0) \vee (n_1 = 1 \wedge n_2 = 1 \wedge n_3 = 1 \wedge n_4 = 1 \wedge n_5 = 1 \wedge n_6 = 1)$ .
- Then it takes the first alternative  $(n_5 = 1 \wedge n_6 = 1 \wedge n_7 = 0)$  and tries to find the rules for the integral  $J(n_1, n_2, n_3, n_4, 1, 1, 0)$ . It starts from the IBP and LI identities at the point  $(n_1, n_2, n_3, n_4, 1, 1, 0)$  and checks for the possibility to shift indices in order to reduce the found rules to the form  $J(n_1, n_2, n_3, n_4, 1, 1, 0) \rightarrow \dots$ . In contrast to all-indeterminate case, this is not always possible, so if the appropriate rule is not found, LiteRed starts to generate and solve identities in the points neighboring  $(n_1, n_2, n_3, n_4, 1, 1, 0)$ . In fact, this search is the Laporta algorithm augmented by the procedure which checks for possible shifts.
- When the appropriate rule is found, the condition of its applicability is constructed. In our case, this condition looks like  $n_5 = 1 \wedge n_6 = 1 \wedge n_7 = 0 \wedge n_4 \neq 1$ .
- Performing the same steps, LiteRed finds 17 rules which reduce all integrals in the sector except  $J(1, 1, 1, 1, 1, 1, 0)$  which it declares a master.

Note that LiteRed successfully finds the reduction rules for much more complicated cases than the one presented above. In particular, it succeeds for the four-loop massless propagators and some other complicated cases.

## 5. Short reference guide for LiteRed package

The package is loaded by the command `<<LiteRed``. Let us describe briefly the most important procedures of LiteRed.

**Declare**[*vars*, *type*] — variable declaration. Here *vars* — either variable name or the list of variable names, *type* is either **Vector** or **Number**. Be sure to declare all variables that enter the basis definition (see below). Both vectors and numbers can be declared in one construct.

*Example:* **Declare**[{**1**, **q**}, **Vector**, **mm**, **Number**].

**NewBasis**[*name*, {*D1*, *D2*, ...}, *loopmoms*, *options*] — definition of the basis. Here *name* is the variable which will be the basis name. It will appear in many commands, associated with a given basis, like **GenerateIBP**[*name*]. Derived objects, like IBP identities, reduction rules, will be associated with this name. Make sure this name is not used anywhere else, but in the appropriate commands. {*D1*, *D2*, ...} is a list of functions  $D_\alpha$ . The scalar product is entered as **sp**[**p1**, **p2**]. Third argument *loopmoms* is a list of loop momenta. The following options can be appended:

- **Directory**->"*dirname*" — determine the directory, where all basis definitions will be saved.
- **Append**->**True** — if the set of the functions  $D_\alpha$ , given in the first parameter, is not complete, append some automatically chosen numerators.
- **GenerateIBP**->**True** — generate IBP identities, see the corresponding procedure below.
- **AnalyzeSectors**->**True** — determine zero and simple sectors, see the corresponding procedure below.

- **FindSymmetries->True** — find equivalent sectors, see the corresponding procedure below.

In case of success, **NewBasis** creates objects **Ds**[*name*], **SPs**[*name*], **LMs**[*name*], **EMs**[*name*], and **Toj**[*name*]. Their meaning is explained in the output of the **NewBasis** procedure. The integrals of the basis are denoted as **j**[*name*, *n1*, *n2*, ...].

*Example:* **NewBasis**[**b1**, {**sp**[1, 1]+**mm**, **sp**[1-**q**, 1-**q**]}, {1}, **Directory**->"**bdir**"]

Convert the explicit expression to **j** form with **Toj**:

**Toj**[**b1**, **sp**[1, **q**] (**sp**[1, 1]+**mm**)<sup>-1</sup> (**sp**[1-**q**, 1-**q**])<sup>-2</sup>]

gives **j**[**b1**, 0, 2]-**j**[**b1**, 1, 1]+(**sp**[**q**, **q**]-**mm**) **j**[**b1**, 1, 2]. The inverse transformation can be done with **Fromj**[*expr*].

**GenerateIBP**[*name*] generates IBP and LI identities for the basis. IBP identities in the point (*n1*, *n2*, ...) can be retrieved by **IBP**[*name*][*n1*, *n2*, ...].

**AnalyzeSectors**[*name*, *pattern*] finds zero sectors and some other objects as it reports in the output. **ZeroSectors**[*name*] is a list of zero sectors, each element have the form **js**[*name*,  $\theta_1$ ,  $\theta_2$ , ...]. The optional parameter *pattern* tells the procedure to analyze only sectors matching pattern. E.g. if the last two  $D_\alpha$  correspond to irreducible numerators, use **AnalyzeSectors**[*name*, {\_\_, 0, 0}]. This procedure should be called before the call of **FindSymmetries**.

**FindSymmetries**[*name*] finds equivalent sectors and forms. It generates several objects as it reports, including the list of unique sectors **UniqueSectors**[*name*] and the list **MappedSectors**[*name*] of sectors which can be mapped onto unique ones. For each mapped sector **js**[*name*,  $\theta_1$ ,  $\theta_2$ , ...] the mapping rules can be retrieved as **jRules**[*name*,  $\theta_1$ ,  $\theta_2$ , ...].

**SolvejSector**[**js**[*name*,  $\theta_1$ ,  $\theta_2$ , ...], *options*] is a procedure which performs a heuristic search of the reduction rules for a given sector. If it succeeds, the list of found rules can be retrieved as **jRules**[*name*,  $\theta_1$ ,  $\theta_2$ , ...]. It returns the number of master integrals found. Typical usage is **SolvejSector**[@**UniqueSectors**[*name*]]. The useful options include

- **Depth** -> *n* set heuristic search depth. Default is *n* = 2
- **SR** -> **True** use internal symmetries of the sector.
- **TimeConstrained** -> *n* set time constraint in seconds.

**DiskSave**[*name*] save all definitions to disk (see option **Directory** in **NewBasis**).

**IBPReduce**[*expr*] performs the IBP reduction of the expression *expr*.

Several additional tools are included in the package:

**Dinv**[**j**[*name*, *n1*, *n2*, ...], **sp**[*p*, *q*]] returns the derivative with respect to the invariant constructed of the external momenta.

**RaisingDRR**[*name*, *n1*, *n2*, ...] returns the right-hand side of the dimensional recurrence relation (4). Note that the factor  $\mu^L = -1$  for Minkovskian metrics and odd number of loops should be taken into account manually.

**LoweringDRR**[*name*, *n1*, *n2*, ...] returns the right-hand side of the dimensional recurrence relation (5).

**FeynParUF**[**js**[*name*,  $\theta_1$ ,  $\theta_2$ , ...]] returns the list {*U*, *F*, {**x1**, **x2**, ...}}, where *U* and *F* are the functions entering the parametric representation of the integrals in the given sector, and **x1**, **x2**, ... are the parameters.

*Learning more* One is encouraged to examine the examples that are included in the distribution. Another good starting point to know more about the functions of the package is to submit a command **?LiteRed`\*`**.



## 6. Conclusion

In this contribution we have reviewed the LiteRed package performing the IBP reduction of the multiloop integrals. We have described a new algorithm of detecting the zero sectors implemented in LiteRed version 1.4.

*Acknowledgments* I am grateful to the organizers of ACAT-2013 for the support and hospitality. This work is supported by the Russian Foundation for Basic Research through grant 11-02-01196 and by the Ministry of Education and Science of the Russian Federation.

## References

- [1] Chetyrkin K G and Tkachov F V 1981 *Nucl. Phys. B* **192** 159
- [2] Tkachov F V 1981 *Phys. Lett. B* **100** 65–68
- [3] Laporta S 2000 *Int. J. Mod. Phys. A* **15** 5087
- [4] Anastasiou C and Lazopoulos A 2004 *J. High Energy Phys.* **0407** 046 (*Preprint* hep-ph/0404258)
- [5] Smirnov A V 2008 *J. High Energy Phys.* **10** 107 (*Preprint* 0807.3243)
- [6] Smirnov A and Smirnov V 2013 (*Preprint* 1302.5885)
- [7] Studerus C 2010 *Comput. Phys. Commun.* **181** 1293–1300 (*Preprint* 0912.2546)
- [8] von Manteuffel A and Studerus C 2012 (*Preprint* 1201.4330)
- [9] Tarasov O V 1998 *Acta Phys. Polon. B* **29** 2655
- [10] Gerdt V P 2004 *Nucl. Phys. Proc. Suppl.* **135** 232–237 (*Preprint* hep-ph/0501053)
- [11] Smirnov A V and Smirnov V A 2006 S-bases as a tool to solve reduction problems for feynman integrals (*Preprint* hep-ph/0606247)
- [12] Smirnov A V 2006 *J. High Energy Phys.* **0604** 026 (*Preprint* hep-ph/0602078)
- [13] Lee R N 2012 (*Preprint* 1212.2685)
- [14] Kotikov A V 1991 *Phys. Lett. B* **254** 158–164
- [15] Kotikov A V 1991 *Phys. Lett. B* **259** 314–322
- [16] Kotikov A V 1991 *Phys. Lett. B* **267** 123–127
- [17] Gehrmann T and Remiddi E 2000 *Nucl. Phys. B* **580** 485 (*Preprint* hep-ph/9912329)
- [18] Gehrmann T and Remiddi E 2001 *Nucl. Phys. B* **601** 248–286 (*Preprint* hep-ph/0008287)
- [19] Gehrmann T and Remiddi E 2001 *Nucl. Phys. B* **601** 287–317 (*Preprint* hep-ph/0101124)
- [20] Derkachov S E, Honkonen J and Pis'mak Y M 1990 *J. Phys. A: Mathematical and General* **23** 5563
- [21] Tarasov O V 1996 *Phys. Rev. D* **54** 6479 (*Preprint* hep-th/9606018)
- [22] Lee R N 2010 *Nucl. Phys. Proc. Suppl.* **205–206** 135–140 (*Preprint* 1007.2256)
- [23] Pak A 2012 *J. Phys.: Conf. Series* **368** 012049 (*Preprint* 1111.0868)
- [24] Lee R N and Pomeransky A A 2013 (*Preprint* 1308.6676)
- [25] Nickel B, Meiron D and Baker G 1977 Compilation of 2-pt. and 4-pt. graphs for continuous spin models  
Tech. rep. University of Guelph
- [26] Lee R N 2008 *J. High Energy Phys.* **07** 031