THE COMPUTATION OF 1–LOOP CONTRIBUTIONS IN Y.M. THEORIES WITH CLASS III NONRELATIVISTIC GAUGES AND REDUCE

A. BURNEL and H. CAPRASSE
Département d' Astronomie et d'Astrophysique
Institut de Physique B5
Université de Liège, B-4000 Sart-Tilman par Liège 1, Belgium
EMAIL: U214001@vm1.ulg.ac.be

1 Introduction

Yang-Mills quantum field theories are now used extensively in high energy physics to describe the interactions of fundamental particles. These theories contain unphysical degrees of freedom which do not contribute to physical processes. However, they are present in all intermediate steps of calculations in perturbation theory. The number of degrees of freedom and the nature of their contributions in a calculation depend on the way the theory is formulated. Choosing a **gauge** amounts to fix the formulation of the theory.

The archetype of such a theory is quantum electrodynamics (QED) which is based on local invariance with respect to the abelian Lie group $U(1)$. This theory is very well understood. Choosing a gauge fixes the expression of the photon propagator. In a relativistic gauge, four degrees of freedom are present. Two of them are unphysical. One may also choose nonrelativistic gauges i.e. gauges which are not stable against Lorentz transformations. The so-called *axial* gauge is one of them. Its interestlies in the fact that unphysical degrees of freedom decouple when one is computing Feynman diagrams. In QED there even exists a particular nonrelativistic gauge : the *Coulomb* gauge in which the unphysical degrees of freedom do not appear anymore inside the photon propagator.

During those last twenty years, more general Y.M. field theories have been constructed. They are based on local invariance with respect to the non-abelian Lie groups $SU(n)$. The archetype of such a theory is quantum chromodynamics (QCD) which is based on local $SU(3)$ invariance. In these theories, nonrelativistic gauge formulations show problems which are not yet completely solved. For instance, in axial gauge, the gluon propagator gets an unphysical pole . It must be regularized to make Feynman integrations but this regularization is not unique. Another problem (of technical origin) which arises is the non-locality of the divergent parts of Feynman integrals. The most important obstacle to tackle these problems is the high complexity of perturbative calculations done in nonrelativistic gauges for all non-abelian theories. This complexity has prevented people to make extensive perturbative calculations. The use of **REDUCE** has made possible to overcome this problem. The perturbative calculations made in the framework of QCD have allowed us to clarify the properties of nonrelativistic gauges formulations.

The present work is based on results obtained in three completed works [1,2,3]. It presents the key aspects of a **REDUCE** program created to handle the required perturbative calculations. We shall insist on those features which may also be useful for other applications .
The high complexity of calculations has led us

- to construct efficient algorithms and to carefully choose the structure representation of the expressions to be computed,
- to minimize the work done on simplifications.

Another important constraint was the necessity to keep a trace of the results obtained at several intermediate stages of the calculations for verification of correctness. This implied the need to divide the process of calculation into several well separated tasks. Finally, in order to apply the program to other Feynman diagram calculations, we wanted to structure it in such a way that modifications can be easily realized. All these requirements led us to give it a high modularity and to

ISSAC '92-7/92/CA, USA

create specific structure representations well adapted to each stage of the calculations. We have experimented its reusability since the set of gauges considered in Ref. 3 is quite different from the one considered in Ref. 2. In section 2, we describe the overall structure of the program and in section 3, we explain its most interesting features. The conclusions which may be drawn from our calculations are presented in section 4. Further remarks on the program are also made.

2 Structure of the Program

The goal is to compute a QCD one-loop integral using a gluon propagator which has an unusual structure and to do this computation in such a way that a detailed analysis of the contributions of the various parts of this propagator may be carried out. Simultaneously, the verification of the validity of the computation must be made possible at various intermediate steps of the calculation. We stress that this is esential since it is the first time that such a type of propagator is used in a perturbative calculation without the use of dubious tricks (see Ref. 3) .

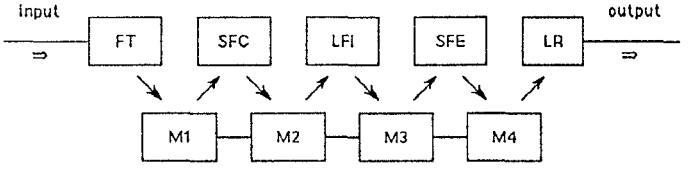

Fig. 1. Organisation of tasks of the program. The lower boxes are code-modules which accomplish four distinct tasks and the upper boxes send input and receive the result of calculations done by the modules.

The structure of the program is depicted in fig.1. The program is divided into four independent modules M1 to M4 which accomplish four distinct tasks. They receive their input and send the result of their computation to their associated data box FT, SFC, LFI, SFE and LR. The output of each box is kept in a file. The initial input is the expression of the integrant of the Feynman integral coresponding to the a specific one-loop Feynman diagram. The output is the result of the Feynman integration before the process of renormalization. We describe briefly here the task of each module:

M1 separates the integrant into a list of contributions corresponding to the various portions of the gluon propagators and also according to their dependence with respect to the integration variable. This list is sent to the box SFC (**S**plitted **F**eynman **C**ontributions). Notice that the box FT (**F**eynman **T**erm) has a passive role.

M2 analyses the dependence in the integration variable of each term and puts it into a canonical form. The output is a list (of lists) of MONOMS. Each monom is in a one-to-one correspondance with a unique Feynman integral which is represented as a KERNEL. Its output is sent to LFI (**L**ist of **F**eynman **I**ntegrals).

To evaluate each Feynman integral, we choose dimensional regularization. The result may be expressed as a symmetrized form of a given expression. For reason of efficiency we do the symmetrization on the abstract (kernel) form of each Feynman integral and transform the various lists of monoms into polynomials. This is the action of M3. The output is sent to the data box SFE (**S**ymmetrized **F**eynman **E**xpressions).

M4 computes the Feynman integrals in terms of generic momenta and subsitutes each kernel expression appearing in the polynomial expressions contained in the box SFE. It makes a term by term simplification and gives the output in LR (**L**oop **R**esult).

3 Description of the Modules

In this section, we want to describe the most interesting features of the program. First, we shall explain some properties of the input. This will give us the occasion to define our notations. Next, we shall explain, how each module is working.

3.1 The Input

The gluon self-energy diagram is given by Fig. 2. The internal lines are gluon or ghost lines.

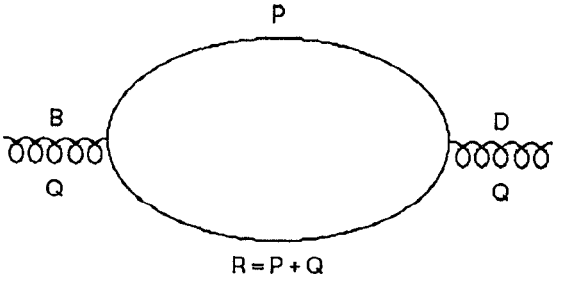

Fig. 2. The gluon self-energy diagram.

External lines denote a gluon with 4-momentum Q. B and D are fixed four-vectors. The projections of Q on them are respectively $Q.B$ and $Q.D$. P is the internal momentum and is also the integration variable of the Feynman integral. The new aspects of our calculation is linked to the new structure of the gluon propagator.

It is given (see Ref. 3) by

$$D_{\mu\nu}(k) = \frac{-i}{k^2+i\epsilon}\left(g_{\mu\nu} - \frac{\hat{k}_\mu k_\nu + \hat{k}_\nu k_\mu}{k_c^2+i\epsilon}\right.$$

$$\left. -k_\mu k_\nu \frac{ak^2 + a'(c.k)^2 - \hat{k}.\hat{k}}{(k_c^2+i\epsilon)^2}\right) \qquad (1)$$

where

$$c = n^*,$$

$$\hat{k}_\nu = C_{\mu\nu}k^\mu,$$

$$(k_c)^2 = C_{\mu\nu}k^\mu k^\nu.$$

n^* is a fixed four-vector. A sufficiently general choice for C is

$$C_{\mu\nu} = \alpha g_{\mu\nu} + n^*_\mu n_\nu$$

where α is a gauge parameter, n a fixed four-vector and

$$\alpha > 0,$$

$$n^2 = n^{*2}.$$

The first term in the R.H.S. of (1) is the usual contribution in the Feynman gauge. The other two terms have a k-dependence which is more complicated than the one of the full Feynman gauge propagator itself. When $\alpha = 0$, the calculations which have been done previously used identities between distributions, which are algebraically not necessarily equal, like

$$\frac{k_c^2}{k_c^2+i\epsilon} = 1$$

and the separation formula

$$\frac{n^*.pn^*.(p+q)}{(n^*.pn.p+i\epsilon)(n^*.(p+q)n.(p+q)+i\epsilon)} =$$

$$\frac{1}{q.n}\left(\frac{n^*.p}{n^*.pn.p+i\epsilon} - \frac{n^*.(p+q)}{n^*.(p+q)n.(p+q)+i\epsilon}\right)$$

to reduce the number of propagators. Such "identities" are proved in distribution theory through the use of **regular** test functions. It may however happen that the regularity conditions are not met in the calculation of Feynman integrals and that the neglected ϵ-terms in fact contribute[4]. On the level of the complexity of the Feynman integral, one propagator in the noncovariant formalism is, so to say, equivalent to *three* Feynman gauge propagators.

From Fig. 2, we see that the integrant contains terms with denominators of the form

$$\frac{1}{(p^2+i\epsilon)^{a1}(r_c^2+i\epsilon)^{a2}(p_c^2+i\epsilon)^{a3}(r^2+i\epsilon)^{a4}} \qquad (2)$$

where $a1$ and $a4$ are equal to 1 while $a2$ and $a3$ are integers between 0 and 2. Happily, one can avoid to consider the above explicit expression inside the computer algebra program. Inside the expression of the integrant , it is unambiguously represented by

$$pc^{a3}rc^{a2}.$$

The data box FT is the algebraic expression of the graph of Fig. 2 calculated from Feynman rules taking into account the above representation of the various denominators. It is given in terms of products of four-vectors as it is needed when one wants to use the **HEPHYS** package of **REDUCE** [5].

3.2 The M1 module

This module takes the expression of the self-energy as given by the data box FT and separates the various parts corresponding to fixed degrees of pc and rc. The subexpressions are decomposed to separate the term which is independent of p and the rest is further decomposed to separate the various parts according to the degree of $p.p$ appearing in them. It introduces the result into an array called FEYN!_TERM whose elements FEYN!_TERM(a3,a2) have the structure

$$\{P_{MAX}, \{\mathbf{deg}_{pcrc} = a3, a2\}, P_{00}, \{P_0, P_1, ..., P_{MAX}\}\} \quad (3)$$

where P_{00} is the part independent of p and where P_i is the coefficient of $(p.p)^i$ in the rest. These coefficients may, of course, still contain other p dependences. The result is sent to the data box SFC.

3.3 The M2 module

The Feynman integral can be put in a canonical form which, after a Wick rotation, is a superposition of integrals of the type

$$I^{a1..a4}_{\mu_1..\mu_n} = \int d^{2\omega}p \, \frac{p_{\mu_1}\cdots p_{\mu_n}}{(p^2)^{a1}(r_c^2)^{a2}(p_c^2)^{a3}(r^2)^{a4}} \qquad (4)$$

The numerator of the expression above, is **linear** with respect to all components of the integration variable. The integration is made in the d dimensional space and

$$\frac{d}{2} = \omega$$

In the module this integral is generated and represented by

$$\mathtt{SYMINTOP}(\{\mu_1 \ldots \mu_n\}) \qquad (5)$$

There is no need to keep track of the denominator since

$$a1 = 1 = a4$$

and $a3, a2$ are fixed and well defined. Let us next explain the linearization process. Each P_i is extracted from (3). It is first put in a distributive form and, next, each monom is placed into a list. The p dependence of each monom is unambiguously identified and its linearization realized in the following algorithmic way:

$$(p.p)^i \Rightarrow (p.p_1)(p_1.p_2)(p_2.p)(p.p)^{i-1}, \quad (6)$$

$$(p.a)^j \Rightarrow (p.a_1)(a.a_1)(p.a)^{j-1} \quad (7)$$

where $p_1, p_2, \ldots, a_1, \ldots$ are generated vectors which will be declared as *indices* later on. The recursions are active until $i = 0$ and $j = 1$. To avoid superfluous simplifications, these manipulations are done in the symbolic mode and the list of arguments of each SYMINTOP directly generated. To give an example, the monom

$$3(p.p)^2(p.r)^3(p.s)(p.t)(r.t)$$

is transformed into

$$3(r.t)(p_1.p_2)(p_3.p_4)(r.r_1)(r.r_2)$$

$$* \texttt{SYMINTOP}(\{p_1, p_2, p_3, p_4, r, r_1, r_2, s, t\}) \quad (8)$$

The result is sent to the data box LFI.

3.4 The M3 module

The operator SYMINTOP must be symmetrized. Though a transformation of each monom (as given by (8) into a sum over all permutations of the indices appearing in SYMINTOP would work, the subsequent number of terms would then be too big. So, we take the underlying intrinsic symmetry of the expression into account. For our present purpose, it is enough to compute the divergent parts of the integrals (4). For these, one finds terms which contain a certain number i of symmetric pairs of indices and $n - 2i$ isolated indices. The number of terms in the sum is

$$number\_of\_terms = \frac{n!}{2^i i!(n-2i)!} \quad (9)$$

For instance, the divergent part of the integral

$$div\ I^{1,2,2,1}_{\mu_1 \quad \mu_{10}}$$

contains the structure

$$(\mu_{10}, \mu_9)(\mu_8, \mu_7)(\mu_6, \mu_5)(\mu_4, \mu_3)(\mu_2)(\mu_1)$$

and the number of terms is 4725 compared to 10!. M3 contains the procedures for the above symmetrization of the monoms.
SYMINTOP becomes a symmetrized sum of another KERNEL called INTOP. The polynomial expression is, finally, reconstructed. The results, for all values of $a3$ and $a2$ are put in an array called POLINTOPEXPR stored in the data box SFE.

3.5 The M4 module

This module effectively computes the value of the Feynman integrals and, subsequently, substitutes the calculated values inside the array elements contained in the data box SFE. The results are put inside the LR . The calculation is delicate because the swelling of intermediate expressions may be very large. First, we further simplified the calculations taking

$$n^2 = 0 = n^{*2} \quad (10)$$

Next, we have made the reduction of each integrant using the explicit expressions of C' and C'^2 (see Ref. 3) by hand and we have given the results to the program as symbolic mode expressions . The program extracts the parametric Feynman integrals called $\texttt{ZINTEG}(i, j, k, l, ln)$ where

$$\texttt{ZINTEG}(i, j, k, l, ln) =$$

$$\int_0^\infty \frac{z^i}{(z+\alpha)^j (z+\alpha+n^{+2})^k (z+\alpha+n^{-2})^l} \quad (11)$$

$$ln \quad = \{n^{+2}, n^{-2}\} \quad (12)$$

and computes them. We observed that the results of integration are often long expressions. For instance, when six arguments are present, each integral in **factorized** form occupies 11K and a factor over 10 more when written in a nonfactorized form. It was hopeless to substitute them *simultaneously*. So, we developped the following strategy:

a. The integrals are, in a first step, computed for generic arguments called

   VECT1, VECT2, ..., VECTN.

   The results are kept (in a factorized form) in an array called INTOPINI.

b. Writing the elements of POLINTOPEXPR in a **distributed** form, a term by term substitution of the integrals is done. The program doing this is given in the appendix C of Ref. 2 . The functions which put the input expression into a distributive form are given in the package ASSIST [6] of the library of REDUCE 3.4. With that method of calculation the intermediate expression has never occupied more than 1.5 Mbytes of heap space for the calculation of the $a2 = 1 = a3$ contributions.

4 Conclusions and Further Remarks

To give the program its powerful features, we had mostly to work within the symbolic mode. The main

reason was that it was vital to avoid superfluous simplifications . The control of the simplification process is possible in RLISP but not on the level of the algebraic mode. This is, in our view, the main drawback of REDUCE. The default full simplification of algebraic expressions made by the function *aeval* should be made suspendable at the level of the algebraic mode and replaced by the *eval-* type lispian evaluation. To keep the expression very compact, in each step of the calculations, we have maximized the *implicit* character of its representation . There is one noticable exception to this rule: this is when we introduce the distributive representation of the multivariate polynomial which describes the Feynman contribution before integration is made and when we reintroduce it to generate a step by step simplification. In each case the increase of the memory occupation was controllable and, in fact, " a priori " known. The adoption of the distributive representation made the algebraic calculations become mainly symbolic manipulations of abstract data.

Feynman integrals have been computed independently of the integration package INT. This last package has been used only to check the validity of our own specialized algorithm. The package HEPHYS was not used for its trace calculation facility (we have no fermion here). It was used to take profit of its abstract 4-vector data structure, to exploit the algebraic properties of the product of two 4-vectors and to use the facility of index summations. The simplification procedures of the package worked automatically and nicely as soon as the results of integration were introduced in the final formal expression given by the module M3. The system was able to realize the important simplifications we expected.

To give an example, in class III nonrelativistic gauges and in the limit $\alpha = 0$ (corresponding to the axial gauge), the divergent part of the contributions of all Feynman integrals for $a2 = 1 = a3$ is **regular** and remarkably simple. It is given by :

```
- ( B.D*Q.Q*NP2**2 + 2*B.Q*D.Q*NP2**2
   + 2*B.D*Q.NN*Q.NSTAR*NP2
   - 6*B.Q*D.NN*Q.NSTAR*NP2
   + 6*B.Q*D.NSTAR*Q.NN*NP2
   + 6*B.NN*D.Q*Q.NSTAR*NP2
   - 2*B.NN*D.NN*Q.NSTAR**2
   - 3*B.NN*D.NSTAR*Q.NN*Q.NSTAR
   + 6*B.NSTAR*D.Q*Q.NN*NP2
   - 3*B.NSTAR*D.NN*Q.NN*Q.NSTAR
  )/(6*NP2**2)
```

where

$$n_+ = \frac{1}{2}(n + n^*),$$

$$NP2 \Leftrightarrow n_+^2 \ , \ NN \Leftrightarrow n \ , \ NSTAR \Leftrightarrow n^*.$$

The fact that we end up with a **local** expression is an important outcome of the calculations.

Another feature which merits further comments is the reusability of the program. In Ref. 3 planar and light-cone gauges with a causal single-pole prescription have been reconsidered. A recent proposal made to regularize the unphysical pole inside the gluon propagator[7] was used to compute the gluon exchange contribution of fig. 2. The Feynman integrals are quite different. To implement the necessary modifications, we had only to make a small addition to the symmetrization program included in M3 and we had to modify the module M4. All calculations were completed within a few days. Again, new interesting results are obtained. They show that that ghosts different from the decoupling Faddev-Popov ghosts must be used so that the main motivation in using true (class II) axial gauges in loop calculations is lost.

Our program is still an incomplete package to compute loop integrals. Its aim was NOT to do that task anyway. In spite of this, we think it offers a unified and efficient strategy to reach that goal. A module should be added which generates all Feynman diagrams of a given order and apply the relevant Feynman rules to generate the input sent to the data box FT. The algorithms necessary to compute the finite parts of the Feynman integrals should be added to M4. Finally a module should be added which apply the renormalization process on the results as given by M4. In that way one isolates the renormalization process. This is important for reusability since one may use many different renormalization schemes.

References

1. A. Burnel, and H. Caprasse, *Phys. Lett.* **265B** (1991) 355.
2. A. Burnel, and H. Caprasse, *Int. J. Mod. Phys.* **C** (1992) (to be published).
3. A. Burnel, and H. Caprasse, *"Planar and Light-Cone Gauges with a Causal Single-Pole Prescription"* University of Liège preprint (To be published 1992).
4. A. Burnel, R. Kobes, G. Kunstatter, and K. Mak, *Ann. Phys.* **204** (1990) 405.
5. *Reduce User's Manual, Version 3.4* (Rand publication **CP78** 1991)
6. H. Caprasse **ASSIST.RED** *Reduce Digital Library*, (1991).
7. I. Lazzizzera, *Nuovo Cim.***102A** (1989) 1385.