JOURNAL OF COMPUTATIONAL PHYSICS 7, 191–200 (1971)

An Attempt to Evaluate Renormalized Radiative Corrections by Computer

J. CALMET AND M. PERROTTET

National Center for Scientific Research, Center of Theoretical Physics, 31 Chemin Joseph Aiguier, 13 Marseille 9e, France

Received December 30, 1969

An attempt is made to apply algebraic computational techniques to renormalization theory. Starting from the functional formalism for a scalar theory as well as for quantum electrodynamics we describe the LISP and FORTRAN programs which generate Feynman diagrams. Within the $g\Phi^3$ scalar model, we integrate the Feynman integrals over the internal momenta and remove the divergences of this superrenormalizable theory. A LISP program performs these operations. The numerical evaluation of the multi-dimensional integrals is briefly discussed in the last part of this paper.

I. INTRODUCTION

The renormalization theory of quantum electrodynamics (QED) and of renormalizable scalar theories in the framework of functional formalism [1–4] gives rise to an algebraic algorithm for the generation of Feynman diagrams. One can determine within this approach the structure and the exact number of counter terms which are needed to get finite contributions from these diagrams. As is well known, the effective evaluation of these contributions is straightforward for the lower orders of the perturbative expansion but involves such an amount of manipulations for higher orders that the use of computers becomes necessary.

The programming language LISP [5, 6] is particularly well adapted to algebraic computations and allows us both to formulate and to perform the above-mentioned algorithm. Moreover, it has already been noted [7–9] that this language is suitable for the evaluation of Feynman diagrams. We have therefore used computers to try to perform the complete scheme of renormalization. This can be split into the following three parts:

(i) Generation of Feynman diagrams and their relative counter terms at a given order;

(ii) Integration over internal momenta and removal of divergences;

(iii) Numerical evaluation of integrals over Feynman parameters.

581/7/2-1

To generate Feynman diagrams one can either use combinatorial analysis or start from the generating functional of propagators. In the first case FORTRAN language has been used to write down a program which works both for a specified interaction and for a mixture of interactions. In this way, one can get the set of graphs describing a given physical process. In the second case we closely follow the functional formalism approach in writing two LISP programs in which the functional derivation is represented by the connection of lines. We thus obtain the whole set of diagrams at a given order. At the present time, these programs work for the following interactions: QED, $g\Phi^3$, and $g\Phi^2\varphi$, where Φ and φ are scalar fields. It is straightforward to generalize them to the $g\Phi^a$ theories ($a > 3$). The results obtained within the first method can be recovered in the functional approach. Section II is devoted to point i and includes a physical introduction.

Point ii, developed in Section III, is presented here for the convergent and logarithmically divergent diagrams of scalar theories. Let us note that the removal of higher divergences could easily be performed. In this approach we closely follow the Feynman method [10]. Although the $g\Phi^3$ theory does not have any physical content, it reproduces the denominators of the integrals appearing in QED. This program is thus a part of a more general one dealing with QED.

The last point (iii) has been carried out by means of Monte-Carlo methods in which the variance is reduced by a stratified sampling technique. Although this method is rough, it is the only one available at the present time which solves our problem. This program, written in FORTRAN, is briefly described in Section IV. All the results related to the $g\Phi^3$ theory amplitudes are obtained within this approach. LISP programs have been run on the IBM 360-67 at Institut de Mathématiques Appliquées de Grenoble and FORTRAN ones on the UNIVAC 1108 at Orsay.

II. Generation of Diagrams

1. *Physical Framework*

Consider the functional u_0 which is defined in QED by

$$u_0[\eta, \bar{\eta}, J; m] = \exp\left\{-\frac{1}{2}\int \bar{\eta}_\alpha(x)\, S^{(c)}_{\alpha\beta}(x - x'; m)\, \eta_\beta(x')\, d^4x\, d^4x'\right\} \times \exp\left\{-\frac{1}{4}\int J_\mu(x)\, \Delta^{(c)}(x - x')\, J_\mu(x')\, d^4x\, d^4x'\right\} \quad (1)$$

and for a scalar theory by

$$u_0[J; \kappa] = \exp\left\{-\frac{1}{4}\int J(x)\, \Delta^{(c)}(x - x'; \kappa)\, d^4x\, d^4x'\right\}. \quad (2)$$

In (1), η, $\bar{\eta}$, and J are the sources corresponding, respectively, to the fermion, antifermion, and photon fields, m is the electron mass, and the photon mass is set equal to zero. In (2) J is the source of the scalar field and κ the square of the mass of the associated particle. $S^{(c)}$ and $\Delta^{(c)}$ are the usual causal Green functions,

$$\Delta^{(c)}(x;\kappa) = -\frac{2i}{(2\pi)^4}\lim_{\epsilon=0}\int d^4k\,\frac{e^{ikx}}{k^2+\kappa-i\epsilon}, \tag{3}$$

$$S^{(c)}(x;m) = -\frac{2i}{(2\pi)^4}\lim_{\epsilon=0}\int d^4k\,\frac{e^{ikx}}{k^2+m^2-i\epsilon}(i\gamma^\mu k_\mu - m). \tag{4}$$

γ^μ are the Dirac matrices; $\mu = 1,\ldots, 4$.

The sources belong to the $\mathscr{D}(R^4)$ space and all the formulas given here are thus mathematically well-defined. The u_0 are the generating functionals of vacuum expectation values of the chronological product of field operators [4]. The generating functional of propagators for a scalar interaction of the type $g\Phi^a$, where g is the coupling constant, takes the form

$$u[J;g,\kappa] = \exp\left\{ig\int \Gamma(\xi)\,d\xi\right\} u_0[J;\kappa]. \tag{5}$$

Γ is the interaction operator defined by

$$\Gamma = \left(\frac{1}{i}\frac{\delta}{\delta J}\right)^a. \tag{6}$$

In the same way u and Γ are defined in QED by

$$u[\eta,\bar{\eta},J;e,m] = \exp\left\{ie\int \Gamma(\xi)\,d\xi\right\} u_0[\eta,\bar{\eta},J;m], \tag{7}$$

$$\Gamma = \frac{\delta}{\delta\eta_\alpha}\gamma^\nu_{\alpha\beta}\frac{\delta}{\delta\bar{\eta}_\beta}\frac{\delta}{\delta J_\nu}, \tag{8}$$

where e is the electron charge and the u functionals are related to the scattering matrix. It is straightforward to show that the functional derivatives of u, once the sources have been set equal to zero, are expressed by the usual Feynman diagrams [4].

The renormalization procedure consists of the introduction into the lagrangian of the theory of enough counter terms to remove all the divergences of this theory. These counter terms appear in u as formal series. Once this procedure has been performed and u has been expanded in a formal series of the coupling constant,

we get the following recurrence formula which is valid both for QED and $g\Phi^a$ ($a < 5$; for $a > 4$, the theory is not a renormalizable one):

$$\mathscr{P}u^{(n+1)} = i\Gamma\mathscr{P}u^{(n)} + \sum_{\rho=1}^{n} \binom{n}{\rho} \mathscr{D}_0^{(\rho)}\mathscr{P}u^{(n-\rho)}, \tag{9}$$

$$\mathscr{P}u^{(0)} = u_0 .$$

$\mathscr{P}u^{(k)}$ denotes the finite part of the approximation of order k of u. The $\mathscr{D}_0^{(\rho)}$ are functional differential operators which depend on the counter terms introduced in u; they are defined in Ref. [1] for QED and in Ref. [2] for $g\Phi^3$. From Eq. (9) one can obtain the finite contributions of the graphs of order k once the $(n-1)$ first finite contributions are known. Equation (9) also gives an algebraic algorithm useful to construct all the renormalized diagrams at a given order; if we set $\mathscr{D}_0^{(\rho)} \equiv 0$ ($\rho = 1,\ldots, n$) unrenormalized graphs are obtained.

2. LISP *programs*

We want to generate all the unrenormalized Feynman diagrams at a given order of perturbative expansion. This means that the interaction and the number of vertices are fixed. It is straightforward to generalize the programs to the case of renormalized diagrams. In Eq. (9) $\mathscr{P}u^{(n)}$ is the product of u_0 with a sum of graphs; thus when Γ is applied to $\mathscr{P}u^{(n)}$ the functional derivatives act either on a graph or on u_0. The algebraic algorithm used to construct graphs can be stated as follows:

(i) A functional derivative applied to u_0 creates an external line;

(ii) When it is applied to an external line of a diagram it changes it into an internal line;

(iii) It has no effect on an internal line;

(iv) All the lines created or transformed by the derivatives involved in Γ are connected at a new vertex.

In order to characterize a diagram we use the following notation. A line of a specified type is associated to each source; for example, in QED one has three different types which are denoted by A, B, C and correspond, respectively, to the fermion (η), antifermion ($\bar{\eta}$) and photon (J) sources. An external line of type A connected to the vertex k is defined by the list (k A). An internal line comes from the derivation of an external line and is thus obtained by connecting two external lines; consequently, a photon line which joins the vertex k to the vertex j will be represented by a list of the form (k C j C), while Eq. (1) shows that ane lectron internal line between the same vertices will be either (k A j B) or (k B j A). Moreover, the fact that types A and B are different means that the electron lines are oriented. By convention, the external lines are placed at the beginning of the list and are

ordered with respect to their type. In QED we have choosen the order A, B, C. For example, the vertex of the third order is

$$((2\,A)(3\,B)(1\,C)(1\,A\,2\,B)(1\,B\,3\,A)(2\,C\,3\,C)), \tag{10}$$

when the sources η, $\bar{\eta}$, and J are associated, respectively, to the vertices 2, 3, 1.

In Eq. (9) $\mathscr{P}u^{(n)}$ is now of the form

$$\mathscr{P}u^{(n)} = \sum_i (G_i u_0), \tag{11}$$

where the G_i are the graphs of order n. Let D be one of the G_i. The list $((n+1\,A)(n+1\,B)(n+1\,C))$ represents the vertex of order $(n+1)$. To obtain all the diagrams of order $(n+1)$ coming from D one has to apply the interaction operator Γ both on D and on u_0. According to the algorithm and to the choosen representation, the generation of these diagrams comes from the following operations:

(i) One of the external lines of the vertex $(n+1)$ is connected to an external line of D and the two others remain external; this occurs when one derivative is applied to D and the two others to u_0.

(ii) Two external lines of the vertex $(n+1)$ are connected to external lines of D and the third one remains external.

(iii) All the external lines of the vertex $(n+1)$ are connected to external lines of D.

(iv) All the lines of the vertex $(n+1)$ remain external (the three derivatives act on u_0) and thus we get a disconnected graph.

A first program, called DIAG I, performs these operations with regard to the following possible connections: C–C, A–B, and B–A. It also includes a function which arranges the lines according to the choosen representation of diagrams. DIAG I gives rise to the whole set of graphs appearing at a given order. With a slight modification, the disconnected diagrams can be deleted.

This program can work under two different circumstances. In the first one, a graph D of order k and the vertex $(k+1)$ are given in the input and we get all the diagrams of order $(k+1)$ coming from D. In the second, a function which builds up the vertices from order 1 to n is added to DIAG I and all the diagrams at every order from 2 to n are obtained. In this case we only have to give the value of n in the input. Diagrams are printed in a form similar to (10).

Several diagrams obtained in this way will be identical; in fact, we again get the numerical coefficients which were obtained by hand for lowers orders [3]. In order to extract the different diagrams another program, called DIAG II, has been written. It acts according to the following scheme. The list of graphs

at a given order is worked out by DIAG I, the first graph of this list is selected, and all its symmetric diagrams are built up by permutations on the labels of the vertices. Then one has only to eliminate from the previous list all the diagrams appearing in the list of symmetric ones and to go on with the next graph. DIAG II also includes a function which cancels tadpole type diagrams and applies Furry's theorem [11]. The input is the same as for DIAG I except that if one wants to read graphs one has to read a whole order. Finally, a function available both with DIAG I or II allows us to extract the diagrams describing a given physical process.

In order to give an idea of the efficiency of this method of enumeration of diagrams let us note that DIAG I needs about 150 sec of machine time to go from the sixth- to the seventh-order connected diagrams, while DIAG II needs 12 sec to give diagrams up to the third order.

These programs have been described for QED. To use them for the $g\Phi^3$ theory we have only to identify the types attached to the three lines. The straightforward generalization to $g\Phi^4$ or $g\Phi^5$ will be written if there is any request for it.

An interesting feature of the above framework is that the generating functional provides a powerful way to solve the problem of generation of diagrams.

3. FORTRAN *programs*

It is obviously possible to construct unrenormalized diagrams by using only combinatorial analysis. This program, called FRENEY, acts in the following way. Given either an interaction or a mixture of interactions, a number of vertices, a number of external lines, and their types it constructs all the corresponding graphs.

One introduces the matrix $N(i, j)$, $1 \leqslant i \leqslant IS$, $1 \leqslant j \leqslant IT$, where i is the index of rows, IS is the number of vertices, j is the index of columns, and $IT/2$ is the number of lines involved in the interaction.

For a mixture of interactions, $IT/2$ is the greater number of lines connected to a vertex. To each kind of field appearing in an interaction is associated a line of definite type which is characterized by a positive integer. A line of type b connected to the vertex k will be labeled by the couple $(k\ b)$. All the lines of the vertex k lie in the k-th row of N, which is simply the juxtaposition of all the couples $(k\ b_{2l})$,

$$(k\ b_2\ k\ b_4 \cdots k\ b_{IT}).$$

The program needs the following ordering of types: $b_{2l} \leqslant b_{2l+2}$. An internal line is taken to be the connection of two couples of a same type; the construction of diagrams will then consist in all the possible connections of such couples. The external lines are defined to be lines which cannot be connected: they are expressed in N by couples (0 0). Let us remark that one has first to write N and

then to fix the external lines. The matrix N' obtained in this way is the one which has to be given in the input. For example, for the interaction $g\Phi^2\varphi$ when φ is said to be of type 1 and Φ of type 2, one gets at the 4th order,

$$N = \begin{pmatrix} 11 & 12 & 12 \\ 21 & 22 & 22 \\ 31 & 32 & 32 \\ 41 & 42 & 42 \end{pmatrix}.$$

If we ask for the self-energies of the Φ field N' has the form

$$N' = \begin{pmatrix} 11 & 00 & 12 \\ 21 & 00 & 22 \\ 31 & 32 & 32 \\ 41 & 42 & 42 \end{pmatrix}.$$

External lines have to be set in the first rows. In order to construct a graph one has to determine its internal lines. In the output a diagram is given by a three-column matrix, denoted by M. A row of M is of the form $(k\ a\ h)$ and represents a line of type a connecting the vertices k to h. It is obtained when the $(k\ a)$ and $(h\ a)$ external lines are connected. When a graph is generated we have to determine whether it is a new one or not. It will be deleted if its associated M matrix only differs from one already obtained by a permutation of the rows. It will also be eliminated if it is a symmetric diagram of a previous one. In order to analyse this possibility we have to generate the diagrams coming from every possible permutations of internal vertices and to compare them at each step of this generation with the diagrams already obtained. To perform these eliminations we have imposed a special ordering on the elements of M [12].

It needs 72 sec on a UNIVAC 1108 to get the vertices of the seventh order in QED and 13 sec to get the electron self-energies of the sixth order. An important feature of this approach is that we get the right number of diagrams appearing in the scattering matrix.

III. Computation of Diagrams

To compute graphs we have closely followed the well-known Feynman method [10]. A LISP program, called ACOFIS, realizes it for the scalar amplitudes which are at most logarithmically divergent. A scalar diagram is expressed by an integral such as

$$\int \frac{d^4k_1 \cdots d^4k_l}{a_1 a_2 \cdots a_n}, \tag{12}$$

where l is the number of independent loops in the graph and the k_i are the internal momenta associated with these loops. The a_i^{-1} are the propagators related to the internal lines. The a_i are of the form

$$a_i = \left(\sum \xi_r p_r + \sum \eta_s k_s\right)^2 + \kappa_i , \tag{13}$$

where the metric has signature $(+ + + -)$, ξ_r and η_s have the values ± 1 or 0, and the p_r are the external momenta. The conservation of momenta at each vertex has been imposed in (12). It determines the p_r and k_s which enter into the a_i . κ_i is the squared mass of the corresponding particle.

Starting from the list of propagators which has to be given in the input, the Feynman parameters α_i are introduced by means of the identity

$$\frac{1}{a_1 \cdots a_n} = (n-1)! \int_0^1 d\alpha_1 \cdots d\alpha_n \frac{\delta(\sum_1^n \alpha_i - 1)}{[\sum_1^n \alpha_i a_i]^n} . \tag{14}$$

Then we look for the terms to be shifted in order to get a quadratic form in the integration variables k_i and perform the corresponding translations on these variables.

At this step we have to consider whether the integral is finite or not. If it is, the formula

$$\int \frac{(k^2)^{m-2}}{(k^2 + A^2)^n} d^4k = \frac{i\pi^2}{[A^2]^{n-m}} B(m, n-m) \qquad (n > m > 0), \tag{15}$$

where $B(p, q)$ is the beta function, immediately gives an integral over the Feynman parameters which is the contribution from the computed graph.

In the $g\Phi^3$ model the only irreducible divergent diagram is the second-order self-energy. It is logarithmically divergent. If $\pi(p^2)$ denotes its contribution, according to renormalization theory its finite part will be defined by

$$\mathscr{P}\pi(p^2) = \pi(p^2) - \pi(p^2 = -\kappa). \tag{16}$$

Then the use of the subtraction identity

$$\frac{1}{a^n} - \frac{1}{b^n} = -n \int_0^1 dz \frac{(a-b)}{[z(a-b) + b]^{n+1}} , \tag{17}$$

which increases the power of the denominator, makes it possible to apply (15).

What has to be noted is that, due to (9), subdiagrams have to be replaced by their finite parts and hence the order of integration over the k_i is irrelevant.

The program consists of 16 simple functions. In the input we need to give the propagators, the internal and external momenta, the masses, and the Feynman

parameters. The convergent or logarithmically divergent diagrams can then be computed for any scalar theory.

The results calculated by hand within the $g\Phi^3$ model [3] have been recovered in this approach. We have also checked, in the same model, the asymptotic behaviour of the Feynman amplitudes at large momentum transfer. Results are plotted in Ref. [12]. To compute all the self-energies and vertices up to the seventh order takes only 80 sec of machine time. This emphasizes the efficiency of the LISP technique.

Apart from the cancellation of logarithmic divergences this program has been extended to the case of QED and used to extract the contributions from irreducible vertices to the anomalous magnetic moment of the electron. ACOFIS is used to compute the denominators.

IV. Numerical Evaluation

The last point to consider is the numerical evaluation of integrals over Feynman parameters. This is done using a FORTRAN program based on Monte-Carlo type methods.

When the Monte-Carlo method does not give a good accuracy, we use the following variance-reducing technique. We split the integration domain into subcubes and ask for an upper limit of the accuracy in each subcube. When the accuracy is not good enough we can either increase the number of random points in the corresponding domain or again split this domain into smaller subcubes.

The accuracy of the values obtained in this way depends obviously on the analytic structure of the integrand, but is at worst 10%. An exact calculation within the $g\Phi^3$ model of the contribution of the fourth-order Delbrück type diagram reported in Ref. [13] has allowed us to check our program. More precisely, we have made use of the equation (11) in Ref. [13]. This equation depends on dilogarithm functions, which have been computed according to the method developed in Ref. [14]. Using the same parametrization in our approach we have checked results with an accuracy of about 1%. For example, a value obtained using 50,000 random points is 0.09264 ± 0.00085, while the correct value is 0.09261. Thus our results are in agreement with those obtained analytically.

Acknowledgments

We would like to thank Professor A. Visconti for suggesting this work to us and for his continuous encouragement. We are grateful to Dr. Y. Siret (I.M.A.G.) for his helpful advice and to Dr. J. V. Corbett for his critical reading of this manuscript.

REFERENCES

1. Y. LE GAILLARD AND A. VISCONTI, *J. Math. Phys.* **6** (1965), 1774–1785.
2. J. SOFFER AND A. VISCONTI, *Phys. Rev.* **162** (1967), 1386–1389.
3. J. CALMET, R. SENEOR, J. SOFFER, AND A. VISCONTI, *Phys. Rev.* **162** (1967), 1390–1393.
4. A. VISCONTI, Proc. of the 4th Winter School of Theoretical Physics in Karpacz, University of Wroclaw, 1967.
5. J. MCCARTHY *et al.*, "LISP 1.5 Programmers Manual" (revised ed.), M.I.T. Press, Cambridge, Mass., 1965.
6. C. WEISSMAN, "LISP 1.5 Primer," Dickenson Publ. Co., Belmont, Calif., 1967.
7. A. C. HEARN, *Comm. ACM* **9** (1966), 573.
8. J. A. CAMPBELL, *Nucl. Phys. B* **1** (1967), 283.
9. A. C. HEARN, "REDUCE User's Manual" (revised version), Stanford Artificial Intelligence Project, Memo 50, 1968, unpublished.
10. R. P. FEYNMAN, *Phys. Rev.* **76** (1949), 769.
11. Any book on quantum electrodynamics.
12. M. PERROTTET, "Thèse de 3e Cycle," Marseille (1970).
13. V. CONSTANTINI, B. DE TOLLIS, AND G. PISTONI, Nota. Int., Vol. 164, Ist. di Fisica G. Marconi, Universita di Roma, 1968.
14. K. MITCHELL, *Phil. Mag.* **40** (1949), 351.