+ corrections

CALT-68-736
DoE RESEARCH AND
DEVELOPMENT REPORT

Feynman Diagram Evaluation by Computer I: General Comments and Gamma Matrix Algebra*

by

STEPHEN WOLFRAM

California Institute of Technology, Pasadena, California 91125

ABSTRACT

This paper is the first in a series which discusses the use of algebraic computer programs as an aid to the calculation of Feynman diagrams in high-energy physics. I first give a general discussion on the nature of the Feynman diagrams to be evaluated and outline the many aspects of their calculation which may conveniently be handled by computer. Then I describe in some detail the program GAMALG, (written in the LISP-based symbolic manipulation language MACSYMA) which performs algebraic manipulations on Dirac gamma matrices (in n dimensions). GAMALG is available for general use, and I give many examples of its application. In addition, I discuss several useful results on gamma matrices which do not appear to be widely known. The next two in this series of papers (written with A. E. Terrano) address the evaluation of the momentum space integrals resulting from Feynman diagrams. We shall describe a program to perform automatically any Feynman parameterization, yielding a form suitable for numerical evaluation, together with a program which performs analytically some integrals in terms of beta functions. For diagrams involving mostly massless particles up to three loops (and with few external lines), the remaining integrals may usually be done analytically by a Chebyshef expansion method, as we shall describe. At the one-loop level, all Feynman integrals may be

*Work supported in part by the U.S. Department of Energy under Contract No. DE-AC-03-79ER0068, and by a Feynman Fellowship.

evaluated completely in terms of dilogarithm functions, and we shall present a program which does this. Finally, we ~~will describe~~ intend to treat the automatic generation of diagrams and of renormalization counterterms.

[Some parts of this paper are based on 'MACSYMA Tools for Feynman Diagram Calculations,' in the proceedings of the 1979 MACSYMA Users' Conference, June 1979].

1. Introduction

The evaluation of Feynman diagrams is essentially the only known method for extracting physical predictions from the relativistic quantized field theories which are believed to describe the interactions of elementary particles. The computation of complicated diagrams is necessary both to allow more precise comparisons with experimental results, and to elucidate the basic structure of the theories investigated. The purpose of this series of papers is to describe in detail programs which have been written by A. E. Terrano and ~~me~~ myself in the algebraic manipulation language MACSYMA to automate most common features of diagram calculations. In Section 2 of this paper, I discuss briefly the classes of Feynman diagrams which have been evaluated to date and delineate those which are likely to be of relevance in the immediate future, and in whose calculation our programs are intended to assist. This paper does not describe the basic aspects of Feynman diagrams; for these we refer the reader to Refs. [1.1] or [1.2]*. In Section 3, I discuss the algorithms and construction of the MACSYMA program GAMALG which handles the gamma matrix algebra aspects of Feynman diagram evaluation, and give examples of its use.

For those unfamiliar with it, a few words should be said concerning MACSYMA. MACSYMA is undoubtedly the most powerful algebraic manipulation computer language available. It was developed by the MATHLAB group of the M.I.T. Laboratory for Computer Science, and is presently generally available only on their KL-10 computer, which may be accessed directly by telephone, or, for example, through the ARPA network. MACSYMA is written in LISP. It manipulates

---

* In any branch of theoretical physics where perturbation theory is used, diagrammatic techniques abound. The diagrammatic methods encountered in studies of many-body systems (atoms, nuclei, solids, spin systems, turbulent fluids, ...) are qualitatively similar to those used in high energy physics, although they are usually more amenable to approximation. We shall, however, discuss only ~~genuine~~ relativistic Feynman diagrams here.

algebraic expressions, and performs symbolic differentiation, integration (of any form whose integral contains only elementary functions, and in some cases, dilogarithm or Spence functions), factoring and so on. The REDUCE, SCHOONSHIP and ASHMEDAI systems often used for high-energy physics calculations perform some gamma matrix algebra efficiently but do not offer such sophisticated ancillary algebraic manipulation capabilities as those available in MACSYMA. The main limitation of MACSYMA at present is its inability to handle more than a few thousand terms; this deficiency should, however, be in part overcome when it is implemented on other computers. For more details of MACSYMA see Ref. [1.3], which may be obtained from the MATHLAB group at M.I.T. Note that in this paper there occasionally appear references to disk files; these are on the M.I.T.-MC KL-10 computer. To use GAMALG (and the programs to be described in later installments of this series), it is not necessary to have a sophisticated knowledge of MACSYMA. In fact, for simple purposes one need only know how to enter MACSYMA. Then GAMALG (and our other programs) may be loaded and used by imitation of the examples given below or those available on-line on the computer, or by following the reference manuals available on the computer.

This series of papers is addressed primarily to those concerned with physics, rather than with computers. However, at least some of the discussion, especially on the purpose of the programs described, is intended to be accessible to all. A more detailed description of ~~the~~ some computer programming aspects of our work will be presented elsewhere.

References for Section 1

1.1. R. P. Feynman, 'The Theory of Fundamental Processes,' Benjamin, 1962.

1.2. J. D. Bjorken and S. D. Drell, 'Relativistic Quantum Mechanics,' McGraw-Hill, 1964.

1.3. MATHLAB Group, M.I.T. Laboratory for Computer Science, 'MACSYMA Primer,' (March 1978) and 'MACSYMA Reference Manual,' Version 9 (Dec. 1977).

2. Applications of Feynman Diagrams

2.1 Introduction

In order to assess what computational tools for Feynman diagram evaluation are necessary or desirable, one must consider the nature of the diagrams to be evaluated and the form of the results required. This consideration is addressed in the present section. After some general comments, I describe the diagrams which have been calculated so far in the various theories presently under widespread investigation and speculate on those which will be of interest in the near future. These speculations will suggest some definite directions for the computer evaluation of Feynman diagrams which I, at least, intend to follow. In this section I ~~could~~ have given many references; ~~instead I have decided to give none.~~ but they are undoubtedly far from complete.

The computation of interaction probabilities in realistic quantized field theories can, at present, be done only as a 'perturbation' expansion in powers of the coupling constant (a parameter which specifies the strength of the interaction) for the theory. The terms in the perturbation expansion may be represented by Feynman diagrams. At the kth order in the expansion, each diagram will typically contain k vertices and up to k/2 loops. (Note that, in estimating the complexity of diagrams, final state phase space integrations should be treated on an equal footing with internal loop integrations.) A total of about 30 sets of diagrams for distinct physical processes have been evaluated at the first (lowest) order; about half of these have been extended to the next order. About four processes have been calculated to third order ($g-2$ [2.1], β_{QED} [2.2], $\sigma_{QCD}(e^+e^-)$ [2.3], deep inelastic scattering operator anomalous dimensions [2.4] and a couple of somewhat derivative and less relevant ones (e.g. [2.5]). The anomalous magnetic moment of the electron is being calculated numerically to fourth order ($O(\alpha^4)$, 4 loops) and involves some 891 diagrams [2.6].

(+ ?)

The complexity of Feynman diagrams grows drastically with order. Typically, at the kth order, k! diagrams contribute and the number of terms in the momentum space integrand for each diagram also grows roughly like k!, while the dimensionality of the final integrals to be evaluated grows as k. The precise number of terms generated depends importantly on the algorithm used, but it is reasonably clear that beyond fifth order, no existing computer memory can be expected to hold the required amount of information. Before this obstruction is met, however, the time required for evaluation of the diagrams by existing computers becomes quite prohibitive. There is no known method for complete systematic analytical evaluation of all the integrals generated in high-order diagram calculations. One must, therefore, resort at least in part to numerical means; the high dimensionality of the integrals required essentially dictates the use of a Monte Carlo method, and huge cancellations require very many sample points. One may estimate that the time required for numerical diagram evaluation on the fastest existing computers grows with order at least like $(k!)^5$, typically crossing one year at $k = 4$. It is clear that a better method than the diagram expansion is called for, but there has so far been no indication that such a method exists. In fact, the vigor with which a new method is sought has dulled somewhat in the thirty years since the invention of Feynman diagrams. The need for a new method of calculation is perhaps the most convincing justification for attempts at analytical, rather than numerical, evaluation of high order diagrams. From analytical results, one may hope to discern patterns and identify the important classes of terms, and thereby be able to guess or even prove the behavior of higher orders. Certainly the comparative simplicity of the analytic forms of many higher-order results encourages this hope. (For example, there are indications that the numerically most important parts of non-logarithmic terms in diagrams

involving only massless particles are proportional to $(\pi^2)^{[j/2]}$ (from changing sheets in log terms), $j!\zeta(j)$ (from end-point singularities in parametric integrals), and sometimes $\psi^{(j-1)}(n)$, where n is a parameter (e.g., moment number) and $j \leq k$. In addition, the fact that all analytical results for diagrams have reduced to polylogarithm functions may be significant.)

Of course, it is not entirely evident that very high order results are relevant. The perturbation expansion is presumably at best an asymptotic series, and if the expansion parameter (presumably effectively the coupling constant) is not sufficiently small that only very low orders need be considered, then the effects of the eventual divergence of the series are probably devastating even if high order terms can be calculated. It appears that no interesting field theory (probably including QED) gives a Borel re-summable perturbation series (i.e., the series are like $\Sigma k!\alpha^k$ rather than $\Sigma(-1)^k k!\alpha^k \sim e^{1/\alpha}\mathrm{Ei}(1/\alpha)/\alpha$). In the case of QED, the expansion parameter ($\alpha \sim 1/137$) is sufficiently small that very high order difficulties are plausibly irrelevant. However, the extreme accuracy of experimental investigations of QED (e.g., one part in 10^{13} for the anomalous magnetic moment of the ~~muon~~ electron) necessitates rather high order calculations to allow theoretical comparisons. (In fact, beyond $O(\alpha^4)$ even the electron anomalous magnetic moment receives sufficiently important contributions from strong and weak interaction effects to render a pure QED calculation inadequate.) In QCD, it is less clear that the effective coupling constant is small enough (at accessible energies) to justify the many perturbative calculations which have been done. Among other difficulties, the effective expansion parameter for QCD perturbation series appears to be α_s or perhaps even $\pi^2\alpha_s$, rather than the α/π typically found in QED.

In the next three sections, I summarize very briefly most of the Feynman diagram calculations which have been done in field theories of apparent relevance to the real world. In addition, I mention some of the more obvious

calculations which remain to be done. With these in mind, Section 2.5 discusses the computer programs for diagram evaluation which are required. The purpose of this series of papers is to describe several computer programs which have been written to serve the majority of these needs.

I shall not mention below the many calculations which have been done in, for example, ϕ^3 and ϕ^4 field theories (in various numbers of spacetime dimension often not close to four). Since such theories are presently not believed to be directly relevant in practice (except perhaps for the Higgs' sector of weak interaction theories), calculations on them are usually in the nature of numerical experiments to investigate the behavior of the perturbation series and are usually simpler than the corresponding ones for more realistic field theories. The foremost of these is probably a recent analytical calculation of the beta function for ϕ_4^4 to fifth order ($O(\lambda^9)$; four loops) [2.10]. This is the highest order to which any complete diagram calculation has been carried. It was performed using methods which we have now implemented in the computer programs to be described in the second and third installments of this series (loop-by-loop extraction of (Euler) beta functions and ~~configuration space Chebyshef~~ Gegenbauer [2.11] expansion). The results were used in an estimate of the prescription-independent quantity $\beta'(\lambda^*)$, $\beta(\lambda^*) = 0$, to all orders in λ using Borel resummation methods [2.12]. Another scalar field theory in which several calculations have been done is ϕ_6^3 (ϕ^3 in 6 spacetime dimensions). These are intended as simple model calculations for QCD (ϕ_6^3 is asymptotically free), but are probably irrelevant, not least because ϕ_6^3 does not exhibit the doubly-logarithmic infrared singularity structure of massless QED and QCD.

2.2 Quantum Electrodynamics (QED)

QED is the relativistic quantized field theory for the interactions of electrons (and muons) with photons. It was for the investigation of this

theory that Feynman diagrams were originally developed [2.13], and until quite recently (~ 1973) it was the only theory in which higher-order calculations appeared to be meaningful. Many complicated QED processes have been evaluated, and in all cases where non-electromagnetic effects are negligible, they have been in precise agreement with experiment; any practical failings of perturbative QED have yet to be uncovered. By now, most QED diagram calculations which are relevant for comparisons with existing and forthcoming experiments have been done, and in recent years there have been few new calculations. In most cases, higher order effects are smaller than (e.g., hadronic) effects not directly calculable from QED.

Of the basic QED processes, μe [2.14], e^-e^- [2.15], e^+e^- [2.16] and $e\gamma$ [2.17] scattering (including $e^+e^- \to \gamma\gamma(\gamma)$ [2.18]) have been evaluated to second order ($O(\alpha^2)$; one loop) while $\gamma\gamma$ scattering is known only to lowest order ($O(\alpha^2)$ in amplitude; one loop) [2.19]. For $\gamma^* \to$ anything (or equivalently, the QED beta function), there are third-order results ($O(\alpha^3)$; two loops) in the high-energy limit [2.2]. (Contributions from diagrams with the maximal number of internal fermion loops, which numerically dominate the $O(\alpha^3)$ result, are known analytically through $O(\alpha^8)$ [2.5].) The photon propagator (vacuum polarization) is known completely to second order ($O(\alpha^2)$; two loops) [2.20]; for massless electrons (i.e., high-energy limit) the three-loop contribution has also been calculated [2.2] (the parametric form for a complete calculation to this order has recently been given [2.21]). The complete electron propagator is known to second order ($O(\alpha^2)$; two loops) [2.22] as is the $e\gamma e$ vertex ($O(e\alpha^2)$; two loops) [2.23]. The $\sigma_{\mu\nu}$ projection of this vertex, which gives the anomalous magnetic moment, has been calculated to third order ($O(e\alpha^3)$; three loops) [2.1] and a fourth-order calculation is in progress [2.6]. All of these results, except parts of the third-order anomalous magnetic moment, are analytical. In most cases there was not even a preliminary numerical calculation made.

In many cases, however, some part or limit of the answer was known before the complete calculation, often by non-diagrammatic techniques (e.g., time-ordered perturbation theory, effective Lagrangians or renormalization group methods). The second class of calculations in QED is of radiative corrections to decay processes. For strongly-interacting initial or final particles, such calculations cannot be made precise within the context of QED, although the relevant diagrams have been computed to one-loop (first) order. For μ decay, complete analytical calculations exist to first order in α ($O(\alpha)$; one loop), while some features of higher order terms (exponentiation) are known ($\mu \to e\nu\nu e^+e^-$ is also known numerically). For other than V $\pm$ A muon-like decays, the effects of higher-order weak interactions must be included; the calculations are complete to the one-loop level. The second-order corrections to the leptonic decays of $W^{\pm}$ have also been calculated. Corrections to radiative decays (in which, to lowest order, a single photon is emitted) may be related to the photon propagator and given (for vanishing electron mass) to third order ($O(\alpha^8)$; three loops), although they are probably not useful.

A third class of QED calculations involves the bound states positronium (e^+e^-), muonium (μ^+e^-) and hydrogen (pe^-). The second-order corrections to the decay rates of ortho ($^3S_1$) and para (^1S_o) positronium are known ($O(\alpha^4)$ and $O(\alpha^3)$, respectively; one loop), for the $^3S_1$ case only numerically. For positronium and muonium, the energy levels have been calculated analytically to third order ($O(\alpha^2 \log\alpha(\alpha^2 Ry))$), and an analytical calculation of the next order is in progress, with some parts completed. The energy levels of the hydrogen atom, and in particular the Lamb splitting, are dominantly determined by QED effects and have been calculated as for muonium. In most higher-order bound state calculations, it is necessary to use sophisticated techniques to treat the multiple interactions which bind the particles into the bound

state, but do not contribute directly to the effect (e.g., decay rate or level splitting) calculated. Typically these methods involve the use of non-plane-wave incoming particle states or application of complicated projections on diagrams to remove iterated effects[2.39]. The programs we describe do not address these manipulations.

A further group of QED calculations give corrections to the scattering of an electron from an external potential. Many of these calculations border on atomic physics and are done numerically. Nevertheless, analytical results exist for Bremsstrahlung ($eV \to e(\gamma)$) to second order ($O(\alpha^2)$; one loop)[2.40], the Bethe-Heitler process ($\gamma V \to e^+e^-(\gamma)$ or $\mu^+\mu^-(\gamma)$) to second order ($O(\alpha^3)$; one loop)[2.41], and for Delbrück scattering ($\gamma VV \to \gamma$) to lowest order ($O(\alpha^4)$; one loop)[2.42]. It is interesting to note that of all the diagram structures in QED, the box (which is responsible for $\gamma\gamma \to \gamma\gamma$ and hence Delbrück scattering) is probably the least studied, despite its numerically dominant contribution to the difference between the electron and muon anomalous magnetic moments to $O(\alpha^3)$ (by virtue of an unusual infrared log (m_μ^2/m_e^2) term not arising from renormalization and therefore not really $\log(m_\mu^2/(m_e^2 + \mu^2))$)[2.26].

There are several further classes of QED calculations. One involves the interactions of polarized electrons and photons, which have typically only been studied to lowest order[2.43]. However, most QED calculations relevant for direct comparison with existing or forthcoming experimental results have been carried to the highest order at which effects of other (e.g., strong) interactions (which cannot usually be calculated accurately) may be neglected. There is, therefore, little experimental incentive for further QED calculations in the immediate future. Moreover, it is unlikely that any meaningful investigation of the structure of the perturbation series for QED could be made by evaluating high-order diagrams using foreseeable computational tools. I,

therefore, do not anticipate many new calculations in QED in the near future, except as trivial by-products of QCD calculations.

2.3 Quantum Chromodynamics (QCD)

QCD is the primary candidate for a field theory of strong interactions. It describes the interactions of quarks and gluons. Almost by definition, strong interactions have coupling constants so large that perturbation theory is usually useless. However, QCD has the remarkable property (which is supported by experimental results) of 'asymptotic freedom,' which leads to small effective coupling constants for processes involving large momentum transfers (short distances)[2.44]. This constraint on the region of applicability of QCD perturbation theory (at least should) determines what quantities are computed in QCD using Feynman diagrams.

The first class of calculations in QCD determines the behavior of the (suitably defined) effective coupling constant at high momentum transfers by evaluating the terms logarithmic in energy resulting from loop corrections to the qqG or GGG vertices, usually by finding the $O(1/\epsilon = 1/(4-n))$ divergent pieces of the diagrams. Second-order ($O(g^3)$ ($g^2 = 4\pi\alpha_s$); one loop) calculations of the beta function, which determine the effective coupling, revealed the property of asymptotic freedom. The analytical result for the beta function is known to third order ($O(g^5)$)[2.45]. Of course, one cannot expect to reveal much about the behavior of QCD at small momentum transfers, where the effective coupling is large, by a perturbation expansion in the strength of the interaction (for example, terms of order $\exp(-1/g^2)$, which have zero asymptotic expansion in powers of g will inevitably be missed). It is, therefore, not entirely clear that a calculation of $\beta(g)$ to higher orders is worthwhile, although such results would presumably help in attempts to find $\beta(g)$ to all orders in g. It would also be amusing to know if the fixed-point condition

$\beta(g^*) = 0$ can be satisfied for $g^* \neq 0$ to higher orders. In the model QED without internal fermion loops necessary to study such eigenvalue conditions, it is known explicitly to $O(e^7)$ that $\beta(e)$ has no zeros away from $e = 0$ and it is proved that if a zero exists, it must be of infinite order. In QCD, $\beta(g)$ can have a zero to $O(g^5)$ only if the number of quark flavors is larger than 8. (It is amusing to note that the $O(g^3)$ term changes sign when $F > 16.5$, while the $O(g^5)$ term becomes positive when $F \geq 8.05$. One wonders whether this increase in the importance of fermion loop terms continues in higher orders.) One unfortunate feature of the $O(g^7)$ and higher contributions to $\beta(g)$ is their dependence on the renormalization prescription used. (This occurs also in the QED beta function, which is known to $O(e^7)$, when diagrams containing extra fermion loops, which give $O(1/\epsilon^2)$ as well as $O(1/\epsilon)$ terms, are included.) This impedes the physical interpretation of such contributions, since cross-sections must also be computed to third order to provide a prescription-independent measure of g. It appears that the programs described in this series of papers should be capable of at least a numerical and probably an analytical calculation of $\beta(g)$ to $O(g^7)$. Most calculations of $\beta(g)$ assume massless quarks; the results for $\beta(g)$ with massive quarks are known to $O(g^3)$, where they are already prescription-dependent.

A class of calculations similar to those yielding $\beta(g)$ derives the 'anomalous dimensions' which give the dependence of the effective (~ current) mass of a quark on the acceleration used to measure it (Q^2). Analytical results here have been obtained to second order ($O(g^4)$; two loops).

The first (and probably theoretically best-founded) class of QCD calculations which allowed direct comparison with experiment made use of the operator product expansion. In this method, the diagrams computed involve 'operator vertices' which are added to the Feynman rules as a formal device to keep

track of the various terms with specific dependences on the external momentum. The process best investigated by these techniques is deep-inelastic scattering for which analytical results to third order ($O(\alpha_s^2\alpha)$; two loops, for anomalous dimensions, one loop for coefficient functions [2.4]) exist, although at third order they are extremely complicated and have yet to be properly used. The second order ($O(\alpha_s\alpha)$) results (for anomalous dimensions) have, however, been used extensively as the basis for leading log approximations to many processes [2.49]. Another use of the operator product expansion method is in calculating QCD corrections to weak decays. These calculations are characterized by the appearance of complicated operator vertices containing γ_5; they have been completed (though not to the satisfaction of all) to the one loop level [2.50] which is quite sufficient in view of the ad hoc nature of the models used to relate results for quarks and gluons to those for hadrons. (Nevertheless, two-loop calculations are apparently being made.)

In the last year or so, there has been a considerable amount of effort devoted to direct calculations of higher-order cross-sections in QCD perturbation theory. These divide basically into two classes: those for which the initial state contains colored particles (i.e., quarks and/or gluons) and those whose initial states consist only of photons (or $W^{\pm}$, z°). The latter class of calculations has the advantage that when the cross-sections for the production of all possible final states accessible from the given initial state are added together, the result is free from infrared divergences. For processes involving initial colored particles, infrared divergences remain but are universal to all processes and may therefore be factorized out (c.f., the discussion of divergences in Section 4.1 of the forthcoming installment) in comparisons between processes. In a given process, the change in the divergent pieces with energy is given by the very same 'anomalous dimensions' as describe

the energy dependence of the deep inelastic scattering cross-section calculated using the operator product expansion. (In fact, beyond lowest order, it is not known whether the anomalous dimensions associated with parton → hadron are the same as those calculated for hadron → parton; an explicit diagram calculation to investigate this would be very interesting.)

Probably the most directly relevant process involving no initial color which has been calculated is the total cross-section for electron-positron annihilation to hadrons, given in perturbation theory by the processes $e^+e^- \to q\bar{q}$, $q\bar{q}G$, $q\bar{q}GG$, $q\bar{q}q'\bar{q}'$, ... This is obtained as the imaginary part of the photon propagator. Calculations of the photon propagator at deep Euclidean momenta ($q^2 \to -\infty$) are at a similar level of rigor to those on deep inelastic scattering of virtual photons (with $q^2 \to -\infty$) from hadrons. (An operator product expansion is, of course, unnecessary for the photon propagator, since at leading twist, there is only one contributing operator.) The photon propagator in QCD has been calculated analytically to second order ($O(\alpha\alpha_s)$; two loops). In fact, to this order, QCD results differ from the corresponding QED only by an overall factor. However, the $O(\alpha\alpha_s)$ finite part of the photon propagator depends on the renormalization prescription used to calculate it, and the original QED calculations had been done for massive electrons using the usual QED renormalization prescription (momentum space subtraction at $q^2 = 0$) rather than the dimensionally regularized minimal subtraction method usually used for QCD. The relevant analytical calculation is, in fact, given in the second of this series of papers as an illustration of the use of our programs. Since QCD perturbation theory is only sensible at energies much higher than the effective masses of quarks 'produced' in e^+e^- annihilation, it is presumably sensible and extremely convenient to neglect their masses in calculations. Unless specified otherwise, all results discussed in the remainder of this section

are for massless quarks. To obtain the imaginary part of the photon propagator, it is only necessary to evaluate $O(1/\varepsilon)$ and more divergent terms in the real part at the order required (the anomalous dimension of the photon field in QCD). The analytical calculation of the necessary terms has recently been completed to third order ($O(\alpha\alpha_s^2)$; three loops). (It was for this calculation that the programs described below were developed. Two other groups have also done the calculation, one numerically and the other analytically [2.3].) In addition to calculations of the total cross-section for e^+e^- annihilation, there have also been several studies of the energy distributions of the final states produced [2.51]. These typically use the $O(\alpha\alpha_s)$ differential cross-section and integrate over final state configurations with some weight function other than the uniform one which would give the total cross-section. The form of the weight function determines the analytical form of the integrals; typically integrals with reasonable weight functions can be done in terms of dilogarithm functions at $O(\alpha\alpha_s)$. A further class of calculations compute the energy spectrum of one or more particles produced in e^+e^- annihilation. The results are not infrared finite, but their divergences may be factorized out. Most calculations in e^+e^- annihilation (as elsewhere) have been done for zero mass quarks. The introduction of small quark masses appears, however, to reveal some interesting results (e.g., $O(m^2/s)$ corrections to cross-sections but $O(\sqrt{m^2/s})$ corrections to final state energy distributions [2.52]).

Another color singlet initial state process computed in QCD is $\gamma\gamma \to$ hadrons. Some ~~Analytical~~ calculations on this have been done to third order ($O(\alpha^2\alpha_s^2)$; two loops [2.53]. A further process in this class is $W^* \to$ hadrons (where the virtual W comes from τ heavy lepton decay $\tau \to W^*\nu_\tau$). Calculations on this give essentially the imaginary part of the W^* propagator (or 'axial vector spectral function') in QCD and have been performed analytically to second order ($O(g_W^2\alpha_s)$;

two loops in real part of propagator[2.54]). They are rendered difficult by the presence of γ_5.

The simplest QCD process involving colored partons in the initial state is probably deep inelastic scattering. This has been analyzed to third order ($O(\alpha\alpha_s^2)$) by operator product expansion methods[2.4]. The correspondence with direct calculations is reasonably clear, but a further direct diagrammatic evaluation of deep inelastic scattering at third order would serve as a useful check on existing results. At lowest order, deep inelastic scattering involves only $\gamma^*q \to q$. In the next order $\gamma^*q \to qG$ and $\gamma^*G \to q\bar{q}$ (and presumably processes thus far not calculated involving extra initial state particles[2.55]) contribute. Using the 'anomalous dimensions' derived from the operator product expansion analysis, or other methods, one may extract the term in $\gamma^*q \to qG$ proportional to $\log(q^2/\mu^2)$ which appears when the gluon is collinear to one of the quarks and sum the tower of such terms arising from multiple independent gluon emissions to all orders in α_s. The next tower of terms is built on the complete result for $\gamma^*q \to qG$ (including the 'constant' part not containing logarithmic factors) and iterates contributions from the $O(\alpha_s^2\alpha)$ 'anomalous dimensions'.

Another process studied directly in QCD perturbation theory is lepton pair (e.g., $\mu^+\mu^-$) production in hadron-hadron collisions (the 'Drell-Yan process'). At lowest-order, this arises from $q\bar{q} \to \gamma^*(\to \mu\bar{\mu})$. The next order corrections due to $q\bar{q} \to \gamma^*G$ and $Gq \to \gamma^*q$ (together with loop corrections to the $q\bar{q} \to \gamma^*$ vertex) have finally (after several incorrect attempts) been evaluated analytically[2.56]. One still higher order correction ($O(\alpha\alpha_s^2)$) due to $qq \to qq\gamma^*$ has now been computed analytically after a numerical result had been obtained[2.57].

The hypothetical alert reader may perceive that the three processes just discussed all involve simple crossings of loop corrections to the $qq\gamma^*$ vertex and to processes involving q, $\bar{q}$, γ^* and G variously in the initial and final

state. The loop diagrams may be crossed directly at this order (up to π^2 terms resulting from reversing the sign of the arguments of $\log^2$ terms), but the phase space integration for the tree diagrams differs for the various initial states, and it is not presently known how to relate the 'crossed' processes directly. Clearly the next set of QCD processes involve q, $\overset{(-)}{q}$, γ^*, $\gamma^{(*)}$ and G. In this class lie $\gamma\gamma \to$ hadrons, γ^*hadron $\to \gamma X$, hadron hadron $\to \gamma\gamma X$ and several others of less immediate experimental relevance. Only $\gamma\gamma \to$ hadrons has been calculated beyond lowest order, but the others should be quite accessible using, for example, the programs discussed below.

I have thus far discussed only QCD processes involving a photon in the initial or final state. QCD perturbation theory cannot describe in detail the formation or structure of hadrons. Whenever the momentum of a quark or gluon produced in a process must be determined, there is always an intrinsic error or smearing introduced because of its fragmentation to the hadrons which are observed. If only photon momenta need be considered, none of these difficulties exists. Nevertheless, processes involving only hadrons and with sufficiently large momentum transfers that use of perturbative QCD is reasonable are more easily accessible experimentally, and it is, therefore, useful to make QCD calculations on them. The lowest-order results for $q\overset{(-)}{q} \to q\overset{(-)}{q}$, $Gq \to qG$, $GG \to q\bar{q}$ and $GG \to GG$ were obtained about three years ago, and recently there have been several attempts to calculate to the next order. Only the tree graphs ($2 \to 3$, e.g., $qq \to qqG$) contributions have been obtained for all the necessary processes (which involve well over 100 diagrams). It will be interesting to check these results and to include one-loop contributions. The lowest-order calculations have been done for massive as well as massless quarks and may be relevant to the production of heavy flavors.

Still another (and perhaps the least justified) class of QCD calculations involves the decays of bound states of heavy quark pairs. At lowest order, these calculations are equivalent to those for positronium. Higher-order corrections, however, depend on the non-Abelian (trilinear gluon) couplings in QCD. They have been evaluated analytically to second order ($O(\alpha_s^3)$; one loop) for the case of $Q\bar{Q}$ annihilation at rest to GG, GGG and $GGq\bar{q}$ (~~no restriction on the quantum numbers of~~ with the initial $Q\bar{Q}$ state restricted to be in a spin 0 state ~~was made~~). [2.61] The corrections here were found to be surprisingly large. It would be interesting to see whether the large corrections to the total decay rate are also manifest in a large spreading of the lowest-order 2-jet final state energy distribution by three-particle production [2.62] and to calculate the effect of initial spin states. In view of the mounting experimental evidence, it would be of considerable interest to compute the higher-order corrections to the three-jet final states expected at lowest order from the $(Q\bar{Q}) \to GGG$ decays of $^3S_1$ $Q\bar{Q}$ bound states. (a part of this calculation was performed in [2.63]) The number of diagrams involved is, however, quite large, and the presence of massive quarks leads to grave difficulties for analytical techniques. Moreover, a proper calculation would have to make use of the bound state techniques mentioned above for QED, which may be difficult to apply precisely to QCD.

In doing QCD perturbation theory, one usually assumes that the various fields have approximately zero value in the vacuum and considers small fluctuations in the fields. However, it is known that, in Euclidean spacetime, the QCD vacuum can exhibit a classical non-vanishing value for the gluon potential (an instanton [one word] solution to the classical pure Yang-Mills field equations). In calculating the perturbations from this vacuum state, one must use complicated propagators for the quarks and gluons in this background field (similar in conception, and some details, to those encountered in calculations of atomic properties, where electrons propagate in the background electromagnetic

field of the nucleus). Such calculations are rather different in character from usual Feynman diagram calculations and are distinguished for the need to work mostly in configuration space and consequently the appearance of Bessel functions [2.64].

From the above discussion, it is clear that there exists a rather large number of interesting calculations in QCD perturbation theory which should yield to computer evaluation in the near future. In many cases, numerical results will be quite sufficient for the phenomenological purposes intended, but in cases involving only a single kinematic invariant (e.g., $\beta(g)$ or the e^+e^- total cross-section), where results will be essentially single numbers, it is clearly desirable to have exact analytical forms, which may help to reveal detailed patterns.

2.4 Quantum Flavordynamics (QFD)

QFD is the name for a class of theories for weak interactions. The simplest and most successful of these is the Weinberg-Salam model. The expansion parameter in the perturbation series for these theories is typically of order 10^{-5} at the energies of present experiments so that calculations beyond the lowest order are rarely necessary. QFD calculations are made difficult by two features: 1) The necessity of treating γ_5 and 2) of retaining many particle masses. The γ_5 couplings are difficult because regularization tends to spoil essential invariances of these couplings (anomalies) while the masses are awkward because they lead to more difficult integrals. Calculations beyond lowest-order in QFD at present energies are few: some rare decays (e.g., $K_L^o \to \mu\bar{\mu}$) [2.65] and mass splittings [2.66] (e.g., $K_L^o - K_S^o$) have been calculated to one-loop order because they vanish at the tree level. In addition, there have been several calculations of higher order effects of eventual relevance to neutrino

scattering. The $O(g_W^2\alpha)$ electromagnetic radiative corrections to neutrino scattering from a quark have been evaluated numerically [2.67], and several analytical calculations to the same order have been done of two-W exchange neutrino scattering [2.68] and of the weak charge radius of the neutrino. (Weak effects give a just measurable contribution to $g_\mu - 2$ at one-loop order, which has been calculated in many weak interaction models [2.69].)

Recently, there has been a number of investigations of the behavior of QFD at very high energies ($\sim 10^{15}$ GeV, etc.), intended to study the possibility of a unification between QED, QFD and QCD at these energies. This would be revealed by the effective couplings from the three theories attaining the same value at very high energies. The necessary calculations (done by considering the momentum dependence of three-point functions or vertices) have been completed to third order ($O(g^5)$; two loops), including massive quarks and bosons [2.71]. Another application for QFD diagram calculations is to estimates of baryon number generation in the very early universe ($kT \sim 10^{15}$ GeV, $t \sim 10^{-36}$ sec), which is proportional to the CP (or time reversal) violation in the decays of superheavy ~~vector~~ scalar (or, perhaps, ~~scalar~~ vector) bosons to baryons [2.72]. In some of these calculations, it is necessary to go to two-loop order, since one-loop effects vanish (by their group theoretic weights). However, these calculations should properly (although they have not, so far) be done with finite-temperature propagators, thereby taking them out of the realm of 'canonical diagrams' which, for example, the programs described in this series of papers could handle directly.

One sector of QFD theories which will undoubtedly receive much attention in the future is the 'Higgs sector'. While direct experimental investigation of this is perhaps remote, it is important in determining the structure of the theory. Since Higgs bosons have spin zero, interactions involving only

these require no gamma matrices: the integrals to be done may be written down directly from diagrams. However, the most relevant quantity associated with the Higgs sector is probably the 'effective potential' which is most conveniently computed without the aid of Feynman diagrams, but still in some cases requires similar integrals. It has been calculated completely to one-loop order [2.73], and the two-loop contributions from internal Higgs scalar loops are known [2.74].

Finally, I mention quantum gravity. No direct experimental investigation of quantized gravity is likely to be performed in the foreseeable future. However, by investigating the structure of quantum gravity theories, it may be possible to constrain the forms of other theories which are amenable to direct experimentation. Feynman diagram calculations in quantum gravity are characterized by the appearances of very large tensors, usually associated with the propagators for particles with high spin (e.g., 3/2 or 2). In most cases, the particles may be taken massless, thus simplifying integrations. Several one loop and a few two-loop calculations have been done in quantum gravity [2.75]. One interesting calculation which could probably be done by the programs described in this serial is of the three-loop beta function for a class of supersymmetric theories in which $\beta(g) = 0$ to $O(g^5)$ [2.76].

2.5 Comments and Preview of the Serial

The last three sections were intended to give some indication of the Feynman diagrams which have been evaluated, and, by dint of the uses to which they have been put, those whose calculation is likely to be of interest in the near future. One feature which should be emphasized is that often a single Feynman diagram appears in many different applications, usually with different interpretations for the participating particles and often with some slight generalization or limit taken. An example of this is the decay of orthopositronium to $\gamma\gamma\gamma$; the same diagrams also give directly the structure expected

for the decay of a $^3S_1$ heavy quark bound state to three gluons. There are many examples of the recycling of QED Feynman diagrams in low-order QCD applications, where the contributing diagrams do not yet involve trilinear gluon couplings. However, it is often rather difficult to locate in the literature a suitable calculation, and even once located, the paper in which it is contained is quite often incomprehensible or at least not in the form or notation close to that desired, and sometimes its results are incorrect. For these reasons, it is common for very similar or even identical calculations to be redone many times. The books on QED (with the notable exception of Landau and Lifshitz) seem almost never to present results for diagram calculations in a useful form. It, therefore, appears that a compilation of Feynman diagram calculations in a reasonably standard form would be of value. We have attempted to carry out such a compilation, and, in fact, the last three sections are in some respects a summary of it. Unfortunately, we found that the results to be found in the literature were very rarely in a useful form (often they had been left as multi-page formulae, and no limits had been considered, or variables appearing in the answer were not defined, and even in some cases the diagrams supposedly evaluated were not drawn). It seems clear that to make a genuinely useful compilation, one must reevaluate all the diagrams to be included. This should in most cases be possible using the programs to be described in this series of papers. No P

Presumably, it would be desirable to evaluate diagrams analytically in as much generality as possible. Then, each particular result desired would require only taking limits in the general expression. By keeping the general results in a computer, one would, of course, be able to take such limits automatically and without error. The value of a compilation of rather general diagram results where limits must be taken to obtain useful forms must be

weighed against a system which would evaluate directly the simpler diagrams. From several points of view, the latter alternative seems desirable, but its desirability depends essentially on the time necessary to obtain each result. Perhaps the most satisfactory compromise would be a compilation (probably not in a computer) of many simpler Feynman diagram results, which would often include the limits (or whatever) required in a particular case. Other results could then be obtained by direct use of the computer programs which generated the results in the compilation.

Feynman diagram evaluation probably represents the major computational burden in modern theoretical high-energy physics. To expect that all aspects of it could be handled by computer is perhaps unjustifiably optimistic. However, it does seem that a major portion of the diagram calculations mentioned in the previous sections could be carried out by the computer programs to be described in this series of papers. The calculations will, nevertheless, probably stretch the programs and the computers on which they are executed to their limit, and it will, therefore, be desirable to simplify the calculations wherever possible. In addition, many special classes of calculations (e.g., those involving bound states) must be treated manually or by further computer programs to cast them into the form required. Clearly, there is a great diversity in the types of Feynman diagrams to be evaluated and in the uses to which they are put. To attempt to write a single computer program which systematically 'evaluates' any Feynman diagram would, therefore, be quite impossible. Rather, it is prudent to develop programs which handle the commoner manipulations required, while allowing the basic organization and method to be determined by the particulars of each calculation. This more flexible approach is the one which we have adopted; the present series of papers describes the programs which we have developed to perform the various manipulations

usually encountered in Feynman diagram calculations, by a selection of methods intended to suit the differing types of results required.

Typically, a calculation starts from a diagram, as represented by a matrix element containing integrals over loop momenta with integrands involving traces of products of gamma matrices arising from fermion lines. The first step is clearly to perform the necessary traces, yielding an integrand which involves only dot products of the various internal and external momenta (or, in some rare cases, totally antisymmetric products of momenta). This simple part of the calculation may always be carried out by GAMALG without difficulty. It is in the evaluation of the momentum space integrals that the main difficulties of Feynman diagram calculations usually lie. Depending on the complication of the diagram considered, and the form of the results required, a number of different approaches to the integration can be taken. It turns out that any one-loop diagram (for which there is but one momentum integration) can be integrated exactly in terms of at most 192 dilogarithm or Spence functions ($Li_2(x) = \int_x^0 \frac{\log(1-t)}{t}\,dt$; see Table 2.1) with distinct arguments. The method works by reducing all integrals to those which appear in the calculation of the general box diagram (and in some cases, the triangle diagram, which requires only 12 dilogarithms), using the fact that at most four vectors can be independent in four dimensions. The program CYCLOPS, which will be described in the fourth installment of this series, should, therefore, be able to give a complete analytical result for any one-loop diagram. The forms obtained will, in general, be very complicated, but easily amenable to numerical evaluation, or to analytical simplification in limiting cases. Note that while the amplitude obtained by integration over the single loop momentum for a one-loop diagram may be expressed in terms of dilogarithm functions, integration over the phase space available for outgoing momenta will almost inevitably introduce

more complicated functions. A simple empirical rule is that each outgoing momentum integration increases the maximum complexity of the functions which appear by incrementing the maximum subscript of the polylogarithm functions by one. Typically, when the high energy limit is taken for the diagrams, so that all masses may be neglected, the polylogarithm functions usually degenerate into $Li_n(\pm 1)$ and, therefore, zeta functions (see Table 2.1). If dimensional regularization is used for the outgoing phase space integrations, and all particle masses are taken to be zero from the outset, then the relevant zeta functions will often appear directly in the expansions of gamma or beta functions. Non-uniform weightings of the final state phase space (as used, for example, in the computation of shape parameters) can give rise to functions more complicated than polylogarithms. ~~It should be pointed out that~~ (certain cases of dilogarithm functions were in fact recently incorporated into the Risch algorithm for indefinite algebraic integration which is implemented in MACSYMA); ~~the systematic inclusion of higher-order polylogarithms apparently presents considerable difficulties, however~~. For diagrams beyond one loop, I know of no systematic algorithm for complete analytical evaluation of the necessary Feynman integrals. Nevertheless, every ordinary Feynman diagram which has ever yielded an analytical result has been expressible in terms of polylogarithm functions (and polynomials). One might suspect that this is simply because no other form could have been found, but it may also be that for some yet unknown reason, all Feynman integrals may be reduced to polylogarithm functions. As mentioned above, this may be proved at the one-loop level by reduction to box diagram integrals, but analogous methods fail for more complicated diagrams. Typically, while dilogs appear in one-loop diagrams, two-loop diagrams give trilogs. However, when all dimensionful parameters in results are removed by taking the limit that all masses go to zero (or

→ and S_{12} ... ?

equivalently, energies go to infinity), the polylog functions always appear to degenerate into zeta functions. (Low-energy limits sometimes introduce $\mathrm{Li}_n(\frac{1}{2})$ as well as $\mathrm{Li}_n(\pm 1)$, whose analytical form is not known except when $n = 2,3$ (see Table 2.1).)

Most of the methods for obtaining analytical results beyond the one loop level are entirely useless for diagrams containing massive particle propagators; they rely on the comparative simplicity of Feynman integrals containing only massless propagators which occur when dimensional regularization is used. The methods also typically fail when the number of external momenta exceeds two or three. In the absence of analytical methods, one must resort to numerical means. In the cases where this is obligatory, it is probably not particularly damaging: the number of independent parameters (internal masses, dot products of external momenta, etc.) is typically sufficiently large that any analytical form would have been too complicated to provide much illumination. One method for numerical evaluation of Feynman integrals which has apparently not been tried is direct numerical (probably Monte Carlo) integration over the various components of the loop momenta. This approach is presumably inappropriate when dimensional regularization is used, since the random vectors to be generated must then have $n \neq 4$ components. However, other regularization schemes, such as analytic regularization (in which n is taken to be 4, but the $1/p^2$ in propagators is replaced by $1/p^{2-\delta}$) or Pauli-Villars regularization (in which $1/p^2$ is replaced by $1/p^2 - 1/(p^2-\Lambda^2)$) should yield Feynman integrals which can be done numerically in a direct manner. One disadvantage of the method is that it is intrinsically not Lorentz covariant: one integrates explicitly over each component of the loop momenta. However, for the otherwise numerical calculations in which such an evaluation would probably be imbedded, this might be quite satisfactory. Of course, once Lorentz covariance is no

object, it is entirely unnecessary to evaluate gamma matrix traces algebraically (as discussed in Section 3); one may just as well choose a particular representation for the gamma matrices and perform traces by numerical evaluation of matrix products. The major failing of the direct numerical method is nevertheless its inapplicability to dimensional regularization. The majority of the higher-order calculations mentioned in the previous sections (especially in QCD) gives results which depend on the renormalization prescription used to derive them. It is only comparisons between results which are prescription-independent and have a physical meaning. Thus, for a set of higher-order calculations to be useful, all must be done using the same renormalization prescription. Dimensional regularization with minimal subtraction (or a subtraction scheme, such as truncated minimal subtraction ($\overline{\text{MS}}$), derived from this) appears to give the cleanest results and to provide the greatest potential for analytical evaluation. Higher-order analytical calculations done by this method would, however, be useless if other numerical calculations were performed using other renormalization schemes.

A more conventional method for numerical evaluation of Feynman integrals, which can be used with dimensional regularization, is by Feynman parameterization. In this approach, extra scalar integration variables (Feynman parameters) are introduced into the original momentum space integrals, which may then be simplified to the extent that the loop momentum integrations may be performed, leaving only the integrations over the Feynman parameters (a few less in number than the propagators in the diagram) to be done. These multidimensional scalar integrals (over simplicial (i.e., hyper-tetrahedral) regions) are usually very complicated. In some cases (see below and the next installment) judicious treatment of the original momentum space form, and of intermediate stages in the Feynman parameterization can yield a complete analytical result, but this

method fails when massive propagators or many external momenta are present. In the general case, little can usually be done with the Feynman parameter integrals except by numerical means. The program VIPER (described in part two of this serial) takes any momentum space integral, introduces Feynman parameters, performs all necessary loop momentum integrations and can output the results in the form of a series of FORTRAN statements suitable for input to a numerical integration program. VIPER can, in principle, perform integrals over an arbitrary number of loop momenta; in its present implementation, it easily handles three-loop integrals (so long as the input forms do not consist of too large a sum of terms). This is achieved by extensive introduction of symbolic names for subexpressions, whose values are assigned only in the output FORTRAN statements.

A significant fraction (often all or all but one or two) of the integrals which result from massless diagrams with one or two independent external momenta can be done directly entirely in terms of beta functions. These are carried out by COBRA, as described in part two of this serial. The basic method is first to use the identity $p_1 \cdot p_2 = \frac{1}{2} (p_1^2 + p_2^2 - (p_1 - p_2)^2)$ in numerator terma to cancel factors in the denominator of the integrand. Then, each loop integration is performed in turn, and if at each stage exactly two denominator factors depend on the particular loop momentum (if only one does, the integral vanishes), then the integral can be performed immediately in terms of beta functions. The final expression is a product of many gamma and beta functions, which are then Taylor expanded about the point $n = 4$, often revealing ultraviolet or infrared divergences in the form of poles at $n = 4$.

For diagrams with a simple (roughly iterative) topological structure, COBRA can usually give complete results. However, in more complicated diagrams, a limited number of integrals remains. Many diagrams leave the same, often

finite, integrals after the application of COBRA. A table of such residual integrals, suitably manipulated by a program to account for rerouting or relabeling of internal momentum may then often suffice. The integrals may usually be derived (either for use in a table, or in the course of calculation) by the Gegenbauer Chebyshev expansion method, which yields analytical (or at least precise numerical) results. This method, together with the programs CHEBP and CHEBX which implement it, will be described in part three of this serial. The basis of the Chebyshef expansion method is the expansion of a massless propagator as an infinite ('multipole') series in terms of the 4 or n-dimensional angular functions (Gegenbauer Chebshef polynomials). These functions are orthogonal when integrated over the surface of a 4- or n-dimensional sphere. If only finite integrals are considered, then all manipulations may be performed in n = 4 dimensions. In this case (handled by CHEBP), the expansion is done directly in the momentum space form of the integral, and the final result is an infinite sum over the index in the Chebshef expansion, which may usually be done simply in terms of zeta functions (when many momenta appear in a diagram, the resulting sums may not be possible in closed form, but a very precise numerical result is then easy to obtain). The 4-dimensional Chebyshef polynomials are suitable for the expansion of the form $1/(p-p')^2$, while n-dimensional Chebyshef polymonials (sometimes known as Gegenbauer polynomials) are the relevant expansion functions for $1/(p-p')^{n-2}$. However, for divergent integrals, all manipulations must be done in $n \neq 4$ dimensions, and so the n-dimensional Chebyshef polynomials must be used. Nevertheless, the momentum space propagator usually remains (except in counterterm diagrams) as $1/(p-p')^2$. I did not succeed in finding a simple expansion of such propagators in terms of n-dimensional Chebyshef polynomials. However, it was recently pointed out by Chetyrkin and Tkachov that if a Fourier transform is applied to the momentum

space form of a diagram, thereby casting it into configuration space, the propagator for a massless particle becomes $1/(x-x')^{n-2}$, just as required for an n-dimensional Chebshef polynomial expansion. (It is clear that such a multipole-like expansion must be possible in configuration space.) In addition to $1/(x-x')^{n-2}$ factors, the amplitude for a Feynman diagram in configuration space will typically contain terms of the form e^{ikx}, where k is some combination of external momenta. The origin in configuration space must be chosen to minimize the number of such factors, but when necessary, they may be expanded in terms of n-dimensional Chebshef polynomials and spherical Bessel functions (the procedure is a simple generalization of the partial wave expansion of a plane wave in terms of Legendre polynomials (3-dimensional Chebyshef polynomials) and spherical Bessel functions familiar in three dimensions). After the expansions have been performed, angular integrations serve to remove the Chebyshef polynomials, leaving simple radial integrations which may be done in terms of gamma functions, yielding infinite sums. These may usually then be done analytically, at least for the various separate terms at each order in the expansion in powers of (n - 4). The momentum space Chebshef expansion method suitable for finite integrals is implemented in the program CHEBP. For divergent integrals, CHEBX first converts the integral to configuration space (at present with some assistance from the user), and then applies the procedure outlined above. In this way, it appears that massless diagrams with two or less independent external momenta and up to three internal loops should yield to complete analytical evaluation. The simplest calculation, after the $O(\alpha\alpha_s^2)$ corrections to the photon propagator for which the methods were designed, appears to be of the $O(g^7)$ contribution to $\beta(g)$ in QCD.

Another method for diagram evaluation which may perhaps be treated in a later installment of this serial uses generalized hypergeometric functions

(otherwise known as Meijer G functions). This technique starts with the Feynman parametric representation of a diagram and performs a formal expansion of the Feynman parametric integrand as a nested power series in the various Feynman parameters. This expanded form may then be integrated term-by-term, and the result written as a generalized hypergeometric function. The necessary infinite sums may often be performed numerically much more easily than would the corresponding complete integral. In addition, the third definition of the polylogarithm functions given in Table 2.1 may allow the generalized hypergeometric functions to be written in terms of polylogs by application of suitable reduction formulae, although the result seems unlikely to be useful. Note that the 192 dilogarithm functions which appear in the general scalar (one-loop) box diagram may be collected into just one generalized hypergeometric function.[1] In many cases the results of diagram calculations will depend on so many parameters (external momentum, masses, forms of couplings, etc.) that any 'analytical' result would almost inevitably be extremely complicated. Such results are presimably of little value in the search for patterns in perturbation series. Nevertheless, there is still some motivation to obtain an 'analytical' (albeit probably complicated and perhaps in terms only of specially-designed functions), rather than a purely numerical result, since such a result is more readily portable and may be reused with less effort than would a purely numerical result or a program designed to obtain it. In this respect, the semianalytical forms obtained in terms of generalized hypergeometric functions may be of value.

Many of the integrals arising from diagrams and computed by any of the methods outlined above will, of course, diverge as $\epsilon = 4 - n \to 0$. These divergences must be canceled by the addition of renormalization counterterms (and possibly some form of infrared 'counterterms' arising in the factorization

of infrared divergences). The necessary counterterms are most easily determined from the structure of the original diagram from which the integrals were derived (by shrinking loops to points, etc.), rather than directly from the integrals themselves. In the methods of integration discussed above, the fundamental objects treated are the momentum space integrals rather than the original diagrams. This approach is clearly more flexible, since it allows for manipulation of the amplitudes before integration, and for the possibility of, say, operator vertices in the original diagram. Moreover, some integrands may cancel between various diagrams, if suitable internal momentum assignments are made. However, if one is to treat only canonical diagrams, as would be involved in about half the calculations outlined in previous sections, then it may be advantageous to consider the diagrams as such, rather than only the integrals which arise from them. With this intention, a later installment of this series is intended to describe the program DIGEN, which will generate the (unevaluated) matrix elements for all diagrams contributing at a certain order to the transition from a specified initial to final state and the program RENOR which subtracts the necessary counterterms. Almost any Hamiltonian may be used in DIGEN, which operates by direct application of Wick's theorem to the relevant terms in the expansion of the time-ordered exponential of the Hamiltonian. The fields in DIGEN may be assigned any propagators and may carry internal indices. For various types of group indices, DIGEN will perform the necessary group traces associated with closed loops. In addition, DIGEN discards disconnected diagrams, and attempts to identify equivalent diagrams, and thereby assigns combinatoric weights. ~~(It tests for the equivalence of two graphs by comparing the characteristic polynomials of their incidence matrices and the alphabetized lists of elements in their incidence matrices.)~~ For most calculations, the use of a computer to generate the contributing diagrams will

be entirely unnecessary and probably unwieldy. However, even for some tree graph calculations (such as the process GG → GGG contributing to large - $p_\perp$ hadron production at $O(\alpha_s^2)$ in QCD), there can be many diagrams, and it is clear that automatic generation of the diagrams, together with their combinatoric and group-theoretic weights, and (when necessary) renormalization counterterms, will be very useful.

$$Li_n(-x) = -\frac{\log^n(x)}{n!} + 2\sum_{k=1}^{[n/2]} \log^{n-2k}(x)\,\frac{(2^{1-2k}-1)\zeta(2k)}{(n-2k)!}$$

$$+ (-1)^{n+1} Li_n(-\tfrac{1}{x})$$

$$\simeq -\frac{\log^n(x)}{n!} \qquad (x \gg 1)$$

$$Li_n(x) \to x \qquad (n \to \infty)$$

$Li_1(x) = -\log(1-x)$

$Li_2(\frac{1}{2}) = \frac{1}{2}(\zeta(2) - \log^2(2)) \simeq 0.582$

$Li_3(\frac{1}{2}) = \frac{7}{8}\zeta(3) + \frac{\log^3(2)}{6} - \frac{\zeta(2)\log(2)}{2} \simeq 0.537$

$\zeta(2) = \frac{\pi^2}{6} \simeq 1.645$

$\zeta(3) \simeq 1.202$

[For further details see:

N. Nielsen, 'Der Eulersche Dilogarithmus und seine Verrallgemeinerungen : eine Monographie,' Halle, Liepzig, 1909; L. Lewin, 'Dilogarithms and associated functions,' Macdonald, 1958; R. Barbieri, J. A. Mignaco and E. Remiddi, Nuovo Cimento 11A, 824 (1972); K. S. Kölbig, J. A. Mignaco, E. Remiddi, BIT 10, 38 (1970).]

Table 2.1: Simple Properties of the Polylogarithm Functions. (Note that $Li_n(x)$ has a branch point at $x = 1$, and is usually taken to have an imaginary part for $x > 1$.)

$$Li_n(x) = \int_0^x \frac{dx_n}{x_n} \cdots \int_0^{x_3} \frac{dx_2}{x_2} \int_0^{x_2} \frac{dx_1}{1-x_1}$$

$$= \frac{(-1)^n}{\Gamma(n-1)} \int_0^1 \frac{\log^{n-2}(t)\log(1-xt)}{t}\, dt \qquad (n \geq 2)$$

$$= \frac{(-1)^n}{\Gamma(n-1)} \left[\frac{\partial^{n-1}}{\partial u^{n-2}\partial v} \frac{1}{u}\, {}_2F_1(v,u;u+1;x)\right]\Big|_{u=v=0}$$

$$= \frac{1}{\Gamma(n)} \int_0^\infty \frac{t^{n-1}}{1-xe^{-t}}\, dt$$

$$= \sum_{k=1}^{\infty} \frac{x^k}{k^n} \qquad (|x| \leq 1)$$

$$= \int_0^x \frac{Li_{n-1}(x_1)}{x_1}\, dx_1$$

$Li_1(x) = -\log(1-x)$

$$Li_2(x) = -\int_0^x \frac{\log(1-t)}{t}\, dt = -\int_0^1 \frac{\log(1-xt)}{t}\, dt$$

$$= \int_0^1 \frac{\log(t)}{t-1/x}\, dt$$

$$Li_n(0) = 0$$

$$Li_n(1) = \zeta(n)$$

$$Li_n(-1) = (2^{1-n}-1)\zeta(n)$$

Section 2 references

2.1 J. Calmet, S. Narison, M. Perrottet and E. de Rafael Rev. Mod. Phys. 49, 21 (1977).

2.2 E. de Rafael and J. Rosner, Ann. Phys. 82, 369 (1974) ($q^2=0$ scheme); ref. [2.3].

2.3 D. M. Ross, A. E. Terrano and S. Wolfram, 'e⁺e⁻ annihilation to $O(\alpha_s^2)$', Caltech preprint in preparation; K. G. Chetyrkin, A. L. Kataev and F. V. Tkachov, Phys. Lett. 85B, 277 (1979); M. Dine and J. Sapirstein, Phys. Rev. Lett. 43, 668 (1979).

2.4 A. J. Buras, 'Asymptotic freedom in Deep Inelastic Processes in leading order and beyond', Fermilab preprint 79/17 THY (1979); E. G. Floratos, D. A. Ross and C. T. Sachrajda, Nucl. Phys. B129, 66 (1977); B139, 545E (1978); B152, 493 (1979); W. A. Bardeen and A. J. Buras; D. W. Duke and T. Muta, Phys. Rev. D18, 3998; W. A. Bardeen and A. J. Buras, Phys. Lett. 86B, 61 (1979).

2.5 S. Wolfram, Talk at 1979 Caltech Workshop on High Energy Physics (Feb. 1979), and in preparation.

2.6 T. Kinoshita and W. B. Lindquist, 'Improving the theoretical prediction of the electron anomalous magnetic moment - progress report', Cornell preprint CLNS-374

(Oct. 1977); 'Eighth Order Magnetic Moment of the electron. I. Second order vertex containing second, fourth and sixth order vacuum polarization subdiagrams', Cornell preprint CLNS-424 (Apr. 1979); 'Eighth order magnetic moment of the electron. II. Fourth order vertices containing second and fourth order vacuum polarization subdiagrams', Cornell preprint CLNS-426 (June 1979).

2.7 e.g. C. M. Bender and S. A. Orszag, 'Advanced Mathematical Methods for Scientists and Engineers', McGraw-Hill 1978, p 381.

2.8 R. S. Van Dyck. et al., Phys. Rev. Lett. 38, 310 (1977). ($g_\mu - 2$ accurate to 10^{-9}: J. Bailey et al., Phys. Lett. 67B, 225 (1977).)

2.9 (e.g.) G. Altarelli, R. K. Ellis and G. Martinelli, 'Large Perturbative Corrections to the Drell-Yan Process in QCD', MIT preprint CTP #776 (Mar. 1979).

2.10 A. A. Vladimirov, JINR preprint E2-11096 (1977) (explicit ultraviolet cutoff); F. M. Dittes, Yu. A. Kubyshin and O. Tarasov, JINR preprint E2-11100 (1977) (R-operation / subtraction at symmetric point); D. I. Kazakov, O. V. Tarasov, A. A. Vladimirov, JINR preprint E2-12249 (1979) ((T)MS scheme).

2.11 A. A. Vladimirov, 'Method for computing renormalization group functions in dimensional renormalization scheme', JINR preprint E2-12248 (1979).

2.12 D. I. Kazakov, O. V. Tarasov and D. V. Shirkov, Teor. Mat. Fiz. 38, 15 (1979); N. N. Khuri, 'The Slope of the Gell-Mann - Low function at the UV fixed point', Rockefeller preprint COO-2232B-169 (1978).

2.13 e.g. ed. J. Schwinger, 'Selected Papers on Quantum Electrodynamics', Dover (1958).

2.14 K. Eriksson, Nuov. Cim. 19, 1029 (1961); 21, 383E (1961).

2.15 $e^- e^-$

2.16 S. I. Eidelman, E. A. Kuraev and V. S. Panin, Nucl. Phys. B148, 245 (1979); F. A. Berends, K. J. F. Gaemers and R. Gastmans, Nucl. Phys. B57, 381 (1973); B63, 381 (1973); R. Gastmans, 'High Energy Tests of Quantum Electrodynamics', Leuven preprint KUL-TF-79/016 (1979). ($e^+e^- \to e^+e^-\mu^+\mu^-$, $e^+e^- \to W^+W^-$, $e^+e^-\pi^+\pi^-$: V. N. Budnev et al., Phys. Rep. 15C, 181 (1975).)

2.17 L. M. Brown and R. P. Feynman, Phys. Rev. 85, 231 (1952).

2.18 S. I. Eidelman and E. A. Kuraev, Nucl. Phys. B143, 353 (1978).

2.19 (4a) B. de Tollis, Nuov. Cim. 32, 757 (1964); 35, 1183 (1965). ($\gamma\gamma \to e\bar{e}e\bar{e}$: T.P. Cheng and T. T. Wu, Phys. Rev. D1, 3414 (1970).)

2.20 Imaginary part:
J. Schwinger, 'Particles, Sources and Fields,' Vol. 2, Addison-Wesley, 1973, p. 407 (also gives result for scalar charged particles);
J. A. Mignaco and E. Remiddi, Nuovo Cim. 60A, 519 (1969).
Real part: G. Källén and A. Sabry, Mat. Fys. Medd. Dan. Vid. Selsk, 29 (1955) No.17;
C. R. Hagen and M. A. Samuel, Phys. Rev. Lett. 20, 1405 (1968).

2.21 T. Kinoshita and W. B. Lindquist, 'Parametric Formula for the Sixth Order vacuum polarization contribution in Quantum Electrodynamics', Cornell preprint CLNS-423 (Apr. 1979).

2.22 A. Sabry, Nucl. Phys. 33, 401 (1962); (Divergent parts: A. Frank, Phys. Rev. 83, 1189 (1951) (explicit cut Π); D. Ross, private communication [illegible]

2.23 R. Barbieri, J. A. Mignaco and E. Remiddi, Nuov. Cim. 11A, 824 (1972); 11A, 865 (1972). [illegible]

2.24 e.g. low-energy ($O(\alpha^{42})$) vacuum polarization: M. Baranger, F. J. Dyson and E. E. Salpeter, Phys. Rev. 88, 680 (1952).

Imaginary part : V. M. Budnev et al., Phys. Rep. 15C, 181 (1975).
Real part : ⟶

98, 1355 (1955).

2.25 e.g. low-energy $\gamma\gamma \to \gamma\gamma$: W. Heisenberg and H. Euler, Z. Phys. 98, 714 (1936). Low energy $\gamma^* \to$ anything : B. Lautrup, Nucl. Phys. B116, 269 (1976).

2.26 e.g. some logs in $g-2$: B. Lautrup and E. de Rafael, Nucl. Phys. B70, 317 (1974).

2.27 $\pi \to l\nu(\gamma)$: T. Goldman and W. J. Wilson, (essential for determination of $\cos\theta_c$) Phys. Rev. D15, 709 (1977). $n \to pe\nu(\gamma)$: A. Sirlin, Phys. 164, 1767 (1967); L. Durand et al., Phys. Rev. 130, 1188 (1963).

$K \to \pi l\nu(\gamma)$: E. S. Ginsburg, Phys. Rev. 142, 1035 (1966); 162, 1570 (1967). $\pi \to \gamma(e\bar{e})$, $\Sigma \to \Lambda(e\bar{e})$ ... : B. E. Lautrup and J. Smith, Phys. Rev. D3, 1122 (1971).

$\eta \to \pi^0 e\bar{e}$: C. H. Llewellyn Smith, Nuovo Cim. 48, 834 (1967). $\pi^0 \to e\bar{e}$: S. Wolfram, Nucl. Phys. B117, 109 (1976).

2.28 T. Kinoshita and A. Sirlin, Phys. Rev. 113, 1652 (1959); N. Cabibbo, G. Corbo and L. Maiani, Nucl. Phys. B155, 93 (1979); W. J. Marciano, G. C. Marques and N. Papanicolaou, Nucl. Phys. B96, 237 (1975); R. E. Marshak, Riazuddin and C. P. Ryan, 'Theory of Weak Interactions', Wiley 1969, p219.

2.29 e.g. D. A. Ross, Nuovo Cim. 10A, 475 (1972). (5a)

2.30 D. Yu Bardin et al., Sov. J. Nucl. Phys. 15, 161 (1972); M. Duong Van, T. Goldman and J. Sapirstein, in preparation.

2.31 D. A. Ross, Nucl. Phys. B51, 116 (1973). (5b)

2.32 W. Marciano, Phys. Rev. D12, 3861 (1975).

2.33 W. E. Caswell, G. P. Lepage and J. Sapirstein, Phys. Rev. Lett. 38, 488 (1977). (Decay $(e^+e^-) \to \gamma^* \to \gamma\gamma\gamma$ via heavy fermion loop: M. Pascual and E. de Rafael, Lett. Nuov. Cim. 4, 1144 (1970). $(e^+e^-) \to \gamma\gamma e\bar{e}$: Z. Kunzst and H. R. Rubinstein, Phys. Lett. 80B, 129 (1978).)

Lowest order calculation: A. Ore and J. L. Powell,

Corrections to e polarization: V. P. Kuznetsov, JETP 12, 1202 (1961).

5b

Correction to weak q decays without QCD:
↳ A. Sirlin, Phys. Rev. Lett. 32, 966 (1974)

Phys. Rev. 75, 1696 (1949). Real part of lowest-order amplitude squared: ref. [2.37].

2.34 I. Harris and L. W. Brown, Phys. Rev. 105, 1656 (1957); ref. [2.18]; ↑ Real part of amplitude: ref [2.36]

Y. Tomozawa, 'Radiative corrections to parapositronium decay', Michigan preprint UM-HE 79-17 (1979).

2.35 e.g. B. E. Lautrup, A. Peterman and E. de Rafael, Phys. Rep. 3C, 193 (1972).

2.36 V. K. Cung, A. Devoto, T. Fulton and W. W. Repko, Phys. Lett. 78B, 116 (1978).

2.37 V. K. Cung et al., Phys Lett 68B, 474 (1977).

2.38 e.g. ref. [2.35]. Lamb shift

2.39 e.g. J. M. Jauch and F. Rohrlich, 'The Theory of Photons and Electrons', 2nd ed., Springer 1976: p. 345.

2.40 e.g. ref. [2.39], p 330. ($eV \to e\mu\bar{\mu}$: Y. Yamamoto, Prog. Theor. Phys. 27, 233 (1962).)

2.41 J. D. Bjorken, S. D. Drell, and S. Frautschi, Phys. Rev. 112, 1409 (1958); N. Parsons, Phys. Rev. 150, 1165 (1966); P. Hulđ, Phys. Lett. 24B, 185 (1967).

2.42 Ref. [2.39] p. 383; T. P. Cheng and T. T. Wu, Phys. Rev. D2, 2444 (1970).

2.43 e.g. $e^-e^- \to e^-e^-$: P. Stehle, Phys. Rev. 110, 1458 (1958).

2.44 e.g. H. D. Politzer, Phys. Rep. 14C, 129 (1974).

2.45 W. Caswell, Phys. Rev. Lett. 33, 244 (1974); D. R. T. Jones, Nucl. Phys. B75, 531 (1974).

2.46 M. Baker and K. Johnson, Phys. Rev. 183, 1292 (1969).

2.47 S. L. Adler, Phys. Rev. D5, 3021 (1972).

2.48 D. V. Nanopoulos and D. A. Ross, Nucl. Phys. B, to be published.

2.49 e.g. S. Wolfram, 'Jet Development in leading log QCD', Caltech preprint CALT-68-740 (Sept. 1979).

2.50 G. Altarelli, R. K. Ellis, L. Maiani, R. Petronzio, Nucl. Phys. B88, 215 (1975).

2.51 e.g. G. C. Fox and S. Wolfram, Nucl. Phys. B149, 413 (1979). Acoplanarity due to $e^+e^- \to q\bar{q}GG$, $q\bar{q}q'\bar{q}'$: A. Ali et al., 'Four-jet production in e^+e^- annihilation', DESY preprint 79/03 (Jan 1979).

2.52 G. C. Fox and S. Wolfram, 'A Gallimaufry of Event Shapes in e^+e^- Annihilation', Caltech preprint CALT-68-723 (August 1979).

2.53 W. A. Bardeen and A. J. Buras, 'Higher-Order Asymptotic freedom corrections to photon-photon scattering', Fermilab preprint THY 78/91.

2.54

2.55 c.f. T. D. Lee and M. Nauenberg, Phys. Rev. 133, B1549 (1964).

2.56 ref. [2.9]; R. D. Field and R. P. Feynman, unpublished.

2.57 Numerical: J. Kripfganz and A. P. Contogouris, 'The Quark-Quark Correction to the Drell-Yan Formalism', McGill preprint (1979); Analytical: A. N. Schellekens and W. L. van Neerven, 'Calculation of the quark-quark contribution to the Drell-Yan process', Nijmegen preprint THEF-NYM-79.8.

2.58 B. L. Combridge, J. Kripfganz and J. Ranft, Phys. Lett. 70B, 234 (1977); R. Cutler and D. Sivers, Phys. Rev. D17, 196 (1978). (Applications: R. D. Field, Phys. Rev. Lett. 40, 997 (1978).)

2.59 Z. Kunszt and E. Pietarinen, 'Production of three large P_T jets in hadron-hadron collisions', DESY preprint 79/34 (June 1979); T. Gottschalk and D. Sivers, 'Basic processes and formalism for the hadronic production of three large-P_T jets', Argonne preprint 79/07 (March 1979) and Erratum; T. Gottschalk, E. Monsay and D. Sivers, 'Process dependence of perturbative QCD jet-structures', Argonne preprint 79-15 (July 1979).

2.60 Z. Kunszt, E. Pietarinen and E. Reya, 'Transverse momenta of hadronically produced heavy quark systems: $2 \to 3$ processes in QCD', DESY preprint 79/28 (May 1979).

2.61 R. Barbieri, G. Curci, E. d'Emilio and E. Remiddi, Nucl. Phys. B154, 535 (1979).

2.62 A. E. Terrano and S. Wolfram, Caltech preprint in preparation.

2.63 A. De Rujula, R. Petronzio, and B. Lautrup, Nucl. Phys. B146, 50 (1978).

2.64 e.g. L. Baulieu, J. Ellis, M. K. Gaillard and W. J. Zakrzewski, 'Non-perturbative QCD effects at large momentum transfers', CERN preprint TH 2482 (1978); T. Appelquist and R. Shankar, 'A General Approach to the computation of instanton effects', Yale preprint 1978.

2.65 M. K. Gaillard and B. W. Lee, Phys. Rev. D10, 897 (1974).

2.66 ref. [2.65]; R. Schrock, Talk at 1979 Caltech workshop on High Energy Physics (Feb. 1979).

2.67 J. Kiskis, Phys. Rev. D8, 2129 (1974); R. Barlow and S. Wolfram, Phys. Rev. D to be published.

2.68 Fujikawa, B. W. Lee and A. I. Sanda, Phys. Rev.

2.69 R. Jackiw and S. Weinberg, Phys. Rev. D5, 2396 (1972).

2.70 G. Passarino and M. Veltman,

2.71 T. J. Goldman and D. A. Ross, Phys. Lett. (1979);
D. A. Ross, Nucl. Phys. (1978)

e.g. R. Jackiw, Phys. Rev. D9, 1686 (1974).

3. Gamma Matrix Manipulation and GAMALG

3.1 Introduction and Basic Manipulations

In this section I discuss what is probably the most mundane feature of Feynman diagram evaluations: the manipulation of Dirac gamma matrices, and I describe in some detail the package GAMALG that A. E. Terrano and I have written in MACSYMA to perform these manipulations. Many examples are given, and the discussion is intended to be sufficiently detailed to provide the information necessary to make use of most features of GAMALG. The basic notations, functions and flags in GAMALG are summarized in Tables 3.1, 3.2 and 3.3, which also indicate its basic capabilities. Note that Section 3.7 gives a comparison between GAMALG and some other gamma matrix manipulators available. It should be pointed out that this section contains a number of results on gamma matrix manipulation that are apparently not well known. Even those who do not use computers may, therefore, find some assistance in their manual calculations below. Most of the particularly powerful results are valid only in n = 4 dimensions and are summarized in Table 3.4.

[the results... are..., not: the number ... is]

GAMALG presently resides in the MIT-MC computer. Some information on its use there is given in the Appendix to this section.

The Feynman amplitude corresponding to fermion lines in a diagram consists of a product of gamma matrices (represented by G( ) in GAMALG). For each internal segment of the fermion line, along which a fermion with mass m carries momentum p, there is associated a 'propagator factor' $1/(\not{p}-m) = (\not{p}+m)/(p^2-m^2)$ [= ZD(p,m) in GAMALG notation], where the 'slashed vector' $\not{p} \equiv \gamma^\mu p_\mu$, and γ_μ is a vector of Dirac gamma matrices. The emission of a vector boson (e.g., a photon) from a fermion line (represented by a three-node in the graph) gives a factor $\not{\varepsilon}$, where ε is the polarization vector of the emitted boson. Typically one sums over the polarization states of the boson, using $\sum_{\text{pols}} \varepsilon_\mu \varepsilon^*_\nu = T_{\mu\nu}$, where

$T_{\mu\nu}$ is a tensor usually determined by the gauge chosen, and depending on the metric tensor $g_{\mu\nu}$, the momentum (say, k_μ) of the boson and sometimes (in axial gauges) a further vector (say, n_μ). In Feynman gauge $T_{\mu\nu} = g_{\mu\nu}$ (where I have removed a spurious (-1) factor from the more usual definition). The gamma matrices which appeared in the $\not{\epsilon}$ factors for the emission and absorption of the boson in this case are constrained to have the same spacetime index, which must be summed over, so that the relevant gamma matrices may both be written just as, say, γ_α [= AL, in GAMALG notation]. This possibility often results in great computational simplifications. It is usually necessary to sum over the polarization states of fermions appearing in Feynman diagrams; this operation corresponds to taking the trace (sometimes called 'spur' and represented by TR( ) or GT( ) in GAMALG) of the product of gamma matrices representing the fermion line, with respect to the internal indices of the gamma matrices.

For tree diagram calculations in which no infrared divergences appear, it is always entirely sufficient to consider gamma matrices in four spacetime dimensions (so that the spacetime index μ which labels the γ_μ runs from 0 to 3). This was the case treated by existing gamma matrix manipulators (see Section 3.7). However, in calculations of loop graphs or of infrared divergent tree graphs, it is usually exceedingly convenient to use the method of dimensional regularization (to be discussed further in Section 4.1 of the next installment), for which one must use gamma matrices generalized to n spacetime dimensions (so that essentially the index μ on γ_μ now runs from 0 to n-1).

The algebra of gamma matrices in n spacetime dimensions may be defined by

$$\{\gamma_\mu, \gamma_\nu\} = 2\, g_{\mu\nu} \qquad (3.1.1a)$$

$$g^{\mu}_{\mu} = n \qquad (3.1.1b)$$

$$\mathrm{Tr}[1] = 4 \qquad (3.1.1c)$$

$$\mathrm{Tr}[\gamma_{\mu_1}\gamma_{\mu_2}\ldots\gamma_{\mu_k}] = 0 \qquad (k \text{ odd}). \qquad (3.1.1d)$$

One might expect that Tr[1] should be generalized to be some function of n. A suitable generalization might be $\mathrm{Tr}[1] = 2^{n/2}$, which is the dimensionality of the spinor representation of the Lorentz group in n dimensions. (This is the extrapolation based, as it presumably should be, on the group SO(2d); the SO(2d+1) groups have a somewhat different structure and would give $\mathrm{Tr}[1] = 2^{(n-1)/2}$.) However, all nonvanishing traces may be reduced by successive application of (3.1.1a) and (3.1.1b) to Tr[1] multiplied by some combination of scalar (dot) products of vectors. If the gamma matrix trace arose from an average over the polarization states of an incoming or outgoing fermion in a diagram, then the average is presumably obtained by dividing the trace by Tr[1], so that any generalization of Tr[1] should cancel out. However, when the trace is associated with the sum over polarizations in an internal fermion loop, Tr[1] will appear in the final result. 1/(n-4) terms arising from loop momentum integrations can reveal O(n-4) terms in Tr[1]. Hence, the results for higher-order Feynman diagrams can depend on the generalization of Tr[1]. Often such generalizations are constrained by the preservation of invariances or by self-consistency requirements, but this does not appear to be so for Tr[1]. Therefore, Tr[1] must be fixed by convention; results for single processes will depend on its value, but suitable physical comparisons will be independent of the choice, as mentioned in Section 2.3. GAMALG always takes Tr[1] = 4.

A further possible ambiguity in the generalization of the gamma matrices to $n \neq 4$ dimensions concerns traces of products of odd numbers of gamma matrices, which vanish when $n = 4$. One may be concerned that, for example, when $n = 3$, the gamma matrices become the Pauli σ matrices, and the trace of a product of three σ matrices does not vanish but is rather proportional to a three-dimensional epsilon symbol. In fact, in any odd number of dimensions, such epsilon symbols appear in traces. They are a direct indication of the different nature of the rotation groups in even and odd numbers of dimensions (e.g., "γ_5" is not distinguished from the identity in SO(2d+1)). There are, therefore, two distinct extrapolations of the gamma matrices, associated with even and odd numbers of dimensions (the extrapolations coincide for vectors but differ for spinors). Since we are concerned with extrapolation away from $n = 4$, we may plausibly choose the "SO(2d) extrapolation" for which eq. (3.1.1d) is valid. Note that the behavior of fermions would be somewhat different in odd numbers of dimensions (and in the "SO(2d+1) extrapolation"), and for example, the Dirac Lagrangian would apparently not be CPT invariant in the usual sense [3.1]. GAMALG always sets the trace of an odd number of gamma matrices to zero.

The traces of products of gamma matrices which appear in diagram calculations usually contain three basic classes of terms. First, there are 'slashed vectors', such as $\not{P} \equiv \gamma^\mu P_\mu$. In some cases it is convenient to perform the contraction of indices between a gamma matrix and a vector only after the trace is performed. A typical case in which this is useful is when the vector in question is the polarization vector for a vector boson. The second class of terms is gamma matrices which appear in a trace uncontracted with a vector (termed 'uncontracted indices'). These may usually be treated just like slashed vectors. However, if the eventual contraction is to be done with a gamma

matrix in another trace, then, at least when n = 4, it is convenient to use the contraction to combine the traces in which the identical gamma matrices appear. For example, $Tr[\gamma_\mu \not{p}]Tr[\gamma_\mu \not{p}' \not{q} \gamma_\nu]$ can be reduced to $2(Tr[\not{p}\not{p}' \not{q}\gamma_\nu] + Tr[\not{p}\gamma_\nu \not{q}\not{p}'])$ if n = 4. The third basic class of terms which occur in traces are 'contracted indices'. These are pairs of gamma matrices in a single trace whose indices are contracted (e.g., the γ_μ in $Tr[\not{p}\gamma_\mu \not{q}\gamma_\mu]$). The presence of contracted indices coming from internal vector boson lines in most traces arising from Feynman diagrams allows considerable simplification in practical gamma matrix manipulations. Of course, this third class may be considered as a special case of the second class, in which the two contracted indices appear in the same trace.

In GAMALG, traces are denoted by $TR(a_1, a_2, \ldots a_k)$, where in the a_i slashed vectors are represented by the names of the vectors and gamma matrices are represented by their indices. For example, $Tr[\not{p}\gamma_\mu \not{q}\gamma_\nu \gamma_\alpha \not{q}\gamma_\alpha]$ is represented as TR(p,mu,q,nu,al,q,al). Indices which are contracted must be declared using CIND to distinguish them from slashed vectors which appear twice in the trace. Hence the α above must be declared by CIND(al). No harm will be done by also declaring μ and ν as indices: CIND(mu,nu); they will be treated just like slashed vectors in the trace, since they appear only once. A product of gamma matrices whose trace is not taken is denoted in GAMALG by $G(a_1, a_2, \ldots, a_k)$ where the a_i are as for traces. Note that 'list brackets' (i.e., [and]) appearing around some of the a_i in G or TR will be ignored, so that, for example, TR([p,mu,q], [nu,aℓ,q],aℓ) is entirely equivalent to TR(p,mu,q,nu,aℓ,q,aℓ). It may be convenient to place some of the a_i in lists, since these may then be given names and manipulated using standard MACSYMA functions. For example, setting ℓ1:[p,mu,q], ℓ2:[nu,aℓ,q], TR(ℓ1, ℓ2) will yield TR(p,mu,q,nu,aℓ,q), while TR(reverse(ℓ1),ℓ2) gives TR(q,mu,p,nu,aℓ,q).

ordinary ℓ's

The output for traces from GAMALG is in the form of dot products of vectors (e.g., p·q, represented by D(p,q)), components of vectors (e.g., p_μ, represented by D(p,mu)) and metric tensors (e.g., $g_{\mu\nu}$, represented by D(mu,nu)). In all cases one may consider that the 'pseudofunction' D(a,b) represents the dot product of the two vectors a and b, since, if either a or b is an 'index', it may be considered to be the basis vector along the direction specified by the 'index'. The commutativity of dot products is essential for the simplification of expressions involving them. It is, however, convenient not to use this commutativity directly, but rather simply to adopt a standard form for dot products. We choose to standardize all dot products by writing the vectors appearing in the pseudofunction D in the alphabetical order of the symbols which represent them. Hence, unless the flag DOF : FALSE, the arguments of all D's will be alphabetized when they are generated, so that, for example, both D(p,q) and D(q,p) will become D(p,q). In addition, sums of vectors, such as p + k/2 or x* p + q may appear as arguments of D's. Scalars such as x in this example must be declared as such using SCALS(x_1,x_2,...x_k) so that they are not confused with vectors. (The scalar property may be removed from a set of variables by doing UNSCALS(x_1,x_2,...x_j).) Dot products involving sums of vectors will be expanded out automatically when, for example, a trace is performed, unless the flag DEF : FALSE. An expression containing dot products of sums of vectors may in any case be expanded by doing DFIX(exp). For example

```
(C2) D(A1+A2,A3+A4);
```

$$\text{(D2)} \qquad D(A2 + A1, A4 + A3)$$

```
(C3) DFIX(%);
```

$$\text{(D3)} \qquad D(A2, A4) + D(A2, A3) + D(A1, A4) + D(A1, A3)$$

```
(C4) EXP:D(P+K/2,Q-K/2)+2/S*D(P,Q+P)*D(K,Q);
```

$$\text{(D4)} \qquad \frac{2\,D(K, Q)\,D(P, Q + P)}{S} + D\left(P + \frac{K}{2}, Q - \frac{K}{2}\right)$$

```
(C5) DFIX(EXP);
```

$$\text{(D5)} \qquad \frac{2\,D(K, Q)\,(D(P, Q) + D(P, P))}{S} + D(P, Q) + \frac{D(K, Q) - \frac{D(K, K)}{2}}{2} - \frac{D(K, P)}{2}$$

```
(C6) SCALS(X1,X2);
```

$$\text{(D6)} \qquad [X1, X2]$$

```
(C7) EXP2:D(Q+X1*P,X2*P-K/2);
```

$$\text{(D7)} \qquad D\left(Q + X1\,P, X2\,P - \frac{K}{2}\right)$$

```
(C8) DFIX(%);
```

$$\text{(D8)} \qquad X2\,D(P, Q) + X1\left(X2\,D(P, P) - \frac{D(K, P)}{2}\right) - \frac{D(K, Q)}{2}$$

The number of terms generated directly at intermediate stages in performing, for example, a trace, will be smaller if the dot products are not expanded out until the end of the calculation. This procedure is applied if BTR, rather than TR, is used for a trace. Note that the D representation of dot products is also ~~used in VIPER (see Section 4).~~ by the programs in the rest of this series.

It is often convenient to replace dot products in a result by scalar (perhaps Mandelstam) variables, for example, D(p,q) → s/2. Such substitutions are defined by doing, e.g., KINDEF(D(p,q) = s/2, D(p,pp) = t/2-mA2,...). In this case (unless the flag DSIM : FALSE), whenever D(p,q) is generated, it

will be replaced by s/2. If certain dot products are defined by KINDEF to be zero, then the number of intermediate terms in the calculation of, for example, a trace, may be smaller, since various dot products will disappear when they are expanded. Replacements for dot products may be removed by doing UNKINDEF(dotp1,....dotpk). For example, UNKINDEF(D(p,q)) will remove any substitutions for D(p,q) previously declared. KINS is the list of substitutions declared by KINDEF. An example of the use of KINDEF is

```
(C2) KINDEF(D(P,Q)=S/2,D(P,P)=M^2);
                                       S            2
(D2)                       [D(P, Q) = -, D(P, P) = M ]
                                       2

(C3) EXP:D(P,Q)*D(P,P)-D(Q,Q)^3/D(P,Q);
                                                3
                                               D (Q, Q)
(D3)                        D(P, P) D(P, Q) - --------
                                               D(P, Q)

(C4) DFIX(%);
                                   2        3
                                  M  S   2 D (Q, Q)
(D4)                              ---- - ----------
                                   2         S

(C5) TR(P,Q);
(D5)                                     2 S
```

Further examples may be found in the calculations given in Section 3.8.

In most Feynman diagram calculations, it is convenient to consider only complete, covariant vectors rather than their components. However, GAMALG does include a facility for assigning particular components to vectors. The command COMPDEF(vec1=[$vec1_o$,$vec1_1$,$vec1_2$,$vec1_3$], ind1=val, vec2=...., vec3=...., ...) assigns components to the vectors vec1, etc., and gives definite values to indices (e.g., μ set to 3). NONCOV(exp) simplifies an expression exp in terms of dot products using, where possible, the components defined by COMPDEF. The component assignments defined by COMPDEF are in the list COMPS; they may

be removed using UNCOMPDEF. An example of the use of COMPDEF and NONCOV is

```
(C2) COMPDEF(Q=[E,0,0,0],P=[X*E/2,0,0,X*E/2],MU=0,NU=3);
                                        E X         E X
(D2)      [Q = [E, 0, 0, 0], P = [----, 0, 0, ----], MU = 0, NU = 3]
                                         2           2

(C3) NONCOV(D(P,P)+D(P,Q));
                                          2
                                         E  X
(D3)                                     ----
                                          2

(C4) CIND(MU,NU);
(D4)                                  [NU, MU]

(C5) TR(P,MU,Q,NU);
(D5)  - 4 D(MU, NU) D(P, Q) + 4 D(MU, P) D(NU, Q) + 4 D(MU, Q) D(NU, P)

(C6) NONCOV(%);
                                           2
(D6)                                  2 E  X
```

Note that the metric is usually assumed to have signature (+---), but may be changed by altering METLIS. Of course, all manipulations performed by NONCOV are valid only when n = 4.

When n = 4 (and to some extent when n ≠ 4), GAMALG can perform traces involving γ_5 (G5), as discussed in Section 3.5. These often result in anti-symmetric products or epsilon symbols (sometimes known as Levi-Civita symbols or alternating tensors) of four vectors, which are represented in GAMALG by $\mathrm{EPS}(p_1,p_2,p_3,p_4) = \epsilon_{\mu_1\mu_2\mu_3\mu_4} p_1^{\mu_1} p_2^{\mu_2} p_3^{\mu_3} p_4^{\mu_4}$. Unless the flag EPSOF : FALSE, all epsilon symbols generated by GAMALG will be put into the standard form in which their arguments appear in alphabetical order. This operation introduces a factor of the signature of the permutation required to get from the initial to the final ordering of the arguments. If two or more arguments of an epsilon sumbol generated are identical then it will automatically be set to zero, as required by the antisymmetry of the epsilon symbol. If the arguments of an epsilon symbol generated are sums of vectors, then (unless EPSEF : FALSE) it will be expanded out as a sum of epsilon symbols. This expansion may be performed

on the epsilon symbols in an expression by doing EPSFIX(exp). For example,

```
(C2) EPS(A2,A1,A3,A4);
(D2)                       EPS(A2, A1, A3, A4)

(C3) EPSFIX(%);
(D3)                     - EPS(A1, A2, A3, A4)

(C4) EPS(P+Q,Q+K/2,PP,QP-P);
                                        K
(D4)                      EPS(Q + P, Q + -, PP, QP - P)
                                        2

(C5) EPSFIX(%);
                                EPS(K, PP, Q, QP) - EPS(K, P, PP, Q)
(D5) - EPS(P, PP, Q, QP) + ---------------------------------------
                                                2

                                                          EPS(K, P, PP, QP)
                                                        - -----------------
                                                                  2
```

An untraced product of gamma matrices is represented in GAMALG by G(a_1, a_2,....a_k). Of course, this product is not simply commutative, and the problem of converting it to a canonical form is non-trivial. Some partial solutions are discussed in Section 3.4. If a term in the product (i.e., an argument of the G) consists of a sum of slashed vectors, then it may be expanded out using GFIX. For example,

```
(C2) EXP:G(P+K/2,MU,Q-R/4,P+Q);
                                 K            R
(D2)                      G(P + -, MU, Q - -, Q + P)
                                 2            4

(C3) GFIX(%);
       G([P, MU, R, Q])   G([P, MU, R, P])
(D3) - ---------------- - ---------------- + G([P, MU, Q, Q]) + G([P, MU, Q, P])
              4                  4

       G([K, MU, R, Q])   G([K, MU, R, P])   G([K, MU, Q, Q])   G([K, MU, Q, P])
     - ---------------- - ---------------- + ---------------- + ----------------
              8                  8                  2                  2
```

Clearly the GFIX expansion of a G product of any complexity will generate a very large number of terms. GFIX should, therefore, be used sparingly.

Another expansion performed internally by GAMALG on some occasions is of $\not{p} + m$ or $1/(\not{p}-m)$ terms (represented by ZN(p,m) and ZD(p,m), respectively) in gamma matrix products or traces. The command ZFIX(exp) will expand all such terms in exp, setting to zero traces (represented by GT's) of odd numbers of gamma matrices.

3.2 Contracting Tensors and Combining Traces

GAMALG can manipulate tensors and performs contractions of indices in $n \neq 4$ dimensions. The representation of tensors in GAMALG was described above. For example, $q^2 q_{\mu\nu} - q_\mu q_\nu$ is represented by D(q,q)* D(mu,nu) - D(q,mu) * D(q,nu). The function CON(exp) contracts out any free pairs of indices in exp which were declared using CIND. For example,

```
(C2) CIND(MU,NU,AL);
(D2)                                  [AL, NU, MU]

(C3) A1:D(Q,Q)*D(MU,NU)-D(Q,MU)*D(Q,NU);
(D3)                  D(MU, NU) D(Q, Q) - D(Q, MU) D(Q, NU)

(C4) CON(A1^2);
                                      2           2
(D4)                             N D (Q, Q) - D (Q, Q)

(C5) A2:D(MU,AL)*D(P+K,AL)+D(MU,P-K);
(D5)                  D(MU, AL) D(P + K, AL) + D(MU, P - K)

(C6) CON(A1*A2);
(D6) D(NU, P + K) D(Q, Q) + D(NU, P - K) D(Q, Q) - D(NU, Q) D(P + K, Q)

                                                        - D(NU, Q) D(P - K, Q)
```

CON will also perform contractions on expressions involving untraced products of gamma matrices (represented by $G(a_1,a_2,....,a_k)$), as in

```
(C7) G1:G(P,MU,Q,NU);
(D7)                                G(P, MU, Q, NU)

(C8) CON(A1*G1);
(D8)                 G(P, NU, Q, NU) D(Q, Q) - G(P, Q, Q, Q)
```

In four dimensions (n = 4), there is a particularly useful set of relations which allows products of gamma matrix traces to be combined into single traces. I shall denote an arbitrary product (string) of gamma matrices by S and products of even and odd numbers of gamma matrices (not counting any γ_5) by E and O, respectively. Then two traces containing indices contracted between them (as from part or all of a vector boson polarization sum) can be combined into a single trace by using the identity [3.2, 3.3]

$$\begin{aligned} \mathrm{Tr}[\gamma_\mu O_1]\mathrm{Tr}[\gamma_\mu O_2] &= 2\ \mathrm{Tr}[(O_1+(O_1)_R)O_2] \\ &= 2\{\mathrm{Tr}[O_1O_2] + \mathrm{Tr}[(O_1)_R O_2]\} \qquad (3.2.1) \\ &= 2\ \mathrm{Tr}[O_1(O_2+(O_2)_R)] \end{aligned}$$

where $(S)_R$ is the reversal of the string S (product of factors in reverse order). The result (3.1) is proved by the methods used to derive the Chisholm identities described in Section 3.4 and is not readily generalizable to $n \neq 4$ dimensions. Of course, if the odd strings in (3.2.1) were instead even strings, then the traces would vanish. Results when more than one index is contracted between the traces may be obtained easily as corollaries to (3.2.1). The relation (3.2.1) is applied to expressions containing products of G's by doing COTR(exp). For example,

```
(C2) NSET(4)$
DIMENSIONS = 4
(D2)                                    4

(C3) CIND(MU,NU)$
(D3)                                 [NU, MU]

(C4) G(MU,P+K,NU,Q)*G(MU,R)$
(D4)                      G(MU, P + K, NU, Q) G(MU, R)

(C5) COTR(%)$
(D5)            2 (GT(Q, NU, P + K, R) + GT(P + K, NU, Q, R))
```

The GT's, rather than G's in the result signify that the simplifications hold only for traces. Note that any G's or GT's in an expression are converted to TR's and evaluated by doing CGT(exp).

In traces, there often appear terms of the form $\not{p} + m$ (where m is implicitly multiplied by the 4 x 4 identity matrix, rather than by gamma matrices), usually arising from massive spin $-\frac{1}{2}$ fermion propagators, and represented in GAMALG by ZN(p,m). Usually, the $\not{p}$ and m must be treated separately, and the trace in which they occur decomposed into a sum of other traces, in which only one of the $\not{p}$ or m appears. Such expansions are carried out by ZFIX. However, in n = 4 dimensions, there are several manipulations for which terms of the form $\not{p} + m$ may be treated as a unit. For example, if $\underline{0}_i$ denotes a product of gamma matrices containing $\not{p} + m$ terms, then [3.4]

$$\mathrm{Tr}[\gamma_\mu \underline{0}_1]\mathrm{Tr}[\gamma_\mu \underline{0}_2] = \mathrm{Tr}[(\underline{0}_1+(\underline{0}_1)_R)(\underline{0}_2+\bar{\underline{0}}_2)], \qquad (3.2.2)$$

where $\bar{\underline{0}}_i$ denotes the product $\underline{0}_i$ with the sign of each in term reversed (so that $\not{p} + m \to \not{p} - m$).

The relations (3.2.1) and (3.2.2) are particularly convenient because they avoid the necessity for explicit contraction of indices in the (usually lengthy) expressions in terms of dot products which result from performing

each trace individually. Another set of relations also exists for combining traces in n = 4 dimensions containing just one identical slashed vector [3.3]:

$$\mathrm{Tr}[\not{p}O_1]\mathrm{Tr}[\not{p}O_2] = \mathrm{Tr}[(O_1+(O_1)_R)(p^2O_2+\not{p}O_2\not{p})]. \qquad (3.2.3)$$

This may be proved trivially using the result (3.3.4a). The relation (3.2.1) may be considered a special case of (3.2.3), obtained by taking p to be a unit vector lying successively along each of the four coordinate axes, and summing the contributions (using the result (3.3.6a) $\gamma_\alpha O_2 \gamma_\alpha = -2\ O_2$). The identity (3.2.3) as such does not, however, appear to be particularly useful and has not been implemented in GAMALG. In most cases, its application would serve ~~simply~~ merely to complicate results.

The identity (3.2.1) is for a product of two traces sharing a contracted index. A similar relation holds in n = 4 dimensions for the product of a trace and a string of untraced gamma matrices [3.5]:

$$\gamma^\mu S_1 \mathrm{Tr}[\gamma_\mu E_2] = 2(E_2+(E_2)_R)S_1. \qquad (3.2.4)$$

Multiple applications of this result may be performed by an algorithm [3.5] analogous to the one devised by Kahane [3.6] for simplifying entirely untraced products of gamma matrices. I have, however, not had occasion to use eq. (3.2.4), and it has presently not been implemented in GAMALG.

When n = 4, the function CON will also perform contractions involving epsilon symbols. An example of this is

```
(C2) NSET(4)$
DIMENSIONS = 4

(C3) CIND(MU1,MU2,MU3,MU4);
(D3)                          [MU4, MU3, MU2, MU1]

(C4) D(MU1,MU2)*EPS(P1,MU1,P2,MU2);
(D4)                  D(MU1, MU2) EPS(P1, MU1, P2, MU2)

(C5) CON(%);
(D5)                                  0

(C6) EPS(P1,MU1,P2,MU2)*EPS(P3,MU2,MU1,P4);
(D6)             EPS(P1, MU1, P2, MU2) EPS(P3, MU2, MU1, P4)

(C7) CON(%),FACTOR;
(D7)          - 2 (D(P1, P3) D(P2, P4) - D(P1, P4) D(P2, P3))

(C8) EPS(MU1,MU2,MU3,MU4)*EPS(MU1,MU3,MU2,MU4);
(D8)           EPS(MU1, MU2, MU3, MU4) EPS(MU1, MU3, MU2, MU4)

(C9) CON(%);
(D9)                                  24
```

Contractions involving epsilon symbols are performed by first converting them to traces using

$$\epsilon_{\mu_1\mu_2\mu_3\mu_4} a_1^{\mu_1} a_2^{\mu_2} a_3^{\mu_3} a_4^{\mu_4} = \frac{1}{4i} \mathrm{Tr}[\not{a}_1 \not{a}_2 \not{a}_3 \not{a}_4 \gamma_5], \qquad (3.2.5)$$

where the a_i may be basis vectors or 'indices', and then combining the traces containing contracted indices by application of the identity (3.2). The final answer is obtained by performing the resulting traces. In Minkowski spacetime, the elements of the raised and lowered-index epsilon symbols differ by a factor (-1) (the product of the diagonal elements of the metric tensor; see e.g., [3.7]). Unless the flag EUCLID : TRUE, GAMALG will, therefore, automatically insert a factor (-1) for each contracted pair of epsilon symbols. The question of whether any remaining epsilon symbol is upper or lower is left to the users' interpretation.

Products of traces in n = 4 dimensions which contain γ_5 may be combined by using the relation [3.3]

$$\mathrm{Tr}[\gamma_5 E_1]\mathrm{Tr}[\gamma_5 E_2] = 2\,\mathrm{Tr}[E_2(E_1+(E_1)_R)] - \mathrm{Tr}[E_1]\mathrm{Tr}[E_2], \qquad (3.2.6)$$

which is also applied by COTR and may be proved from the result [3.3, 3.5]

$$E + E_R = \frac{1}{2}\,(\mathrm{Tr}[E] + \gamma_5\mathrm{Tr}[\gamma_5 E]). \qquad (3.2.7)$$

If E_1 and E_2 do not share contracted indices, then the traces on the left-hand side of (3.2.6) cannot be combined using (3.2.1), and each will involve epsilon symbols. However, the product of any two epsilon symbols can always be reduced to dot products, since the product of two pseudotensors must be expressible in terms of ordinary tensors (and hence the metric tensor). The necessary reduction of epsilon symbol products is carried out by CON using the special case of (3.2.6). (Note, as always, the Minkowski space sign):

$$(\varepsilon_{\mu_1\mu_2\mu_3\mu_4}{}^{\mu_1\mu_2\mu_3\mu_4}\, a_1^{\mu_1} a_2^{\mu_2} a_3^{\mu_3} a_4^{\mu_4})(\varepsilon^{\nu_1\nu_2\nu_3\nu_4}\,(b_1)_{\nu_1}(b_2)_{\nu_2}(b_3)_{\nu_3}(b_4)_{\nu_4})$$

$$= -\frac{1}{16}\,\mathrm{Tr}[\gamma_5 \not{a}_1\not{a}_2\not{a}_3\not{a}_4]\mathrm{Tr}[\gamma_5 \not{b}_1\not{b}_2\not{b}_3\not{b}_4]$$

$$= -\frac{1}{16}\,\{2\,\mathrm{Tr}[\not{b}_1\not{b}_2\not{b}_3\not{b}_4(\not{a}_1\not{a}_2\not{a}_3\not{a}_4+\not{a}_4\not{a}_3\not{a}_2\not{a}_1)]$$

$$- \mathrm{Tr}[\not{a}_1\not{a}_2\not{a}_3\not{a}_4]\mathrm{Tr}[\not{b}_1\not{b}_2\not{b}_3\not{b}_4]\} \qquad (3.2.8)$$

$$= -\begin{vmatrix} a_1.b_1 & a_1.b_2 & \cdots & a_1.b_4 \\ \vdots & & & \vdots \\ a_4.b_1 & \cdots & & a_4.b_4 \end{vmatrix}$$

$$\equiv -\,\mathrm{Det}(a_i.b_j).$$

3.3 Evaluation of Traces

From the definitions (3.1.1), one may formally write down a recursion relation for the trace of k gamma matrices (e.g., [3.8, 3.9])

$$\mathrm{Tr}[\not{a}_1\not{a}_2\ldots\not{a}_k] = \sum_{j=2}^{k} (-1)^j (a_1\cdot a_j)\mathrm{Tr}[\not{a}_2\ldots\not{a}_{j-1}\not{a}_{j+1}\ldots\not{a}_k] \quad (k \text{ even})$$

$$= 0 \quad (k \text{ odd}). \tag{3.3.1}$$

The recursion may be carried out, to obtain

$$\mathrm{Tr}[\not{a}_1\not{a}_2\ldots\not{a}_k]$$

$$= \sum_{\text{pairs}} \varepsilon_{i_1 i_2\ldots i_k} (a_{i_1}\cdot a_{i_2})(a_{i_3}\cdot a_{i_4})\ldots(a_{i_{k-1}}\cdot a_{i_k}) \tag{3.3.2}$$

where $\varepsilon_{i_1\ldots i_k}$ is the rank-k totally antisymmetric tensor. This form is similar to a determinant and is known as a Pfaffian [3.10]. When no cancellations occur, eqs. (3.3.1) and (3.3.2) yield $(k-1)!!$ terms for the trace of a product of k gamma matrices, ~~except by an overall factor set consistently to 4 in eq. (3.1.1c) for all n.~~ Note that (3.3.1) and (3.3.2) do not depend on n, the dimensionality of spacetime. If all the vectors a_i are distinct, then there is no choice but to use (3.3.1) or (3.3.2). However, if some of the vectors a_i are identical, then use of the result $\not{a}\not{a} = a\cdot a \equiv a^2$ will simplify traces, particularly if $a\cdot a$ is specified to vanish by KINDEF. One could scan a trace for identical vectors and then tailor the recursion (3.3.1) so that they would meet. However, it turns out to be more efficient in GAMALG to use the standard form (3.3.1) but to check at each level of recursion for adjacent identical vectors in the traces and to remove them if present, yielding traces shorter

by two gamma matrices. It is possible (using the properties of Pfaffians) to derive a formula analogous to eq. (3.3.1) but which makes use of any identical vectors in the trace [3.10]. Nevertheless, the simple recursive method appears to be better in practice. (Note that in GAMALG, setting the flag PLATU : TRUE will cause templates to be used for the evaluation of all traces up to length 8. Since with this method, no simplifications occur with identical vectors, it is generally less efficient than the recursive method.) For traces of length less than 6 containing no contracted indices, GAMALG uses the special cases of (3.3.2)

$$\mathrm{Tr}[\not{a}_1\not{a}_2] = 4(a_1\cdot a_2)$$

$$\mathrm{Tr}[\not{a}_1\not{a}_2\not{a}_3\not{a}_4] = 4[(a_1\cdot a_2)(a_3\cdot a_4) \qquad (3.3.3)$$

$$- (a_1\cdot a_3)(a_2\cdot a_4) + (a_1\cdot a_4)(a_2\cdot a_3)].$$

In n = 4 dimensions, there exist reduction formulae which allow identical vectors in a trace to be eliminated (as in Section 3.2, E and O denote products of even and odd numbers of gamma matrices (not counting any γ_5 which appear) respectively) [3.3, 3.11]:

$$\not{p}O\not{p} = -p^2 O_R + \frac{1}{2}\not{p}\{\mathrm{Tr}[\not{p}O_R] + \gamma_5\mathrm{Tr}[\gamma_5\not{p}O_R]\} \qquad (3.3.4a)$$

$$\not{p}E\not{p} = -p^2 E_R + \frac{1}{2}\gamma_\alpha\not{p}\mathrm{Tr}[\not{p}\gamma_\alpha E_R], \qquad (3.3.4b)$$

where (again as in Section 3.2), S_R is the reversal of the string S. The results (3.3.4) are proved by methods analogous to those used in deriving the

Chisholm identities discussed below (and which are in some forms special cases of (3.3.4)) and cannot be generalized readily to $n \neq 4$ dimensions. In trace calculations, eq. (3.3.4) is rarely helpful, although in some cases, GAMALG may invoke (3.3.4b).

In most traces encountered in practice, some fraction of the elements of the trace are pairs of gamma matrices whose indices are contracted together. The presence of such elements allows traces to be evaluated with far less terms than the $k!!$ of eqs. (3.3.1) and (3.3.2). In four dimensions ($n = 4$) there exist many elegant methods for making optimal use of contracted indices. Away from four dimensions, such methods fail, terms proliferate and one must apparently resort to less elegant algorithms.

The basic strategy for evaluating traces with contracted indices is to eliminate the indices by using relations of the form

$$\gamma_\mu S \gamma^\mu = f(S), \tag{3.3.5}$$

where S represents a product of gamma matrices, and f(S) is some sum of reorderings of S. In four dimensions, contracted indices may be reduced out using the "Chisholm" identities [3.2, 3.3, 3.10, 3.11, 3.12]

$$\gamma_\mu O \gamma^\mu = -2\, O_R \tag{3.3.6a}$$

$$\gamma_\mu E \gamma^\mu = \mathrm{Tr}[E] - \gamma_5 \mathrm{Tr}[\gamma_5 E] \tag{3.3.6b}$$

$$= 2\{\mathrm{Tr}[E] - E - E_R\}$$

$$\gamma_\mu (O\gamma_\alpha)\gamma^\mu = 2\{(\gamma_\alpha O) + (O_R \gamma_\alpha)\}, \tag{3.3.6c}$$

where, as always, O denotes an odd and E an even string of gamma matrices, and R signifies reversal. In addition, as for previous similar relations, γ_5 may appear any number of times in any of the products and without affecting the counting of their length. (γ_5 may be written as the product of four distinct gamma matrices and, therefore, behaves as an even string.) The only exception to this rule is when the string consists solely of γ_5, in which case the form (3.3.6b), rather than (3.3.6c) must be used. GAMALG reduces even strings according to the third form (3.3.6c) of the even string Chisholm identity. The relations (3.3.6a) and (3.3.6c), and some corollaries to them, are summarized in the Kahane algorithm [3.6]. This algorithm is used by the CRUNCH function (see Section 3.4) in GAMALG whenever possible. However, the Kahane algorithm is designed for products of gamma matrices, not their traces. It, therefore, does not make use of the cyclic property of traces ($\mathrm{Tr}[\not{a}_1\not{a}_2\ldots\not{a}_k] = \mathrm{Tr}[\not{a}_2\ldots\not{a}_k\not{a}_1]$). For this reason (unless KAHAF : TRUE) GAMALG simply uses directly the Chisholm identities (3.3.6a) and (3.3.6c) for traces in four dimensions.

The proof of the Chisholm identities is interesting and reveals the reason that no such similar identities hold in $n \neq 4$ dimensions. The methods used may also be employed in the derivation of most of the other four-dimensional gamma matrix relations summarized in Table 3.1. I know of three proofs ~~of~~ for the Chisholm identities [(3.2, 3.12), 3.3, 3.10]; all are ultimately equivalent but are superficially rather different. Here I follow the method given in Refs. [3.2, 3.6, 3.12]. The other two proofs are in terms of Pfaffians [3.10] and by decomposition of the gamma matrices to Pauli spin matrices [3.3]. The first form of (3.3.6b) is also proved in [3.13] as an example of the reduction of a "V product". The relation (3.3.6c) for even strings may be derived trivially by use of the anticommutation relations (3.1.1a) and the result (3.3.6a)

for odd strings. I shall, therefore, discuss only the proof of (3.3.6a). The basis for the proof I shall discuss [3.1, 3.4, 3.8] is the representation of any product S of gamma matrices in n dimensions in terms of the elements of the Clifford algebra with 2^n components (strictly for the "SO(2n) extrapolation") in n classes given by [3.9, 3.10]

$$\Gamma^i = \gamma_{[\mu_1}\gamma_{\mu_2}\cdots\gamma_{\mu_i]}, \tag{3.3.7}$$

where [ ] symbolizes antisymmetrization. In four dimensions, these classes of elements are simply

$$\Gamma = \{1, \gamma_\mu, \sigma_{\mu\nu}, \gamma_5\gamma_\mu, \gamma_5\}. \tag{3.3.8}$$

Any string S may then be expanded in the form (the Γ^i are orthogonal in the sense that $Tr[\Gamma^i\Gamma^j] = 4\delta_{ij}$)

$$S = \sum \lambda_i \Gamma^i, \tag{3.3.9}$$

where the sum runs over all 2^n elements of the Clifford algebra, but by Lorentz invariance, all the elements in each class may be treated identically in our later manipulations, and so one may consider the index i to label the n classes of elements rather than the individual 2^n elements of the Clifford algebra. We wish to consider

$$\gamma_\mu S \gamma^\mu = \sum \lambda_i \gamma_\mu \Gamma^i \gamma^\mu. \tag{3.3.10}$$

Elementary combinatorics suffice to show that

$$\gamma_\mu \Gamma^i \gamma^\mu \equiv a_i \Gamma^i = (-1)^i (n-2i) \Gamma^i, \tag{3.3.11}$$

so that in n = 4 dimensions

$$a_i = \{4, -2, 0, 2, -4\}, \tag{3.3.12}$$

as usual. The expansion of any odd string will involve only the Γ^i with i odd, and we, therefore, need consider only these. Defining (analogous to charge conjugation)

$$B \gamma_\mu B^{-1} \equiv (\gamma_\mu)^T, \tag{3.3.13}$$

where T denotes transpose, one sees that

$$\begin{aligned} B \Gamma^1 B^{-1} &= +(\Gamma^1)^T = (\gamma_\mu)^T \\ B \Gamma^i B^{-1} &= -(\Gamma^i)^T \qquad (i \text{ odd}), \end{aligned} \tag{3.3.14}$$

where

$$B S B^{-1} = (S_R)^T, \tag{3.3.15}$$

so that

$$0_R = \lambda^o_1 \Gamma^1 - \sum \lambda^o_{2i+1} \Gamma^{2i+1}, \tag{3.3.16}$$

where the λ_i^o are the expansion coefficients for the odd string 0. It is now clear that in n = 4 dimensions, $\gamma_\mu 0 \gamma^\mu$ can be written in terms of 0_R, as in (3.3.6a). However, in n ≠ 4 dimensions, it does not appear possible to obtain relations analogous to the Chisholm identity (3.3.6a) which expresses $\gamma_\mu 0 \gamma^\mu$ as a sum of rearrangements of 0 whose number does not increase as the length of 0 increases. This is true even in 4 dimensions for products of the form $\sigma_{\mu\nu} S \sigma^{\mu\nu}$ (for which the a_i in eq. (3.3.12) become {12, 0, -4, 0, 12}). The Chisholm identity (3.3.6a), therefore, appears to be a fortuitous accident. In general, $\gamma_\mu S \gamma^\mu$ could be a sum of k rearrangements of S, where k is the length of S, although, in fact, it is easy to show that only k-1 rearrangements need appear. GAMALG uses the relations (e.g., [3.14])

$$\gamma_\alpha \gamma^\alpha = n$$

$$\gamma_\alpha \not{a}_1 \gamma^\alpha = (2-n) \not{a}_1$$

$$\gamma_\alpha \not{a}_1 \not{a}_2 \gamma^\alpha = (n-2) \not{a}_1 \not{a}_2 + 2 \not{a}_2 \not{a}_1$$

$$= 4(a_1 \cdot a_2) + (n-4) \not{a}_1 \not{a}_2 \qquad (3.3.17)$$

$$\gamma_\alpha \not{a}_1 \not{a}_2 \not{a}_3 \gamma^\alpha = -2 \not{a}_3 \not{a}_2 \not{a}_1 - (n-4) \not{a}_1 \not{a}_2 \not{a}_3$$

$$\gamma_\alpha \not{a}_1 \not{a}_2 \not{a}_3 \not{a}_4 \gamma^\alpha = 2(\not{a}_3 \not{a}_2 \not{a}_1 \not{a}_4 + \not{a}_4 \not{a}_1 \not{a}_2 \not{a}_3) + (n-4) \not{a}_1 \not{a}_2 \not{a}_3 \not{a}_4$$

$$\gamma_\alpha \not{a}_1 \not{a}_2 \not{a}_3 \not{a}_4 \not{a}_5 \gamma^\alpha = 2(\not{a}_2 \not{a}_3 \not{a}_4 \not{a}_5 \not{a}_1 - \not{a}_1 \not{a}_4 \not{a}_3 \not{a}_2 \not{a}_5 - \not{a}_1 \not{a}_5 \not{a}_2 \not{a}_3 \not{a}_4)$$

$$- (n-4) \not{a}_1 \not{a}_2 \not{a}_3 \not{a}_4 \not{a}_5 .$$

Note that we have no proof that the reductions given here are the shortest possible. When n ≠ 4, GAMALG first performs the reductions (3.3.17) with respect to the closest pair of the contracted indices in the trace and then repeats the procedure for each of the resulting sum of shorter traces. If a pair of contracted indices is separated by more than five slashed vectors, then when n ≠ 4 GAMALG simply treats the indices as slashed vectors and then uses eq. (3.3.1), until the indices become sufficiently close to apply eq. (3.3.17).

We now give some examples of trace evaluations in GAMALG. First we declare the indices μ,ν to be contracted

```
(C2) CIND(MU,NU,AL,BE);
(D2)                                [BE, AL, NU, MU]
```

We begin with some simple traces (any list brackets in TR arguments are ignored):

```
(C3) TR(P,Q);
(D3)                                4 D(P, Q)

(C4) TR(MU,P,MU,Q);
(D4)                              (8 - 4 N) D(P, Q)

(C5) TR([MU,P],MU,Q);
(D5)                              (8 - 4 N) D(P, Q)

(C6) TR(MU,NU,P,Q,NU,R,MU,P);
                    2
(D6)   (- 4 N  + 24 N - 32) D(P, P) D(Q, R) + (32 - 16 N) D(P, Q) D(P, R)

(C7) TR(MU,P,Q,AL,P,NU,Q,AL,Q,P,MU,NU);
                         3        2
(D7)          (- 4 N  + 24 N  - 48 N + 32) D(P, P) D(P, Q) D(Q, Q)

(C8) TR(MU,NU,AL,MU,BE,NU,AL,BE);
                              4        3        2
(D8)                       - 4 N  + 40 N  - 96 N  + 64 N
```

Recall that N is the dimensionality of spacetime assumed. Note that even indices declared by CIND will not be contracted if they are unpaired. So, for example:

```
(C9) TR(MU,P,Q,NU)$
(D9)     4 D(MU, NU) D(P, Q) + 4 D(MU, P) D(NU, Q) - 4 D(MU, Q) D(NU, P)
```

This can be contracted with the metric tensor by doing

```
(C10) CON(%*D(MU,NU))$
(D10)                                   4 N D(P, Q)
```

Now we let the vector p be equal to k + ℓ:

```
(C11) EV(%,P=K+L)$
(D11)                                 4 D(L + K, Q) N
```

and then expand the dot products

```
(C12) DFIX(%)$
(D12)                              4 (D(L, Q) + D(K, Q)) N
```

The last four steps could have been done directly as follows:

```
(C13) TR(MU,K+L,Q,MU)$
(D13)                             (4 D(L, Q) + 4 D(K, Q)) N
```

Just for illustration, we now do a long trace:

```
(C14) TR(MU,AL,P1,P2,NU,MU,P2,P3,AL,P1,NU,P3,P1,P2)$
                  3          2
(D14) ((- 4 N  + 80 N  - 416 N + 672) D(P1, P1) D(P1, P2) D(P2, P2)

                2                  3
 + (- 32 N  + 256 N - 512) D (P1, P2)) D(P3, P3)

                2                                  2
 + (- 32 N  + 256 N - 512) D(P1, P1) D(P1, P2) D (P2, P3)

                3          2
 + ((- 8 N  + 112 N  - 416 N + 448) D(P1, P1) D(P1, P3) D(P2, P2)

          2                    2
 + (64 N  - 512 N + 1024) D (P1, P2) D(P1, P3)) D(P2, P3)

          3          2                                2
 + (16 N  - 224 N  + 896 N - 1152) D(P1, P2) D (P1, P3) D(P2, P2)
```

The examples above were all done in n-dimensional space. If instead, one requires only traces in four dimensions, GAMALG can be restructured slightly

so as to be more efficient for these by doing

```
(C2) NSET(4)$
DIMENSIONS = 4
```

First, we redo a trace given above in n dimensions.

```
(C3) TR(MU,NU,P,Q,NU,R,MU,P)$
(D3)                              - 32 D(P, Q) D(P, R)
```

As an illustration of the internal workings of GAMALG, we now repeat another trace done above in n dimensions but now set the flag BORED : TRUE:

```
(C5) TR(MU,AL,P1,P2,NU,MU,P2,P3,AL,P1,NU,P3,P1,P2)$
ENTERING TRO WITH  [MU, AL, P1, P2, NU, MU, P2, P3, AL, P1, NU, P3, P1, P2]
ENTERING TRO WITH  [P2, P1, AL, NU, P2, P3, AL, P1, NU, P3, P1, P2]
ENTERING TRO WITH  [P3, P2, NU, P1, NU, P3, P1, P2, P2, P1]
ENTERING TRO WITH  [P1, P3, P1, P2, P2, P1, P3, P2]
ENTERING TRO WITH  [P1, P3, P1, P1, P3, P2]
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 850 MSEC
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 881 MSEC
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 914 MSEC
ENTERING TRO WITH  [NU, AL, P1, P2, P2, P3, AL, P1, NU, P3, P1, P2]
ENTERING TRO WITH  [P2, P1, P3, AL, P1, P2, P2, P3, AL, P1]
ENTERING TRO WITH  [P2, P2, P1, P3, P1, P2, P1, P3]
ENTERING TRO WITH  [P1, P3, P1, P2, P1, P3]
EXITING TRO WITH 2 TERMS. 39 BLOCKS CORE FREE. TIME = 2182 MSEC
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 2246 MSEC
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 2314 MSEC
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 2381 MSEC
EXITING TRO WITH 2 TERMS. 39 BLOCKS CORE FREE. TIME = 2483 MSEC
EXITING TRO WITH 3 TERMS. 39 BLOCKS CORE FREE. TIME = 2519 MSEC
EXITING TRO WITH 3 TERMS. 39 BLOCKS CORE FREE. TIME = 2551 MSEC
ENTERING TRO WITH  [P3, P1, P2, P2, P1, P2, P1, P3]
ENTERING TRO WITH  [P3, P1, P1, P2, P1, P3]
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 2771 MSEC
EXITING TRO WITH 1 TERMS. 39 BLOCKS CORE FREE. TIME = 2805 MSEC
(D5) 32 D(P1, P1) D(P1, P2) D(P2, P2) D(P3, P3)

 + 64 D(P1, P1) D(P1, P3) D(P2, P2) D(P2, P3)

                     2
 - 128 D(P1, P2) D (P1, P3) D(P2, P2)

(C6) TIME (%)$
TIME or [TOTALTIME, GCTIME] in msecs.$
(D6)                                [[2925, 508]]
```

Note that the times quoted include garbage collection* and can vary considerably according to the load on the computer, and occasionally become appreciably faster due to developments in the GAMALG code.

*i.e. the periodic automatic removal of dead LISP trees in storage.

Several shortened versions for combinations of gamma matrices commonly appearing in diagram calculations may be used in GAMALG (for G and GT as well as TR). Fermion propagators may be written as

$$ZN(p,m) = \not{p} + m$$

$$ZD(p,m) = 1/(\not{p}-m) = (\not{p}+m)/(p^2-m^2),$$

where, if the mass m is omitted, it will be assumed to be zero. (If in all cases, m is to be assumed zero, then ZERM : TRUE will make GAMALG more efficient.) Some examples of the use of ZN and ZD are (note that TR RATSIMPs but does not FACTOR the expression it returns)

```
(C2) CIND(MU,NU);
(D2)                              [NU, MU]

(C3) TR(ZD(P,M),ZD(Q,MP));
                            4 D(P, Q) + 4 M MP
(D3)             ------------------------------------------------
                           2                 2                2   2
                 (D(P, P) - M ) D(Q, Q) - MP  D(P, P) + M   MP

(C4) FACTOR (%);
                              4 (D(P, Q) + M MP)
(D4)                    ------------------------------
                                    2                2
                        (D(P, P) - M ) (D(Q, Q) - MP )

(C5) TR(ZN(P,M),MU,ZN(Q,M),NU,ZN(PP,M),MU,ZN(Q,M),NU);
              2                               2  2           2          2
(D5) ((4 N  - 40 N + 64) D(P, PP) - 4 M  N  + 24 M  N - 16 M ) D(Q, Q)

            2                                  2  2          2          2
 + ((- 8 N  + 48 N - 64) D(P, Q) + 8 M  N  - 32 M  N + 32 M ) D(PP, Q)

        2  2          2           2                     2  2          2           2
 + (8 M  N  - 32 M  N + 32 M ) D(P, Q) + (- 4 M  N  + 24 M  N - 16 M ) D(P, PP)

      4  2      4
 - 4 M  N  + 8 M  N
```

In general, GAMALG computes a trace involving $\not{p} + m$ factors by first expanding it out into a sum of traces in which such objects have been broken up. This procedure often generates more terms than are necessary, especially if the various $\not{p} + m$ factors are similar. In four dimensions, it is in some cases possible to treat $\not{p} + m$ as a single unit until the very end of the calculation of a trace and, thereby, generate few intermediate terms. In particular, the Chisholm identities (3.3.6) may be extended to allow $\not{p} + m$ terms to be treated as single factors. In the notation defined for eq. (3.2.2) [3.4]:

$$\gamma_\mu \underline{O} \gamma^\mu = 2\{\mathrm{Tr}[\gamma_5 \underline{O}]\gamma_5 - \underline{O}_R\} - \underline{O} + \bar{O}$$

$$\gamma_\mu \underline{E} \gamma^\mu = 2\{\mathrm{Tr}[\underline{E}] - E_R\} - \underline{E} - \bar{E}. \qquad (3.3.18)$$

It is clear that these results reduce to eq. (3.3.6) when all the $m_i = 0$ (so that $\underline{O} = \bar{O} = 0$, $\underline{E} = \bar{E} = E$). The relations (3.3.18) are proved by appending the 4 x 4 identity matrix to the set of gamma matrices. They are, of course, not easily generalized to $n \neq 4$ dimensions. Even after contracted indices have been removed, there may remain many $\not{p} + m$ factors in a trace. In suitable cases, GAMALG uses the result [3.15]

$$\mathrm{Tr}[(\not{p}_1+m_1)S_1(\not{p}_2+m_2)S_2 \ldots (\not{p}_k+m_k)S_k] = \mathrm{Tr}[\not{q}_1 S_1 \not{q}_2 S_2 \ldots \not{q}_k S_k],$$

$$(k + \mathrm{length}(S_i) \text{ even}) \qquad (3.3.19)$$

$$q_i \cdot v = p_i \cdot v \qquad (v \neq q_j)$$

$$q_i \cdot q_j = p_i \cdot p_j + (-1)^{\ell_{ij}} m_i m_j,$$

where ℓ_{ij} is the number of gamma matrices between $\not{p}_i$ and $\not{p}_j$ in the original trace. The trace in terms of the $\not{q}_i$ may be treated just as if the q_i were ordinary 4-component vectors. The formula (3.3.19) is very convenient when it applies, because it provides results where terms from the numerators of propagators can easily be canceled with those from their denominators. There exist several further algorithms for use with $\not{p} + m$ terms. When there are more than 6 independent such factors in a trace, an efficient reduction formula is given in Ref. [3.4].

3.4 Simplification of Gamma Matrix Products

It is sometimes useful to manipulate and simplify products of gamma matrices whose trace is not taken. In GAMALG, the function CRUNCH simplifies the gamma matrix products (represented by G's) in an expression by reducing out pairs of contracted indices (and pairs of adjacent identical vectors). One problem here is that there often exist many equivalent forms for the results. In four dimensions, the Kahane algorithm [3.6] (which is used in this case) purports to yield one of the shortest forms. In $n \neq 4$ dimensions, I do not know an algorithm which definitely does this. If the flag COF : FALSE (its default value) then GAMALG attempts to return the shortest result from CRUNCH. However, two forms returned in this way may appear different although they are, in fact, equal. To overcome this difficulty, when COF : TRUE CRUNCH alphabetizes all vectors appearing in products of gamma matrices by anticommuting them. This procedure is somewhat inelegant but does yield a standard form. For example, with COF : FALSE, one has, declaring μ and ν to be contracted indices:

```
(C2) CIND(MU,AL);
(D2)                                  [AL, MU]

(C3) EXP:G(R,MU,Q,P,MU)+2*G(MU,P,AL,MU,Q,R,AL);
(D3)          G(R, MU, Q, P, MU) + 2 G(MU, P, AL, MU, Q, R, AL)

(C4) CRUNCH(%);
(D4) (N - 4) G(R, Q, P) + 4 D(P, Q) G(R) + 8 G(Q, R, P)

                                                              2
                     + (8 N - 32) G(P) D(Q, R) + (2 N  - 16 N + 32) G(P, Q, R)
```

while if COF : TRUE, then

```
(C5) COF:TRUE$

(C6) CRUNCH(EXP);
(D6) (2 N - 20) D(P, Q) G(R) + (10 N - 40) G(P) D(Q, R)

                                                              2
                     + (24 - 2 N) D(P, R) G(Q) + (2 N  - 17 N + 44) G(P, Q, R)
```

This was done in n dimensions. In four dimensions, CRUNCH uses the Kahane algorithm. With COF : FALSE, for example, one then finds

```
(C7) COF:FALSE$

(C8) NSET(4)$
DIMENSIONS = 4

(C9) CRUNCH(EXP);
(D9)                      4 D(P, Q) G(R) + 8 G(Q, R, P)

(C10) CIND(NU,BE);
(D10)                              [BE, NU, AL, MU]

(C11) EXP2:G(AL,P1,BE,P2,MU,P3,NU,AL,P4,MU,NU,P5,BE,P6);
(D11)     G(AL, P1, BE, P2, MU, P3, NU, AL, P4, MU, NU, P5, BE, P6)

(C12) CRUNCH(%);
(D12)   - 16 G(P5, P1, P4, P3, P2, P6) - 16 G(P1, P5, P4, P3, P2, P6)

(C13) COF:TRUE$

(C14) CRUNCH(EXP2);
(D14) - 64 D(P1, P5) D(P2, P3) G(P4, P6) + 64 D(P1, P5) D(P2, P4) G(P3, P6)

           - 64 D(P1, P5) G(P2, P6) D(P3, P4) + 32 D(P1, P5) G(P2, P3, P4, P6)
```

In n = 4 dimensions, the identities (3.3.4) allow some reduction of gamma matrix products by making use of identical pairs of vectors but often at the cost of generating γ_5 and epsilon symbols. If the flag ALCRU : TRUE then these relations will be applied, together with the first form in (3.3.6b) and the relation

$$\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3} = \gamma_{\mu_1} g_{\mu_2\mu_3} - \gamma_{\mu_2} g_{\mu_1\mu_3} + \gamma_{\mu_3} g_{\mu_1\mu_2} + \epsilon_{\mu_1\mu_2\mu_3\alpha}\gamma_5\gamma^{\alpha}. \tag{3.4.1}$$

This result apparently plays a central role in the algorithms used by SCHOONSHIP [3.11].

In some cases, one considers an untraced product of gamma matrices sandwiched between on-shell Dirac spinors. Then it may be convenient to reorder the gamma matrices by anticommutation in such a way that the Dirac equation may be used to remove those adjacent to the spinors. If suitable names are chosen for the gamma matrices, this order will be the alphabetical one generated by CRUNCH when COF : TRUE. Some algorithms for treating products of gamma matrices sandwiched between spinors have been devised (e.g., [3.25]).

GAMALG ~~does not,~~ at present, treats SIG(mu,nu) = $\sigma_{\mu\nu} = \frac{1}{2i}[\gamma_\mu,\gamma_\nu]$ only by expanding it out and does not yet directly allow, for example, the projection of the $\sigma_{\mu\nu}$ part from a product of gamma matrices, as would be required in calculations of an anomalous magnetic moment.

3.5 Squaring Amplitudes

In addition to performing traces, GAMALG can also form the square of an amplitude consisting of a sum of products of gamma matrices sandwiched between spinors. The amplitude associated with a particular fermion line is represented

by SQ(spnl,amp,spn2), where spnl is the incoming spinor and spn2 the outgoing one. amp is an expression consisting of a sum of G's representing the sum of products of gamma matrices to be sandwiched between the spinors. A fermion spinor is denoted by UV(p,m). All spinors are assumed to be for fermions, rather than antifermions, so that if the latter are required, the signs of momenta or masses (depending on the user's conventions) must be changed. SQ performs the sum over fermion polarization states using the standard completeness relation (this defines our spinor normalization)

$$\sum_{\text{pols}} \bar{u}(p,m)u(p,m) = (\not{p}+m), \qquad (3.5.1)$$

denoted by ZN(p,m) in GAMALG. If one does not require a sum over the polarization states of the incoming or outgoing fermion, then it is, as usual, most convenient simply to insert the projection operator $(1+\gamma_5\not{S})/2$ in the sum (3.5.1). In GAMALG, a polarized spinor is represented by UVS(p,m,s). For both UV and UVS, if the mass is omitted, then it will be assumed to be zero. If only zero mass fermions are to be considered, setting the flag ZERM : TRUE will make GAMALG more efficient for this case.

l.c.

If the flag NTR : TRUE, then SQ will generate GT's rather than producing TR's which are then performed, SQ(uv(pi,mi), amp, ZN(pf,mf), (amp)*), and, for example, (recall that all brackets ("[" and "]") in G's etc. are entirely irrelevant; they are inserted by SQ as shown for easier assimilation)

```
(C2) NTR:TRUE$

(C3) SQ(UV(PI,MI),G(A1,A2),UV(PF,MF))$
(D3)              GT(ZN(PI, MI), [A1, A2], ZN(PF, MF), [A2, A1])

(C4) SQ(UV(PI,MI),2*G(A1,A2)+G(A1,A3)+G(A4),UV(PF,MF))$
(D4) GT(ZN(PI, MI), [A4], ZN(PF, MF), [A4])

 + 2 GT(ZN(PI, MI), [A1, A3], ZN(PF, MF), [A4])

 + GT(ZN(PI, MI), [A1, A3], ZN(PF, MF), [A3, A1])

 + 4 GT(ZN(PI, MI), [A1, A2], ZN(PF, MF), [A4])

 + 4 GT(ZN(PI, MI), [A1, A2], ZN(PF, MF), [A3, A1])

 + 4 GT(ZN(PI, MI), [A1, A2], ZN(PF, MF), [A2, A1])

(C5) SQ(UV(P),G(ZD(Q-P),K),UV(PP))$
(D5)                GT(P, [ZD(Q - P), K], PP, [K, ZD(Q - P)])
```

That GT's, rather than G's, are returned, signifies that the traces of the gamma matrix products are to be taken. Note that for an amplitude consisting of j terms, SQ will return $j(j+1)/2$ traces. The reduction from j^2 traces assumes charge conjugation invariance. Note that SQAM(spn1,amp1,spn2,amp2) will form the amp1(amp2)* term in the square of an amplitude.

The G's which appear in amplitudes to be squared will typically have coefficients which represent combinatoric weights (and sometimes residual signs from the implementation of Fermi antisymmetrization). Especially in non-Abelian gauge theories, such weights are often not commutative so that one may not simply multiply each term (diagram) in the amplitude by some number. Rather, one must, for example, assign a weight wi to the i^{th} diagram and then set wi*wj (perhaps using RATSUBST) to the combinatoric factor for the term in the amplitude squared consisting of the i^{th} term (diagram) in the amplitude and the j^{th} term in the conjugate of the amplitude. The automatic computation of combinatoric and group-theoretic weights for diagrams ~~is discussed in Section 5.3.~~ will be discussed in a later installement.

When a vector boson line connects two points on the same fermion line, one may perform a Feynman gauge polarization sum for it simply by inserting a pair of identical indices at the relevant two positions in the G's representing the amplitude in SQ. For other gauges, it is usually better to perform the polarization sum before, rather than after the traces are taken.

External vector boson lines may be treated in two ways by SQ. If an index that has been declared contracted by CIND appears only once in an amplitude, then in the conjugate of the amplitude, it will be replaced by the concatenation of its name with 'PRIME' unless the index appears in the list NPIND. In the latter case, the index in the conjugate amplitude will be the same as in the amplitude, and when traces are done a contraction will be performed between the two appearances of the index. This gives a Feynman gauge polarization sum for the corresponding external vector boson. ~~This may be done for all~~ indices which appear will be treated in this manner if the flag ~~by setting~~ NOP : TRUE. For example,

```
(C6) CIND(MU,NU)$
(C7) SQ(UV(PI),G(MU,NU),UV(PF));
(D7)                GT(PI, [MU, NU], PF, [NUPRIME, MUPRIME])
(C8) NPIND:[MU]$
(C9) SQ(UV(PI),G(MU,NU),UV(PF));
(D9)                  GT(PI, [MU, NU], PF, [NUPRIME, MU])
(C10) NOP:TRUE$
(C11) SQ(UV(PI),G(MU,NU),UV(PF));
(D11)                    GT(PI, [MU, NU], PF, [NU, MU])
```

When two fermion lines appear in an amplitude, they are usually connected by one or more bosons. If these are vector bosons, and Feynman gauge is used, then the two traces corresponding to the two fermion lines will contain ~~two~~ a pairs of indices which are contracted between the traces. If n = 4, then by

applying COTR to the corresponding product of GT's, one may usually combine the traces using the relations (3.2.1) and (3.2.2). Some examples of this convenient procedure are given in Section 3.8.

3.6 γ_5

In many calculations it is convenient to introduce the pseudoscalar gamma matrix γ_5 (denoted G5 in GAMALG). This appears in expressions for vertices which violate parity (as in weak interactions) or in conjunction with spin axial vectors (as in the fermion spin projection operator). In four dimensions, the treatment of γ_5 is simple. It may be defined as (e.g., [3.8])

$$\begin{aligned}\gamma_5 &= \frac{i}{4!}\,\gamma_{[\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4]} \\ &= i\gamma^0\gamma^1\gamma^2\gamma^3,\end{aligned} \tag{3.6.1}$$

where in the second form the explicit components of γ_μ are used (equivalently, these could be obtained by dotting γ_μ into unit vectors along the various axes). It is clear from eq. (3.6.1) that

$$\begin{aligned}\{\gamma_\mu,\gamma_5\} &= 0 \\ (\gamma_5)^2 &= 1,\end{aligned} \tag{3.6.2}$$

and so that the trace of a product of less than four gamma matrices and γ_5 vanishes. The simplest non-vanishing trace involving a single γ_5 is, therefore,

$$\mathrm{Tr}[\not a_1\not a_2\not a_3\not a_4\gamma_5] = 4\,i\varepsilon_{\mu_1\mu_2\mu_3\mu_4}a_1^{\mu_1}a_2^{\mu_2}a_3^{\mu_3}a_4^{\mu_4}. \tag{3.6.3}$$

The totally antisymmetric product of four vectors appearing here is represented as EPS(a1,a2,a3,a4) in GAMALG. The symbols in an EPS are usually alphabetized for simplification, as described in Section 3.2. The trace of a product of more than four gamma matrices with γ_5 may be reduced by the relation (e.g., [3.4, 3.13])

$$\begin{aligned}
\mathrm{Tr}[\not{a}_1 \ldots\ldots \not{a}_k \gamma_5] &= \sum_{i_1<i_2<i_3<i_4} (-1)^{i_1+i_2+i_3+i_4} \\
&\quad \times \mathrm{Tr}[\not{a}_{i_1} \not{a}_{i_2} \not{a}_{i_3} \not{a}_{i_4} \gamma_5] \mathrm{Tr}[\not{a}_{i_5} \ldots\ldots \not{a}_{i_k}] \\
&= 4 \sum_{\{i_j\}} \varepsilon_{i_1 i_2 \ldots\ldots i_k} (\varepsilon_{\mu_1\mu_2\mu_3\mu_4} a_{i_1}^{\mu_1} a_{i_2}^{\mu_2} a_{i_3}^{\mu_3} a_{i_4}^{\mu_4}) \\
&\quad \times (a_{i_5} \cdot a_{i_6}) \ldots\ldots (a_{i_{k-1}} \cdot a_{i_k}) \qquad (k \text{ even}) \\
&= 0 \qquad (k \text{ odd}).
\end{aligned} \tag{3.6.4}$$

Some simple examples of traces involving γ_5 done using GAMALG are (RHP = $(1 + \gamma_5)/2$, LHP = $(1 - \gamma_5)/2$):

```
(C2) NSET(4)$
(C3) CIND(MU,NU)$
(C4) TR(P1,G5,P2,P3,P4)$
(D4)                          - 4 %I EPS(P1, P2, P3, P4)
(C5) TR(RHP,P3,P4,P2,P1)$
(D5) 2 D(P1, P2) D(P3, P4) + 2 D(P1, P3) D(P2, P4) - 2 D(P1, P4) D(P2, P3)
                                                    - 2 %I EPS(P1, P2, P3, P4)
(C6) TR(P1,MU,NU,P2,G5,P3,MU,P4,NU)$
(D6)                          - 16 %I EPS(P1, P2, P3, P4)
(C7) TR(P1,P2,P3,P4,P5,P6,G5)$
(D7) 4 %I EPS(P1, P2, P3, P4) D(P5, P6) - 4 %I EPS(P1, P2, P3, P5) D(P4, P6)
 + 4 %I EPS(P1, P2, P3, P6) D(P4, P5) + 4 %I EPS(P1, P2, P4, P5) D(P3, P6)
 - 4 %I EPS(P1, P2, P4, P6) D(P3, P5) + 4 %I D(P1, P2) EPS(P3, P4, P5, P6)
 + 4 %I EPS(P1, P2, P5, P6) D(P3, P4) - 4 %I EPS(P1, P3, P4, P5) D(P2, P6)
 + 4 %I EPS(P1, P3, P4, P6) D(P2, P5) - 4 %I D(P1, P3) EPS(P2, P4, P5, P6)
 - 4 %I EPS(P1, P3, P5, P6) D(P2, P4) + 4 %I D(P1, P4) EPS(P2, P3, P5, P6)
 - 4 %I D(P1, P5) EPS(P2, P3, P4, P6) + 4 %I D(P1, P6) EPS(P2, P3, P4, P5)
 + 4 %I EPS(P1, P4, P5, P6) D(P2, P3)
```

The generalization of γ_5 to $n \neq 4$ dimensions is not as straightforward as for the usual gamma matrices. Many prescriptions for the generalization of γ_5 exist, and great care must be taken to ensure that physical results obtained in the limit $n \to 4$ are independent of the one chosen. (Recently, several results on chiral symmetry breaking in QCD deduced from the behavior of the axial vector spectral function have suffered from difficulties associated with the generalization of γ_5.) One of the major advantages of dimensional regularization is that it preserves invariances in theories (most importantly, gauge invariance). However, it is believed that the chiral invariances of the Lagrangian which lead to axial vector Ward identities are genuinely violated in perturbation theory (leading to 'anomalies'), and that there is no

method of regularization which leaves them intact. Different choices for the generalization of γ_5 make the violations apparent in different places. It is thought that only theories for which the 'anomalous' violations of the axial Ward identities appearing in perturbation theory cancel (usually between sets of fermions with suitable charge assignments) may be made renormalizable, so that higher order calculations are meaningful. Thus, the generalization of γ_5 should be arranged so that the cancellations are most easily achieved. Several prescriptions for the construction of γ_5 in n dimensions have been proposed; the more common are

1. Take $\{\gamma_5,\gamma_\mu\} = 0$ for $\mu = 0, 1, 2, 3$ and $[\gamma_5,\gamma_\mu] = 0$ for $\mu = 4, \ldots, n-1$ [3.16].

2. Take $\gamma_5 = \Gamma^n = \gamma_{[\mu_1}\gamma_{\mu_2}\ldots\gamma_{\mu_n]}$ (c.f. eq. (3.3.7)) [3.17].

3. Take $\gamma_5 = \Gamma^4 = \gamma_{[\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4]}$ (covariant form of (1)) [3.17].

4. Take $\{\gamma_5,\gamma_\mu\} = 0$ for all μ and set $\mathrm{Tr}[\gamma_5\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4}]$ to the four-dimensional epsilon symbol plus arbitrary terms of order (n-4) which should cancel in calculations [3.18].

5. Perform gamma matrix algebra associated with γ_5 in $n = 4$ but let most loop momenta, as usual, have n components (may not work) [3.19].

6. Treat separately the left- and right-handed components of fields (which in $n = 4$ are $\frac{1}{2}(1-\gamma_5)\psi$ and $\frac{1}{2}(1+\gamma_5)\psi$, respectively), so that γ_5 never appears explicitly. (Not convenient for calculation.)

All of these prescriptions, of course, reduce to the usual definition (3.22) when $n = 4$, but $1/(n-4)$ terms from loop integrations can reveal their small differences when $n \neq 4$. Any of these, if applied with sufficient care, apparently give correct results at the one-loop level. Alternative (4) is probably the most convenient and gives most directly the required cancellations, especially for traces involving an even number of γ_5. It is not clear how the

various alternatives will fare at the two-loop level. At present, GAMALG uses the prescription (4) (after printing a warning message) for treating γ_5 in $n \neq 4$ dimensions. A program to implement (2) or (3) will probably soon be included in GAMALG (and will be invoked by setting the flag NG5 : CGN or CG4, respectively [3.20]).

3.7 Comparison of Gamma Matrix Manipulators

As an example of the various gamma matrix manipulators available*, we give their input and output for the trace

$$\mathrm{Tr}[(\not{p}'+m)\gamma_\mu(\not{p}+\not{k}+m)\gamma_\nu(\not{p}+m)\gamma^\nu(\not{p}+\not{k}+m)\gamma^\mu]$$

which corresponds to the square of the s-channel pole diagram for γe scattering.

GAMALG

```
This is MACSYMA 281

FIX281 18 DSK MACSYM being loaded
Loading done

(C1) loadfile (gamalg,>,share)$

GAMALG 24 DSK SHARE being loaded
Loading done

(C2) cind(mu,nu)$

(C3) nset(4)$
DIMENSIONS = 4

(C4) tr(zn(pp,m),mu,zn(p+k,m),nu,zn(p,m),nu,zn(p+k,m),mu)$

                                  2                                  4
(D4) (16 D(P, P) - 48 M  - 16 D(K, K)) D(P, PP) + 32 D(K, PP) D(P, P) + 64 M

                                                                  2
          + (- 64 D(K, PP) + 64 D(K, P) + 64 D(K, K)) M  + 32 D(K, P) D(K, PP)
```

*I am grateful to D. Ross and J. Babcock for providing the SCHOONSHIP and ASHMEDAI outputs, respectively. The REDUCE example was run on the Caltech PDP-10.

SCHOONSHIP [3.11] (usually run in a batch mode and written in CDC assem-bler) (?)

```
SCHOONSCHIP, VERSION OF JANUARY 1, 1972

TIME      .09 SECONDS

                 V PP,P,K
                 I MU,NU,J
                 S M
                 Z TR=(G(J,PP)+M*GI(J))*G(J,MU)*(G(J,P)+G(J,K)+M*GI(J))*G(J,NU)*(G(J,P)
                   +M*GI(J))*G(J,NU)*(G(J,P)+G(J,K)+M*GI(J))*G(J,MU)
         L 2     ID,TRICK,TRACE,J
                 * END

SYMBOLS        I=I, M.

INDICES        MU, NU, J.

VECTORS        PP, P, K.

FUNCTIONS      D, EPF=I, G=I, GI, G5=I, G6=C, G7, UG=C, UBG, DD=U, D6, DT,
               D3=U, DX, DK, DP, DF=U.

RUNNING TIME         .02 SEC.
NUMBER OF TERMS       19
EQUAL TERMS            9
CANCELLATIONS          1     INPUT SPACE  NR. OF EXPR.      ID. REG.    FUNC. REG.     MULT.    OUTPUT SPACE
RECORDS                0     497 ( 2063)     38 (200)        4 (500)      61 (750)       378     2090 ( 6953)

  TR =

    - 46.*M**2*PPDP - 64.*M**2*PPDK + 64.*M**2*PDK + 64.*M**2*KDK + 64.*M**4 + 16.*PPDP*PDP - 16.*PPDP*KDK

   + 32.*PPDK*PDP + 32.*PPDK*PDK  +0.


END OF RUN. TIME      .12 SECONDS
```

stet.

REDUCE [3.21] (usually run in an interactive mode and written in LISP) (REDUCE divides all results for traces by 4)

```
REDUCE 2 (NOV-30-73) ...

*VECTOR PP,K,P$

*INDEX MU,NU$

*OPERATOR Z$

*FOR ALL P LET Z(P)=G(L,P)+M$

*Z(PP)*G(L,MU)*Z(P+K)*G(L,NU)*Z(P)*G(L,NU)*Z(P+K)*G(L,MU);

                 2                                     4         2           2
4*( - 4*PP.K*M  + 2*PP.K*P.K + 2*PP.K*P.P + 4*M  + 4*M *P.K + 4*M *

          2
   K.K - 3*M *P.PP - K.K*P.PP + P.P*P.PP)
```

```
   0.025    11 32000 * SUBST,KINC KI,KI=0 $ KF,KF=0 $ PI,PI=M*M $ PF,PF=M*M $ KI,EI=0 $
   0.026    11 32000  2 SUBST (1549393988)
 891125612
CR SYM 1    10 KINC   ( 891125612)
CR SYM 3     1 KI     ( 890353072)
CR SYM 3     2 KF     ( 884097664)
CR SYM 3     3 PI     (1286528912)
CR SYM 2     1 M      (1030057184)
CR SYM 3     4 PF     (1280273504)
CR SYM 3     5 EI     ( 914992064)
   0.027    76 32000 *       1KF,EF=0$ EI,EI=-1 $ EF,EF=-1 $$
CR SYM 3     6 EF     ( 908686656)
   0.028   100 32000 * SUBST,KIN PS(J) = KI(J)+PI(J) $$
   0.028   100 32000  2 SUBST (1549393988)
 891121280
CR SYM 1     9 KIN    ( 891121280)
CR SYM 3     7 PS     (1307380272)
CR SYM 4    21 J      ( 792351580)
   0.029   133 32000 * SUBST,KIN1 EI,PI=0 $ EF,PI=0 $$
   0.030   133 32000  2 SUBST (1549393988)
 891161712
CR SYM 1    2F KIN1   ( 891161712)
   0.030   150 32000 * SUBST,KIN2 KI,PF=KF,PI $ EF,PF=EF,KI $$
   0.030   150 32000  2 SUBST (1549393988)
 891163156
CR SYM 1    20 KIN2   ( 891163156)
   0.031   169 32000 * C
   0.031   169 32000 * INPUT SPUR((PF/+M)*EF/*(PS/+P)*EI/*(PI/+M)*EI/*(PS/+P)*EF/)
   0.032   169 32000  1 INPUT ( 793217452)

CR SYM 1    25        (          0)
   0.032   210 32000 * TRACE DOT COLCT LOOK
   0.033   224 31991  1 TRACE (1622295202)

   0.033   224 31995  1 DOT    ( 349315152)

   0.033   224 31997  1 COLCT ( 269646100)

CR SYM 1    29 COLCT1( 269646137)
FLIP AUT COLCT1.    122 WORDS,     2 TREES,     22 CALLS,     18 TERMS,      0 ZEROES,      68 TESTS.
   0.037   283 31993  1 LOOK   ( 982938020)

+4*M**2*PF.PI*EI.EI*EF.EF -8*M**2*EI.PI*EI.PF*EF.EF +16*M**2*EI.PI*EF.PF*EF.EI+16*M**2*EI.PI*PS.EI*EF.EF-8*M**2*EF.PI*EF.PF*EI.EI
-8*M**2*PS.PI*EI.EI*EF.EF +16*M**2*EI.PF*EI.PI*PS.EF-8*M**2*PS.PF*EI.PI*EF.EF +4*M**2*EI.EI*EF.EF*PS.PS -4*PF.PI*EI.EI*EF.EF*PS.PS
+8*EI.PI*EI.PF*EF.EF*PS.PS-16*EI.PI*EF.PF*EF.EI*PS.PS                                  +32*EI.PI*EF.PF*PS.EI*PS.EF
-16*PI.PI*PS.PF*PS.EI*EF.EF                                  +8*EF.PI*EF.PF*EI.EI*PS.PS-16*PS.PI*EF.PF*EI.EI*PS.EF
+8*PS.PI*PS.PF*EI.EI*EF.EF+M**4*EI.EI*EF.EF
   0.040   283 31993 * SUBST,KIN SS COLCT LOOK
   0.041   283 31993  2 SUBST (1549393988)
 891121280
   0.041   283 31988  1 COLCT ( 269646100)

CR SYM 1    2F COLCT*( 269646137)
FLIP AUT COLCT*.    230 WORDS,     2 TREES,     49 CALLS,     36 TERMS,      3 ZEROES,     200 TESTS.
   0.047   341 31993  1 LOOK   ( 982938020)

+4*M**2*PI.KI*EI.EI*EF.EF -8*M**2*PF.KI*EI.PI*EF.EF +16*M**2*EI.KI*EI.PI*EF.EF+16*M**2*EI.KI*EF.PF*EI.EI-4*M**2*PI.PI*EI.EI*EF.EF
-4*M**2*PF.PI*EI.EI*EF.EF +16*M**2*EI.PI*EI.PI*EF.EF-8*M**2*EI.PI*EI.PF*EF.EF +16*M**2*PI.PI*EF.PF*EF.EI+8*M**2*EF.PI*EF.PF*EI.EI
-4*KI.KI*PF.PI*EI.EI*EF.EF+8*KI.KI*PI.PI*EI.PF*EF.EF-16*KI.KI*EI.PI*EF.PF*EF.EI                    +8*KI.KI*EF.PI*EF.PF*EI.EI
+8*PI.KI*PF.KI*EI.PI*EF.EF-16*PI.KI*EF.KI*EF.PF*EI.EI                    +16*PI.KI*EI.PI*EI.PF*EF.EF
-32*PI.KI*EI.PI*EF.PF*EF.EI                    -16*PF.KI*EI.KI*EI.PI*EF.EF                    +8*PF.KI*PI.PI*EI.EI*EF.EF
-16*PF.KI*EI.PI*EI.PI*EF.EF                    +32*EI.KI*EF.KI*EI.PI*EF.PF                    -16*EI.KI*PF.PI*EI.PI
```

For a discussion of other gamma matrix manipulators, see Ref. [3.23]. To give CPU times for each of the examples would be quite meaningless, since each was run on a different computer. Nevertheless, in the implementations available to me, SCHOONSHIP ran about a hundred times quicker than the other three, which all took a few seconds for the trace.

REDUCE is probably the most widely used gamma matrix manipulator. Its main merit appears to be portability. It is only able to handle pure traces - it cannot, for example, square amplitudes directly. SCHOONSHIP's primary advantage is speed. The fact that it is written in assembler means that it can compute traces often two orders of magnitude faster than REDUCE or GAMALG. SCHOONSHIP can square amplitudes. ASHMEDAI is in many respects similar to SCHOONSHIP, although it is not as fast. For SCHOONSHIP, REDUCE and ASHMEDAI, a large fraction of the programming effort and code was devoted to providing facilities comparable to the simpler functions of MACSYMA. Since GAMALG was written in MACSYMA, we were able to start at a much higher level and to implement very easily many complicated algorithms. The primary motivation for the construction of GAMALG was the necessity of computing traces in $n \neq 4$ dimensions. As discussed at length in Section 3, algorithms developed for $n = 4$ fail when $n \neq 4$. It turns out that a simple set of tricks (which was discovered empirically) suffices to allow SCHOONSHIP to evaluate traces of up to 8 gamma matrices in $n \neq 4$ dimensions. However, these fail for larger traces, and, to our knowledge, GAMALG is the only program presently available for their evaluation.

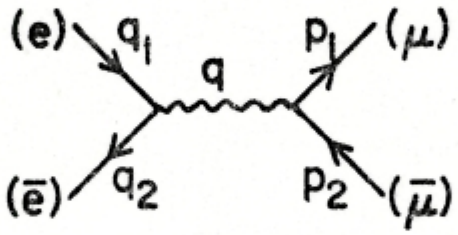
One important limiting factor in a gamma matrix manipulator is the length of intermediate expressions which it can handle. As on many other counts, SCHOONSHIP wins here; it is reputed never to have been stopped by too many intermediate terms in any sensible calculation and appears to have successfully

dealt with in excess of 10^5 terms. On the other hand, REDUCE (at least in the implementation available to me) cannot manipulate more than a few hundred terms. GAMALG has successfully handled slightly larger intermediates. In both cases, the main limitation appears to come from the fact that most present implementations of LISP cannot address sufficient memory* for very large expressions to be handled. Forthcoming implementations (e.g., [3.24]) should allow much larger intermediates (perhaps $\gtrsim 10^5$ terms) but probably at the cost of a considerable reduction in speed.

3.8 Examples of the use of GAMALG

3.8.1 $e^+e^- \to \mu^+\mu^-$

The diagram considered here is



The calculation is to be done in four dimensions, so do

```
(C2) NSET(4);
DIMENSIONS = 4
(D2)                                    4
```

We first specify some kinematics by doing

```
(C3) Q2:Q-Q1;
(D3)                                 Q - Q1

(C4) P2:Q-P1;
(D4)                                 Q - P1

(C5) KINDEF(D(Q,Q)=S,D(Q,P1)=S/2,D(Q,Q1)=S/2,D(P1,P1)=MMU^2,D(Q1,Q1)=MMME^2);
                                      S                 S                2
(D5) [D(Q, Q) = S, D(Q, P1) = -, D(Q, Q1) = -, D(P1, P1) = MMU ,
                              2             2

                                                                  2
                                                      D(Q1, Q1) = ME ]
```

* Typically a CONS cell is 18 bits, allowing $\sim 2^{18} \simeq 250$K of memory to be addressed.

We declare the indices μ and ν to be contracted when possible:

```
(C6) CINDD(MU,NU)$
(D6)                                        [NU, MU]
```

We shall do this simple calculation by several methods, to illustrate various features of GAMALG.

First, we construct the initial lepton tensor directly by doing a trace

```
(C7) LEPI:TR(MU,ZN(Q1,ME),NU,ZN(Q2,ME))$
(D7) - 2 D(MU, NU) S + (4 D(MU, Q) - 8 D(MU, Q1)) D(NU, Q1)

                                                    2
                                  + 4 D(MU, Q1) D(NU, Q) + 8 ME  D(MU, NU)
```

and similarly for the final lepton tensor

```
(C8) LEPF:TR(MU,ZN(P1,MMU),NU,ZN(P2,MMU))$
(D8) - 2 D(MU, NU) S + 4 D(MU, P1) D(NU, Q)

                                                    2
                       + (4 D(MU, Q) - 8 D(MU, P1)) D(NU, P1) + 8 MMU  D(MU, NU)
```

Then we contract indices

```
(C9)       CON(LEPI*LEPF) ,FACTOR  $
                 2                  2          2          2          2    2
(D9) 8 (S  - 4 D(P1, Q1) S - 2 MMU  S - 2 ME  S + 8 D (P1, Q1) + 16 ME  MMU )
```

The traces above could have been constructed by squaring amplitudes. For example:

```
(C10) LEPI:SQ(UV(Q1,ME),G(MU),UV(Q2,ME))$
(D10) - 2 D(MU, MUPRIME) S + (4 D(MU, Q) - 8 D(MU, Q1)) D(MUPRIME, Q1)

                                                    2
                                + 4 D(MU, Q1) D(MUPRIME, Q) + 8 ME  D(MU, MUPRIME)
```

Note the generation of the new index muprime. The weight of the amplitude here was taken as 1. To see the form which was passed to TR by SQ, do

```
(C11) NTR:TRUE$

(C12) SQ(UV(Q1,ME),G(MU),UV(Q2,ME));
(D12)          GT(ZN(Q1, ME), [MU], ZN(Q - Q1, ME), [MUPRIME])
```

The G's could be turned in TR's and done by doing CGT(%).

A slightly more efficient way to do the calculation is as follows:

```
(C13) NTR:TRUE$

(C14) SQ(UV(Q1,ME),G(MU),UV(Q2,ME))*SQ(UV(P1,MMU),G(MU),UV(P2,MMU));
(D14) GT(ZN(P1, MMU), [MU], ZN(Q - P1, MMU), [MUPRIME])

                                     GT(ZN(Q1, ME), [MU], ZN(Q - Q1, ME), [MUPRIME])
(C15) COTR(%);
(D15) 2 (GT([ZN(Q - P1, MMU), MUPRIME, ZN(P1, MMU), ZN(Q - Q1, ME), MUPRIME,

ZN(Q1, ME)]) + GT([ZN(P1, MMU), MUPRIME, ZN(Q - P1, MMU), ZN(Q - Q1, ME),

MUPRIME, ZN(Q1, ME)]))

(C16) CGT(%),FACTOR;
             2                          2                         2             2
(D16) 8 (S   - 4 D(P1, Q1) S - 2 MMU   S + 2 ME MMU S - 2 ME   S + 8 D (P1, Q1)

                                                                       2     2
                                                              + 16 ME   MMU  )
```

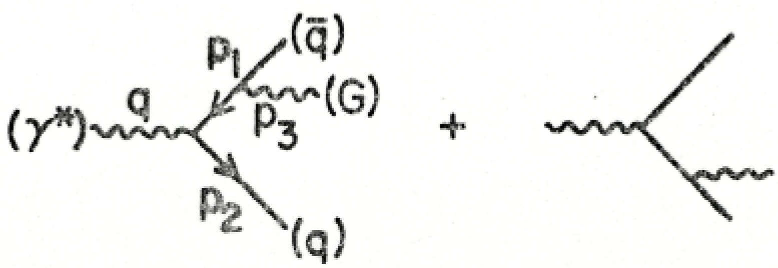
In order to reproduce a familiar result, let us set the lepton mass to zero and define some more kinematics. CT is the μ^+e^- c.m.s. angle.

```
(C17) EV(%,ME=0,MMU=0,D(P1,Q1)=S/4*(1-CT),FACTOR);
                                  2      2
(D17)                     4 (CT   + 1) S
```

To obtain the final answer, this result must be multiplied by a flux factor, and by the γ^* propagator.

3.8.2 $\gamma^* \to q\bar{q}G$

We consider the diagrams



and will sum over the virtual photon and gluon polarizations using Feynman gauge. We, therefore, set

```
(C2) NOP:TRUE;
(D2)                                                    TRUE
```

so that no primed indices are generated by SQ and declare indices:

```
(C3) CIND(MU,AL);
(D3)                                                 [AL, MU]
```

Once again, we work in four dimensions:

```
(C4) NSET(4);
DIMENSIONS = 4
(D4)                                                     4
```

We take the q and G masses to be zero and, therefore, write

```
(C5) ZERM:TRUE;
(D5)                                          TRUE

(C6) KINDEF(D(Q,Q)=S,D(P1,P1)=0,D(P2,P2)=0,D(P3,P3)=0);
(D6)       [D(Q, Q) = S, D(P1, P1) = 0, D(P2, P2) = 0, D(P3, P3) = 0]
```

So as to see the expression before traces are taken, set

```
(C7) NTR:TRUE;
(D7)                                                    TRUE
```

Then, to generate the squared amplitude,

```
(C8) SQ(UV(P1),G(MU,ZD(Q-P2),AL)-G(AL,ZD(Q-P1),MU),UV(P2));
(D8) GT(P1, [MU, ZD(Q - P2), AL], P2, [AL, ZD(Q - P2), MU])

 + GT(P1, [AL, ZD(Q - P1), MU], P2, [MU, ZD(Q - P1), AL])

 - 2 GT(P1, [AL, ZD(Q - P1), MU], P2, [AL, ZD(Q - P2), MU])
```

The traces are done using CGT:

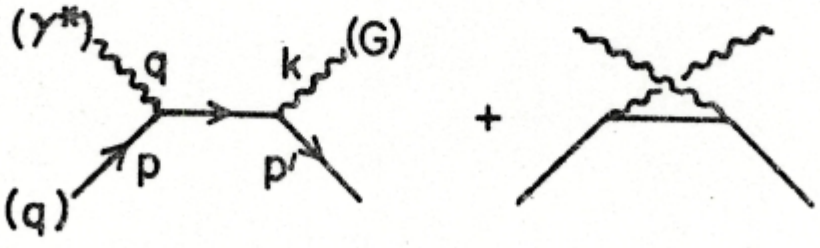
```
(C9) ANS:CGT(X);
          16 D(P1, P2) S - 32 D(P1, Q) D(P2, Q)
(D9) - ------------------------------------
                    2                      2
              S  - 4 D(P2, Q) S + 4 D (P2, Q)

 + 2 (32 D(P1, P2) S - 32 D(P1, P2) D(P2, Q) - 32 D(P1, P2) D(P1, Q)

                2                 2
 + 32 D (P1, P2))/(S  + (- 2 D(P2, Q) - 2 D(P1, Q)) S + 4 D(P1, Q) D(P2, Q))

   16 D(P1, P2) S - 32 D(P1, Q) D(P2, Q)
 - ------------------------------------
                    2                      2
              S  - 4 D(P1, Q) S + 4 D (P1, Q)
```

We may write this result in a more digestible form by making some good kinematic substitutions (and using DFIX to apply them):

```
(C10) KINDEF(D(Q,P1)=X1*S/2,D(Q,P2)=X2*S/2,D(P1,P2)=S*(1-X3)/2);
(D10) [D(Q, Q) = S, D(P1, P1) = 0, D(P2, P2) = 0, D(P3, P3) = 0,

                      S X1                 S X2                 S (X2 + X1 - 1)
          D(Q, P1) = ----, D(Q, P2) = ----, D(P1, P2) = ----------]
                       2                    2                    2

(C11) FACTOR (EV(DFIX(ANS), X3=2-X1-X2) );
                                  2    2
                              8 (X2  + X1 )
(D11)                         -------------
                              (X1 - 1) (X2 - 1)
```

3.8.3 <u>$\gamma^* q \to qG$</u>

We consider the diagrams



At first, we will not sum over the virtual photon polarization states, so as to obtain the 'hadron tensor' corresponding to these diagrams. We perform a Feynman gauge polarization sum for the gluon.

```
(C3) NSET(4)$
DIMENSIONS = 4
(D3)                                    4

(C4) ZERM:TRUE$
(D4)                                   TRUE

(C5) CIND(MU,AL)$
(D5)                                [AL, MU]

(C6) NPIND:[AL]$
(D6)                                 [AL]

(C7) KINDEF(D(P,P) = 0,DS(Q) = Q2,DS(K) = 0,D(Q,P) = (S-Q2)/2,D(Q,K) = (Q2-T)/2
,D(P
,K) = -U/2)$
                                                                  S - Q2
(D7) [D(P, P) = 0, D(Q, Q) = Q2, D(K, K) = 0, D(Q, P) = ------,
                                                                    2

                                                                  Q2 - T          U
                                                  D(Q, K) = ------, D(P, K) = - --]
                                                                    2             2

(C8) PP:Q+P-K$
(D8)                                Q + P - K

(C9) T:-S-U+Q2$
(D9)                              - U - S + Q2

(C10) NTR:TRUE$

(C11) ANS:SQ(UV(P),G(MU,ZD(Q+P),AL)+G(AL,ZD(P-K),MU),UV(PP))$
(D11) GT(P, [MU, ZD(Q + P), AL], Q + P - K, [AL, ZD(Q + P), MUPRIME])

 + GT(P, [AL, ZD(P - K), MU], Q + P - K, [MUPRIME, ZD(P - K), AL])

 + 2 GT(P, [AL, ZD(P - K), MU], Q + P - K, [AL, ZD(Q + P), MUPRIME])

(C12) TEN:CGT(%)$

(C13) FACTOR (EV(TEN))$
                                                                    2
(D13) - 4 (D(MU, MUPRIME) U  - 2 D(MU, MUPRIME) Q2 U

 + 4 D(MU, Q) D(MUPRIME, Q) U + 4 D(MU, P) D(MUPRIME, Q) U

 - 4 D(K, MU) D(MUPRIME, Q) U - 2 D(K, MU) D(MUPRIME, P) U

                                                                  2
 + 2 D(K, MUPRIME) D(MU, P) U + D(MU, MUPRIME) S  - 2 D(MU, MUPRIME) Q2 S

 - 2 D(K, MU) D(MUPRIME, Q) S - 4 D(MU, Q) D(MUPRIME, P) S

 - 2 D(K, MU) D(MUPRIME, P) S + 2 D(K, MUPRIME) D(MU, Q) S

                                                                    2
 + 2 D(K, MUPRIME) D(MU, P) S + 2 D(MU, MUPRIME) Q2

 + 4 D(MU, P) D(MUPRIME, Q) Q2 + 4 D(MU, Q) D(MUPRIME, P) Q2

 + 8 D(MU, P) D(MUPRIME, P) Q2 - 4 D(K, MUPRIME) D(MU, Q) Q2

 - 8 D(K, MUPRIME) D(MU, P) Q2 + 4 D(K, MU) D(K, MUPRIME) Q2)/(S U)
```

Now we check gauge invariance (the INFEVAL flag in the EV is merely to guarantee that all assignments such as (C9) are carried out in a single line)

```
(C14) FACTOR (EV(  CON(D(Q,MU)*D(Q,MUPRIME)*TEN) ,INFEVAL ));
(D14)                                  0
```

and then project out two components of the tensor TEN (the first is roughly the transverse component, and the second is the longitudinal component):

```
(C15) FACTOR (EV(  CON(D(MU,MUPRIME)*TEN) ,INFEVAL ));
                          2                2                2
                   8 (U  - 2 Q2 U + S  - 2 Q2 S + 2 Q2 )
(D15)            - ------------------------------------
                                  S U
(C16) FACTOR (EV(  CON(D(P,MU)*D(P,MUPRIME)*TEN) ,INFEVAL ));
(D16)                          - 4 (U + S - Q2)
```

3.8.4 Muon Decay

Here we calculate the differential cross-section for muon decay (with massless product particles and a V-A coupling). XNE, in terms of which the answer is most simply expressed, is the fractional energy of the outgoing ν_e.

Appendix: Using GAMALG on the MIT-MC Computer

GAMALG may be loaded into a MACSYMA job on the MIT-MC computer by doing

```
LOADFILE(GAMALG,>,SHARE);
```

This will access the latest version of GAMALG, which may differ in some respects from that discussed ~~below~~ above. All the operations described above should, however, work; it is merely possible that GAMALG will be developed further in the future. An updated version of a manual for GAMALG may be printed from the MIT-MC computer by doing (outside MACSYMA)

```
:PR SHARE;GAM USAGE
```

and this should contain all the latest information. Examples of GAMALG may be run in a MACSYMA by doing

```
BATCH(GAM,DEMO,SHARE);
```

In the unlikely event that GAMALG misbehaves, a message should be sent to Terrano and me using (outside MACSYMA)

```
:MAIL SWOLF,TREX
complaint
control C
```

Any suggestions for improvements or modifications will also be welcome.

Table 3.1

Basic GAMALG Notations

N	Dimensionality of spacetime
D(a1,a2)	Dot product of a1 and a2 (which may be 'indices' or 'basis vectors')
DS(a1)	D(a1,a1), = i.e. the square of the four vector a1
EPS(a1,a2,a3,a4)	Totally antisymmetric product of 4-vectors a1, a2, a3 and a4 (which may be 'indices' or 'basis vectors')
G(a1,a2,....,ak)	Product of gamma matrices represented by a1, ..., ak
GT(a1,a2,....,ak)	Undone trace of gamma matrices represented by a1, ..., ak
G5	γ_5 (in n = 4)
LHP	Left-hand projection operator $(1-\gamma_5)/2$
RHP	Right-hand projection operator $(1+\gamma_5)/2$
SIG(mu,nu)	$\sigma_{\mu\nu}$
UV(p,m)	Fermion spinor with momentum p and mass m
UVS(p,m,s)	Polarized fermion spinor with spin s
ZN(p,m)	Numerator of fermion propagator $\not{p} + m$
ZD(p,m)	Fermion propagator $1/(\not{p}-m) = (\not{p}+m)/(p^2-m^2)$

Table 3.2

Summary of Basic GAMALG Functions

CIND(mul,...,muk) adds mul through muk to the list of contracted indices

CGT(exp) converts G's to TR's and does them

COMPDEF(vec1=list1,vec2=list2,ind1=val1,ind2=val2,vec3=...) defines lists as the components of vectors and values for indices, for use by NONCOV

CON(exp) contracts all free indices in exp (including epsilon symbols)

COTR(exp) reduces (in n=4) products of traces with contracted indices or containing γ_5 to single traces

CRUNCH(exp) simplifies untraced products of gamma matrices in exp

DFIX(exp) expands all dot products in exp

EPSFIX(exp) expands all epsilon symbols in exp

FLAGS() displays the values of flags and information lists

GFIX(exp) expands sums of vectors appearing in untraced products of gamma matrices in exp

GLUE3(i1,i2,i3) gives the tensor corresponding to the three-gluon vertex represented by its arguments

KINDEF(dotp1=rep1,dotp2=rep2,...) defines kinematics substitutions dotp1=rep1,...

NONCOV(exp) substitutes the non-covariant components specified by COMPDEF for vectors and indices in dot products in exp

NSET(dim) sets the dimensionality of spacetime to dim

SCALS(x1,..,xk) adds x1 through xk to the list of scalars

SQ(spn1,amp,spn2) squares the amplitude amp sandwiched between the spinors spn1 and spn2

TR(a1,a2,...) takes the trace of gamma matrices represented by its argument

UNCIND(mul,...,muk) removes mul through muk from the list of contracted indices

UNCOMPDEF(vec1,ind1,vec2,vec3,...) removes the components defined for its arguments

UNKINDEF(dotp1,...) removes simplifications defined for dot products dotp1 through dotpk

Table 3.2 (continued)

UNSCALS(x1,...,xk) removes x1 through xk from the list of scalars

ZFIX(exp) expands all $\not{p} + m$ terms appearing in G's and GT's in exp

Table 3.3

Basic GAMALG Flags (Default Values in Brackets) and Information Lists (Initially Empty)

ALGRU[FALSE] if TRUE uses further relations for G reduction in n = 4

BORED[FALSE] if TRUE prints intermediate stages in calculations

COF[FALSE] if TRUE alphabetizes CRUNCH outputs by anticommutation

DEF[TRUE] if FALSE will prevent the expansion of dot products as they are generated

EPSEF[TRUE] if FALSE will prevent expansion of epsilon symbols as they are generated

KAHAF[FALSE] if TRUE will cause the Kahane algorithm to be used on traces with many contracted indices in n = 4

NOP[FALSE] if TRUE causes SQ to generate no primed indices (does Feynman gauge polarization sums)

NTR[FALSE] if TRUE causes SQ to generate G's rather than TR's

VIRED[FALSE] if TRUE generates VIPER-compatible output (see next installment)

ZERM[FALSE] if TRUE assumes all particle masses to be zero

COMPS is the list of components defined by COMPDEF

IND is the list of contracted indices (which will be uncontracted if unpaired)

KINS is the list of kinematic substitutions defined by KINDEF

NPIND is the list of indices automatically summed over by SQ (or SQAM)

SCALARS is the list of scalars

Table 3.4

$$\{\gamma_{\mu_1},\gamma_{\mu_2}\} = 2\, g_{\mu_1\mu_2}$$

$$g^{\mu}_{\mu} = n$$

$$\mathrm{Tr}[1] = 4$$

$$\mathrm{Tr}[\not{a}_1\not{a}_2\ldots\not{a}_k] = 0 \qquad (k \text{ odd})$$

$$\mathrm{Tr}[\not{a}_1\not{a}_2\ldots\not{a}_k] = \sum_{j=2}^{k} (-1)^j (a_1\cdot a_j)\,\mathrm{Tr}[\not{a}_2\ldots\not{a}_{j-1}\not{a}_{j+1}\ldots\not{a}_k]$$

$$= \sum_{\{ij\}} \varepsilon_{i_1 i_2\ldots i_k} (a_{i_1}\cdot a_{i_2})(a_{i_3}\cdot a_{i_4})\ldots(a_{i_{k-1}}\cdot a_i) \qquad (k \text{ even})$$

$$\mathrm{Tr}[\not{S}] = \mathrm{Tr}[S_R]$$

In n = 4 dimensions:

$$\{\gamma_\mu,\gamma_5\} = 0$$

$$(\gamma_5)^2 = 1$$

$$\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3} = \gamma_{\mu_1} g_{\mu_2\mu_3} - \gamma_{\mu_2} g_{\mu_1\mu_3} + \gamma_{\mu_3} g_{\mu_1\mu_2} + \varepsilon_{\mu_1\mu_2\mu_3\alpha}\gamma_5\gamma^{\alpha}.$$

Table 3.4 (continued)

$$\mathrm{Tr}[\not a_1 \ldots \not a_k \gamma_5] = \sum_{i_1<i_2<i_3<i_4} (-1)^{i_1+i_2+i_3+i_4}$$

$$\times \mathrm{Tr}[\not a_{i_1} \not a_{i_2} \not a_{i_3} \not a_{i_4} \gamma_5] \mathrm{Tr}[\not a_{i_5} \ldots \not a_{i_k}]$$

$$= 4 \sum_{\{i_j\}} \varepsilon_{i_1 i_2 \ldots i_k} (\varepsilon_{\mu_1\mu_2\mu_3\mu_4} a_{i_1}^{\mu_1} a_{i_2}^{\mu_2} a_{i_3}^{\mu_3} a_{i_4}^{\mu_4})$$

$$\times (a_{i_5} . a_{i_6}) \ldots\ldots (a_{i_{k-1}} . a_{i_k}) \qquad (k \text{ even})$$

$$= 0 \qquad (k \text{ odd}).$$

$$\gamma_\mu 0 \gamma^\mu = -2\, 0_R$$

$$\gamma_\mu E \gamma^\mu = \mathrm{Tr}[E] - \gamma_5 \mathrm{Tr}[\gamma_5 E]$$

$$= 2\{\mathrm{Tr}[E] - E - E_R\}$$

$$\gamma_\mu (0\gamma_\alpha)\gamma^\mu = 2\{(\gamma_\alpha 0) + (0_R \gamma_\alpha)\}.$$

$$\not p 0 \not p = -p^2 0_R + \frac{1}{2} \not p \{\mathrm{Tr}[\not p 0_R] + \gamma_5 \mathrm{Tr}[\gamma_5 \not p 0_R]\}$$

$$\not p E \not p = -p^2 E_R + \frac{1}{2} \gamma_\alpha \not p \mathrm{Tr}[\not p \gamma_\alpha E_R].$$

$$\gamma_\mu \underline{0} \gamma^\mu = 2\{\mathrm{Tr}[\gamma_5 \underline{0}]\gamma_5 - \underline{0}_R\} - \underline{0} + \bar{0}$$

$$\gamma_\mu \underline{E} \gamma^\mu = 2\{\mathrm{Tr}[\underline{E}] - E_R\} - \underline{E} - \bar{E}.$$

Table 3.4 (continued)

$$\gamma_\mu S_1 \mathrm{Tr}[\gamma^\mu E_2] = 2\{E_2+(E_2)_R\}S_1.$$

$$\mathrm{Tr}[\gamma_\mu O_1]\mathrm{Tr}[\gamma_\mu O_2] = 2\{\mathrm{Tr}[(O_1+(O_1)_R)O_2]\}.$$

$$\mathrm{Tr}[\gamma_5 E_1]\mathrm{Tr}[\gamma_5 E_2] = 2\ \mathrm{Tr}[E_2(E_1+(E_1)_R)] - \mathrm{Tr}[E_1]\mathrm{Tr}[E_2].$$

$$\mathrm{Tr}[\not{p}O_1]\mathrm{Tr}[\not{p}O_2] = \mathrm{Tr}[(O_1+(O_1)_R)(p^2 O_2+\not{p}O_2\not{p})].$$

$$\mathrm{Tr}[\gamma_\mu \underline{O}_1]\mathrm{Tr}[\gamma_\mu \underline{O}_2] = \mathrm{Tr}[(\underline{O}_1+(\underline{O}_1)_R)(\underline{O}_2+\bar{\underline{O}}_2)].$$

Some Gamma Matrix Identities used by GAMALG. S denotes any product (or 'string') of gamma matrices. O and E represent products of gamma matrices which contain an odd and even number of factors (not counting γ_5), respectively. $\underline{S}$ denotes a product in which some of the factors can be of the form $\not{p} + m$, while in $\bar{S}$, all such factors are replaced by $\not{p} - m$.

Acknowledgments

First, I thank the MATHLAB group of the MIT Laboratory for Computer Science for the use of MACSYMA. I am grateful to many people for discussions and encouragement, particularly R. P. Feynman, R. D. Field, G. C. Fox, H. D. Politzer, D. A. Ross and A. E. Terrano. Much of GAMALG was written in collaboration with A. E. Terrano; we will give a more computer-orientated description of its internal workings elsewhere.

References for Section 3:

3.1 S. P. Rosem, J. Math. Phys. 9, 1593 (1968).

3.2 J. S. R. Chisholm, Nuovo Cimento 30, 428 (1963).

3.3 S. Sarkar, Int. J. Theor. Phys. 8, 171 (1973).

3.4 S. Sarkar, Int. J. Theor. Phys. 10, 21 (1974).

3.5 J. S. R. Chisholm, Comp. Phys. Comm. 0, 205 (1972).

3.6 J. Kahane, J. Math. Phys. 9, 1732 (1968).

3.7 L. D. Landau and E. M. Lifshitz, 'The Classical Theory of Fields,' 4th ed. Pergamon 1975, p. 17.

3.8 J. D. Bjorken and S. D. Drell, 'Relativistic Quantum Mechanics,' McGraw-Hill 1964, p. 104.

3.9 J. M. Jauch and F. Rohrlich, 'The Theory of Photons and Electrons,' 2nd ed., Springer 1976, Appendix A2.

3.10 E. R. Caianiello and S. Fubini, Nuovo Cimento 9, 1218 (1952); E. R. Caianiello, 'Combinatorics and Renormalization in Quantum Field Theory,' Benjamin 1973, p. 15; N. R. Ranganathan and R. Vasudevan, Nuovo Cimento 37, 172 (1965).

3.11 M. Veltman, ~~CERN preprint~~ unpublished (1967); H. Strubbe, 'Manual for SCHOONSHIP,' CERN preprint DD/74/S (April 1974).

3.12 J. S. R. Chisholm, Ph.D. thesis, Cambridge University, 1952.

3.13 N. Salingaros and M. Dresden, Phys. Rev. Lett. 43, 1 (1979).

3.14 C. T. Hill, Ph.D. Thesis, Caltech, 1977; T. L. Curtright, private communication.

3.15 W. Becker and Th. Schott, J. Phys. A 10, 125 (1977).

3.16 G. 't Hooft and M. Veltman, Nucl. Phys. B44, 189 (1972).

3.17 D. A. Akyeampong and R. Delbourgo, Nuovo Cimento 17A, 578 (1973).

3.18 W. A. Bardeen, R. Gastmans and B. Lautrup, Nucl. Phys. B46, 319 (1972); M. Chanowitz, M. Furman and I. Hinchliffe, 'The Axial Current in Dimensional Regularization,' LBL preprint LBL-8855 (1979); S. Gottlieb and J. T. Donohue, 'The Axial Vector Current and Dimensional Regularization,' Argonne preprint ANL-HEP-PR-79-08 (1979).

3.19 W. A. Bardeen in Proceedings of the XVI International Conference on High Energy Physics, edited by J. D. Jackson and A. Roberts (National Accelerator Laboratory (1972)).

3.20 A. D. Kennedy, 'Clifford Algebra in 2ω Dimensions,' Sussex preprint (1979) and private communication.

3.21 A. C. Hearn, 'REDUCE Users Manual,' Stanford Artificial Intelligence project memo (1968); J. Comp. Phys. 5, 280 (1970).

3.22 M. J. Levine, J. Comp. Phys. 1, 454 (1967); R. C. Perisho, 'ASHMEDAI Users Guide,' Carnegie-Mellon preprint COO-3066-44 (January 1975).

3.23 D. Barton and J. P. Fitch, Rep. Prog. Phys. 35, 235 (1972).

3.24 J. L. White, 'NIL: A Perspective,' in Proc. 1979 MACSYMA Users' Conf., Washington, D.C. (June 1979).

3.25 J. S. R. Chisholm, J. Comp. Phys. 8, 1 (1971); Nuovo Cimento 1A, 777 (1971).

Co-authored with Tony Terrano

IV. Feynman Integrals: VIPER and COBRA

4.1 Introduction

Whenever a closed loop appears in a Feynman diagram, the corresponding amplitude contains an integral over the undetermined momenta of the virtual particles in the loop. The evaluation of these integrals is the major task in Feynman diagram calculations. To prepare the momentum space integrals for numerical evaluation or for some types of further algebraic mainpulation, it is usually necessary to convert them to Feynman parametric form (see sec. 4.4, however, for a discussion of a direct method which does not use this representation). In this form, auxilliary scalar integration variables called Feynman parameters are introduced, one for each propagator or internal line in the diagram, so that the actual momentum integrals may be performed easily, leaving the integrations over the Feynman parameters. The program VIPER described in sec 4.3 below conberts any momentum space integral to Feynman parametric form, performing the necessary momentum space integrations usingdimensional regularization. The remaining Feynman parameter integrals are often not amenable to analytical evaluation, and must therefore be caluculated numerically, typically using a Monte Carlo method. If required, VIPER can generate a file contaioning the FORTRAN form of the Feynman-parameterized integral, suitable for direct insertion into a numerical integration program. For diagrams with a simple topological structure and in which all the internal particles are massless, it is possible to perform the Feynman prarmetric integrations analytically in terms of beta functions. The program COBRA descibed in sec 4.4 below will give complete analytical results for most momentum integrals which can be evaluated in terms of beta functions.

VIPER and COBRA introduce Feynman parameters sequentially for each loop integration momentum. In practice, they will easily handle integrals over up to three or more momenta, so long as the integrals doe not contain too may terms. Larger ones. may often be treated by multiple applications of VIPER or COBRA.

Co-authored with Tony Terrano

4.2 Dimensional Regularization and Feynman Parametrization

The basic form of the integrals arising from a Feynman diagram is

$$\int d^n p_1 \cdots d^n p_\ell \; \frac{N(k_i\cdot k_j, p_i\cdot k_j, p_i\cdot p_j)}{\prod_i \left[(\sum_j \eta_{ij} p_j + \xi_{ij} k_j)^2 - m_i^2\right]^{\alpha_i}} \tag{4.2.1}$$

where ℓ is the number of closed loops in the diagram. A convenient method for regularizing the divergences in these integrals is to perform the loop momentum (p_i) integrations in (4.2.1) in a spacetime with arbitrary dimensionality n (rather than 4) as indicated by $d^n p_i$. In (4.2.1) the numerator function N of the momenta typically comes directly from performing the gamma matrix traces in a diagram, also in n dimensions. The k_i appearing in it are the momenta of extermal (incoming or outgoing) particles in the diagram. The denominator factors come from the propagators of particles appearing in the diagram. The η_{ij} and ξ_{ij} are matrices depending on the routing and labelling of the momentum variables in the diagram.

The fundamental result for dimensionally regularized integrals is

$$\int d^n p \; \frac{1}{(p^2 + V)^\alpha} = i(-\pi)^{n/2} \frac{\Gamma(\alpha - n/2)}{\Gamma(\alpha)} \; V^{n/2-\alpha} \tag{4.2.2}$$

$$= 0 \quad \text{if } V = 0$$

where we have assumed a Minkowski metric with signature (+,−,−,−). The vanishing of the integral if V=0 must be assumed for all values of n and α. ~~It~~ this assumption is known to be consistent when α is an integer and believed always to be satisfactory. Terms linear in p (e.g. $k\cdot p$) may easily be included in the denominator of (4.4.2) by translating p, as is always permissable in dimensionally-regularized integrals, and for example

$$\int d^n p \; \frac{1}{(k^2 + 2p\cdot k + V)^\alpha} = i(-\pi)^{n/2} \frac{\Gamma(\alpha - n/2)}{\Gamma(\alpha)} (V - p^2)^{n/2-\alpha} \tag{4.2.3}$$

Co-authored with Tony Terrano

The possibility of shifting the variable of integration without altering the value of the integral (which runs over all possible k) will be used extensively below. Any numerator term odd under $k \to -k$ introduced into (4.2.2) clearly gives zero upon integration. For numerator terms containing an even number of k's the generalization of (4.2.2) is found easily (e.g. by induction)

$$\int d^n p \, \frac{p_{\mu_1} p_{\mu_2} \cdots p_{\mu_{2j}} T_{\mu_1 \mu_2 \cdots \mu_{2j}}}{(p^2 + V)^\alpha}$$
$$= \frac{(-\pi)^{n/2} \Gamma(\alpha + j - n/2)}{2^{2j} \Gamma(\alpha)} V^{n/2 + j - \alpha}$$
$$\times \sum_{\text{distinct}\{\lambda_i\}} g_{\mu_{\lambda_1} \mu_{\lambda_2}} \cdots g_{\mu_{\lambda_{2j-1}} \mu_{\lambda_{2j}}} T_{\mu_1 \cdots \mu_{2j}} \qquad (4.2.4)$$
$$= 0 \quad \text{if } V = 0$$

The tensor T in (4.2.4) does not depend on p, but may contain $g_{\mu_i \mu_j}$ factors, thus forming any p^2 terms in the numerator of the integrand. The restriction to distinct sets of λ_i in (4.2.4) forbids terms which differ only by the irrelevant interchange of indices on the individual metric tensors. Some readers may be amused by the similarity of the sum in (4.2.4) to the Pfaffian form (3.3.2) obtained for the trace of a product of gamma matrices. The form (4.2.4) is symmetric, while (3.3.2) is antisymmetric under transpositions of the μ_i.

To cast the integrand of (4.2.1) into a form to which (4.2.2) can directly be applied, it is first necessary to combine the denominator factors in (4.2.1). The most common and in nearly all cases most convenient method for doing this is to introduce auxilliary Feynaman parameter integrals using the Feynmna identity

$$\frac{1}{D_1^{\alpha_1} D_2^{\alpha_2} \cdots D_K^{\alpha_K}} = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int_0^1 dx_1 \cdots \int_0^1 dx_K \, \frac{\delta(1 - \sum_i x_i) \prod_i x_i^{\alpha_i - 1}}{[\sum_i D_i x_i]^{\sum_i \alpha_i}} \qquad (4.2.5)$$

For each of the loop momenta k_i in turn, the resulting combined denominator may be written in the form ($\alpha = \sum \alpha_i$)

Co-authored with Tony Terrano

$$\left[\lambda\left(p_i^2 + 2q\cdot p_i + V\right)\right]^{\alpha} \tag{4.2.6}$$

where q and V depend on the other $p_{j\neq i}$ but not on p_i. Then, by making the change of variables $p_i \to \tilde{p}_i = p_i + q$, the denominator form (4.2.6) becomes

$$\left[\lambda\left(\tilde{p}_i^2 + V - q^2\right)\right]^{\alpha} \tag{4.2.7}$$

so that the linear term in k is thus removed. At this stage, the integration over the loop momentum p_i may be performed by direct application of (4.2.2). The reuslt of this integration may then be integrated succesively over each further p_j in (4.2.1) by repeated application of the same proceedure. This iterative technique is usually termed 'loop-by-loop' Feynman parametrization; it is the method used by VIPER and COBRA as described in some detail below.

There are, in fact, several algorithms for performing the Feynman parametrization and integration described above. All must ultimately give equivalent results although the complexity of intermediate expressions generated differs. An alternative algorithm does not proceed iteratively through each loop, but treats all together. The algorithm begins by combining all of the factors in the denominator according to (4.2.5), and writing the result in the form

$$\int d^n p_1 \cdots \int d^n p_\ell \; \frac{N(k_i, p_i)}{\left(P^T\cdot U\cdot P + 2R^T\cdot K + V\right)^{\alpha}} \tag{4.2.8}$$

where R is a vector consisting of the ℓ loop momenta p_i, and U is an $\ell x \ell$ matrix of Feynman parameters. Completing the square in the denominator of (4.2.8) and diagonalizing with respect to U, the integral becomes

$$\left[\det(U)\right]^{-n/2} \int d^n \bar{p}_1 \cdots d^n \bar{p}_\ell \; \frac{N\left(k_i, (R^{-1}\cdot\bar{P})_i/\sqrt{\lambda_i} - (U^{-1}\cdot P)_i\right)}{\left(\bar{P}^2 + \bar{V}\right)^{\alpha}} \tag{4.2.9}$$

where R is the orthogonal matrix that diagonalizes U (i.e. $R.U.R^T = U_D$, $R^T.R = I$), and λ_i is the i^{th} eigenvalue of U ($(U_D)_{ii}$). The form (4.2.9) may be evaluated by successive direct applications of (4.2.4). One advantage of this method is that

$$\bar{K} = R\cdot(K + U^{-1}\cdot P)$$
$$\bar{V} = V - P^T\cdot U^{-1}\cdot P$$

since the denominator in (4.2.9) is even under $\mathbf{p}_i \rightarrow -\mathbf{p}_i$, all terms in the numerator of (4.2.9) which are odd under $\mathbf{p}_i \rightarrow -\mathbf{p}_i$ may immediately be discarded, since they will vanish upon integration. However, the method leading to (4.2.9) appears to have many disadvantages compared to the simpler loop-by-loop method described above. The foremost of these is that the algebraic functions of the Feynman parameters reuslting from the diagonalization of U are typically extremely complicated for multiloop diagrams (involving many radicals). Mostly as a consequence of this, the final analytical form of the Feynman-parametrized integral is in general much more complicated, thereby hindering further analytical or numerical treatment. In addition, the method by which simple parametric integrals are evaluated analytically in terms of beta functions by COBRA (described in sec. 4.4) relies on loop-by-loop parametrization.

VIPER and COBRA take the momentum space form for a diagram (generated, for example, by GAMALG) and then perform the momentum integrations by introducing Feynman parameters. It is also possible to construct the parametric representation for a diagram directly, without considering the momentum space form. We did not use this method in VIPER because it is rather inflexible; it must udergo fundamental modifications to treat any changes in the momentum space forms of vertices in diagrams. In addition, is usually simpler to perform renormalizations in the momentum space representation. The most sophisticated algorithm for obtaining the Feynman parametric form of a diagram directly was constructed by Cvitanovic and Kinoshita, which requires only the specification of the diagram in terms of graph-thoeretical matrices (corresponding essentially to η and ξ in (4.2.1)). Of course this method is ultimately equivalent to the more straightforward one used by VIPER and COBRA and it certainly generates no fewer terms. The most difficult aspect of the method is the application of parametric differentiation operators to the results of momentum space integrals with numerators equal to 1, as in deriving (4.2.4). The Cvitanovic-Kinoshita algorithm was implemented by Cvitanovic in and

Co-authored with Tony Terrano

elegant TECO/SCHOONSCHIP program and was used in their calculation of the $O(\alpha^3)$ contribution to the muon anomalous magnetic moment. Cvitanovic and Kinoshita have also deised an alogorithm for performing the renormalizations necessary in any ordinary QED graph directly in Feynman parameter space.

The integration region for Feynman parameters in (4.2.5) is an m-simplex (m-dimensional generalization of a tetrahedron). For numerical evaluation, it is convenient to implement the δ function condition $z_i=1$, and to change variables so that the integration region is transformed into an (m-1) diensional hypercube, and the modified (m-1) Feynman parameters all run from 0 to 1. A suitable transformation are the x_i, defined by

$$
\begin{aligned}
z_1 &= (1-x_1) \\
z_2 &= x_1(1-x_2) \\
z_3 &= x_1 x_2 (1-x_3) \\
&\vdots \\
z_j &= \prod_{i=1}^{j-1} x_i (1-x_j) \\
&\vdots \\
z_m &= \prod_{i=1}^{m-1} x_m
\end{aligned}
\qquad (4.2.10a)
$$

This transformation has Jacobian

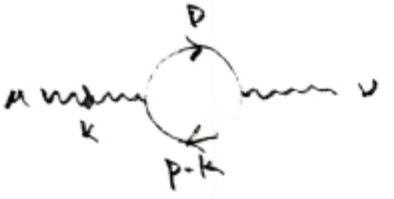
$$
J = \prod_{i=1}^{m-1} x_i = z_m \qquad (4.2.10b)
$$

Co-authored with Tony Terrano

4.3 VIPER

The function VIPER (exp,[p1,p2,...]) Feynman parametrizes the momnetum space integrand exp, and integrates it with respect to the loop momenta p1,p2,... in the order given. The expression exp, which is often generated by using GAMALG, must be written in terms of dot products of momenta (e.g. p·k1, represented by D(p,k1) as in GAMALG), massive propagators (written as ZDEN(p,m) = $p^2 - m^2$), and numerical constants such as the dimensionality of spacetime N. If constants other than N appear, they must be declared using SCALS as in GAMALG. Note that N has the default value 4-2E; it may be reset to any expression exp (e.g. 4-E) by NSET(exp), also as in GAMALG).

As a first example of the use of VIPER we consider the one loop correction to the photon propagator in QED:



$$= -e^2 \int \frac{d^N p}{(2\pi)^N} \mathrm{tr}\left[\gamma_\mu (\not{p}-m)^{-1} \gamma_\nu (\not{p}-\not{k}-m)^{-1}\right] \tag{4.3.1}$$

The trace has been performed using GAMALG; the result is called inp. In the following output, inp is first printed as D1, and then Feynamn parametrized by VIPER.

to be redone

Co-authored with Tony Terrano

When an integral is fed to VIPER, is is simplified by collecting together all terms with the same denominator. Each term in the resulting sum is processed seperately; if the flag INTP:TRUE, VIPER wil print each integrand with a distinct denominator (2) along with the loop momentum over which it is being integrated.

VIPER begins by finding all the terms in the integrand whcih depend on the integration momentum p. Those occuring in the denominator are combined using the Feynman identity (4.2.5) and the coefficient λ of p^2 is extracted. The translation of the loop momentum necessary to complete the square in the denominator is found. Details of this operations are printed if the flag DENF:TRUE (3). The Feynman parameters introduced are named z_{ij} wher i numbers the loop and j labels the parameters for a given loop; for each loop i the parameters are constained by $\sum z_{ij}=1$. The first entry in the output list is the parametrized denominator, the second gives its exponent, and the third gives the remaining factor in (4.2.5). Thus the integrand

is

Co-authored with Tony Terrano

integrand is $$\frac{\text{FACTOR}}{[\lambda\cdot\text{DENOM}]^{\text{POWER}}}\ (\text{numerator}) \qquad (4.3.2)$$

Finally the Feynman parameters which have been introduced are listed. The translation vector QVEC is given, specifying the shift of variables $p \to p+QVEC$ in the loop integration. Next, VIPER constructs this shifted integrand. In the denominator, the δ function in (4.2.2) is used to simplifiy any occurences of the sum of all of the Feynman parameters introduced for a given loop to one. After the numerator terms are shifted, those containing an odd number of p's are dropped, since they vanish upon integration according to (4.2.4). If the flag NUMP:TRUE then the result of shifting each term in the numerator will be printed (3). Next, the formula (4.2.4) is applied to give the integral of each term. Squares of the loop momentum are first written as $g_{\mu\nu}k^{\mu}k^{\nu}$ etc. (Note that if a method of regularization other than the dimensional one is required, only this (rather small) part of the proceedure need be changed.) Adding together the results for each term, the integration over the loop momentum p is completed and in this one loop example the answer is returned (4). Since VIPER works loop-by-loop, its operation for each loop in a multiloop case is analogous to this example: the result after integrating over the first loop momentum becomes the input for the next integration.

In general, the ouput from VIPER should not be factored-- the MACSYMA function FACTOR will usually complicate large expressions rather than simplify them. Here however, the result is simple and by factoring it we see that it is proportional to $g_{\mu\nu} - k_{\mu}k_{\nu}/k^2$ as is required by gauge invariance. In the answer, GAMMA(x) is the gamma function of x, %I = $\sqrt{-1}$, and PI = π . If the flag PIFAC is set to FALSE, a factor $(-4\pi)^{(-n/2)}$ will be omitted for each loop and the final answer will be free of PI and $(-4)^{\epsilon}$.

VIPER applies the recursion relation for Γ functions, $z\Gamma(z) = \Gamma(z+1)$, until the argument of each one is of the form 1+f(N) with f(4)=0. This operation may be prevented by setting GAMSIM:FALSE, but usually leads to important simplifications.

~~Finally we note that the parametric integral in this case can be pefformed~~

Co-authored with Tony Terrano

Finally we note that in thsi case the parametric integral can be performed analytically in terms of hypergeometric functions, yielding the result

check!

$$-\frac{i}{60\pi^2}\left(\frac{4\pi}{m^2}\right)^{\epsilon}\frac{1}{\epsilon}\,{}_2F_1\left(\epsilon,2;5/2;\frac{k^2}{4m^2}\right)\left(k^2 g_{\mu\nu}-k_\mu k_\nu\right) \tag{4.3.3}$$

Alternatively, it may first be Taylor expanded around $\epsilon = 0$. The result to $O(1)$ is sufficiently simple for the MACSYMA function INTEGRATE to perform the final parametric integral over Z11.

As a second example, we consider the integral

$$\int\frac{d^n p}{(2\pi)^n}\frac{d^n q}{(2\pi)^n}\,\frac{k\cdot p}{p^2(p-k)^2(p-q)^2(q-k)^2q^2} \tag{4.3.4}$$

which occurs in the two loop contribution to the photon propagator in massless QED. For most integrals over more than one loop momentum, the expressions which reuslt from the application of eqn. 4.2.2 can be quite cumbersome, and often exceed the maximum expression size which MACSYMA can handle. In addition, most FORTRAN compilers have a limit on the length of individual arithmetic input expressions. For these reasons it is essential to introduce dummy symbolic names for the many repeated subexpressions which are generated during Feynman parametrization. Doing this minimizes the number of computations required in the numerical evaluation of the integral. If DUMMY:TRUE then dummy names will be introduced using the following notation:

DCi is the coefficient of the square of the i^{th} integration variable in the denominator of the $(i-1)^{th}$ integration. The values of the DCi are given as a set of equations in the list DCL.

SCij is the coefficient of the j^{th} momentum in the translation vector QVEC for the i^{th} integration. If a coefficient is sufficiently simple, the corresponding SCij will be left unassigned. The values of the assigned SCij

Co-authored with Tony Terrano

are given in the list SCL.

NCi is the coefficient of the i^{th} term generated in the shifted numerator. Again, some NCi may be left unassigned. The values of those used are givne in the list NCL.

FID is the final denominator. Its value is givne as an equation in the list FIDL.

When saving a result in a disk file, all of the lsits above and the list ZL of Feynman parameters as well as the final answer must be saved. The function EGG(NAME, ANS,fn1,fn2,dir) will save all required expressions in the disk file [fn1,fn2,dir], assigning the answer ANS to the variable NAME.

The Feynman parametrization of the integral (4.3.4) (whose integrand is denoted by inp) is performed as follows:

```
(C6) inp;
                                    D(K, P)
(D6)  ------------------------------------------------------------
      D(P, P) D(P - K, P - K) D(P - Q, P - Q) D(Q, Q) D(Q - K, Q - K)

(C7) ans2:viper(inp,[P,Q]);
INTEGRATING

                                D(K, P)
  ------------------------------------------------------------  OVER  P
  D(P, P) D(P - K, P - K) D(P - Q, P - Q) D(Q, Q) D(Q - K, Q - K)

[DENOM,POWER,FACTOR,PARAMETERS]:
 [Z13 (D(Q, Q) - 2 D(P, Q)) + D(P, P) + Z12 (D(K, K) - 2 D(K, P)), 3, 2, Z11,
Z12, Z13]

QVEC= - Z13 Q - Z12 K

TRANSLATED NUMERATOR    Z13 D(K, Q) + D(K, P) + Z12 D(K, K)
INTEGRATING

    - E - 1                       - E - 1
  DC2        Z13 GAMMA(E + 1) FID        D(K, Q)
  ------------------------------------------  OVER  Q
            D(Q, Q) D(Q - K, Q - K)

[DENOM,POWER,FACTOR,PARAMETERS]:

                          2
            Z23 ((Z12 - Z12 ) D(K, K) - 2 Z12 Z13 D(K, Q))
 [D(Q, Q) + ----------------------------------------------
                                DC2

                                                          E
 + Z22 (D(K, K) - 2 D(K, Q)), E + 3, (E + 1) (E + 2) Z23 , Z21, Z22, Z23]

        (Z12 Z13 Z23 + DC2 Z22) K
QVEC= - -------------------------
                   DC2
```

(1) (2) (3)

Co-authored with Tony Terrano

(4) TRANSLATED NUMERATOR D(K, Q) - SC21 D(K, K)
INTEGRATING

$$\frac{DC2^{-E-1}\ Z12\ \mathrm{GAMMA}(E+1)\ \mathrm{FID}^{-E-1}\ D(K,K)}{D(Q,Q)\ D(Q-K,Q-K)}\ \text{OVER}\ Q$$

[DENOM,POWER,FACTOR,PARAMETERS];

$$[D(Q,Q) + \frac{Z23\ ((Z12 - Z12^2)\ D(K,K) - 2\ Z12\ Z13\ D(K,Q))}{DC2}$$

$$+ Z22\ (D(K,K) - 2\ D(K,Q)), E+3, (E+1)(E+2)\ Z23^E, Z21, Z22, Z23]$$

$$\mathrm{QVEC} = -\frac{(Z12\ Z13\ Z23 + DC2\ Z22)\ K}{DC2}$$

TRANSLATED NUMERATOR D(K, K)

$$(D8)\ DC2^{-E-1}\ SC21\ Z13\ Z23^E\ \mathrm{GAMMA}(2E+1)\ \mathrm{FID}^{-2E-1}\ D(K,K)$$

$$- DC2^{-E-1}\ Z12\ Z23^E\ \mathrm{GAMMA}(2E+1)\ \mathrm{FID}^{-2E-1}\ D(K,K)$$

The integration (1) over the first loop proceeds as in the previous example. The dummy variables make their appearance in the second loop integration. The coefficient of Q^2 in the denominaotr factor reuslting from the first loop integration (over P) is extracted and called DC2 in the input to the second loop integration (over Q). The exact expression for QVEC is printed (3), however the dummy SC21 has been substituted for the coefficient of K in the translated numerator (4). The values for the dummy variables can be found in the lists specified above:

(D9) dc1;

$$(D9)\ [DC2 = -2\ (Z13 - 1)\ Z13]$$

(C10) sc1;

$$(D10)\ [SC21 = -\frac{Z12\ Z13\ Z23 + DC2\ Z22}{DC2}]$$

(C11) fid1;

$$(D11)\ [FID = -\frac{Z12^2\ Z13^2\ Z23^2\ D(K,K)}{DC2^2} - \frac{Z12\ (2\ Z13\ Z22 + Z12 - 1)\ Z23\ D(K,K)}{DC2}$$

$$- (Z22 - 1)\ Z22\ D(K,K)]$$

Even for this relatively simple case, the use of the dummies simplifies the answer considerably. The values of the dummies may be substituted into the final answer ANS by using the function UNDUM(ANS).

Co-authored with Tony Terrano

to be redone

where EXPT(A,B) is MACSYMA's print representation for A^B when A is more than a single line long.

Once the parametric form of the integrand has been found, the parametric integrations must be uncoupled. Since the Feynman parameters were introduced independently for each loop integration, the transformation 4.2.10 must be applied to each set seperately. The function CUBIFY(exp,flag) will transform exp, a function of the Zij into a function of variables Xi all of which are to be integrated from 0 to 1. If flag=J then the result will be multiplied by the Jacobian of the transformations; if flag is omitted, the Jacobians will be omitted as well.

The funciton MONTE(NAME,exp,fn1,fn2,dir,flag) prepares a FORTRAN input file suitable for immediate insertion into a numerical integration program. The VIPER ouput exp is given the name NAME, and a file with the name fn1,fn2 in directory dir is created containing the result of applying the transformation 4.2.10 to the value lists and exp. If necessary, additional dummy variables are defined so that each arithmetic statement is less than ten lines long, as required by most FORTRAN compilers. If the final argument to MONTE is W a new file will be created; if it is A, the results will be appended to the existing file with the given name. Note that anything typed by the user also is written into the file. The integral we have been considering in the second example is finite, so we may

Co-authored with Tony Terrano

set $\epsilon = 0$ before proceeding.

```
(C13) ans2:ev(ans2,e=0);
                          SC21 Z13 D(K, K)   Z12 D(K, K)
(D14)                     ---------------- - -----------
                               DC2 FID           DC2 FID

(C15) monte(ansf,ans2,viper,test,trex,w);
      FORTRAN
      DC2 = -2*X1*X2*(X1*X2-1)
      SC21 = -(X1**2*(1-X2)*X2*X4*X5+DC2*X4*(1-X5))/DC2
      FID = -D(K,K)*X1**4*(1-X2)**2*X2**2*X4**2*X5**2/DC2**2-D(K,K)*X1*(
     1   1-X2)*X4*(2*X1*X2*X4*(1-X5)+X1*(1-X2)-1)*X5/DC2-D(K,K)*X4*(X4*(
     2   1-X5)-1)*(1-X5)
      EX1 = SC21*D(K,K)*X1**2*X2**2*X4*X5/(DC2*FID)
      EX2 = -D(K,K)*X1**2*(1-X2)*X2*X4*X5/(DC2*FID)
      ANSF = EX2+EX1
                              [VIPER, TEST, DSK, TREX]
```

The contents of the file begins with the line following the word FORTRAN, and ends with the line ANSF = ... The final line gives the name of the file into which the preceeding lines have been written.

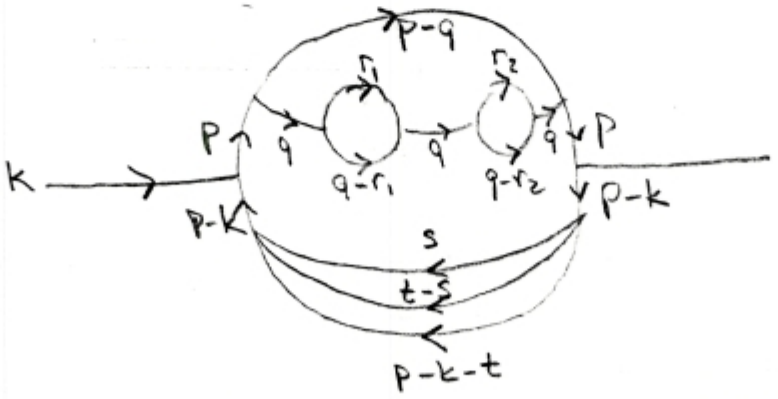
4.4 COBRA

When a Feynman diagram involves only massless internal particles, many of the integrals involved may be performed analytically. The function COBRA(exp,[p1,p2,...,pn]) attempts to integrate the expression exp, typically output from GAMALG, over the loop momenta p1,p2,...,pn. The order in which the pi are given is unimportant. The integrand will in geneeral be a sum of terms, each with a different product of dot products in the denominator. These factors in the denominator must not be expanded (i.e. DFIXed), for expample, D(p-q,p-q) cannot be written as D(p,p)-2D(p,q)+D(q,q). If the flag VIRED:TRUE, all output from GAMALG will be compatible with COBRA.

The algorithm of COBRA is based on the formula

$$\int \frac{d^n p}{(p-k_1)^{2\alpha_1}(p-k_2)^{2\alpha_2}} = (-1)^{n/2}\,\frac{\Gamma(\alpha_1+\alpha_2-n/2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}\,\frac{\Gamma(n/2-\alpha_1)\,\Gamma(n/2-\alpha_2)}{\Gamma(n-\alpha_1-\alpha_2)}\,\frac{1}{[(p_1-p_2)^2]^{\alpha_1+\alpha_2-n/2}} \qquad (4.4.1)$$

In diagrammatic terms this states thath the result of integrating a subdiagram of the form  is a propagator raised to a non-integer power multiplied by a constant factor (usually a product of gamma functions). Since eqn 4.4.1 holds for arbitrary α_1 and α_2, it can be applied iteratively. As an example, consider the (useless) diagram



corresponding to the integral:

$$\frac{1}{[(2\pi)^n]^\epsilon}\int \frac{d^n p\, d^n q\, d^n r_1\, d^n r_2\, d^n s\, d^n t}{(p^2)^2 (q^2)^3 (p-q)^2 r_1^2 (q-r_1)^2 r_2^2 (q-r_2)^2 ((p-k)^2)^2 s^2 (t-s)^2 (p-k-t)^2}$$

(4.4.2)

If the integrations are done in the order r_1, r_2, q, s, t, p then this integral can be evaluated completely using eqn 4.4.1. Denoting the integrand of (4.4.2) by inp, the integral (4.4.2) is performed by COBRA as follows:

```
(C1) inp; PIFAC ; CAPS
(D1) 1/(D 2(P - K, P - K) D(- T + P - K, - T + P - K) D 2(P, P) D 3(Q, Q)
 D(P - Q, P - Q) D(R1, R1) D(Q - R1, Q - R1) D(R2, R2) D(Q - R2, Q - R2)
 D(S, S) D(T - S, T - S))
(C2) cobra(inp,[s,r1,p,q,r2,t]); ← order

SENDING  1/(D 2(P, P) D 2(P - K, P - K) D(P - Q, P - Q) D 3(Q, Q)
 D(Q - R1, Q - R1) D(R1, R1) D(Q - R2, Q - R2) D(R2, R2) D(S, S)
 D(- T + P - K, - T + P - K) D(T - S, T - S))   TO EULER [R1, R2, Q, S, T, P]

INTEGRATING    1/(D 2(P, P) D 2(P - K, P - K) D(P - Q, P - Q) D 3(Q, Q)
 D(Q - R1, Q - R1) D(R1, R1) D(Q - R2, Q - R2) D(R2, R2) D(S, S)
 D(- T + P - K, - T + P - K) D(T - S, T - S))   OVER    R1

INTEGRATING    GAMMA 2(1 - E) GAMMA(E) D(Q, Q)
                                                    - E - 3
/(GAMMA(2 - 2 E) D 2(P, P) D 2(P - K, P - K) D(P - Q, P - Q) D(Q - R2, Q - R2)
 D(R2, R2) D(S, S) D(- T + P - K, - T + P - K) D(T - S, T - S))   OVER    R2

INTEGRATING    GAMMA 4(1 - E) GAMMA 2(E) D(Q, Q)
                                                    - 2 E - 3
       2              2              2
/(GAMMA (2 - 2 E) D (P, P) D (P - K, P - K) D(P - Q, P - Q) D(S, S)
 D(- T + P - K, - T + P - K) D(T - S, T - S))   OVER    Q

INTEGRATING    GAMMA(- 3 E - 1) GAMMA 5(1 - E) GAMMA 2(E) GAMMA(3 E + 2)
          - 3 E - 4       2
 D(P, P)          /(GAMMA (2 - 2 E) GAMMA(- 4 E) GAMMA(2 E + 3)
  2
 D (P - K, P - K) D(S, S) D(- T + P - K, - T + P - K) D(T - S, T - S))
  OVER    S

INTEGRATING    GAMMA(- 3 E - 1) GAMMA 7(1 - E) GAMMA 3(E) GAMMA(3 E + 2)
          - 3 E - 4       3
 D(P, P)          /(GAMMA (2 - 2 E) GAMMA(- 4 E) GAMMA(2 E + 3)
  2                                                  E
 D (P - K, P - K) D(- T + P - K, - T + P - K) D (T, T))   OVER    T
```

Co-authored with Tony Terrano

```
                                                    8          2
(4)  INTEGRATING    GAMMA(- 3 E - 1) GAMMA (1 - E) GAMMA (E) GAMMA(2 E - 1)

                                    1 - 2 E          - 3 E - 4
     GAMMA(3 E + 2) D(K - P, K - P)          D(P, P)

                                    2                                  2
     /(GAMMA(3 - 3 E) GAMMA (2 - 2 E) GAMMA(- 4 E) GAMMA(2 E + 3) D (P - K, P - K))

       OVER    P
                                                    8          2
(D2) - GAMMA(- 3 E - 1) GAMMA(1 - 3 E) GAMMA (1 - E) GAMMA (E) GAMMA(2 E - 1)

                                                             - 6 E - 3
     GAMMA(3 E + 2) GAMMA(4 E - 2) GAMMA(6 E + 3) D(K, K)

                                                    2
     /(GAMMA(- 7 E - 1) GAMMA(3 - 3 E) GAMMA (2 - 2 E) GAMMA(4 E) GAMMA(2 E + 1)

      GAMMA(2 E + 3) GAMMA(3 E + 4))
```

Here the dimensionality of spacetiem has been taken to be N = 4 -2E but may be re-assigned to any expression by NSET(exp). Also a factor $(-4\pi)^{(-N/2)}$ has been suppressed (by setting PIFAC:FALSE as in VIPER). COBRA begins by searching for an order in which the integrations may be performed using eqn. 4.4.1. If it succeeds and the flag SENP:TRUE then it will display the integrand and the order of integration (1), saying thath the integrand had been sent to EULER -- in this case the parametric integrals are just beta functions (i.e. Euler integrals of the first type)/ If no ordering suffices, then the integrand is displayed as sent to GEGENBAUER, in which case a table of results obtained using Gegenbauer expansions is searched (see below). If the required result does not appear in the table, then COBRA simply returns the integrand. (When the input is a sum of terms, the final output will consist of the results of those whcih can be integrated analytically added to any intractable integrands). In the example (4.4.2) there are 6! = 720 possible orderings of the integrations. These orderings are searched by first permuting the last two momenta in the input list, then the last three, and so on. The time spent in seraching for an acceptible ordering can be significantly reduced by specifying the loop momenta in a usable order. ~~In addition, the list of permutations of five of more momenta will take up a considerable amount of space in core, perhaps even so much that the rest of the calcualtion cannot be completed.~~

Co-authored with Tony Terrano

For integrals involving more than four loops, the integration will be attempted in the order given in the input list; the permutations will be ~~generated~~ attempted only if this fails.

In the example (4.4.2) the integrations over r_1 and r_2 are performed first (2). Denoting $(K^2)^{-\beta}$ by $\overset{\beta}{\text{———}}$ if $\beta \neq 1$ the result may be indicated schematically by

Next the integrations over q,s,t are performed (3):

Finally, the p integration is performed (4).

Eqn 4.4.1 only guarantees complete evaluation of diagrams with a rather simple topological structure. Nevertheless, a large fraction of the integrals arising in massless QED and QCD from diagrams with a more complicated topological structure can still be evaluated using only eqn 4.4.1. The basic proceedure is to expand numerator factors involving the loop momenta as sums of terms which will cancel factors in the denominator using the algebraic identity

$$p\cdot q = \tfrac{1}{2}\left(p^2 + q^2 - (p-q)^2\right) \qquad (4.4.3)$$

For example the integral $\int d^n p\, d^n q \, \frac{p\cdot q}{p^2 (p-k)^2 (p-q)^2 (q-k)^2 q^2}$

which occurs in evaluation of the diagram 

~~would be written as~~:

would be written as:

$$\frac{1}{2}\int d^4p\, d^4q \left\{ \frac{1}{(p-L)^2 (p-q)^2 (q-L)^2 q^2} - \frac{1}{p^2 (p-L)^2 (q-L)^2 q^2} + \frac{1}{p^2 (p-L)^2 (p-q)^2 (q-L)^2} \right\}$$

COBRA first implements the identity (4.4.3) for each possible dot product, expanding the result. It then combines those terms which have the same denominator. In a few cases, it may fail to make a possible substitution and as a result be unable to integrate a particular term. The user may define the required expansion with the function NUMEV(epx,replacement) where replacement is exp written as a sum of squares.

As an example of the use of COBRA, we consider the evaluation of the one-loop correction to the vacuum polarization in massless QED. (The calculation of the massive one loop vacuum polarization was discussed as an example of VIPER above.) The integrand is obtained from GAMALG and called inp1.

```
(C1) pifac:false$
                    CAPS
(C2) inp1;
(D2) (8 D(MU, P) D(NU, P) - 4 D(K, MU) D(NU, P) - 4 D(K, NU) D(MU, P)

                                                                  4 D(MU, NU)
          + 4 D(K, P) D(MU, NU))/(D(P, P) D(P - K, P - K)) - ---------------
                                                             D(P - K, P - K)

(C3) cobra(inp1,[p,q]);
INTEGRATING    (8 D(MU, P) D(NU, P) - 4 D(K, MU) D(NU, P)
(1)
 - 4 D(K, NU) D(MU, P) + 4 D(K, P) D(MU, NU))/(D(P, P) D(P - K, P - K))

  OVER    P

[DENOMINATOR,POWER,FACTOR,QVEC]:
(2)
 [D(P, P) + Z11 Z12 D(K, K), 2, 1, - Z11 K]

TRANSLATED NUMERATOR    D(MU, P) D(NU, P) + Z11 D(K, MU) D(NU, P)

                                                    2
 + Z11 D(K, NU) D(MU, P) + Z11  D(K, MU) D(K, NU)
(3)
TRANSLATED NUMERATOR    D(K, MU) D(NU, P) + Z11 D(K, MU) D(K, NU)

TRANSLATED NUMERATOR    D(K, NU) D(MU, P) + Z11 D(K, MU) D(K, NU)

TRANSLATED NUMERATOR    D(K, P) D(MU, NU) + Z11 D(K, K) D(MU, NU)

(D3) - 4 %I (E - 1) GAMMA(1 - E) GAMMA(E + 1)
                          2
 (D(K, K) D(MU, NU) - D(K, MU) D(K, NU))

                                                    E
/(E (2 E - 3) (2 E - 1) GAMMA(1 - 2 E) D (K, K))
```

If the flag INTP:TRUE then each term being integrated over each momentum will be printed (1). COBRA works on each term in turn and begins by identifying the denom-

Co-authored with Tony Terrano

inator factors which involve the first integration momentum. If the denominator contains only one such factor, then the integral vanishes according to eqn 4.4.1. If there are more than two factors, the integral cannot be performed using eqn 4.4.1 and the integrand is returned as intractable. If there are just two factors in the dneominator, then the factos are combined by introducing two Feynman parameters and the translation of the origin required to complete the square in the resulting denominator is found. If the flag DENP:TRUE, a list of details of the denominator's treatment is printed (2), as in VIPER. The integrand is given in terms of the elements of this list by $\frac{\text{FACTOR}}{[\text{DENOM}]^{\text{POWER}}}$ (numerator). The Feynamn parameters introduced for the i^{th} loop inetgration are denoted by Zij where j = 1 or 2: Zi2 = 1-Zi1. The translation vector is called QVEC. The loop momentum is translated by p → p+QVEC in the numerator: the resulting numerator terms are printed if the flag NUMP:TRUE (3). Terms containing an odd number of factors of the loop momentum are dropped since they vanish upon integration; the remaining terms are then integrated over p using eqn 4.2.4. The necessary parametric integrations are performed by application of

$$\int_0^1 z_{i1}^{\alpha_1} z_{i2}^{\alpha_2} \delta(1-z_{i1}-z_{i2})\, dz_{i1}\, dz_{i2} = B(1+\alpha_1, 1+\alpha_2)$$

where B(x,y) is the Euler beta function. In results from COBRA, GAMMA(x) and %I denote $\Gamma(x)$ and $\sqrt{-1}$ respectively.

The proceedure for evaluating multiloop integrals differs form that in the one loop case only at the beginning and end. Those integrals which will not yield to repeated applications of eqn 4.4.1 are singled out and sent to a seperate set of routines called GEGENBAUER (see below). And those amenable to eqn 4.4.1 must be passed iteratively to COBRA for each successive loop integration. Note finally that as in VIPER the reucrrence relation for gamma funcitons is used fo write the argument of each gamma function as 1 + f(N) with f(4)=0. This simplification may be prevented by setting GAMSIM:FALSE.

Some massless integrals not tractable with eqn 4.4.1 may be evaluated by

Co-authored with Tony Terrano

expanding each propgator in a seies of n (=2j+2) dimensional hyperspheical harmonics C_j^λ (Gegenbauer polynomials):

$$[(p-q)^2]^{-\rho} = (p^2-2p\cdot q+q^2)^{-\rho}$$

$$= \frac{1}{(|p||q|)^\rho}\frac{\Gamma(\lambda)}{\Gamma(\rho)}\sum_{j=0}^{\infty}\frac{\Gamma(j+\rho)}{\Gamma(j+\lambda)}T(p,q)^{j+\rho}\,{}_2F_1(-j,\rho;\tfrac{1}{2}j+\lambda;\tfrac{1}{2}j+\lambda+\tfrac{1}{2};T^2(p,q))\,C_j^\lambda(\hat{p}\cdot\hat{q}) \qquad (4.4.4)$$

where $T(p,q) = \min(\frac{p}{q},\frac{q}{p})$. Some properties of the hyperspherical polynomials are given in table 4.1. The orthogonality relations for the hyperspherical harmonics allow the angular part of the loop momentum integrations to be done easily. The remaining radial integrals become trivial when the hypergeometric funcitons are replaced by their power series expansions:

$${}_2F_1(a,b;c;z) = \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)}\sum_{j=0}^{\infty}\frac{\Gamma(a+j)\Gamma(b+j)}{j!\,\Gamma(c+j)}z^j \qquad (4.4.5)$$

Ocaisionally the resulting series may be summed exactly in terms of known functions. Usually, however, the series must first be expanded around $\epsilon = 0$. The coefficients of each power of ϵ will then be a series which, in relatively simple diagrams, can be summed.

As an exapmle of these methods, consider the integral:

$$\int d^n p\, d^n q\, \frac{1}{(p^2)^\rho((p-k)^2)^\sigma(p-q)^2(q-k)^2 q^2}$$

Particular cases of this arise in the calculation of vaccum polarization at three loops in QED. This integral is evaluated by COBRA as follows:

```
(C1) inp;     caps
                                        1
(D1) ------------------------------------------------------------------
      RHO         SIG
     D   (P, P) D   (P - K, P - K) D(P - Q, P - Q) D(Q, Q) D(Q - K, Q - K)

(C2) scals(rho,sig)$

(C3) cobra(inp,[p,q])$
SENDING

                                      1
------------------------------------------------------------------------
 RHO         SIG
D   (P, P) D   (P - K, P - K) D(P - Q, P - Q) D(Q, Q) D(Q - K, Q - K)

TO ~~CHEBYSHEF~~ GEGENBAUER
```

Co-authored with Tony Terrano

```
INTEGRATING

                                    1
------------------------------------------------------------------------ OVER
 RHO         SIG
D   (P, P) D    (P - K, P - K) D(P - Q, P - Q) D(Q, Q) D(Q - K, Q - K)

[P, Q]

                2 E   2 E       2
(D3) - (- 4)     PI    GAMMA (1 - E) GAMMA(E)

 (GAMMA(- SIG - RHO - 2 E + 3) (GAMMA(- RHO - 2 E + 2) GAMMA(RHO)

 GAMMA(- SIG - E + 1) GAMMA(SIG + E) + GAMMA(- RHO - E + 1) GAMMA(RHO + E)

 GAMMA(- SIG - 2 E + 2) GAMMA(SIG)) GAMMA(SIG + RHO + 2 E - 2)

 + GAMMA(- RHO - E + 1) GAMMA(RHO + E) GAMMA(- SIG - E + 1)

 GAMMA(- SIG - RHO - 3 E + 3) GAMMA(SIG + E) GAMMA(SIG + RHO + E - 1))

         - SIG - RHO - 2 E + 1          2
D(K, K)                          /(256 PI2  GAMMA(- 2 (E - 1)) GAMMA(RHO)

GAMMA(RHO + E) GAMMA(- SIG - RHO - 3 E + 3) GAMMA(- SIG - RHO - 2 E + 3)

GAMMA(SIG) GAMMA(SIG + E))
```

When COBRA finds that the required integrations cannot be performed using eqn 4.4.1, it prints that the integrand (1) has been sent to GEGENBAUER; the printout may be prevented by setting SENP:FALSE. If ITNP:TRUE then GEGENBAUER itself will print each integrand (2) it receives and list the momenta being integrated.

The GEGENBAUER ruotines do not evaluate integrals by an algorithm; they merely use the table of results given in table 4.2. Note that some of the table entries are just the first few temrs in the Laurent expansion around N = 4, and not the exact answer for arbitrary N. For example, consider the integral:

$$\int d^n p\, d^n q \; \frac{1}{p^2 (p-k)^2 ((p-q)^2)^{1+\epsilon} (q-k)^2 q^2}$$

which arises from the diagram:

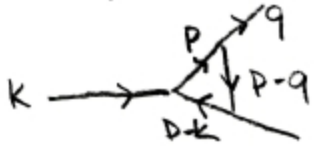
and is multiplied by a factor of $\Gamma(\epsilon)$ resulting from the integration over the internal fermion line (see the first example in this section above). The relevant integral may be evaluated by COBRA as follows.

Co-authored with Tony Terrano

```
(C1) inp1;
                                                    - E - 1
                                  D(P - Q, P - Q)
(D1)          ------------------------------------------------------
              D(P, P) D(P - K, P - K) D(Q, Q) D(Q - K, Q - K)

(C2) cobra(inp1,[p,q]);
                                                - E - 1
                               D(P - Q, P - Q)
SENDING       ------------------------------------------------------
              D(P, P) D(P - K, P - K) D(Q, Q) D(Q - K, Q - K)

TO CHEBYSHEF


                                                - E - 1
                               D(P - Q, P - Q)
INTEGRATING   ------------------------------------------------------ OVER [P, Q]
              D(P, P) D(P - K, P - K) D(Q, Q) D(Q - K, Q - K)
                 2 E   2 E      2
(D2) - (- 4)    PI    (4 E  DUM1 + 2 E (- 8 ZETA(4) + (6 - 6 %GAMMA) ZETA(3)

                                                        - 3 E - 1          2
                          2
                  + 5 ZETA (2)) + 6 ZETA(3)) D(K, K)              /(256 PI2 )
```

Thus to get the correct finite part for the entire diagram, the result for the final two loops must be known to $O(\epsilon)$. Similarly when a counterterm diagram is being calculated, the integral also will be multiplied by factors containing poles. If the expansion in the table is to $O(\epsilon^j)$ ans the factor contains k^{th} order poles, their product will be correct only to $O(\epsilon^{j-k})$. To avoid mistakes, each expanded answer has added to it a term DUMiϵ^{j+1}. the expansion of the final answer is then correct to the highest order in ϵ whose coefficient contains no DUMi.

We have not attempted to write a compete systematic program impelmenting the hyperspherical expansion technique. In the first place, MACSYMA has only a very primative facility for performing summations, which is quite inadequate for those required. In addition, the terms in the final sum which contribute to a given order in ϵ may be identified manually without difficulty; a systematic program to find them would be quite complex. Nevertheless, we have written several programs which perform some of the required expandsions. As an example we consider the one loop scalar vertex disgram.



The required integral is

$$\int d^n p \, \frac{1}{p^2 (p-k)^2 (p-q)^2}$$

The function GEGEX(exp,[p1,...]) makes the expansions 4.4.4 and 4.4.5 for each propa-

Co-authored with Tony Terrano

gator which depends upon one of the integration variables p_i.

```
(C1) inp:1/ds(k)/ds(k-p)/ds(k-q)$

                                  1
(D1)          ------------------------------------
              D(K, K) D(K - P, K - P) D(K - Q, K - Q)

(C2) ans:sesint(inp,[k],[p,q])$

                                                                    2
(D2) (M1 - E + 1) (M2 - E + 1) GAMMA (1 - E) GAMMA(J1 + E) GAMMA(J2 + E)

                                                  2 (1 - E) - 3          M1 + 2 J1 + 1
 GAMMA(M1 + J1 + 1) GAMMA(M2 + J2 + 1) K                       T(K, P)

                          M2 + 2 J2 + 1
 CH(K, P, M1) T(K, Q)                    CH(K, Q, M2)

       2
/(GAMMA (E) GAMMA(J1 + 1) GAMMA(J2 + 1) GAMMA(M1 + J1 - E + 2)

 GAMMA(M2 + J2 - E + 2) P Q)
```

Here N = 4-2E so λ = 1-E, $C^{\lambda}_{m_1 m_2}(p\cdot q)$ is denoted by CH(P,Q,M1), and P = p^2 etc. A summation variable m_i running from 0 to infinity is introduced for each propagator. Additionally, each hypergeometric function is expanded in its power series (4.4.5), introducing a further variable j_i each also running from 0 to infinity. Note that the answer is proportional to $\Gamma(j_1+\epsilon)\Gamma(j_2+\epsilon)/\Gamma^2(\epsilon)$. This factor is $O(\epsilon^2)$ if j_1 and j_2 are both greater than 0, $O(\epsilon)$ if either vanishes, and O(1) if both do. The final ~~radi~~ radial integration may introduce poles for particular values of j_1 and j_2. The isolation of the coefficient each power of ϵ is left to the user.

The function CHINT(exp,var) performs the angular integration of exp over var. For example

```
(C3) chint(ch(k,p,m1)*ch(k,q,m2),k)$

                                  2 - E
                           2 PI        CH(P, Q, M1)
(D3)                       ------------------------
                           (M1 - E + 1) GAMMA(1 - E)
```

Which is just the orthogonality relation ( ) in Table 4.1

Continuing with the scalar one-loop vertex diagram,

```
(C4) chint(ans,k)$

                2 - E
(D4) 2 PI             (M1 - E + 1) GAMMA(1 - E) GAMMA(J1 + E) GAMMA(J2 + E)

 GAMMA(M1 + J1 + 1) GAMMA(M1 + J2 + 1) K^(2 (1 - E) - 3)          M1 + 2 J1 + 1
                                                    T(K, P)

          M1 + 2 J2 + 1                                  2
 T(K, Q)                  CH(P, Q, M1)/(GAMMA (E) GAMMA(J1 + 1) GAMMA(J2 + 1)

 GAMMA(M1 + J1 - E + 2) GAMMA(M1 + J2 - E + 2) P Q)
```

If more than two of the C_m depend on var, CHINT simply returns the input. There is

Co-authored with Tony Terrano

no known way to perform the integral of three C_m^{λ}'s with different arguments. If two of the arguments are the same, eqn.3 in table 4.2 may be used.

Finally the function RADINT(exp,[k1,...,kr],[p1,...,ps]) will perform the radial integrations over the k_i; the p_i are the external momenta. RADINT uses a table of results which at present contains the value of the integral for the cases r=1,2,3,s=1.

Co-authored with Tony Terrano

Table II

$$\int d^n p\, d^n q \; \frac{1}{[p^2]^\alpha [(p-k)^2]^\beta (p-q)^2 (q-k)^2 q^2}$$

$$= -(-\pi)^n (k^2)^{1-\alpha-\beta-2\epsilon} \frac{\Gamma(\epsilon)\Gamma^2(1-\epsilon)}{\Gamma(2-2\epsilon)} \left\{ \frac{\Gamma(2-\alpha-2\epsilon)\Gamma(1-\beta-\epsilon)\Gamma(-2+\alpha+\beta+2\epsilon)}{\Gamma(\alpha+\epsilon)\quad \Gamma(\beta)\quad \Gamma(3-\alpha-\beta-3\epsilon)} \right.$$

$$+ \frac{\Gamma(1-\alpha-\epsilon)\Gamma(2-\beta-2\epsilon)\Gamma(-2+\alpha+\beta+2\epsilon)}{\Gamma(\alpha)\quad \Gamma(\beta)\quad \Gamma(3-\alpha-\beta-3\epsilon)}$$

$$\left. + \frac{\Gamma(1-\alpha-\epsilon)\Gamma(1-\beta-\epsilon)\Gamma(-1+\alpha+\beta+\epsilon)}{\Gamma(\alpha)\quad \Gamma(\beta)\quad \Gamma(3-\alpha-\beta-2\epsilon)} \right\}$$

$\alpha = \beta = 1$

$\alpha = 1+\epsilon, \quad \beta = 1$

$\alpha = \epsilon, \quad \beta = 1$

$$\int d^n p\, d^n q \; \frac{1}{p^2 (p-k)^2 [(p-q)^2]^{1+\epsilon} (q-k)^2 q^2}$$

$$= \frac{-(-\pi)^n}{(k^2)^{1+3\epsilon}} \left\{ 6\zeta(3) + 2\epsilon\left(\frac{9}{5}\zeta^2(2) + 6(1-\gamma)\zeta(3)\right) + O(\epsilon^2) \right\}$$

$$\int d^n p\, d^n q \; \frac{1}{p^2 (p-k)^2 [(p-q)^2]^{\epsilon} (q-k)^2 q^2}$$

$$= -(-\pi)^n (k^2)^{-3\epsilon} \left\{ \frac{1}{3\epsilon^2} + \frac{5-2\gamma}{3\epsilon} + \frac{17 - 5\zeta(2) + 2\gamma^2 - 10\gamma}{3} \right.$$

$$\left. + \frac{\epsilon}{9}\left(123 - (15-6\gamma)\zeta(2) + 28\zeta(3) - 4\gamma^3 + 30\gamma^2 - 102\gamma\right) + O(\epsilon^2) \right\}$$

$$\int d^n p\, d^n q \; \frac{[(p-q)^2]^{1-\epsilon}}{p^2 (p-k)^2 (q-k)^2 q^2 [\Gamma^{1+\epsilon}]} = -(-\pi)^n (k^2)^{1-3\epsilon} \left\{ -\frac{1}{6\epsilon^2} - \frac{9-4\gamma}{12\epsilon} - \frac{49 - 45\zeta(2) + 8\gamma^2 - 36\gamma}{24} \right.$$

$$\left. - \frac{\epsilon}{144}\left(981 - (108-48\gamma)\zeta(2) - 352\zeta(3) - 32\gamma^3 + 216\gamma^2 - 588\gamma\right) + O(\epsilon^2) \right\}$$

Co-authored with Tony Terrano

Table III

$$\int d^n p \frac{N}{[(p-k_1)^2]^{a_1}[(p-k_2)^2]^{a_2}}$$

$N = 1$

$$\int = \frac{i(-1\pi)^{n/2}}{(k^2)^{a_1+a_2-2+\epsilon}} \frac{\Gamma(a_1+a_2-2+\epsilon)}{\Gamma(a_1)\Gamma(a_2)} \frac{\Gamma(2-a_1-\epsilon)\Gamma(2-a_2-\epsilon)}{\Gamma(4-a_1-a_2-2\epsilon)}$$

$\leftarrow \int \overline{[\ ][\ ]}$ $\quad a_1 = a_2 = 1$

$N = p_\mu$

$$\frac{i(-4\pi)^{n/2}}{(k^2)^{a_1+a_2-2+\epsilon}} \frac{\Gamma(a_1+a_2-2+\epsilon)}{\Gamma(a_1)\Gamma(a_2)} \frac{\Gamma(2-a_1-\epsilon)\Gamma(2-a_2-\epsilon)}{\Gamma(5-a_1-a_2-2\epsilon)} \quad a_1 = 1+\epsilon,\ a_2 = 1$$

$$\times \{ (2-a_1-\epsilon) k_{2\mu} + (2-a_2-\epsilon) k_{1\mu} \}$$

$N = p_\mu p_\nu$

$$\frac{i(-4\pi)^{n/2}}{(k^2)^{a_1+a_2-2+\epsilon}} \frac{\Gamma(a_1+a_2-2+\epsilon)}{\Gamma(a_1)\Gamma(a_2)} \frac{\Gamma(3-a_1-\epsilon)\Gamma(3-a_2-\epsilon)}{\Gamma(6-a_1-a_2-2\epsilon)}$$

$$\times \{ \frac{k^2 g_{\mu\nu}}{2(a_1+a_2-3+\epsilon)} + \frac{3-a_2-\epsilon}{2-a_1-\epsilon} k_{1\mu} k_{1\nu}$$

$$+ \frac{3-a_1-\epsilon}{2-a_2-\epsilon} k_{2\mu} k_{2\nu} + k_{1\mu} k_{2\nu} + k_{1\nu} k_{2\mu} \}$$