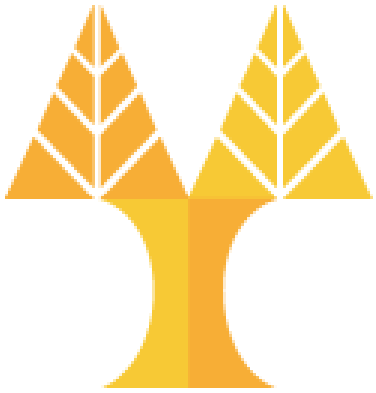
Gauge Invariant Renormalization scheme for Composite Operators

Theodosis Pafitis



Department Of Physics
University of Cyprus
May 2020

Acknowledgements

I would like to express my sincerest gratitude to my senior thesis advisor, professor Haralambos Panagopoulos. The completion of this project would not have been possible without his continuous support and nurturing. I would also like to extend my sincere thanks to Apostolos Skouroupathis, Marios Kosta, Ioannis Karpasitis and Gregoris Spanoudes for their assistance and contribution to the project, as well as the moral support they provided.
Last but not least, I would like to thank my family and friends for their unceasing encouragement and love, not only during the development of this thesis, but throughout every moment of my life.

Introduction

Quantum field theory is a theoretical framework that combines the concepts of quantum mechanics, special relativity and classical field theory. Earlier attempts to unite quantum mechanics and relativity were proven unsuccessful, as relativistic one-particle equations, such as the Klein-Gordon and the Dirac equations presented many inconsistencies (negative probabilities, infinite towers of negative energy states, violation of causality). In order to overcome those obstacles, we have to introduce the concept of fields and treat those equations as field equations.

An outline of the thesis

This thesis consists of four chapters: In the first chapter we introduce the basic elements of classical field theory. Then we shall introduce the quantization procedures, using path integrals, that lead us to quantum field theory and the fundamental quantum fields in their continuum formulation. Next, we review the basic principles of renormalization and we present the $\overline{\text{MS}}$ renormalization scheme. On this occasion we calculate the two-point fermion-antifermion Green's function in dimensional regularization and we compare the scheme with the RI$'$. The main body of this thesis is embedded in the fourth chapter: Initially, we define the composite operators and then, the new renormalization scheme. Next, we present our steps in evaluating the tree level and one loop diagrams, as well as our final results for the Green's functions, renormalization and conversion factors from our scheme to the $\overline{\text{MS}}$.

Contents

Chapter 1

Introduction to Quantum Field Theory

1.1 Concept of fields and their necessity in Relativistic Quantum Mechanics

A field is a natural quantity defined at every point of space and time. It appeared as a concept back when physicists tried to explain forces that acted at a distance, such as the electromagnetic and the gravitational ones. It was assumed that this quantity, the field, existed in the space between two or more objects, and \transferred" the force those objects exerted upon each other. This quantity had to be defined everywhere for it to be able to explain the interaction between those objects. Thus, the Classical Field Theory emerged, where fields are conceptualized as functions of space and fully explain interactions in a classical system of bodies.

Further research on matter, in smaller scales, showed that Classical Mechanics was not sufficient to predict accurately several quantities that could be measured in a laboratory, like position and momentum. A new theory was needed to describe interactions in the \small world". This brought about Quantum Mechanics, in which every particle ceases to behave as a \body" with a well defined set of measurable parameters and acquires \wave" properties. As a result, physical quantities cannot be measured without uncertainty, a kind of which cannot be avoided since it is embedded by nature itself. This theory sounded crazy but scientists already had \hints" of such a behaviour from Maxwell's equations where electromagnetic fields travelled at the speed of light, like photons. Erwin Schrodinger successfully formulated an equation which introduced the quantization of particles, while describing them as wavefunctions, that could not actually be measured but were linked to the probability of measuring different physical quantities. The wavefunction was enough for describing non relativistic elemen-

tary particles, maintaining the classical distinction between particle and field.

Nevertheless, this theory failed in describing relativistic particles. Relativistic Quantum Mechanics studies the behaviour of such objects by combining the principles of Quantum Mechanics and those of the Special Theory of Relativity. A first step into such a formulation is to think of the wavefunction of a particle as a field, since it spreads over space (just as electromagnetic fields describe photons). Defining the different fields though is not enough. One must proceed by quantizing those fields, in the same manner this was needed for non-relativistic Quantum Mechanics. By doing this, Quantum Field Theory is born and successfully describes three out of the four fundamental forces of nature, the strong interaction, the weak interaction and the electromagnetic forces.

The necessity of introducing the quantum field can be broken down into three separate reasons. Firstly, it is impossible to describe relativistic single particle states. The energy squared E^2 of a system is $E^2 = m^2c^4 + p^2c^2$, leading to no upper bound for the energy, which isn't really a problem, but also leading to no lower bound. That would seem wrong because it could only mean that there is no ground state. By describing such states as states of many particles and antiparticles we can overcome the problem of the infinitely deep energy well, and regard as ground state of the system, the one which all single-particle states of negative energy have been filled, compatibly with the Pauli exclusion principle. Leaving empty one of these negative energy states may now be interpreted as the presence of an antiparticle. This description makes the creation, as well as the annihilation, of a particle-antiparticle pair possible, even if the system has not enough energy, being consistent with Heisenberg's uncertainty principle $E \quad t \quad \frac{h}{2}$. That makes the number of particles in our system variable, concluding that the wavefunction is unsuited for describing it, since it only does so for a fixed number of bodies, like in the classical approach. Therefore replacing the wavefunction with a field that can describe a system consisting of infinite degrees of freedom seems sensible.

Secondly, working with a field instead of a wavefunction, made it possible for us to understand the connection between spin and statistics of identical particles. In Quantum Mechanics, such particles are described by constructing a wavefunction to be symmetric or anti-symmetric, meaning, adding the statistics of those particles by hand. For bosons (integer spin) in a quantum state, an exchange between two of them, returns the state to its initial form, therefore the symmetric wavefunction is suited for their description, whereas for fermions (half-integer spin), performing a switch of two of them results in a sign change of the state so, the anti-symmetric wavefunction would characterize those particles. With the introduction of the concept of fields though, no manual addition of statistics is needed, since the spin-statistics relation can be proven directly from Quantum Field Theory.

Lastly, but perhaps most importantly, Quantum Field Theory maintains causality. Let us analyze this part a bit by looking at the propagator of a free particle. In Quantum Mechanics, the propagator is a function that specifies the probability amplitude for a particle to travel from a point in space-time $(x_0; t_0)$ to another one $(x; t)$. In Quantum Mechanics' language:

$$K(x; t; x_0; t_0) = \langle x; t | x_0; t_0 \rangle = \langle x | e^{-(i/\hbar)\hat{H}(t - t_0)} | x_0 \rangle \tag{1.1}$$

where $\hat{H} = \sqrt{\hat{p}^2c^2 + m^2c^4}$ the Hamiltonian operator of a free relativistic particle, and $\hat{p}$ its momentum operator. By writing the Hamiltonian explicitly in the above propagator we get:

$$K(x; t; x_0; t_0) = \langle x; t | x_0; t_0 \rangle = \langle x | e^{-(i/\hbar)\sqrt{\hat{p}^2c^2 + m^2c^4}(t - t_0)} | x_0 \rangle \tag{1.2}$$

Let us define the identity operator by using a complete basis of momentum eigenstates as:

$$1 = \int d^3p \, |p\rangle\langle p| \tag{1.3}$$

and insert it in eq. (1.2):

$$K(x; t; x_0; t_0) = \int d^3p \, e^{-(i/\hbar)\sqrt{\hat{p}^2c^2 + m^2c^4}(t - t_0)} \langle x | p \rangle\langle p | x_0 \rangle \tag{1.4}$$

From basic Quantum Mechanics we know that:

$$\langle x | p \rangle = \frac{1}{2\hbar}^{3/2} e^{(i/\hbar)p \cdot x} \tag{1.5}$$

$$\langle p | x \rangle = \frac{1}{2\hbar}^{3/2} e^{-(i/\hbar)p \cdot x} = \langle x | p \rangle \tag{1.6}$$

Combining eq. (1.4),(1.5) and (1.6) we get:

$$K(x;t;x_0;t_0) = \int \frac{d^3p}{(2\ h)^3}\, e^{\ (i=h)\ p\sqrt{p^2c^2+m^2c^4}(t\ \ t_0)+(i=h)p\ (x\ \ x_0)} \tag{1.7}$$

$$= \frac{1}{(2\ h)^3}\int_0^{2} d \int_0^{1} dp\, p^2 \int_0^{} d\ \sin\ \ e^{(i=h)pjx\ \ x_0j\cos}\ e^{\ (i=h)\ p\sqrt{p^2c^2+m^2c^4}(t\ \ t_0)} \tag{1.8}$$

$$= \frac{i}{(2\ h)^2 jx\ \ x_0j}\int_0^{1} dp\, p\ \ e^{(i=h)\ pjx\ \ x_0j\ \ p\sqrt{p^2c^2+m^2c^4}(t\ \ t_0)} \\ e^{\ (i=h)\ pjx\ \ x_0j+\ p\sqrt{p^2c^2+m^2c^4}(t\ \ t_0)} \tag{1.9}$$

and by substituting p ! -p in the second term of the integral:

$$K(x;t;x_0;t_0) = \frac{i}{(2\ h)^2 jx\ \ x_0j}\int_1^{1} dp\, p\, e^{(i=h)\ pjx\ \ x_0j\ \ p\sqrt{p^2c^2+m^2c^4}(t\ \ t_0)} \tag{1.10}$$

Looking at the integral in eq. (1.10), we observe that it can be solved in the limit where $jx\ \ x_0j >> c(t\ \ t_0)$. The phase function has a stationary point at $p_s = p\frac{imcjx\ \ x_0j}{jx\ \ x_0j^2\ \ c^2(t\ \ t_0)^2}$, thus:

$$K(x;t;x_0;t_0)\ \ \ \frac{mc}{2\ h}^{(3=2)}\frac{c(t\ \ t_0)}{jx\ \ x_0j^{5=2}}\, e^{\ (mc=h)jx\ \ x_0j} \tag{1.11}$$

The above result shows that the propagator of a free particle from point $(x_0;t_0)$ to $(x;t)$ calculated in the limit where $jx\ \ x_0j >> c(t\ \ t_0)$ might be small, but not zero! This means that the particle can travel outside the light cone, violating casualty. Hence, quantum mechanical description of any relativistic particle will always violate casualty. In a eld theoretical treatment of particles and antiparticles, moving in opposite directions along a space-like interval ($j\ \ xj > cj\ \ tj$), their respective propagators are indistinguishable. This means that when studying the dependence of an observation at a point in space-time, on an observation at another point in space-time, the contributions of a particle and an antiparticle to the probability amplitude cancel against each other if the two points happen to be separated by a space-like interval. Thus causality is

maintained.

Overall, Quantum Field Theory can explain transitions among states with a non xed number of particles, proves the relation between spin and statistics, and maintains causality by introducing antiparticles. This theory also provides the tools for calculating scattering cross sections, particle lifetimes and other observable quantities.

1.2 Action, Lagrangian and Hamiltonian Mechanics

The action of a system is an attribute of its dynamics from which the equations of motion can be derived through the principle of stationary action. Mathematically, the action is a functional that takes a path as its argument and has a real number as a result. Di erent paths, generally, correspond to di erent values of the action. In Classical Mechanics, using the calculus of variations, the evolution of a physical system corresponds to a stationary point of the action; for every choice of initial conditions, this procedure leads to a single physical path. In the description of particles the action S involves an integral over time. However, as we will show later, the action of a eld requires integration over spatial variables as well. For a single particle in one spatial dimension we write:

$$S[q(t)] = \int_{t_1}^{t_2} L[q(t); \dot{q}(t); t]dt \tag{1.12}$$

where L is the Lagrangian, which is a function of position $q(t)$, the path of the system between two points in time and velocity $\dot{q}(t)$, the derivative of the path with respect to time. The true evolution of a physical system, as stated above, is the one corresponding to the stationary point of the action. Let $q_{true}(t)$ be the true evolution of the system whereas, $q_{per}(t)$ a slightly perturbed version of it. Suppose the di erence between those two evolutions is in nitesimal:

$$dq(t) = q_{per}(t) \quad q_{true}(t) \tag{1.13}$$

and vanishes at the endpoints:

$$dq(t_1) = dq(t_2) = 0 \tag{1.14}$$

The difference between the action integrals for these two evolutions is:

$$S = \int_{t_1}^{t_2} L[q_{true}(t) + dq(t); \dot{q}_{true}(t) + d\dot{q}(t); t] \quad L[q_{true}(t); \dot{q}_{true}(t); t] \; dt \tag{1.15}$$

$$S = \int_{t_1}^{t_2} dq \frac{\partial L}{\partial q} + d\dot{q} \frac{\partial L}{\partial \dot{q}} \; dt \tag{1.16}$$

Integrating the second term of the above integral by parts and combining it with eq. (1.14) we get:

$$S = \int_{t_1}^{t_2} dq \frac{\partial L}{\partial q} \quad dq \frac{d}{dt} \quad \frac{\partial L}{\partial \dot{q}} \quad dt \tag{1.17}$$

By stationary point of the action we imply that the first-order change must be zero for any possible perturbation $dq(t)$:

$$S = 0$$

This can only be true if the evolution is a solution to the differential equation:

$$\frac{\partial L}{\partial q} \quad \frac{d}{dt} \quad \frac{\partial L}{\partial \dot{q}} \quad = 0 \tag{1.18}$$

The above equation is known as the Euler-Lagrange equation, which one must solve to determine the path a physical system will follow, with $q(t)$ being the path and $\partial L/\partial \dot{q}$ being its conjugate momentum $p(t)$. This formulation is called Lagrangian Mechanics.

In most cases of physical interest, the Lagrangian consists of the kinetic energy T of the system and its potential energy V:

$$L = T \quad V = \frac{m\dot{q}^2}{2} \quad V(q) \tag{1.19}$$

Another formulation which also predicts the same outcomes, referred to as Hamiltonian Mechanics, makes use of the generalized coordinates $q_i(t)$ and their conjugate momenta $p_i(t) \quad @L=@\dot{q}_i$ as follows:

$$H(q_i(t); p_i(t); t) \quad \sum_i p_i \dot{q}_i \quad L \tag{1.20}$$

where H is the Hamiltonian, that replaces the generalized velocity variables with generalized momenta variables. By doing so, it is possible to handle certain systems, such as aspects of Quantum Mechanics (which we will explain later), that would otherwise be even more complicated. Di erentiating the Hamiltonian we get:

$$dH = \sum_i \quad \frac{@H}{@q_i} \; dq_i + \quad \frac{@H}{@p_i} \; dp_i \quad + \quad \frac{@H}{@t} \; dt \tag{1.21}$$

$$= \sum_i \; \dot{q}_i dp_i + p_i d\dot{q}_i \quad \frac{@L}{@q_i} \; dq_i \quad \frac{@L}{@\dot{q}_i} \; d\dot{q}_i \quad \frac{@L}{@t} \; dt \tag{1.22}$$

Substituting the above de nition of conjugate momenta and the Euler-Lagrange equation into eq. (1.22) and also matching coe cients of eq. (1.21) and eq. (1.22) leads us to the equations of motion for the canonical coordinates $(q_i; p_i)$:

$$\frac{@H}{@q_i} = \quad \dot{p}_i \tag{1.23}$$

$$\frac{@H}{@p_i} = \dot{q}_i \tag{1.24}$$

$$\frac{@H}{@t} = \quad \frac{@L}{@t} \tag{1.25}$$

also known as the canonical equations of Hamilton.

There are some advantages in using the Hamiltonian over the Lagrangian. The Hamiltonian formalism leads to $2n$ first-order differential equations, while Lagrange's equations consist of n second-order equations. Both the Lagrangian and Hamiltonian approaches provide the groundwork for deeper results in the theory of Classical Mechanics, and for formulations of Quantum Mechanics and Quantum Field Theory, as will be shown right away.

Let us consider a system whose equations of motion are written in Hamiltonian form. The Poisson brackets [1] among the canonical coordinates are:

$$\{q_i(t), q_j(t)\} = 0 \tag{1.26}$$

$$\{p_i(t), p_j(t)\} = 0 \tag{1.27}$$

$$\{q_i(t), p_j(t)\} = \delta_{ij} \tag{1.28}$$

When formulating Quantum Mechanics, we must replace those generalized coordinates and their momenta with hermitian operators which act on a Hilbert space and satisfy these commutation relations (in Heisenberg's picture):

$$[\hat{q}_i(t), \hat{q}_j(t)] = 0 \tag{1.29}$$

$$[\hat{p}_i(t), \hat{p}_j(t)] = 0 \tag{1.30}$$

$$[\hat{q}_i(t), \hat{p}_j(t)] = i\hbar\,\delta_{ij} \tag{1.31}$$

in accordance with the classical relations. The $\hat{}$ above the quantities denotes that they are operators. By determining a set of operators compatible with eqs. (1.29-1.31), we can substitute them in the Hamiltonian, turning it into an operator as well. This procedure leads to a quantization of the system, with the property that expectation values of physical observables obey the same equations of motion as their classical analogs. In a field theoretical treatment of particles though, a generalized coordinate $q_i(t)$ must be replaced with a field $(\vec{x}, t)$ while, its conjugate momentum $p_i(t)$ with a conjugate field $(\vec{x}, t)$. The essential difference when working with fields is that they are functions (operators in a Quantum Mechanical view) of space and time, unlike the position of a particle which is only a function of time. Indeed, the field is well defined at

[1] The Poisson brackets between two functions f and g are defined as:

$$\{f, g\} = \sum_{i=1}^{N} \left[\frac{\partial f}{\partial q_i} \frac{\partial g}{\partial p_i} - \frac{\partial g}{\partial q_i} \frac{\partial f}{\partial p_i} \right]$$

every point in space whereas the coordinate of a particle is de ned at one point only. Hence, the i index is replaced with a continuous variable x. Therefore, working with a eld demands that the commutation relations (1.29),(1.30) and (1.31) must be reformulated.

Consider a eld, with nite degrees of freedom, in a three-dimensional space, which is divided into cells of volume V_i. Also the eld's value $_i(t)$ is de ned as the average value of $(x;t)$ in the i_{th} cell:

$$_i(t) \quad \frac{1}{V_i} \int_{i_{th}cell} d^3x \quad (x;t) \tag{1.32}$$

In the same manner, $-_i(t)$ is de ned as the average value of the derivative with respect to time of the eld in the i_{th} cell. The Lagrangian, which used to be a function of position and velocity, now becomes a function of elds, and must therefore be de ned as the average value of the Lagrangian density L in the i_{th} cell, summed over all the cells:

$$L = \sum_i V_i L_i \tag{1.33}$$

According to Classical Mechanics, the eld's conjugate momentum is:

$$p_i(t) = \frac{@L}{@-_i(t)} = V_i \frac{@L}{@-_i(t)} \qquad V_i \ _i(t) \tag{1.34}$$

where $_i(t)$ is the momentum density in the i_{th} cell. In the continuum limit:

$$(x;t) = \frac{@L(\ ;@ \ )}{@-(x;t)} \tag{1.35}$$

at the continuous limit, and plays the role of the conjugate eld of $(x;t)$. Thus the commutation relations for the above system in discretized space are:

$$[\ _i(t);\ \ _j(t)] = 0 \tag{1.36}$$
$$[\ _i(t);\ \ _j(t)] = 0 \tag{1.37}$$
$$[\ _i(t);\ \ _j(t)] = ih\frac{_{ij}}{V_i} \tag{1.38}$$

which in turn become:

$$[\ (x;t);\ \ (y;t)] = 0 \tag{1.39}$$
$$[\ (x;t);\ \ (y;t)] = 0 \tag{1.40}$$
$$[\ (x;t);\ \ (y;t)] = ih\ ^3(x\ \ y) \tag{1.41}$$

in the continuum limit ($V_i\ !\ 0$).

This formalism is used for the quantization of bosons. One immediate consequence is that bosonic states are symmetric with respect to exchange of the elds which create them from the vacuum. On the contrary, quantization of fermions leads to inconsistent results unless one replaces the commutators in eqs.(1.39-1.41) with anti-commutators. In this section we introduce the elds and the Lagrangian density, by analogy with corresponding concepts in the description of particles. We also hint at the standard procedure for quantizing elds, referred to as Canonical Quantization. It will be later shown that, by determining the Classical action of the elds, we can quantize them in a more e cient way , in which relativistic invariance is manifest, by using the Path Integral for the calculation of di erent propagators and Green's functions, as advocated by Richard Feynman.

1.3 Conservation Laws and Noether's Theorem

The relation of symmetries and conservation laws in Classical Field Theory can be described by Noether's Theorem. This theorem stands unaltered even in Quantum Field Theory, with the sole di erence that the elds are replaced with operators, which satisfy commutation relations as above. According to the theorem, for every continuous transformation, which is a symmetry of a system (meaning the equations of motion remain una ected), a quantity named Noether's current exists and is conserved. The fact that the conservation of the Noether current holds at every point in spacetime, is a useful guide in determining the form of all possible interaction terms in a Lagrangian density,

compatibly with the symmetries which one wishes to preserve.

Let us de ne the four vector $x \quad x = (ct; x)$ where an index with values $(0; 1; 2; 3)$. We are doing this since relativity treats time and space equally as coordinates in a Minkowski space. Let us also employ units in which $c \quad 1$ for simplicity. Assume an in nitesimal change of a eld, which generally can be written as follows:

$$(x) \; ! \quad ^{0}(x) = \quad (x) + a \quad (x)$$

where a an in nitesimal constant and (x) a distortion of the eld. For this transformation to be a symmetry, the Lagrangian density must generally change by a total derivative of a function J , so that the action and the equations of motion as mentioned above, remain una ected.

$$L(\; (x); @ \quad (x)) \; ! \; L^{0}(\; (x); @ \quad (x)) = L(\; (x); @ \quad (x)) + a@ \; J \; (x) \qquad (1.42)$$

In general, a change in the Lagrangian density mathematically means:

$$\begin{aligned} L^{0}(\; (x); @ \quad (x)) &= L(\; (x); @ \quad (x)) + a \quad L(\; (x); @ \quad (x)) \\ &= L + a\frac{@L}{@} \qquad + a\frac{@L}{@(@ \quad)}@ \; (\quad) \\ &= L + a@ \quad \frac{@L}{@(@ \quad)}(\quad) \; + a \; \frac{@L}{@} \quad @ \quad \frac{@L}{@(@ \quad)} \end{aligned}$$

The third term of the above expression vanishes if (x) satis es the equations of motion, therefore:

$$L^{0}(\; (x); @ \quad (x)) = L(\; (x); @ \quad (x)) + a@ \quad \frac{@L}{@(@ \quad)}(\quad) \qquad (1.43)$$

and by looking at eq. (1.42) and (1.43):

$$@\ \frac{@L}{@(@\ )}(\quad) = @\ J$$
$$) @\ j = 0$$
$$) j \quad \frac{@L}{@(@\ )}(\quad)\ J \tag{1.44}$$

where j is Noether's current, and satisfies the continuity equation; thus, it is conserved. As for every current, its respective \charge":

$$Q = \int_{all\ space} j^0\, d^3x$$

is time-independent. For each symmetry of a system, this charge is a different conserved quantity.

Consider a shift in space-time coordinates :

$$x \ ! \ x^0 = x \quad a$$

which in turn brings about the following transformation of a scalar field:

$$(x) \ ! \ {}^0(x^0) = (x) = (x^0) + a\ @\ (x^0)$$

The Lagrangian density is also a scalar, therefore the change is similar to that of the field:

$$L \ ! \ L^0 = L + a\ @\ L = L + a\ @\ (\quad L)$$

Comparing eq.(1.42) with the above expression, $J = L$. According to eq.(1.44), the conserved currents (one current for each parameter a) are:

$$j = \frac{@L}{@(@\ )}@ \qquad L \tag{1.45}$$

The time components of these currents (j^0) lead to four conserved quantities; the Hamiltonian of the system:

$$H = \int d^3x\, j^{00} = \int d^3x\, [\ (x)\dot{-}(x) \quad L] \qquad \int d^3x\, H \tag{1.46}$$

and the three components of the system's momentum:

$$P^i = \int d^3x\, j^{0i} = \int d^3x \quad (x)@^i \quad (x)$$
$$\vec{P} = \int d^3x \quad (x)\vec{\tau} \quad (x) \tag{1.47}$$

1.4 The Action of Classical Fields

As we have already mentioned by the end of section (1.2), it is of great importance to determine the action of di erent elds for e ciently quantizing them. Therefore, for the rest of this chapter, we are going to dive into the derivation of those actions, as well as their respective Lagrangian densities for both free and interacting elds.

1.4.1 The Klein-Gordon Action for Free Scalar Fields

The Klein-Gordon equation, which describes a free real scalar eld, is:

$$2 + \left(\frac{mc}{h}\right)^2 \quad (x) = 0 \tag{1.48}$$

where the box notation indicates the D'Alembertian operator

$$2 \qquad \sum_{=0}^{3} @\ @$$

This equation is the result of an attempt to replace Schrodinger's equation, with one that could describe relativistic quantum states, by interpreting (x) as the state's wave function. One can produce the above equation by taking the relativistic expression for the energy of a free particle $E^2 = p^2c^2 + m^2c^4$ and replacing its energy and momentum with their respective operators from Quantum Mechanics ($E \; ! \; ih\frac{@}{@t}$, $p \; ! \; ih\gamma$). The only di erence from Schrodinger's equation is that we have abandoned the classical expression of a free particle's energy and we have adopted the relativistic one. This procedure eventually failed to ful l its original purpose due to di erent shortcomings. Firstly, it is a second-order di erential equation with respect to time, which means one initial condition i.e. $(x; t_0)$, is not enough to fully describe the system. Moreover, it has negative eigenvalues without a lower bound (which was later dealt with in cases where the Pauli exclusion principle can be invoked, by introducing a multi-particle interpretation of states as mentioned in section 1.1) and also leads to negative probability densities. Ultimately the Klein-Gordon equation remained as a classical theory for the free scalar eld.

Let us de ne units where $h = c \quad 1$. The Lagrangian density of the scalar eld is:

$$L_{K \; G} = \frac{1}{2}@ \quad @ \qquad \frac{1}{2}m^2 \; ^2 \tag{1.49}$$

while the Euler-Lagrange equation stemming from eq.(1.49) gives the correct Klein-Gordon equation:

$$\begin{aligned}
\frac{@L}{@ \quad} \quad @ \quad \frac{@L}{@(@ \quad)} \quad &= 0 \\
) \quad m^2 \quad @ \quad \frac{@}{@(@ \quad)}(\frac{1}{2} \quad @ \quad @ \quad) \quad &= 0 \\
) \quad m^2 \quad @ \quad \frac{1}{2} \qquad @ \quad + \frac{1}{2} \qquad @ \quad &= 0 \\
) \quad m^2 \quad @ \quad \frac{1}{2}@ \quad + \frac{1}{2}@ \quad &= 0 \\
) \; (@ \; @ \; + m^2) \quad &= 0 \\
) \; (2 + m^2) \; (x) \; &= 0
\end{aligned}$$

The equivalent action of the K-G eld is:

$$
\begin{aligned}
S_{K\ G} &= \int d^4x\, L_{K\ G} \\
&= \int d^4x\, (\frac{1}{2}\partial\ \ \partial\ \ \ \ \frac{1}{2}m^2\ \ ^2) \\
&= \frac{1}{2}\int d^4x\, \partial\ (\ \partial\ \ ) \ \ \frac{1}{2}\int d^4x\ \ (x)(2 + m^2)\ \ (x)
\end{aligned}
$$

By turning the rst term of the above integral into a surface integral over the boundary of four-dimensional space-time, we can conclude that it goes to zero, since the eld's value along that surface is zero. Therefore the action is:

$$
S_{K\ G} = \frac{1}{2}\int d^4x\ \ (x)(2 + m^2)\ \ (x) \tag{1.50}
$$

1.4.2 The Dirac Action for Free Fermions

The free Dirac equation that describes the spinor eld of free fermions is:

$$
(i\partial\!\!\!/\ \ m)\ \ (x) = 0 \tag{1.51}
$$

where $\partial\!\!\!/ = \ \ \partial$ and are 4x4 Dirac matrices, which satisfy the anti-commutation relation $f\ \ ;\ \ g = 2\ \ $. This equation was the result of an attempt to improve the Klein-Gordon equation, by eliminating the problems it posed for describing relativistic quantum states. Indeed, it is a rst-order di erential equation with respect to time therefore, one initial condition is enough for solving it. It also leads to positive probability densities but still has negative eigenvalues without a lower bound. Nevertheless, even though it solved two out of the three problems of the Klein-Gordon equation, the idea of describing relativistic quantum states using it, got rejected. It did nd a use eventually in describing multi-particle states, regarding as ground state of the system the one in which all single-particle states of negative energy have been lled, compatibly with the Pauli exclusion principle (the Dirac \sea"), thus, solving the problem posed by the in nitely deep well of states. It also introduces the concept of anti-particles as states of negative energy which stay empty (holes), when one particle is lifted out of the Dirac sea and becomes a positive-energy particle. Each empty state behaves like a positive energy particle with its electric charge reversed. Some bosons also have antiparticles, but since bosons do not obey the Pauli exclusion principle (only fermions do), this hole theory does not work for them. A uni ed interpretation of anti-particles is now available in Quantum Field Theory, which

solves both these problems by describing antimatter as negative energy states of the same underlying matter eld, i.e. particles moving backwards in time. Back to the classical interpretation of the eld, it is important to note that the Dirac equation contains a \spinor" with four components, describing a fermionic eld. The Lagrangian density is:

$$L_F = \quad (x)(i@\!\!\!/ \quad m) \quad (x) \tag{1.52}$$

where $(x) \quad ^{y\ 0}$ is the conjugate eld of (x). Indeed, the Euler-Lagrange equation of the above Lagrangian produces the Dirac equation for the eld (x):

$$\begin{aligned} \frac{@L}{@} \quad @ \quad \frac{@L}{@(@ \quad)} &= 0 \\ ) \quad (i@\!\!\!/ \quad m) \quad (x) &= 0 \end{aligned}$$

whereas, an equation for the conjugate eld can also be generated:

$$\begin{aligned} \frac{@L}{@} \quad @ \quad \frac{@L}{@(@ \quad)} &= 0 \\ ) \quad m \quad @ \ (i \quad) &= 0 \\ ) \ i@ \quad + m \quad &= 0 \\ ) \quad (i@\!\!\!/ + m) &= 0 \end{aligned}$$

which is equivalent to eq. (1.51). Hence, the action of the Dirac eld is:

$$S_F = \quad \int d^4x \ L_F = \quad \int d^4x \quad (x)(i@\!\!\!/ \quad m) \quad (x) \tag{1.53}$$

1.4.3 The Action of Quantum Electrodynamics (QED)

Quantum Electrodynamics (QED) describes the interactions between electrons through photons. The respective action consists of two parts; the photon action

and the electron action. Furthermore, QED is a gauge invariant theory, meaning that it remains unchanged under local gauge transformations of the abelian group U(1). Therefore, both parts of the action must remain unaffected under such transformations.

According to eq.(1.53), the free electron action is:

$$S_{EL} = \int d^4x \quad (x)(i\not{@} \quad m) \quad (x)$$

since electrons are fermions. Consider a local transformation $G(x) \quad e^{i \ (x)}$ (where (x) is a function of x) of the abelian U(1) group. The transformed spinors are:

$$\begin{aligned} (x) &\ ! \ G(x) \quad (x) \\ (x) &\ ! \quad (x)G^{\ 1}(x) \end{aligned}$$

Obviously, the above expressions do not leave the action unaltered. To accomplish this local U(1) symmetry, we must substitute the simple partial derivative $@$, with the covariant derivative:

$$D \quad @ \ + ieA$$

where e is the electron charge, while A is the electromagnetic four-potential. We will refer to it as the photonic field. The presence of the covariant derivative thus adds an interaction term between the photons and the electronic field. The covariant derivative indeed transforms in a covariant fashion, as can be deduced from the tranformation of A :

$$\begin{aligned} A \quad &! \ A \quad \frac{1}{e}@ \quad (x) \\ ) \ D \quad &! \ G(x)D \ G^{\ 1}(x) \end{aligned}$$

We conclude that the new invariant electron action is:

$$S_{EL} = \int d^4x \quad (x)(i\not{D} \quad m) \quad (x) \tag{1.54}$$

We have yet to determine the photon action, which will only contain a kinetic term because the photon-electron interaction term has been considered and also because photons do not interact with each other since they do not have an electric charge. It can be shown that, for this theory to be gauge invariant, it should not contain any photon mass terms either. Hence the term we are looking for is:

$$S_{PH} = \frac{1}{4} \int d^4x \; F \quad F \tag{1.55}$$

where $F \quad = @ \; A \quad @ \; A$. The tensor F is also gauge invariant. The above expression coincides with Maxwell's classical action for the electromagnetic field. The photon action is also gauge invariant as we demanded.

Finally, the total action of QED is the sum of the electron and photon actions:

$$S_{QED} = S_{PH} + S_{EL} = \frac{1}{4} \int d^4x \; F \quad F \quad + \int d^4x \quad (x)(i\not{D} \quad m) \quad (x) \tag{1.56}$$

One can check that, invoking the Euler-Lagrange equation by differentiating with respect to and , leads to the equations of motion for the respective fields:

$$(i\not{@} \quad e\not{A} \quad m) \quad (x) = 0 \tag{1.57}$$

$$(x)(i\not{@} + e\not{A} + m) = 0 \tag{1.58}$$

The above are equivalent forms of the Dirac equation, coupled to the electromagnetic field.

There is one last equation of motion that we can derive for the photon field:

$$
\begin{aligned}
& \frac{@L_{QED}}{@A} \quad @ \quad \frac{@L_{QED}}{@(@\ A\ )} = 0 \\
) \quad & \frac{@(\ e \quad A\ )}{@A} + \frac{1}{4}@ \quad \frac{@(\ {}_0 \quad {}_0 F^{\ 0\ 0} F}{@(@\ A\ )} = 0 \\
) \quad & e \quad + \frac{1}{2}@ \quad F \quad \frac{@F}{@(@\ A\ )} = 0 \\
) \quad & e \quad + @\ F \quad = 0 \\
) \quad & e \quad + \quad @\ F \quad = 0 \\
) \quad & e \quad + @\ F \quad = 0 \\
&) \ @\ F \quad = e
\end{aligned}
\tag{1.59}
$$

The above expression is the equation of motion for the photon eld and as expected; indeed, we have recovered Maxwell's equations ($@\ F \quad = j\ ,\ j \quad = (\ ;\vec{j})$) written in units where $c = {}_0 = 1$. Therefore the QED action we constructed correctly describes electromagnetic interactions.

1.4.4 The Action of Quantum Chromodynamics (QCD) for Strong Interactions

Quantum Chromodynamics (QCD) describes the interactions between quarks through gluon exchange. The QCD action, similar to QED, consists of two parts: the gluon action and the quark action. Moreover, QCD is also a gauge invariant theory, with the only di erence being that it remains invariant under local transformations of the non-abelian group SU(3) (while QED was gauge invariant under local transformations of the abelian group U(1)). Hence, each part of the total action must remain una ected under such transformations.

Generally, non-abelian gauge theories SU(N), involve N Dirac elds of the same mass. In this case, quarks and anti-quarks are detected in N=3 di erent types, which we refer to as \colors". Therefore, the eld must be replaced with a 3 1 column vector:

$$
\_ = \begin{matrix} 2 \ {}_1 3 \\ 6 \quad 7 \\ 6 \quad 7 \\ 6 \ {}_2 7 \\ 6 \quad 7 \\ 4 \quad 5 \\ {}_3 \end{matrix}
$$

whereas its conjugate will be a 1 3 row vector:

$$\bar{\psi} = \quad ^{1}; \quad ^{2}; \quad ^{3}$$

Respectively, the gluonic eld substitutes the photonic one with a 3 3 $\underline{A}$ matrix, so that it can be coupled to two interacting quarks. The quark action therefore is:

$$S_Q = \int d^4x \, \bar{\psi}(x)(i\not{D} \quad m)\psi(x) \tag{1.60}$$

where $\not{D} = \quad D \ , D = @ + ig\underline{A}$ and g is the coupling constant of the strong interactions. The action above is invariant under these local transformations:

$$\begin{aligned} \psi(x) &! \underline{G}(x)\psi(x) \\ \bar{\psi}(x) &! \bar{\psi}(x)\underline{G}^{1}(x) \\ \underline{A} \ (x) &! \underline{G}(x)\underline{A} \ (x)\underline{G}^{1}(x) \quad \frac{i}{g}\underline{G}(x)@ \ \underline{G}^{1}(x) \end{aligned}$$

where $\underline{G}(x)$ is a transformation of the non-abelian group SU(3), which means that $\underline{G}(x) = e^{i\theta(x)}$ and $\theta(x)$ a traceless hermitian matrix that belongs to the Lie algebra of SU(3). It is worth noticing that if $\underline{G}(x)$ becomes a 1 1 matrix, then the above transformations become the correct QED transformations, as expected. That is why, when $\underline{G}(x)$ is an N N matrix, local transformations generally translate to what we have written for $N = 3$.

The gluon action is similar to the photon one with one crucial di erence. The eld is now a 3 3 matrix, which is why its respective tensor components must become 3 3 matrices $\underline{F}$. Additionally, we demand that the tensor is at least covariant under local transformations of SU(3) ($\underline{F} \quad ! \underline{G}(x)\underline{F} \quad \underline{G}^{1}(x)$). The gluon action can therefore be constructed from the trace of $\underline{F} \quad \underline{F}$, since it is the simplest scalar quantity, that is gauge invariant. For the tensor to become covariant, we must de ne it di erently from the one in QED:

$$\underline{F} \quad = @ \ \underline{A} \quad @ \ \underline{A} \ + ig[\underline{A} \ ; \underline{A} \ ] \tag{1.61}$$

Furthermore, the appearance of three di erent colors means di erent types of gluons, which can interact amongst them. Taking this into account, one can show that the gluon action is the following:

$$S_G = \frac{1}{2} \int d^4x \ tr(\underline{F} \quad \underline{F} \quad) \tag{1.62}$$

We then conclude that the QCD total action is:

$$S_{QCD} = S_G + S_Q = \frac{1}{2}\int d^4x \; tr(\underline{F} \quad \underline{F} \quad) + \int d^4x \; \_(x)(i\not{D} \quad m)\_(x) \quad (1.63)$$

It is worth mentioning that there are also di erent kinds of quarks with di erent masses, categorized by \ avor". Currently, six of them are known; the Up (u), Down (d), Charm (c), Strange (s), Top (t) and Bottom (b) quarks. Taking avors into consideration the action from eq.(1.61) can be written as:

$$S_{QCD} = \frac{1}{2}\int d^4x \; tr(\underline{F} \quad \underline{F} \quad) + \sum_{f=1}^{\infty}\int d^4x \; \_f(x)(i\not{D} \quad m_f)\_f(x) \quad (1.64)$$

Chapter 2

The Functional Form of Quantum Field Theory

2.1 The Concept of the Path Integral

As we all know, a common integral, $\int f(x)\,dx$, involves a function $f(x)$ and a continuous range of values x. We now introduce the concept of a functional integral which, as the name suggests, is the integration of a functional $G[f(x)]$, over a continuous range of functions $f(x)$, at every point in space-time:

$$\int Df\, G[f] \qquad \int \prod_x df(x)\, G[f(x)]$$

In a discretized space-time the above integral can be written as follows:

$$\int \prod_i df_i\, G[f_i]$$

where the integer i ranges over all spacetime points.

Functional integrals are useful for determining propagators in Quantum Field Theory, instead of trying to do so by the method of Canonical Quantization. Therefore we do not need to worry about operator formalism nor about using the Hamiltonian. Instead we make use of the Lagrangian and the actions we have derived in chapter 1. Moreover, through functional integration, we can manifestly preserve the gauge invariance of physical properties. Finally, the functional quantization of fields is used for non-perturbative calculations in interactive theories.

The simplest application of functional integration in Quantum Mechanics is the derivation of a particle propagator between two space-time points. What

we accomplish by doing so is that we calculate the probability of the particle to be observed at point $(t_f; x_f)$, given that it was initially observed at $(t_i; x_i)$, by taking into account every possible path it can follow, with each path being statistically weighted.

The propagator of a non relativistic particle (in Heisenberg's picture) is:

$$K(x_f; t_f; x_i; t_i) = \langle x_f; t_f | x_i; t_i \rangle \tag{2.1}$$

By inserting a unit operator in eq.(2.1) we get:

$$K(x_f; t_f; x_i; t_i) = \int dx_1 \langle x_f; t_f | x_1; t_1 \rangle \langle x_1; t_1 | x_i; t_i \rangle \tag{2.2}$$

The above translates as the probability amplitude of the particle reaching point $(t_f; x_f)$ given that it has passed through any point x_1 (with probability amplitude $\langle x_1; t_1 | x_i; t_i \rangle$ for each point) at time t_1. Inserting unit operators at times t $(t_i < t < t_f)$ recursively, therefore adding more intermediate steps, we successfully construct a Path Integral, where the particle can follow literally in nite paths to its destination, each with a probability as mentioned above. Hence the total probability amplitude can been written as:

$$K(x_f; t_f; x_i; t_i) = \int \prod_{n=1} dx_n \langle x_f; t_f | x_N; t_N \rangle \langle x_N; t_N | x_{N-1}; t_{N-1} \rangle ::: \langle x_1; t_1 | x_i; t_i \rangle \tag{2.3}$$

Now, one can calculate the propagator between two consecutive points in time:

$$K(x_{n+1}; t_{n+1}; x_n; t_n) = \langle x_{n+1}; t_{n+1} | x_n; t_n \rangle = \langle x_{n+1} | e^{-(i/\hbar)\hat{H}(t_{n+1}-t_n)} | x_n \rangle \tag{2.4}$$

where $\hat{H}$ is the Hamiltonian operator:

$$\hat{H} = \frac{1}{2m} \sum_{d=1} \hat{p}_d^2 + V(\hat{x})$$

and $\hat{p}_d^2$ the conjugate momentum operator in the d^{th} direction of the operator $\hat{x}$. We may now split the exponential into a product of two terms; this is valid to $O(\Delta t^2)$ $(\Delta t << 1)$. Hence eq.(2.4) becomes:

$$K(x_{n+1}; t_{n+1}; x_n; t_n) = \langle x_{n+1} | \prod_{d=1} e^{-(i/\hbar)\frac{1}{2m}\hat{p}_d^2 \Delta t} | x_n \rangle e^{-(i/\hbar)V(\hat{x})\Delta t}$$

Inserting the unit operator once again, but now using the momentum eigenstates, and given that $\langle x | p \rangle = (\frac{1}{2\pi\hbar})^{D/2} e^{(i/\hbar)x \cdot p}$, we get a Gaussian integral.

Performing the integration results to [1]:

$$K(x_{n+1};t_{n+1};x_n;t_n) = \left(\frac{m}{2\,ih\,t}\right)^{D=2} exp\,(i=h)\;t\;\frac{m}{2}\sum_{d=1}^{D}\left(\frac{x_{n+1;d}-x_{n;d}}{t}\right)^2 - V(x_n)$$

The total propagator from eq.(2.3) combined with the above result becomes:

$$K(x_f;t_f;x_i;t_i) = \int\prod_{n=1}^{N-1}\prod_{d=1}^{D} dx_{n;d}\left(\frac{m}{2\,ih\,t}\right)^{D=2} exp\,(i=h)\;t\sum_{n=1}^{N}\frac{m}{2}\sum_{d=1}^{D}\left(\frac{x_{n+1;d}-x_{n;d}}{t}\right)^2 - V(x_n)$$

At the continuous limit ($N \;!\; 1$ and $t \;!\; 0$) our propagator is:

$$K(x_f;t_f;x_i;t_i) = \lim_{N\;!\;1} N\int\prod_{n=1}^{N-1}\prod_{d=1}^{D} dx_{n;d}\; exp\,(i=h)\int_{t_i}^{t_f} dt\;\frac{m}{2}\sum_{d=1}^{D}\dot{x}^2(t) - V(x(t)) \tag{2.5}$$

$$N = \left(\frac{m}{2\,ih\,t}\right)^{(N+1)D=2}$$

We notice that an exponential of the action is involved so we can write our expression as the following functional integral:

$$K(x_f;t_f;x_i;t_i) = \int Dx\; e^{iS[x]=h} \tag{2.6}$$

thus proving the statement, that the probability amplitude of observing a particle at point $(t_f;x_f)$, given that it was initially observed at $(t_i;x_i)$, is the sum of the probability amplitudes of the particle following any one of the in nite available paths, with each path being weighted by a factor $e^{iS[x]=h}$.

2.2 Regularization and renormalization

Objects of fundamental importance in Quantum Field Theory are Green's functions (correlation functions). These functions, if appropriately chosen, contain all physical information one wishes to know about particle physics. Each Green's function can be expressed, in perturbation theory, as an in nite sum of terms that can be visually represented by Feynman diagrams. Usually, a su cient approximation of the Green's function is obtained by calculating diagrams up to a certain order in the coupling constant. Diagrams beyond leading order normally exhibit ultraviolet divergences and special care is needed to treat those singularities. In order to make the Feynman diagrams nite, we rstly need to

[1] The index n refers to the n^{th} point in space, whereas d refers to the d^{th} component of the vector

modify our theory by introducing a regulator.

An option is the dimensional regularization (which will be discussed in Chapter 3), where the integrals appearing in Feynman diagrams are regularized in d dimensions, an analytic continuation of the 4 spacetime dimensions. Typically, we choose d = 4-2 , with being the regulator in this method. In order to ensure that singularities will not appear as the regulators vanish, one must apply a renormalization procedure, which involves rede nition of our elds and parameters so that the Green's functions remain nite.

2.3 The Klein-Gordon Propagator

Now that we have seen an application of the functional integral in Quantum Mechanics, let us proceed by deriving the various eld propagators. The fact that we are dealing with elds, forces us to generalize the method we used for the non-relativistic particle, to one that can apply to a Field Theory. Firstly, we must consider the vacuum ($j0i$) as the normalized ($h0j0i = 1$) ground state of the system. By following the Canonical Quantization method (Chapter 2 of Ref.[3]) one can show that:

$$\hat{}(x)j0i = \int d^3p=(2\ )^3 \frac{1}{2E_p} e^{\ ip\cdot x} jpi \tag{2.7}$$

the interpretation of the above state, by ignoring the factor $1=(2E_p)$ (which is constant at small values of p), is the nonrelativistic expression for the eigenstate jxi. Therefore we can claim that $\hat{}(x)$, acting on the vacuum, creates a particle at position x. The eld propagator between two points x_1 and x_2 is:

$$D_{K\ \ G} = h0j\hat{}(x_2)\hat{}(x_1)j0i \tag{2.8}$$

where x_2 is a point at time $t_2 > t_1$. Furthermore, it can be shown (Ref.[4], pp.15-20) that any Green's function, which involves elds, can be written as a functional integral as follows:

$$hO(\ )i = \frac{\int D\ \ O(\ )e^{iS=h}}{\int D\ \ e^{iS=h}} \tag{2.9}$$

In our case, where we want to express eq.(2.8) as a functional integral similar to eq.(2.9), $O(\ )$ is $\hat{}(x_2)\hat{}(x_1)$ and $S\ !\ S_{K\ \ G}$:

$$D_{K\ \ G} = \frac{\int D\ \ \hat{}(x_2)\hat{}(x_1)e^{iS_{K\ \ G}=h}}{\int D\ \ e^{iS_{K\ \ G}=h}} \tag{2.10}$$

The above equation has the form of a polynomial mean value, weighted by

a gaussian. By discretizing our space-time and performing a Wick rotation ($t \to it^E$) from Minkowski to Euclidean space, we get:

$$D_{K\ G}(x_2 \quad x_1) = \frac{\int \prod_i d\ (x_i)\quad (x_2)\ (x_1) exp\quad \frac{1}{2}\sum_{i;j}\ (x_i)K_{ij}\ (x_j)}{\int \prod_i d\ (x_i)\ exp\quad \frac{1}{2}\sum_{i;j}\ (x_i)K_{ij}\ (x_j)} \tag{2.11}$$

where $K_{ij} = (@\ @\ + m^2)_{ij}$. The solution of the integral in eq.(2.11) can be proven (Chapter 1 of Ref.[6]) to be equal to the inverse of K_{ij}:

$$D_{K\ G}(x_2 \quad x_1) = (K^{\ 1})_{2;1} \tag{2.12}$$

Going back to the continuous Minkowski space, we use the following expression:

$$(@\ @\ + m^2)D_{K\ G}(x_2 \quad x_1) = \quad i\ ^{(4)}(x_2 \quad x_1) \tag{2.13}$$

with a i multiplying the four dimensional delta function, because we preserve real valued time components. By performing a Fourier transformation on both sides of eq.(2.13) we get:

$$\int \frac{d^4p}{(2\ )^4}\ (@\ @\ + m^2)\tilde{D}_{K\ G}(p)e^{\ ip(x_2\ \ x_1)} = i\int \frac{d^4p}{(2\ )^4}\ e^{\ ip(x_2\ \ x_1)}$$

$$) \int \frac{d^4p}{(2\ )^4}\ ((\ ip\ )(\ ip\ ) + m^2)\tilde{D}_{K\ G}(p)e^{\ ip(x_2\ \ x_1)} = i\int \frac{d^4p}{(2\ )^4}\ e^{\ ip(x_2\ \ x_1)}$$

$$) \int \frac{d^4p}{(2\ )^4}\ (\ p^2 + m^2)\tilde{D}_{K\ G}(p)e^{\ ip(x_2\ \ x_1)} = i\int \frac{d^4p}{(2\ )^4}\ e^{\ ip(x_2\ \ x_1)}$$

$$) \tilde{D}_{K\ G}(p) = \frac{i}{p^2 \quad m^2}$$

To avoid any singularities on the real axis we shall add a term i to the denominator:

$$\tilde{D}_{K\ G}(p) = \frac{i}{p^2 \quad m^2 + i} \tag{2.14}$$

consequently leading to the Klein-Gordon propagator in position-space:

$$D_{K\ G}(x_2 \quad x_1) = \int \frac{d^4p}{(2\ )^4}\ \frac{i}{p^2 \quad m^2 + i}e^{\ ip(x_2\ \ x_1)} \tag{2.15}$$

Eq.(2.14) is the equivalent of eq.(2.15) in momentum-space.

2.4 The Dirac Propagator

The functional representation of the Dirac propagator can be achieved exactly like the Klein-Gordon case. Generally a fermion Green's function involving $N = 2I$ space-time points acquires the following functional form:

$$G(x_1; x_2; :::; x_I; y_1; y_2; :::y_I) =$$

$$= \frac{\int D \quad D \quad {}_{a_1}(x_1) \quad {}_{a_2}(x_2):::: \quad {}_{a_I}(x_I) \quad {}_{b_1}(x_1) \quad {}_{b_2}(x_2):::: \quad {}_{b_I}(x_I) \quad e^{iS_F[\ ;\ ]}}{\int D \quad D \quad e^{iS_F[\ ;\ ]}}$$

where $(\ ;\ )$ are Grassmann variables (Chapter 1 of Ref.[6]), $S_F(\ ;\ )$ the Dirac action from eq.(1.53) and $D \quad D \quad = \prod_{x;y} \prod_{a;b=1}^{4} d \quad {}_b(y) d \quad {}_a(x)$. Any Green's function involving fermion operators $O(\ ;\ )$ is:

$$h O(\ ;\ ) i = \frac{\int D D \quad O(\ ;\ ) e^{iS_F[\ ;\ ]}}{\int D \quad D \quad e^{iS_F[\ ;\ ]}} \tag{2.16}$$

The free Dirac eld propagator $D_{F(a;b)} = h0j \quad {}_a(x_2) \quad {}_b(x_1)j0i$ therefore can be written:

$$D_{F(a;b)}(x_2 \quad x_1) = \frac{\int D \quad D \quad {}_a(x_2) \quad {}_b(x_1) e^{iS_F[\ ;\ ]}}{\int D \quad D \quad e^{iS_F[\ ;\ ]}} \tag{2.17}$$

Similar to the method used for the Klein-Gordon propagator, we discretize space-time ($x \ ! \ x_i$) and perform a Wick roatation ($t \ ! \quad it^E$) from Minkowski to Euclidean space:

$$D_{F(a;b)}(x_2 \quad x_1) =$$

$$= \frac{\int \prod_{i;j} \prod_{c;d=1}^{4} d \quad {}_c(x_i) d \quad {}_d(x_j) \quad {}_a(x_2) \quad {}_b(x_1) exp \quad \sum_{i;j} \sum_{c;d=1}^{4} \quad {}_c(x_i) K_{cd}(x_i; x_j) \quad {}_d(x_j)}{\int \prod_{i;j} \prod_{c;d=1}^{4} d \quad {}_c(x_i) d \quad {}_d(x_j) \, exp \quad \sum_{i;j} \sum_{c;d=1}^{4} \quad {}_c(x_i) K_{cd}(x_i; x_j) \quad {}_d(x_j)} \tag{2.18}$$

where $K_{cd}(x_i; x_j) = \quad (i \quad @ \quad m)_{cd}$. Like the integral in eq.(2.11), the above integral can be solved:

$$D_{F(a;b)}(x_2 \quad x_1) = K_{ab}^{\ 1}(x_2; x_1) \tag{2.19}$$

Going back to the continuous Minkowski space, we use the following expression:

$$(i \quad @ \quad m)_{ab} D_{F(b;c)} = \quad i \ {}^{(4)}(x_2 \quad x_1) \ {}_{ac} \tag{2.20}$$

Once again, by performing Fourier transformations in eq.(2.20) we get:

$$\int \frac{d^4p}{(2\ )^4}(i\quad @\qquad m)_{ab}\tilde{D}_{F\,(b;c)}(p)\,e^{\ ip(x_2\ \ x_1)} = i\int \frac{d^4p}{(2\ )^4}\,e^{\ ip(x_2\ \ x_1)}{}_{ac}$$

$$)\quad \int \frac{d^4p}{(2\ )^4}(i\quad (\ \ ip\ )\quad m)_{ab}\tilde{D}_{F\,(b;c)}(p)\,e^{\ ip(x_2\ \ x_1)} = i\int \frac{d^4p}{(2\ )^4}\,e^{\ ip(x_2\ \ x_1)}{}_{ac}$$

$$)\quad (\quad p\qquad m)\tilde{D}_F(p)_{\ ac} = i\ _{ac}$$

By multiplying the above result with $(\quad p\ + m)_{ba}$ on the left and solving for $\tilde{D}_F(p)$:

$$\tilde{D}_{F\,(b;c)}(p) = \frac{i(\not{p} + m)_{bc}}{p^2\quad m^2} \tag{2.21}$$

We now have the free fermion two-point propagator in momentum-space, which in turn becomes:

$$D_{F\,(a;b)}(x_2\quad x_1) = \int \frac{d^4p}{(2\ )^4}\,\frac{i(\not{p} + m)_{ab}}{p^2\quad m^2 + i}\,e^{\ ip(x_2\ \ x_1)} \tag{2.22}$$

in position-space, with the addition of the term i to the denominator to avoid singularities on the real axis.

2.5 The Photon Propagator

The derivation of the photon propagator cannot be done by only following the standard procedure for the Klein-Gordon and the free Dirac eld, due to having to pick a gauge at some point. Firstly, we shall retrace the steps we have done so far for the previous elds. Afterwards, we will see what problems arise and then, we will give the correct answer by following the procedure proposed by Faddeev and Popov. The two-point propagator can be written:

$$D_{P\,H(\quad)}(x_2\quad x_1) = \frac{\int DA\ A_{\ }(x_2)A_{\ }(x_1)\,e^{iS_{P\,H}[A]}}{\int DA\ e^{iS_{P\,H}[A]}} \tag{2.23}$$

where $DA = \prod_x \prod_{=0}^{3} dA_{\ }(x)$ and $S_{P\,H}[A] = \frac{1}{4}\int d^4x F_{\quad}F_{\quad}(F_{\quad} = @_{\ }A_{\ } - @_{\ }A_{\ })$. Integrating by parts the photon Lagrangian density we get:

$$S_{P\,H}[A] = \frac{1}{2}\int d^4x\ A_{\ }(x)(@_{\ }@_{\ }\qquad\quad @_{\ }@_{\ })A_{\ }(x) \tag{2.24}$$

showing that, like before, generating the propagator is indeed reduced to a polynomial mean value problem with gaussian weight. Once again we discretize space-time ($x \to x_i$) and perform a Wick rotation ($t \to it^E$) from Minkowski to Euclidean space:

$$D_{PH(\ \ )}(x_2 \quad x_1) = \\ = \frac{\int \prod_{i=0}^{3} dA\ (x_i)\ A\ (x_2) A\ (x_1) exp\left[-\frac{1}{2} \sum_{i,j;\ =0}^{3} A\ (x_i) K_{ij}\ A\ (x_j) \right]}{\int \prod_{i=0}^{3} dA\ (x_i)\ exp\left[-\frac{1}{2} \sum_{i,j;\ =0}^{3} A\ (x_i) K_{ij}\ A\ (x_j) \right]} \tag{2.25}$$

with $K_{ij} = (@\ @ \quad @\ @\ )_{ij}$. The above propagator is once again the following:

$$D_{PH(\ \ )}(x_2 \quad x_1) = (K^{\ 1})_{2;1} \tag{2.26}$$

Going back to the continuous Minkowski space, we use the following expression:

$$(@\ @ \quad @\ @\ ) D_{PH(\ \ )}(x_2 \quad x_1) = \ i\ ^{(4)}(x_2 \quad x_1) \tag{2.27}$$

Using Fourier transformations on the above we get:

$$\int \frac{d^4p}{(2\ )^4} (@\ @ \quad @\ @\ ) \Theta_{PH(\ \ )}(p) e^{\ ip(x_2 \quad x_1)} = \ i \int \frac{d^4p}{(2\ )^4} e^{\ ip(x_2 \quad x_1)}$$

$$) \int \frac{d^4p}{(2\ )^4} ((\ ip\ )(\ ip\ ) \quad (\ ip\ )(\ ip\ )) \Theta_{PH(\ \ )}(p) e^{\ ip(x_2 \quad x_1)} =$$

$$= \ i \int \frac{d^4p}{(2\ )^4} e^{\ ip(x_2 \quad x_1)}$$

$$) \ (p^2 \quad p\ p\ ) \Theta_{PH(\ \ )}(p) = \ i \tag{2.28}$$

Eq.(2.28) has no solution, due to the 4 4 matrix $(p^2 \quad p\ p\ )$ being non-invertible. Indeed, p is an eigenvector with eigenvalue zero, which means the determinant of the matrix is zero, therefore the statement that the matrix is non-invertible holds true. Consequently, when the photon eld becomes $\hat{A}\ (p) \quad p\ \theta(p)$ or equivalently $A\ (x) \quad @\ (x)$ in position-space, where $\theta(p);\ (x)$ are scalar functions, the photon action becomes zero. Because the photon eld and its action obey gauge symmetry, it would be desirable to split DA into two:

$$DA = DA_{GI} DA_{GE} \tag{2.29}$$

where \GI" and \GE" mean gauge inequivalent and gauge equivalent, respectively. Integration over A_{GI} involves fields that are gauge inequivalent, whereas integration over A_{GE} contains all the gauge transformations of a field. One can imagine a vector space where every point represents a configuration of field the A , with all the gauge equivalent fields residing on a \sheet" (gauge orbit) in this configuration space. In particular, one of these sheets contains a subset of configurations A $(x) = (1=e)@$ (x), which are gauge transformations of $A = 0$. The components of this subset have an action $S_{PH} = 0$ thus, integration over A_{GE} diverges since $e^{iS_{PH}(A)} ! 1$ and $\int DA_{GE}$ is performed for every possible configuration. With the configurations being infinite and the former integral diverging, the total integral $\int DAe^{iS_{PH}(A)}$ diverges too, consequently leading us to the conclusion that the definition we adopted for the photon propagator is indeed problematic. One solution to this problem is to only choose one \representative" field configuration from each sheet, to avoid integrating over the infinite equivalent ones that would otherwise lead to a divergent functional integral.

Faddeev and Popov came up with a way to produce the correct answer. To isolate a representative field configuration, selected a \gauge- fixing" functional, $G[A]$, such that the condition $G[A] = 0$ be obeyed by only one configuration from each gauge orbit. To this end, they inserted into the numerator and the denominator of eq.(2.23) the identity:

$$1 = \int D \quad (G[A(\ )])det \ \frac{G[A(\ )]}{} \tag{2.30}$$

The transformation of the field is:

$$A^0 = A \quad \frac{1}{e}@ \tag{2.31}$$

The simplest gauge fixing functional one may choose to use corresponds to the generalized Lorentz gauge. Properly handling the redefined expression of the propagator, it can be shown that the propagator can be written as follows:

$$D_{PH(\ )}(x_2 \quad x_1) = \frac{\int DA \ A \ (x_2)A \ (x_1) \ e^{\left[i \ S_{PH}[A]+S_{GF}[A] \right]}}{\int DA \ e^{\left[i \ S_{PH}[A]+S_{GF}[A] \right]}} \tag{2.32}$$

where S_{GF} is a gauge fixing action which stems from the Faddeev and Popov procedure and can be written as:

$$S_{GF} = \int d^4x A \ (x) \ \frac{1}{}@ \ @ \quad A \ (x) \tag{2.33}$$

and is any finite constant.
We can see that the functional integrals in eq.(2.32) are again reduced to a

polynomial with gaussian weight, with the only difference being an extra gauge fixing term in the total photon action. Hence retracing our steps from the beginning of this section leads us to the corrected form of eq.(2.28):

$$p^2 \quad 1 \quad \frac{1}{} \; p\ p \quad \tilde{D}_{PH(\ )}(p) = \quad i \tag{2.34}$$

The matrix multiplying $\tilde{D}_{PH(\ )}(p)$ is now reversible:

$$\frac{1}{p^2} \quad (1 \quad a)\frac{p\ p}{p^4} \quad p^2 \quad 1 \quad \frac{1}{} \; p\ p \quad \tilde{D}_{PH(\ )}(p) =$$

$$= \quad i \; \frac{1}{p^2} \quad (1 \quad a)\frac{p\ p}{p^4}$$

$$) \quad \tilde{D}_{PH(\ )}(p) = \quad i \; \frac{1}{p^2} \quad (1 \quad a)\frac{p\ p}{p^4} \tag{2.35}$$

Eq.(2.35) is the correct photon propagator with its equivalent form in position-space being:

$$\tilde{D}_{PH(\ )}(x_2 \quad x_1) = \int \frac{d^4p}{(2\ )^4} \; \frac{i}{p^2 + i} \quad (1 \quad a)\frac{p\ p}{p^2 + i} \; e^{\ ip(x_2 \quad x_1)} \tag{2.36}$$

2.6 The Gluon Propagator

The derivation of the gluon propagator demands a similar procedure to that of the photon propagator. We should keep in mind though that gauge transformations in this case are non-abelian. Because the term $i=g\underline{G}(x)@\ \underline{G}^{\ 1}(x)$, that occurs when locally transforming the gluon field, is a traceless hermitian component of the Lie algebra of SU(3) , the hypothesis that the gluon field is also a component of the Lie algebra, is self-consistent. Therefore one can express the field $\underline{A}\ (x)$ as follows:

$$\underline{A}\ (x) = \sum_{a=1}^{8} A^a(x)\frac{^a}{2} \tag{2.37}$$

where $A^a(x)$, consists of eight different gluon fields, each corresponding to one of the eight generators ^{a} of SU(3). These generators are 3 3 Gell-Mann matrices and satisfy the following commutation and orthogonality relations:

$$[\ ^a;\quad] = 2i \sum_{c=1}^{8} f^{abc}\ \ ^c$$
$$tr(\ ^a\ ^b) = 2\ ^{ab}$$

Following the photon propagator procedure, for the gluon propagator, we arrive at a point where the same problems arise. At that same point, we must insert eq.(2.30) to the numerator and denominator of the expression we first adopted for the gluon propagator (which is similar to eq.(2.23) but the field is a 3 3 matrix now). The determinant in the identity we are about to use is not independent of the field. Fadeev and Popov treated this determinant by expressing it as a functional integral over a new set of Grassmann variables:

$$det\ \frac{G[A\ ]}{} = \int DcDc\ exp\ i \int d^4x\ c\quad g\frac{G[A\ ]}{}\ \ c \tag{2.38}$$

The transformation of the field is:

$$(A^a)^{\prime} = A^a + \frac{1}{g}@\quad ^a + f^{abc}A^b\ \ ^c = A^a + \frac{1}{g}D\quad ^a \tag{2.39}$$

These two new Grassmann variables are referred to as ghost fields, and are used as additional fields in the calculation of Feynman diagrams. As Grassmann variables they satisfy anti-commutation relations event hough they act similar to scalar fields under Lorentz transformations. This leads to a faulty spin-statistics relationship. Nevertheless this does not bother us since they do not represent existent particles. The ghost action (Chapter 16 of Ref.[3]) stemming from eq.(2.38) combined with eq.(2.39) is:

$$S_{ghost} = \int d^4x\ c^a(\ \ @^2\ ^{ac}\quad g@\ \ f^{abc}A^b)c^c \tag{2.40}$$

with the first term giving us the ghost propagator:

$$D^{ab}_{ghost}(x_2\quad x_1) = \int \frac{d^4p}{(2\ )^4}\ \frac{i\ ^{ab}}{p^2}e^{\ ip(x_2\quad x_1)} \tag{2.41}$$

When placing all the pieces together one can see that the total action contributing to a Green's function contains the gluon action, a gauge fixing action just like the case of photons, and an action containing ghost fields. The former two terms only differ from the respective photon expression by a factor of a Kronecker delta of the color indices. To calculate the gluon propagator one must ignore non-quadratic terms in the total action, leaving only the gluon and the gauge fixing action. Hence, the gluon propagator is similar to the photon propagator, multiplied by a Kronecker delta of the color indices:

$$D^{ab}_{G(\ \ )}(p) = \frac{i\ \ ^{ab}}{p^2} \qquad (1 \quad a)\frac{p\ \ p}{p^2} \tag{2.42}$$

The position-space expression is:

$$D^{ab}_{G(\ \ )}(x_2 \quad x_1) = \int \frac{d^4p}{(2\ \ )^4}\ \frac{i\ \ ^{ab}}{p^2 + i} \qquad (1 \quad a)\frac{p\ \ p}{p^2 + i}\ e^{\ ip(x_2 \quad x_1)} \tag{2.43}$$

Chapter 3

Dimensional Regularization

3.1 Dimensional Regularization Formulation

Now that we have calculated the propagators of the different fields, it is time to discuss interacting theories. One may need to calculate a Green's function of an operator $O(\ ;\ ;\ ;\ldots)$ that involves various fields and their derivatives, at a point in space-time. This becomes complicated even for a two-point Green's function since the action, which acts as Gaussian weight in the integral, does not only contain the free field term but also contains interaction terms. In perturbation theory, the exponential of the action is expanded as a power series of the coupling constant: the first term gives us the propagator and the rest are perturbative corrections. Those integrals can be diagrammatically represented as Feynman diagrams. Usually UV-divergences show up when calculating these diagrams and therefore the need to renormalize our theory arises. In this chapter we will present a powerful method to perform the above calculations, the dimensional regularization. Although this method is not applicable in the complete theory, it is consistent and well-defined within the framework of perturbation theory and in most cases simpler to use in calculations. Also, keep in mind that we will introduce dimensional regularization in the form which applies to Euclidean spacetime.

Invoking this method, one works in $d = 4 \quad 2$ dimensions and then expands in powers of . However, since working in a non-integer number of dimensions is abstract, we have to be careful with our definitions in order to avoid inconsistencies. Firstly, let us define an "operation" that acts as integration in d-dimensions, expressed in spherical coordinates[1]:

$$\int d^d x f(x) = \prod_{i=1}^{d\Upsilon 1} \int sin^{i\ 1}(\ _i) d\ _i \int djxj\, jxj^{d\ 1} f(x) \tag{3.1}$$

[1] It is understood that the limits of all $d^d x$ integrals are $(\ 1\ ;\ 1\ )$

with $_1 \in [0; 2\ ]$, $_{j\neq 1} \in [0;\ ]$ and $jxj = \sqrt{x_1^2 + x_2^2 + \ldots + x_d^2} \in [0; 1)$. We shall refer to jxj as x for simplicity. In case $f(x) = f(x)$, meaning it does not depend on the direction of x, integration over angles $_i$ can immediately be performed by using (p.106 of Ref.[5]):

$$\int_0 \sin^m(\ )d\ = \sqrt{\ } \frac{(\frac{m+1}{2})}{(\frac{m+2}{2})} \tag{3.2}$$

consequently leading to an integral of the following form:

$$\int d^d x\ f(x) = \frac{2^{\frac{d}{2}}}{(\frac{d}{2})} \int_0 dx\ x^{d\ 1} f(x) \tag{3.3}$$

Here, is the gamma function which is usually de ned as:

$$(d) = \int_0 x^{d\ 1} e^{\ x} dx \tag{3.4}$$

The function's argument can be a complex number as long as its real part is positive. We therefore de ne the gamma function as the analytical continuation of eq.(3.4) to a meromorphic[2] function that is holomorphic[3] in the whole complex plane except the non-positive integers, where the function has simple poles. This behavior, as we will see later on, will be one of the manifestations of divergences in Feynman diagrams as $\ \to\ 0$.

With eq.(3.3) in mind, let us proceed by imposing the following axioms[2]:

1. Linearity: For any complex numbers a and b:

$$\int d^d x\ [af(x) + bg(x)] = a \int d^d x\ f(x) + b \int d^d x\ g(x) \tag{3.5}$$

2. Scaling: For any number s:

$$\int d^d x\ f(sx) = s^{\ d} \int d^d x f(x) \tag{3.6}$$

3. Translation invariance:

$$\int d^d x\ f(x + y) = \int d^d x\ f(x) \tag{3.7}$$

[2] A meromorphic function on an open subset D of the complex plane is a function that is holomorphic on all of D except for a set of isolated points, which are poles of the function.

[3] A holomorphic function is a complex-valued function of one or more complex variables that is, at every point of its domain, complex di erentiable in a neighborhood of the point. The existence of a complex derivative in a neighbourhood is a very strong condition, for it implies that any holomorphic function is actually in nitely di erentiable and equal, locally, to its own Taylor series.

4. Invariance under rotations.

Linearity is true for any integration. Translation and rotation invariance are basic properties of Euclidean space, while the d-dimensionality is embedded in the scaling property. It can be proven that these four axioms ensure that the desired integration operation is uniquely de ned.

Eq.(3.3) satis es the above requirements thus, we can associate it with d-dimensional integration, as it holds for any d. Obviously, this is only true if $f(x) = f(x)$ and therefore, it will be su cient only for integrals that are only dependent on the magnitude of the vector argument. Nevertheless, for cases where separate components of the argument appear, a d-dimensional integration also exists and we will provide the relevant result when necessary.

3.1.1 Gamma Matrices in d-Dimensions

As we are no longer in four dimensions, we need to de ne a new set of gamma matrices that are compatible with the d-dimensional regularization. The algebra needs to be represented by in nite-dimensional matrices. However, because we desire that the representation behaves as its d-dimensional counterpart, we de ne the trace operation on the identity matrix as

$$Tr(1) = 4 \tag{3.8}$$

To be consistent, we demand that the d-dimensional gamma matrices satisfy the anticommutation relation:

$$f \quad ; \quad g = 2 \quad 1 \tag{3.9}$$

Therefore, we arrive at the following results:

$$= \frac{1}{2} f \quad ; \quad g = d1 \tag{3.10}$$

$$= 2 \qquad\qquad = (2 \quad d) \tag{3.11}$$

$$Tr(\qquad) = \quad Tr(\qquad) + 2 \quad Tr(1)$$

leaving us with:

$$Tr(\qquad) = 4 \tag{3.12}$$

We must also rede ne $_5$ (which was $_5 = _1 _2 _3 _4$ in four-dimensional Euclidean space). A consistent de nition [2] is obtained when writing:

$$ {}_5 = \frac{1}{4!} \tag{3.13} $$

where the Levi-Civita symbol, or the totally antisymmetric tensor:

$$ = \begin{cases} 1 \text{ if } (\quad) \text{ is an even permutation of (0123),} \\ 1 \text{ if } (\quad) \text{ is an odd permutation of (0123)} \\ 0 \text{ otherwise.} \end{cases} \tag{3.14} $$

Consequently, the d-dimensional ${}_5$ satisfies:

$$ \begin{aligned} f\ {}_5;\quad g = \mathbb{0} \text{ if } &= 0;1;2;3 \\ [\ {}_5;\quad] = \mathbb{0} &\text{ otherwise} \\ ({}_5)^2 = 1;\quad \tfrac{y}{5} = {}_5: & \end{aligned} \tag{3.15} $$

It is obvious that this definition is not Lorentz invariant on the full space, but only on the first four dimensions. However, even though this lack of full Lorentz invariance is a nuisance, we will use this definition, as it gives the correct axial current anomaly.

3.1.2 QCD in d-Dimensions

Having given the necessary definitions, let us now continue to the study of the dimensional regularization of QCD. We will use the continuum action with the only difference that we are no longer integrating over the four-dimensional space-time, but over the abstract concept of $4\quad 2$ dimensions. Before we proceed with the evaluation of Green's functions, let us first perform a simple dimensional analysis. Since we want the action to be dimensionless, every term of the Lagrangian has to be of dimension d. From that, we can work out the `mass' dimensions of the fields and coupling constant:

$$ \begin{aligned} @ \qquad &! \ 2[\ ] + 1 \qquad &= d \ ! \ [\ ] = \tfrac{d\ 1}{2}. \\ (@\ A\ )^2 \ &! \ 2[A] + 2 \qquad &= d \ ! \ [A] = \tfrac{d\ 2}{2}. \\ g_0\ A \quad &! \ [g_0] + 2[\ ] + [A] &= d \ ! \ [g_0] = \tfrac{4\ d}{2}. \end{aligned} $$

For $d = 4\quad 2$, the coupling constant g_0 has dimensions of . Let us define a dimensionless quantity:

$$ g_0 = g \tag{3.16} $$

where is an arbitrary mass scale.

For convenience in calculations, we also define a scale through:

$$= \left(\frac{4}{e^{\ E}} \right)^{\frac{1}{2}} \tag{3.17}$$

where $_E = 0.57721\ldots$ is the Euler-Mascheroni constant.

The perturbative method of evaluating Green's functions by summing Feynman integrals is as follows:

Expanding our exponential of the action as a power series of the coupling constant, leads to the appearance of terms containing di erent vertices (parts of the action consisting of a product of three or more elds and/or their derivatives). Invoking Wick's theorem, we form loop diagrams by contracting the vertices (i.e. substituting pairwise elds by the respective propagators). We de ne a one particle irreducible (1PI) diagram to be any diagram that cannot be split in two by removing a single line. Let $(\not{q})$ denote the sum of all 1PI diagrams with two external fermion lines. In order to form $(\not{q})$ we consider amputated diagrams, that is diagrams without Feynman propagators for the external lines. It can be shown that the two-point fermion Green's function assumes the form of a geometric series, hence:

$$\begin{aligned} h \quad (q) \quad (q^0)i &= (2 \quad)^4 \ (q \quad q^0) \ \frac{1}{i\not{q} + m} + \frac{1}{i\not{q} + m} \quad (\not{q}) \frac{1}{i\not{q} + m} + \ldots \\ &= (2 \quad)^4 \ (q \quad q^0) \ \frac{1}{i\not{q} + m \qquad (\not{q})} \end{aligned} \tag{3.18}$$

and by writing $(\not{q})$:

$$(\not{q}) = \ _0(q) + i\not{q} \ _1(q) \tag{3.19}$$

since $\not{q}^2 = q^2$, we then get:

$$h \quad i = (2 \quad)^4 \ (q \quad q^0) \ \frac{1}{i\not{q}[1 \qquad _1(q)] + m[1 \qquad _0(q)]} \tag{3.20}$$

By translational invariance, the multiplicative delta function over momenta will always be present when considering the Green's function of two operators. Requiring that the factor $[1 \quad _1(q)]$ be nite will lead to the renormalization of the fermion eld; further, requiring that the term involving the mass in eq.(3.20) be also nite leads to the fermion mass renormalization.

3.2 The $\overline{\text{MS}}$ Renormalization Sceme

In dimensional regularization, the divergences arise from poles in . Speci cally, when performing a perturbative calculation, the Laurent series of the Green's

function at one loop has the general form:

$$\langle O(q)\rangle = \frac{1}{\epsilon}(\ldots) + {}^{0}(\ldots) + {}^{1}(\ldots) + \ldots \tag{3.21}$$

To return to four dimensions, we take the limit $\epsilon \to 0$ and we are left with the first two terms, a divergent term and a finite term, which we refer to as the leading and sub-leading terms. The $\overline{\text{MS}}$ (Minimal Subtraction) renormalization scheme eliminates the $1/\epsilon$ pole that appears in the first term while leaving the finite term unaltered.

To illustrate the application of $\overline{\text{MS}}$, let us again calculate the two point Green's function up to order $O(g^2)$. In dimensional regularization, we have only a single one-loop diagram to consider:

Figure 3.2.1: One loop contribution to the two-point function in the continuum.

We need to contract two antiquark-gluon-quark vertices, each being:

$$ig(2\pi)^4 \quad (p_1 \quad p_2 + p_3) \quad T^a \tag{3.22}$$

The delta function suggests the conservation of momentum in the vertex. The convention here is that $p_1; p_2; p_3$ are the momenta of the gluon, antiquark and quark respectively. After summing over the colors and renaming indices we are left with:

$$(q) = \frac{N^2 \quad 1}{2N} g^2 \int d^4p_1 \int d^4p_2 \quad (p_1 \quad p_2 + q_1) \; (p_1 \quad p_2 + q_2) \quad D_F(p_2) \quad D_G(p_1) \tag{3.23}$$

where

$$\begin{aligned} D_F(p) &= \frac{ip\!\!\!/ + m}{p^2 + m^2} && \text{fermion propagator} \\ D_G(p) &= \frac{1}{p^2}\left[g \quad \frac{p\ p}{p^2} \right] && \text{gluon propagator} \end{aligned} \tag{3.24}$$

$= 1$ and the factor $C_F \quad (N^2 \quad 1)/(2N)$ appears upon summing over color generators. We are only interested in the amputated diagram therefore we do not multiply by external fermion propagators.

To perform integration over the loop momentum in d-dimensions we make use of the following formulae [1]:

$$\int \frac{d^d p\, P_n(p)}{(2\ )^d p^2\ \ (p\quad k)^2} = \frac{(k^2)^{2\ \ ''}}{(4\ )^2} \sum_0^{\ } G(\ ;\ ;n;\ )\, k^2 \quad \frac{1}{!}\ \frac{2_p}{4} \quad P_n(p)\ \Big|_{p=k} \tag{3.25}$$

where $2_p = @^2 = @p\ @p$, and

$$G(\ ;\ ;n;\ ) = (4\ )^{''} \frac{(\ +\ \ \ \ \ \ \ \ \ 2+'')}{(\ )\ (\ )} B(2\ \ ''\ \ +n\ \ ;2\ \ ''\ \ +\ ) \tag{3.26}$$

where $P_n(p)$ a polynomial in p such that $P_n(\ p) = \ ^n P_n(p)$ and B is the beta-function:

$$B(p;q) = \frac{(p)\ \ (q)}{(p+q)}: \tag{3.27}$$

We may focus on the massless case, setting the mass to be zero. Expanding in a Laurent series around $= 0$ we get:

$$_1(q) = \frac{1}{16\ ^2} g^2(1\quad\ )C_F \left[\ \frac{1}{\ } \quad 1 + \ln\ \frac{^2}{q^2}\right] \tag{3.28}$$

Plugging the above in eq.(3.20) (bear in mind that we set m=0), one arrives at the Green's function result up to $O(g^2)$.

In order to eliminate the $1=$ pole, we must multiply the elds with the $\overline{\text{MS}}$ renormalization factor:

$$_B = \sqrt{Z}\ \ _R$$

where B and R indicate bare and renormalized elds, respectively.

The $\overline{\text{MS}}$ renormalization factor Z is necessarily of the form:

$$Z\ \ = 1 + \frac{1}{\ } g^2 c \tag{3.29a}$$

We may now derive a value for the constant c, in such a way that the $1=$ term vanishes, leading us to a nite Green's function:

$$\begin{aligned} Z^{DR;\overline{\text{MS}}}[1\quad\ (q)] &= \left[1 + \frac{1}{''} g^2 c\right]\left[1 + \frac{1}{''}\frac{1}{16\ ^2} g^2(1\quad\ )C_F\right] \\ &= 1 + \frac{1}{''}\frac{1}{16\ ^2} g^2(1\quad\ )C_F + \frac{1}{''} g^2 c + O(g^4) \end{aligned} \tag{3.29b}$$

If $c = (1=16\ ^2)(\quad 1)C_F$, then the $1=$ terms cancel out and thus we are left with

$$Z^{DR;\overline{MS}} = 1 \quad \frac{1}{"}\frac{1}{16\ ^2}g^2(1\quad)C_F \tag{3.30}$$

An alternative renormalization scheme may include in the de nition of Z, in addition to $1=$ terms, also nite contributions: A widely employed renormalization scheme is the RI' (Regularization Invariant), in which Z is de ned by the requirement that the two-point function of the renormalized fermion eld coincide with the tree-level expression at a certain value q of the external momentum. To achieve this, by following steps analogous to eqs.(3.29a, 3.29b), we are led to:

$$Z^{DR;RI'} = 1 + \frac{1}{16\ ^2}g^2(1\quad)C_F \quad \frac{1}{"} \quad 1 + \ln \frac{^2}{q^2} \tag{3.31}$$

We can better express it:

$$\begin{aligned} Z^{DR;RI'} &= 1 \quad \frac{1}{16"\ ^2}g^2(1\quad)C_F \quad 1 + \frac{1}{16\ ^2}g^2(1\quad)C_F \quad 1 + \ln \frac{^2}{q^2} \\ Z^{DR;RI'} &= Z^{DR;\overline{MS}}\ 1 + \frac{1}{16\ ^2}g^2(1\quad)C_F \quad 1 + \ln \frac{^2}{q^2} \end{aligned} \tag{3.32}$$

leaving us with an expression which shows that converting the renormalization factor from one scheme to another is possible, with the use of a nite conversion factor:

$$C^{\overline{MS};RI'} = \frac{Z^{DR;RI'}}{Z^{DR;\overline{MS}}} = \frac{Z^{L;RI'}}{Z^{L;\overline{MS}}} = 1 + \frac{1}{16\ ^2}g^2(1\quad)C_F \quad 1 + \ln \frac{^2}{q^2} \tag{3.29}$$

Clearly the following reverse conversion also holds:

$$C^{RI';\overline{MS}} = \frac{1}{C^{\overline{MS};RI'}} \tag{3.30}$$

The usefulness of the RI' scheme stems from its applicability beyond perturbation theory, and beyond dimensional regularization.

Chapter 4

Gauge-Invariant Renormalization Scheme (GIRS)

As we desire to use a gauge-invariant renormalization scheme, we could focus exclusively on the $\overline{\text{MS}}$ renormalization scheme, which is the most widely used and is also gauge-invariant. Indeed, a renormalization of the fields and parameters is possible perturbatively, without need for any other intermediate schemes. To do this, one must calculate bare Green's functions in the continuum and by subtracting poles in ε end up with $\overline{\text{MS}}$-renormalized Green's functions or calculate the same bare Green's functions on the lattice, and introduce renormalization constants in such a way that the resulting Green's functions coincide with those in $\overline{\text{MS}}$.

Although these tasks ensure that we are left with a gauge-invariant renormalization scheme, this procedure entailed performing perturbative calculation both on the lattice and in the continuum. While perturbative calculations are easier in the continuum (and also unavoidable by the very nature of the $\overline{\text{MS}}$ scheme), they become exceedingly complicated on the lattice, and consequently, calculations beyond two loops are practically unfeasible.[1] In addition, even some two-loop calculations become prohibitive for some "improved" actions, such as the ones used in many large-scale simulations nowadays. In practice, lattice perturbative results are typically limited to one loop, and this can lead to large systematic errors. An alternative scheme, which does not necessitate lattice perturbation theory, is the RI' scheme which can be used on any Green's function, including, e.g. two-point Green's functions. This renormalization scheme, although it can be used in simulations and thus lead to more precise results, is not gauge-invariant as we wish.

[1] The only three-loop calculations on the lattice existing thus far regard "vacuum" diagrams, that is diagrams without external lines and momenta.

In this chapter we aim to construct a renormalization scheme that has both of the desired properties: being gauge-invariant and nonperturbative. We will not use this scheme to renormalize the fields (which are gauge-variant by their very definition), but instead to renormalize gauge-invariant composite operators. In section 4.2 we will define the new scheme and in the sections that follow we will obtain a value for conversion factors from this scheme to $\overline{\text{MS}}$ by evaluating the relevant Green's function up to one loop.

4.1 Composite operators

We will study composite operators of the following form

$$O^{\Gamma}(x) = \bar{\psi}^{f}(x)\,\Gamma\,\psi^{f'}(x) \tag{4.1}$$

where $\Gamma = 1, \gamma_\mu, \gamma_5, \gamma_5\gamma_\mu$, or $\Gamma = [\gamma_\mu, \gamma_\nu]$.

The indices f and f' refer to other quantum numbers, such as flavor, which may be present in the definition of the elementary fermion fields. The renormalization constants for flavor-singlet and -nonsinglet operators will differ in general; however, given that this difference will only appear beyond one loop, we will not consider it any further from this point on, and we will drop flavor indices from our notation.

Composite operators can also involve derivatives of fields:

$$O^{\Gamma}_{\mu_1\mu_2\ldots}(x) = \bar{\psi}(x)\,\Gamma\,D_{\mu_1}D_{\mu_2}\ldots\psi(x), \tag{4.2}$$

but we will limit our attention to those of eq. (4.1).

Depending on the choice of Γ, the operators behave under a Lorentz transformation and under parity as[2] [3]:

1	scalar
γ_μ	vector
γ_5	pseudoscalar
$\gamma_5\gamma_\mu$	axial vector
$[\gamma_\mu, \gamma_\nu]$	tensor

[2] The pseudoscalar and axial vector operators acquire an extra minus sign under parity, as compared to the scalar and vector operators, respectively.

The above composite operators or \Dirac bilinears", appear frequently in the study of the eigenstates of a spectrum of a theory (i.e. hadrons) and therefore it is meaningful to construct a renormalization scheme for them.

Given that the limit of the product of two fields typically diverges (i.e. the Green's function containing such products diverge) as these fields are brought near each other, one must treat such cases carefully to remove these divergences and obtain the renormalized composite operator. To renormalize them, one studies Green's functions involving these operators and a number of fundamental fields.

Another possibility is to use a Green's function that contains only the product of two composite operators, defined at different points, as given by

$$G(x - y) = \langle O_{\Gamma}(x) O_{\Gamma'}(y) \rangle ; \tag{4.3}$$

where Γ' can, in principle, differ from Γ. Therefore, since composite operators are gauge-invariant, choosing eq.(4.3) ensures that both the Green's function and the renormalized operators are independent of the gauge. The condition which will define the new renormalization scheme stems from $G(x - y)$, summed over all spatial directions:

$$\int d^3\vec{x}\, \langle O_{\Gamma}(x) O_{\Gamma'}(y) \rangle : \tag{4.4}$$

Although this expression involves an undesired separation between space and time, we benefit from the fact that it limits our parameters.

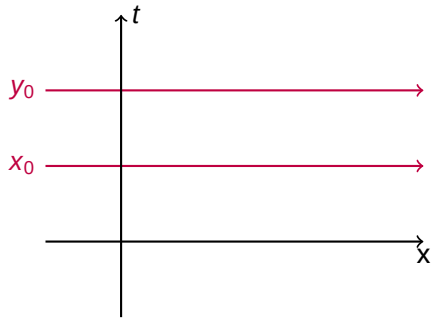


Figure 4.1.1: Integration region for the Green's function. The time interval $t = y_0 - x_0$ remains fixed, while the spatial components of x are integrated over.

For any speci c values of the and ^{0} matrices, the resulting expression (4.4) will depend only on the time component t of the four-vector $y \quad x$: $t \quad (y_0 \quad x_0)$. Since we will be focusing on the massless case no dependence on the masses, that would otherwise exist, will appear. In addition, the integration over the spatial components of x produces less statistical error in simulations. Clearly, we integrate over all spatial components while keeping a constant non-zero time interval between points x & y, to avoid the region $jx \quad yj \; ! \; 0$, which possibly contains further \contact" singularities.

4.2 GIRS definition

Let us de ne this scheme. We rstly demand that:

$$Z \quad Z_{0} \quad \int d^3x \; O \; (x)O_{0}(y) \quad = \text{tree level value} \tag{4.5}$$

for a certain value of $jx_0 \quad y_0j = jtj$. This condition is similar to that of the RI' scheme. One can satisfy the described condition in a perturbative calculation or in a nonperturbative one. Indeed, in a lattice simulation, one can calculate the integral in (4.5) and nd the renormalized operators:

$$O^{\text{GIRS}} = Z^{L;\text{GIRS}} O^{L} \tag{4.6}$$

As O^{GIRS} is independent of the regulator, we can also calculate it using dimensional regularization:

$$O^{\text{GIRS}} = Z^{DR;\text{GIRS}} O^{DR} \tag{4.7}$$

However, since $\overline{\text{MS}}$ is the most widely used renormalization scheme, we want to convert our results to that scheme. So, similarly with eqs. (4.6) and (4.7), we have:

$$\begin{aligned} O^{\overline{\text{MS}}} &= Z^{L;\overline{\text{MS}}} \, O^{L} \\ &= Z^{DR;\overline{\text{MS}}} \, O^{DR} \end{aligned} \tag{4.8}$$

$O^{\overline{\text{MS}}}$ and O^{GIRS} di er by a nite multiplicative factor:

$$O^{\overline{\text{MS}}} = \frac{Z^{L;\overline{\text{MS}}}}{Z^{L;\text{GIRS}}} \, O^{\text{GIRS}} = \frac{Z^{DR;\overline{\text{MS}}}}{Z^{DR;\text{GIRS}}} \, O^{\text{GIRS}} \tag{4.9}$$

and therefore we obtain:

$$Z^{L;\overline{\mathrm{MS}}} = Z^{L;\mathrm{GIRS}} \frac{Z^{DR;\overline{\mathrm{MS}}}}{Z^{DR;\mathrm{GIRS}}} \quad Z^{L;\mathrm{GIRS}} C^{\mathrm{GIRS};\overline{\mathrm{MS}}} \tag{4.10}$$

The renormalization factor $Z^{L;\mathrm{GIRS}}$ can be evaluated directly in a lattice simulation using the GIRS condition of eq.(4.5) . As we wish to obtain results in the $\overline{\mathrm{MS}}$ scheme, we are interested in the conversion factor $C^{\mathrm{GIRI};\overline{\mathrm{MS}}}$ that converts the renormalization factor from the GIRS scheme to the $\overline{\mathrm{MS}}$ scheme. By comparing eq.(4.10) with eq.(4.9), we see that the conversion factor is independent of the regularization method. With that knowledge we will proceed by calculating it perturbatively using dimensional regularization, as it is more convenient for calculations.

4.3 Tree Level

4.3.1 Tree Level in 4-dimensions

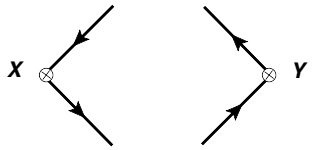
The Green's function that we will use to obtain the conversion factor is the following:

$$\begin{aligned} G(t) &= \int d^3x \; O^{f f^0}(x) O^{f^0 f}_{0}(y) \\ &= \int d^3x \quad {}^{f}(x) \quad {}^{f^0}(x) \; {}^{f^0}(y) \; {}^{0} \; {}^{f}(y) \end{aligned} \tag{4.11}$$

where the avor indices $f;\ f^0$ are chosen in a way that we will get a non-zero result. The Green's function can be expressed perturbatively as[3]:

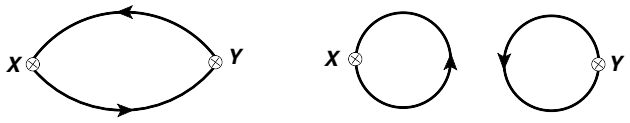
$$G(t) = G^{(0)}(t) + g_0^2 G^{(1)}(t) + g_0^4 G^{(2)}(t) + :::: \tag{4.12}$$

At tree level the vertices for contraction are



where the arrows pointing inwards and outwards indicate fermions and antifermions respectively. After contraction, we are left with two possibilities:

[3] Beyond tree level, the Green's function depends also on a scale parameter .



The disconnected diagram on the right vanishes for avor-nonsinglet operators. For avor-singlets it takes the form:

$$G_b(0) = N^2 \mathrm{Tr}\ \frac{i\not{q}_1}{q_1^2} \quad \mathrm{Tr}\ \frac{i\not{q}_2}{q_2^2}\ {}^{0} \tag{4.13}$$

which is zero for massless fermions, and for any value of and ^{0}. In the massive case, a nonzero contribution will arise if both and ^{0} are unit matrices, i.e. for the avor-singlet scalar operator. However, since we are interested in normal ordered operators[4], we will neglect this contribution and instead focus solely on the connected diagram. We perform a Fourier expansion on the elds:

$$\begin{aligned} (x) &= \int \frac{d^4p_1}{(2\ )^4}\,\tilde{}(p_1)e^{\ ip_1\ x} \\ (x) &= \int \frac{d^4p_2}{(2\ )^4}\,\tilde{}(p_2)e^{ip_2\ x} \end{aligned} \tag{4.14}$$

and similarly for (y), (y). Thus, we obtain

$$\begin{aligned} G^{(0)}(t) = \quad & N \int d^3x \int \frac{d^4p_1}{(2\ )^4}e^{\ ip_1x} \int \frac{d^4p_2}{(2\ )^4}e^{ip_2x} \int \frac{d^4p_3}{(2\ )^4}e^{\ ip_3y} \int \frac{d^4p_4}{(2\ )^4}e^{ip_4y} \\ & (2\ )^4\ (p_2 \quad p_3)(2\ )^4\ (p_4 \quad p_1)Tr\ \frac{1}{i\not{p}_2}\ {}^{0}\frac{1}{i\not{p}_1} \\ = \quad & N \int d^3x \int \frac{d^4p_1}{(2\ )^4} \int \frac{d^4p_2}{(2\ )^4}e^{i(p_2 \quad p_1)(x \quad y)}Tr\ \frac{1}{i\not{p}_2}\ {}^{0}\frac{1}{i\not{p}_1} \end{aligned} \tag{4.15}$$

where N is the number of colors which appears by summing over color indices. In order to understand how the trace emerges, let us write eq. (4.11) in matrix form (ignoring integration):

[4] In quantum eld theory, an operator (product of quantum elds) is in normal ordering if all creation operators in the product are to the left of the annihilation operators. For example, $a^\dagger a^\dagger aa$ is the normal order of $aa^\dagger aa^\dagger$. In general, a normal ordered operator is de ned as:

$$:O: = O \quad hOi:$$

With this de nition, disconnected diagrams do not contribute to the evaluation of Green's functions.

$$i \quad ij \quad j \quad k \quad \overset{0}{kl} \quad l$$

To form a connected diagram, $_k$ must contract with $_j$ and $_i$ with $_l$. To do so, $_l$ has to move to the left of $_i$. It anticommutes with three objects therefore a combinatorial factor -1 arises.

$$\begin{aligned} l \quad i \quad ij \quad j \quad k \quad \overset{0}{kl} &= \frac{1}{i\not{p}_1} \quad li \quad ij \quad \frac{1}{i\not{p}_2} \quad jk \quad \overset{0}{kl} \\ &= Tr \quad \frac{1}{i\not{p}_1} \quad \frac{1}{i\not{p}_2} \quad \overset{0}{} \end{aligned}$$

Let us calculate the tree level contribution for the massless case. Without loss of generality, we choose the x and y coordinates to be $x = (x_1; x_2; x_3; 0)$ and $y = (0; 0; 0; t)$. Let us rename $ = X$ and $^0 = Y$, while $p_1 = p$ and $p_2 = q$:

$$G^{(0)}(t) = N \int d^3x \int \frac{d^4p}{(2\ )^4} \int \frac{d^4q}{(2\ )^4} e^{i(q\ \ p)(x\ \ y)} \frac{1}{p^2\ q^2} Tr(\not{p} X \not{q} Y) \qquad (4.16)$$

Making use of Schwinger parameters [3] we may express p^2 and q^2 as follows:

$$\frac{1}{p^2} = \int_0^{\ 1} d \quad e \quad {}^{p^2} \qquad (4.17)$$

We are left with thirteen integrations, with the integrations over 4-momenta p and q and their respective Schwinger parameters being equivalent. First, we integrate over an individual component of momentum p, using

$$\frac{1}{2} \int dp \quad e \quad {}^{\ p(p\ )^2 \quad i(x\ \ y)p} = \frac{e \quad {}^{(x\ \ y\ )^2=(4\ \ p)}}{2\sqrt{p}} \qquad (4.18)$$

$$\frac{1}{2} \int dp \quad p \quad e \quad {}^{\ p(p\ )^2 \quad i(x\ \ y)p} = i\frac{e \quad {}^{(x\ \ y\ )^2=(4\ \ p)}(x\ \ y)}{4\ \ {}^{3=2}p} \qquad (4.19)$$

To compute the four-dimensional p integral we multiply eq.(4.19), for the slashed component, with one instance eq.(4.18) raised to the third power, for each of the other three components. Then, the integration over the Schwinger parameter $_p$ reads:

$$\int_0^1 d_p \frac{ie^{(x\quad y)^2=(4\ _p)}(x\quad y\ )}{32\ ^2\ ^3_p} = \frac{i(x\quad y\ )}{2\ ^2\ (x\quad y)^2\ ^2}$$

Similarly, we perform the integration over q and $_q$. We obtain the same result as above, but with an i instead of a i. Therefore multiplying the two four-dimensional integrals over p and q we get:

$$\int \frac{d^4p}{(2\ )^4} \int \frac{d^4q}{(2\ )^4} e^{i(q\quad p)(x\quad y)} \frac{1}{p_1^2 p_2^2} p\ q\ = \frac{(x\quad y\ )(x\quad y\ )}{4\ ^4\ (x\quad y)^2\ ^4} \tag{4.20}$$

It is worth noting that the above expression multiplied by a $Tr(\quad X\quad Y)$ is the Green's function $G(x\quad y)$. Remember that due to the components being slashed, summations over and are implied. Interestingly, non-diagonal terms (6=) contribute to the function. Also, there are non-vanishing contributions from some choices of X 6= Y.

Upon integration over the space-like directions of x, only diagonal terms will survive, that is when = . First, we take the time-like components. In that case, $(x\quad y\ )^2 = t^2$:

$$\int \frac{d^3x\ t^2}{4\ ^4(x_1^2 + x_2^2 + x_3^2 + t^2)^4} = \frac{1}{32\ ^2t^3} \tag{4.21}$$

whereas the space-like components give

$$\int \frac{d^3x\ x_i^2}{4\ ^4(x_1^2 + x_2^2 + x_3^2 + t^2)^4} = \frac{1}{96\ ^2t^3} \tag{4.22}$$

with $i = 1;2;3$ (remember that $y_1; y_2; y_3 = 0$). Thus, the full massless tree level contribution to the Green's function is

$$G^{(0)}(t) = \frac{N}{96\ ^2t^3}\ 2Tr(\ _0X\ _0Y) + \sum_{=0}^{3} Tr(\quad X\quad Y) \tag{4.23}$$

We could have foreseen the t-dependence of this result since we chose a Green's function (expression (4.4)) which will only be dependent on the time interval t, while by dimensional analysis we can see that it will behave as $t^{\ 3}$. We then set

$X = Y$ (since the above traces vanish for $X \neq Y$) and we work out the expressions for the eight different options for X: $\{1, \gamma_5, \gamma_0, \gamma_i, \gamma_5\gamma_0, \gamma_5\gamma_i, \gamma_0\gamma_i, \gamma_i\gamma_j\}$. The results are presented in Table 4.1:

X = Y	$G^{(0)}(0;t)$
1	$\frac{N}{4\pi^2 t^3}$
γ_5	$\frac{N}{4\pi^2 t^3}$
γ_0	0
γ_i	$\frac{N}{6\pi^2 t^3}$
$\gamma_5\gamma_0$	0
$\gamma_5\gamma_i$	$\frac{N}{6\pi^2 t^3}$
$\gamma_0\gamma_i$	$\frac{N}{12\pi^2 t^3}$
$\gamma_i\gamma_j$	$\frac{N}{12\pi^2 t^3}$

Table 4.1: Value of tree level contribution to the Green's function G(t) in 4 dimensions. Here $i, j = 1, 2, 3$ and $i \neq j$.

4.3.2 Tree Level in d-dimensions

As the tree level contribution is strictly finite, we were able to perform our calculations in 4 dimensions. However, because we will afterwards divide the one loop contribution (that has a divergent part) by the tree level, we need to express the latter in d-dimensions as well:

$$G^{(0)}(t) = N \int d^{d-1}x \int \frac{d^dp}{(2\pi)^d} \int \frac{d^dq}{(2\pi)^d} e^{i(q-p)(x-y)} \frac{1}{p^2 q^2} Tr(\not{p} X \not{q} Y) \qquad (4.24)$$

As before, we insert two Schwinger parameters α_p and α_q and we then integrate over the 4-momenta $p; q$. The d-dimensional p integral is obtained by using eq. (4.19) for the slashed component and an instance of eq. (4.18) for each of the remaining components. Then, the integration over the Schwinger parameter α_p reads:

$$\int_0^1 d\alpha_p \frac{ie^{-(x-y)^2/(4\alpha_p)}(x-y)}{2^{1+d-d/2}\alpha_p^{1+d/2}} = \frac{i(x-y)}{2^{d/2}(x-y)^{2-d/2}} \quad (d/2)$$

Integrations over q and α_q give the same result as above but with a different sign. Therefore the the two d-dimensional integrals read:

$$\int \frac{d^dp}{(2\pi)^d} \int \frac{d^dq}{(2\pi)^d} e^{i(q-p)(x-y)} \frac{1}{p^2 q^2} (p) \quad (q) = \frac{(x-y)^2 \quad (d/2)^2}{4^{d} (x-y)^{2(d}} \qquad (4.25)$$

where we immediately set the two directions of x to coincide, as non-diagonal terms will vanish upon spatial integration. For the time-like components, where $(x-y)^2 = t^2$, we make use of the following integral [5]:

$$\int d^dp \frac{1}{(p^2 + 2kp + m^2)^a} = \frac{i^{d/2}}{(m^2-k^2)^{a-d/2}} \frac{(a-d/2)}{(a)} \qquad (4.26)$$

while for the space-like components, $(x-y)^2 = x_i^2$, we use:

$$\int d^dp \frac{p\ p}{(p^2 + 2kp + m^2)^a} = \frac{i^{d/2}}{(m^2-k^2)^{a-d/2}} \left(\frac{1}{(a)} (a-d/2)k\ k + (a-1-d/2)\frac{1}{2} (m^2-k^2) \right) \qquad (4.27)$$

with $=$. In both have used the Euclidean version of what is found in [5], and we replaced $d \to d-1$, $a \to d$, $p \to x$, $k \to 0$ and $m \to t$. By setting $d = 4-2\varepsilon$, we finally obtain:

$$G^{(0)}(t) = \frac{N \quad (2 \quad \varepsilon)^2}{8 \quad ^{5=2} \quad \varepsilon \, t^{3 \quad 2} \quad (4 \quad 2\varepsilon)}$$

$$(3=2 \quad \varepsilon) + 2 \ (5=2 \quad \varepsilon) \ Tr(\ _0 X \ _0 Y) + \quad (3=2 \quad \varepsilon) Tr(\quad X \quad Y) \tag{4.28}$$

Our next step is to substitute $X = Y$, considering all 8 possibilities for X and perform the Dirac algebra. However, since we are not in 4 dimensions, we have to be careful with how we handle the gamma matrix $_5$, as, by de nition, it anticommutes only with the four rst Dirac matrices (= 1; 2; 3; 4) and commutes with the other d 4. Thus:

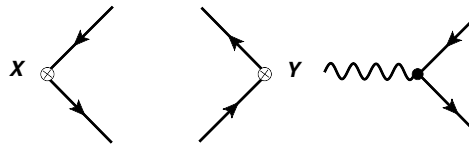
X = Y	$G^{(0)}(0; t)$
1	$\frac{N(\ 3 + 2\varepsilon)\ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$
$_5$	$\frac{3N \ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$
$_0$	0
$_i$	$\frac{N(\ 2 + 2\varepsilon)\ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$
$_5 \ _0$	$\frac{2\varepsilon N \ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$
$_5 \ _i$	$\frac{2N \ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$
$_{0i}$	$\frac{N \ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$
$_{ij}$	$\frac{N(\ 1 + 2\varepsilon)\ (3=2 \quad \varepsilon)\ (2 \quad \varepsilon)^2}{^{5=2} \quad \varepsilon \, t^{3 \quad 2\varepsilon} \quad (4 \quad 2\varepsilon)}$

Table 4.2: Value of tree level contribution to the Green's function G(t) in $d = 4 \quad 2\varepsilon$ dimensions.

Observe that the Green's functions containing the operators $_0$ and $_5\ _0$ vanish in 4 dimensions, in agreement with Table 4.1.

4.4 One-loop order contributions

We are now ready to examine the one-loop contributions to the Green's function G(t). As we have already stated, we will express our results in $d = 4\quad 2$ dimensions. Firstly, we consider the vertices to be contracted at order $O(g^2)$:



Upon contraction three diagrams are formed:

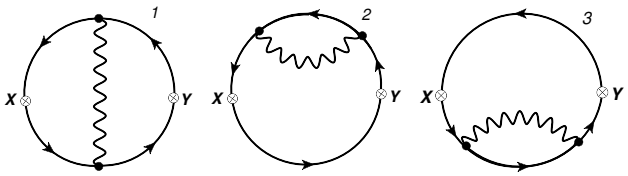
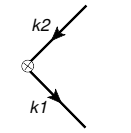
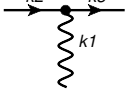


Figure 4.1.1: One loop Feynman diagrams for the composite operator Green's function.

Of course, these diagrams consist of two loops, but the name `one loop' indicates that they are of order $O(g^2)$. All three diagrams have a combinatorial factor of 1. As mentioned before, disconnected diagrams will not contribute. By Fourier transformation of the elds, the vertices read as follows:



$$= \int \frac{d^d k_1}{(2\ )^d} \int \frac{d^d k_2}{(2\ )^d}\, e^{\ ik_1 x} e^{ik_2 x}\ {}^{f\,a}_{i}(k_1)\ X_{ij}\ {}^{\sim f' a}_{j}(k_2);$$

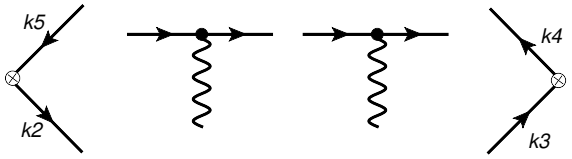


$$= \sum_f \int \frac{d^d k_1}{(2\ )^d} \int \frac{d^d k_2}{(2\ )^d} \int \frac{d^d k_3}{(2\ )^d} (2\ )^d\ (k_1 \quad k_2 + k_3)(ig)$$

$$ {}^{f\,a}_{i}(k_2)\ \ {}_{ij}\ {}^{\sim f' b}_{j}(k_3)\, A^c(k_1)\ T^c_{ab}; \tag{4.29}$$

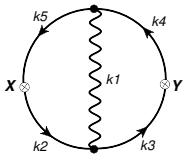
where $f;\ f'$ are avor indices and $a;\ b;\ c$ are color indices. We then name the

momenta as follows:



Then, upon contraction, the vertices form the following three diagrams (from now on all integrations and all factors of $(2\ )^{\ d}$ are left implicit):

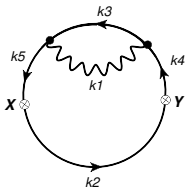
Diagram 1



$$= \frac{g_0^2}{2}(N^2 \quad 1)\mathrm{Tr}\ \ D_F(k_4)Y\,D_F(k_3) \quad D_F(k_2)X\,D_F(k_5)$$

$$D_G(k_1) \quad (k_1 + k_2 \quad k_3)\ (k_4 \quad k_1 \quad k_5)e^{iy(k_3 \quad k_4)}e^{ix(k_5 \quad k_2)} \tag{4.30}$$

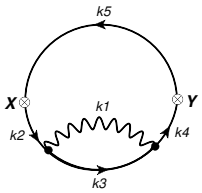
Diagram 2



$$= \frac{g_0^2}{2}(N^2 \quad 1)\mathrm{Tr}\ \ D_F(k_2)X\,D_F(k_5) \quad D_F(k_3) \quad D_F(k_4)Y$$

$$D_G(k_1) \quad (k_1 + k_4 \quad k_3)\ (k_3 \quad k_1 \quad k_5)e^{iy(k_2 \quad k_4)}e^{ix(k_5 \quad k_2)} \tag{4.31}$$

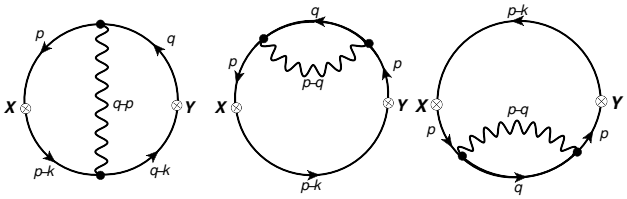
Diagram 3



$$= \frac{g_0^2}{2}(N^2 \quad 1)\mathrm{Tr}\ \ D_F(k_5)Y\,D_F(k_4) \quad D_F(k_3) \quad D_F(k_2)X$$

$$D_G(k_1) \quad (k_1 + k_2 \quad k_3)\ (k_3 \quad k_1 \quad k_4)e^{iy(k_4 \quad k_5)}e^{ix(k_5 \quad k_2)} \tag{4.32}$$

After we apply the replacements indicated by the delta functions and we rename the remaining momenta appropriately we have:



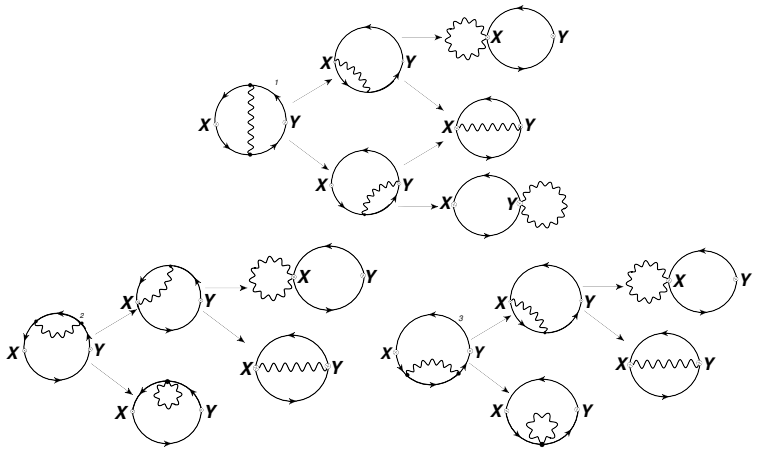
Our next step is to verify that the one-loop contribution is indeed gauge invariant, that is, that the -terms cancel out when we sum the three diagrams. To do so, we make use of the following identity:

$$D_F(p_a + p_b)\not{p}_a D_F(p_b) = \quad iD_F(p_b) + iD_F(p_a + p_b): \tag{4.33}$$

One can easily verify this identity by writing the propagators explicitly:

$$\begin{aligned} \frac{1}{i(\not{p}_a + \not{p}_b)}(\not{p}_a)\frac{1}{i\not{p}_b} &= \frac{1}{i(\not{p}_a + \not{p}_b)}\Big[(\not{p}_a + \not{p}_b) \quad (\not{p}_b)\Big]\frac{1}{i\not{p}_b} \\ &= \frac{1}{\not{p}_b} + \frac{1}{(\not{p}_a + \not{p}_b)} \end{aligned} \tag{4.34}$$

Double application of the above identity can be diagramatically depicted:



Tadpole diagrams vanish in dimensional regularization. After implementation of the identity, we are left with four non-vanishing diagrams, all of them with

the same propagator structure: diagram 1 has two contributions, while diagram 2 and diagram 3 have one each. One can prove that the result is gauge-independent, and therefore we can proceed our calculations using any gauge. In our case this will be the Feynman gauge.

The complete expressions of our diagrams in this gauge are:

Diagram 1:

$$\int \frac{d^dk}{(2\pi)^d} \int \frac{d^dp}{(2\pi)^d} \int \frac{d^dq}{(2\pi)^d} \frac{g_0^2}{2}(N^2-1)e^{ik(x-y)} \frac{1}{(p-q)^2p^2(p-k)^2q^2(q-k)^2}\mathrm{Tr}(X\not{p}\not{q}Y(\not{q}-\not{k})(\not{p}-\not{k})) \tag{4.35}$$

Diagram 2:

$$\int \frac{d^dk}{(2\pi)^d} \int \frac{d^dp}{(2\pi)^d} \int \frac{d^dq}{(2\pi)^d} \frac{g_0^2}{2}(N^2-1)e^{ik(x-y)} \frac{1}{(p-q)^2(p^2)^2(p-k)^2q^2}\mathrm{Tr}(X\not{p}\not{q}\not{p}Y(\not{p}-\not{k})) \tag{4.36}$$

Diagram 3:

$$\int \frac{d^dk}{(2\pi)^d} \int \frac{d^dp}{(2\pi)^d} \int \frac{d^dq}{(2\pi)^d} \frac{g_0^2}{2}(N^2-1)e^{ik(x-y)} \frac{1}{(p-q)^2(p^2)^2(p-k)^2q^2}\mathrm{Tr}(X(\not{p}-\not{k})Y\not{p}\not{q}\not{p}) \tag{4.37}$$

Our main task will be to compute the bare one-loop Green's functions in d-dimensions, for arbitrary values of the vectors x and y. Once this is done, we can extract the $\overline{\mathrm{MS}}$-renormalized Green's functions, and express them in 4 dimensions. At that point, in order to obtain the conversion factor between $\overline{\mathrm{MS}}$ and GIRS, we will integrate over the spatial components of x.

The integrals over p and q in diagrams 2 and 3 may be computed by repeated application of the formulae in (3.25) and (3.26). Diagram 1 has a more complicated \diamond" topology. In order to evaluate it, we may use the formula for the scalar diamond:

$$I(a;b;l;m;n) = \int \frac{d^dp}{(2\pi)^d} \int \frac{d^dq}{(2\pi)^d} \frac{1}{p^{2a}q^{2m}(p-q)^{2l}(p-k)^{2b}(q-k)^{2n}} \tag{4.38}$$

In our case we must evaluate $I(1;1;1;1;1)$. It can be shown that:

$$I(1;1;1;1;1) = \frac{1}{}\frac{(k^2)^{\ 1\ 2}}{(4\ )^4}(4\ )^2\ \ (\ )\frac{(1\ \ \ )^2}{(2\ \ 2\ )}\ \frac{(1+\ )\ (\ \ )\ (1\ \ \ )}{(1\ \ 2\ )}$$
$$\frac{(1+2\ )\ (\ \ )\ (1\ \ 2\ )}{(1+\ )\ (1\ \ 3\ )} \tag{4.39}$$
$$= \frac{6\ (3)}{(4\ )^4k^2} + O(\ )$$

Let us rstly label part of the integral in diagram 1 as I , where its indices denote the direction of the components involved in its numerator. Given that the integral in question is not scalar, we must rst reduce it to scalar expressions. To do so, we make use of the fact that, by rotational invariance and by symmetry with respect to indices ; and ; , the result for the integral necessarily will have the form :

$$I\ \ \ (k)\ \ \int\frac{d^dp}{(2\ )^d}\int\frac{d^dq}{(2\ )^d}\frac{p\ \ q\ \ (q\ \ \ k)\ \ (p\ \ \ k)}{p^2\ q^2\ (p\ \ \ q)^2\ (p\ \ \ k)^2\ (q\ \ \ k)^2}$$
$$= k\ \ k\ \ k\ \ k\ \ \frac{1}{(k^2)^2}f_1(k) + \ \ \ \ \ f_2(k) + \ \ \ \ \ f_3(k) + \ \ \ \ \ f_4(k) +$$
$$(\ \ \ k\ k\ \ +\ \ \ \ k\ \ k\ )\frac{1}{k^2}f_5(k) + (\ \ \ k\ \ k\ \ +\ \ \ \ k\ \ k\ )\frac{1}{k^2}f_6(k) +$$
$$(\ \ \ k\ \ k\ \ +\ \ \ \ k\ \ k\ )\frac{1}{k^2}f_7(k) \tag{4.40}$$

where $f_1; f_2; :::; f_7$, by dimensional analysis must equal $(k^2)^{1\ 2}$ multiplied by a constant $c_1; c_2; :::; c_7$. These constants can be calculated at a later time by taking seven appropriate inner products of I with components of k or Kronecker deltas, and solving for those constants using these seven equations.

Once the integrals over p and q are performed, there remains an integral over k. By dimensional arguments this integral will necessarily have one of the following forms:

$$\int d^dk e^{ik(x\ \ y)}(k^2)^{1\ \ 2}$$
$$\int d^dk e^{ik(x\ \ y)}(k^2)^{\ 2}\ k\ \ k$$
$$\int d^dk e^{ik(x\ \ y)}(k^2)^{\ 1\ 2}\ k\ \ k\ \ k\ \ k$$

The rst of these integrals can be performed using a Schwinger representation of $k^{\ b}$:

$$k^{\ b} = \int_0^{\ 1} d\ \frac{^{\frac{b}{2}\ 1} e^{\ k^2}}{(\frac{b}{2})} \tag{4.41}$$

Performing the integral over k and then over :

$$\int d^d k e^{ik(x\ \ y)} k^{\ b} = \frac{^{2+}\ (x\ \ y)^{\ 4+b+2}\ (2\ \ \frac{b}{2}\ \ )}{2^b\ (\frac{b}{2})} \tag{4.42}$$

The second and third integrals may be computed by successive di erentiation of the rst one with respect to $(x\ \ y)$, $(x\ \ y)$, $(x\ \ y)$ and $(x\ \ y)$. Parameter b takes the following values for the rst,second and third integral respectively:

$$\begin{matrix} 2+4 \\ 4 \\ 2+4 \end{matrix}$$

Adding all the contributions together and determining the constants $c_i; i =$ 1 7 we are led to the bare Green's functions, containing simple poles $1=$ plus nite terms. From the bare Green's functions we obtain the multiplicative renormalization factors $Z^{DR;\overline{\text{MS}}}$ by demanding the left-hand side ($\overline{\text{MS}}$ renormalized) of the renormalization condition in eq. (4.43) equals the bare Green's functions multiplied by the renormalization factor squared. To avoid heavy notation we have omitted coordinate arguments, as well as Dirac indices on $hO_X O_X i$:

$$hO_X O_X i^{\overline{\text{MS}}} = (Z^{DR;\overline{\text{MS}}})^2 hO_X O_X i^B \tag{4.43}$$

From eq. (4.43), we conclude that $Z^{DR;\overline{\text{MS}}}$ equals:

$$Z_S^{DR;\overline{\text{MS}}} = 1\ \ \frac{g^2 C_F}{16^{\ 2}} \frac{3}{\ } \tag{4.44}$$

$$Z_P^{DR;\overline{\text{MS}}} = 1\ \ \frac{g^2 C_F}{16^{\ 2}} \frac{3}{\ } \tag{4.45}$$

$$Z_V^{DR;\overline{\text{MS}}} = 1 \tag{4.46}$$

$$Z_A^{DR;\overline{\text{MS}}} = 1 \tag{4.47}$$

$$Z_T^{DR;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16^{\ 2}} \frac{1}{\ } \tag{4.48}$$

The renormalization factor $Z^{DR;\overline{\text{MS}}}$ for the vector and axial vector operators is 1 as the divergent part of the three one-loop diagrams cancels out. As a consistency check, we note that the factor $Z_T^{DR;\overline{\text{MS}}}$ (eq. 4.48) is su cient to render the tensor Green's functions nite for all possible choices of the Lorentz indices.

The above results are in agreement with the well-known results in the literature [8],[9]. Thus the $\overline{\text{MS}}$- renormalized Green's functions may be expressed directly in 4 dimensions, as follows:

$$\begin{aligned}
\langle O_{\mathbb{1}}(x)O_{\mathbb{1}}(y)\rangle^{\overline{\text{MS}}} &= \frac{N}{\pi^4((x-y)^2)^3} \\
&\quad 1+\frac{g^2C_F}{16\pi^2}\left(-2+6\log(\mu^2(x-y)^2)+12\gamma_E-12\log(2)\right) \\
\langle O_{\pi^5}(x)O_{\pi^5}(y)\rangle^{\overline{\text{MS}}} &= \frac{N}{\pi^4((x-y)^2)^3} \\
&\quad 1+\frac{g^2C_F}{16\pi^2}\left(-18+6\log(\mu^2(x-y)^2)+12\gamma_E-12\log(2)\right) \\
\langle O_{\pi^1}(x)O_{\pi^2}(y)\rangle^{\overline{\text{MS}}} &= \frac{N}{\pi^4((x-y)^2)^4} \\
&\quad 2(x_{\pi^1}-y_{\pi^1})(x_{\pi^2}-y_{\pi^2})-\delta_{\pi^1\pi^2}(x-y)^2\left(-1+3\frac{g^2C_F}{16\pi^2}\right) \\
\langle O_{\pi^5\pi^1}(x)O_{\pi^5\pi^2}(y)\rangle^{\overline{\text{MS}}} &= \frac{N}{\pi^4((x-y)^2)^4} \\
&\quad 2(x_{\pi^1}-y_{\pi^1})(x_{\pi^2}-y_{\pi^2})-\delta_{\pi^1\pi^2}(x-y)^2\left(-1+11\frac{g^2C_F}{16\pi^2}\right) \\
\langle O_{\pi^1\pi^2}(x)O_{\pi^3\pi^4}(y)\rangle^{\overline{\text{MS}}} &= \frac{N}{\pi^4((x-y)^2)^4} \\
&\quad 2\,\delta_{\pi^1\pi^3}(x_{\pi^2}-y_{\pi^2})(x_{\pi^4}-y_{\pi^4})-2\,\delta_{\pi^1\pi^4}(x_{\pi^2}-y_{\pi^2})(x_{\pi^3}-y_{\pi^3}) \\
&\quad -2\,\delta_{\pi^2\pi^3}(x_{\pi^1}-y_{\pi^1})(x_{\pi^4}-y_{\pi^4})+2\,\delta_{\pi^2\pi^4}(x_{\pi^1}-y_{\pi^1})(x_{\pi^3}-y_{\pi^3}) \\
&\quad (x-y)^2(\delta_{\pi^1\pi^3}\delta_{\pi^2\pi^4}-\delta_{\pi^1\pi^4}\delta_{\pi^2\pi^3}) \\
&\quad 1+\frac{g^2C_F}{16\pi^2}\left(-6-2\log(\mu^2(x-y)^2)-4\gamma_E+4\log(2)\right)
\end{aligned} \tag{4.49}$$

where $C_F = (N_c^2-1)/(2N_c)$ is the quadratic Casimir operator in the fundamental representation.

Similarly, in order to find the renormalization factor in the *GIRS* scheme we use the condition:

$$\frac{Z^{DR;GIRS}}{Z^{DR;\overline{\text{MS}}}}\int d^3x\ \langle O_X(x)O_X(y)\rangle^{\overline{\text{MS}}} = \int d^3x\, G^{tree}(x-y) \tag{4.50}$$

(at a particular value for $t = y_0 - x_0$, of our choice)

As previously mentioned, the spatial integration separates further these cases into 8 possibilities (Scalar (S), Pseudoscalar (P), Vector temporal (Vt), Vector spatial (Vs), Axial temporal (At), Axial spatial (As), Tensor temporal(Tt), Tensor spatial (Ts)), which depend on whether operator indices are temporal or not, and they correspond to $X = f1;\ _5;\ _0;\ _i;\ _5\ _0;\ _5\ _i;\ _0\ _i;\ _i\ _j g$. Two of them vanish, corresponding to Vt and At ; given that the lattice is isotropic we are free to use for Vt and At the same renormalization factors as for Vs and As, respectively. Given that the conversion factor from $GIRS$ to $\overline{\text{MS}}$ is given by:

$$C^{GIRS;\overline{\text{MS}}} = \frac{Z^{DR;\overline{\text{MS}}}}{Z^{DR;GIRS}} \tag{4.51}$$

we nally obtain:

$$C_S^{GIRS;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16\ ^2}\ \ 6\log(t\ ) + 6\ _E\ \ \frac{1}{2} \tag{4.52}$$

$$C_P^{GIRS;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16\ ^2}\ \ \frac{15}{2} + 6\log(t\ ) + 6\ _E \tag{4.53}$$

$$C_{Vs}^{GIRS;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16\ ^2}\ \ \frac{3}{2} \tag{4.54}$$

$$C_{As}^{GIRS;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16\ ^2}\ \ \frac{11}{2} \tag{4.55}$$

$$C_{Ts}^{GIRS;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16\ ^2}\ \ 2\log(t\ )\ \ 2\ _E + \frac{25}{6} \tag{4.56}$$

$$C_{Tt}^{GIRS;\overline{\text{MS}}} = 1 + \frac{g^2 C_F}{16\ ^2}\ \ 2\log(t\ )\ \ 2\ _E + \frac{25}{6} \tag{4.57}$$

4.5 Conclusions and outlook

We have studied the renormalization of all dimension-3 fermion bilinears. In particular, we computed the conversion factors from the GIRS scheme to $\overline{\text{MS}}$. These conversion factors, being, regularization independent, can be computed in perturbation theory using dimensional regularization which, by virtue of its simplicity as compared to lattice perturbation theory, allows us to extend the calculation to higher loops. The GIRS renormalized Green's function may be computed directly in numerical simulations, without reference to perturbation theory. A rather strong consistency check of the proceduce is the veri cation that the $\overline{\text{MS}}$-renormalized Green's functions, obtained by multiplying nonperturbative GIRS-renormalized Green's functions with the corresponding conversion factor, will turn out to be independent of the intermediate scale t. We intend to apply this renormalization scheme to the study of operators related to the energy-momentum tensor of QCD. These operators are used in simulations

to extract physical quantities, such as the fraction of nucleon momentum carried by gluons. The main composite operator of interest in this case is: $O_1 = G^a G^a$ (δ) This operator is known to mix with another gauge-invariant operator: $O_2 = \frac{1}{4}$ $_f$(^{f} D ^{f} + ^{f} D ^{f} $^f D$ ^{f} $^f D$ ^{f}), but, according to general theorems on renormalization, O_1 can also mix, and indeed it does mix, with non-gauge-invariant operators as well. The latter (a total of 4 further operators, containing also ghost fields in their definition) will not appear in the evaluation of matrix elements of O_1 in physical states; however, they affect the evaluation of the mixing coefficients between O_1 and O_2. Use of the GIRS scheme is thus ideally suited in order to eliminate the presence of such operators right from the start: there are still mixing coefficients to be calculated in this scheme, however the mixing matrix is 2x2, rather than 6x6, and, due to the fact that it is manifestly gauge-invariant, may be calculated in whichever gauge appears more convenient.

Appendix A

Mathematical Formulae and Identities

A.1 Gaussian Integrals

In this section we brie y present some Gaussian integrals that are useful in the evaluation of Green's functions in the functional integral quantization of eld theories [6]. Some prototype integrals which are involved in Green's functions have the form:

$$\langle x_{k_1} \cdots x_{k_n} \rangle = N \int \prod_{i=1}^{n} dx_i \, x_{k_1} \cdots x_{k_n} \exp\left(-\sum_{i,j=1}^{n} \frac{1}{2} x_i A_{ij} x_j \right); \qquad \text{(A.1)}$$

where x is an n-component vector with real elements and the normalization constant N is de ned in a way that $\langle 1 \rangle = 1$.

Let us rst consider the general Gaussian integral

$$I(A;b) = \int \prod_{i=1}^{n} dx_i \exp\left(-\sum_{i,j=1}^{n} \frac{1}{2} x_i A_{ij} x_j + \sum_{i=1}^{n} b_i x_i \right); \qquad \text{(A.2)}$$

where A is a symmetric matrix with eigenvalues λ_i satisfying

$$\mathrm{Re}(\lambda_i) \geq 0; \qquad \lambda_i \neq 0. \qquad \text{(A.3)}$$

By changing variables, $x = y + A^{-1}b$, we obtain

$$I(A;b) = \int \prod_{i=1}^{n} dy_i \exp\left[-\sum_{i,j=1}^{n} \frac{1}{2}\left(y_i + b_h A_{hi}^{-1} \right) A_{ij} \left(y_j + A_{jk}^{-1} b_k \right) + \sum_{i=1}^{n} b_i \left(y_i + A_{jk}^{-1} b_k \right) \right]$$
$$= \exp\left[\sum_{i,j} \frac{1}{2} b_i A_{ij}^{-1} b_j \right] \int \prod_i dy_i \exp\left[\sum_{i,j} \frac{1}{2} y_i A_{ij} y_j \right] : \tag{A.4}$$

After diagonalizing A in order to calculate the last integral, we finally obtain

$$I(A;b) = (2\pi)^{n=2} (\det A)^{-1=2} \exp\left[\sum_{i,j} \frac{1}{2} b_i A_{ij}^{-1} b_j \right] : \tag{A.5}$$

Repeated differentiation of (A.5) with respect to b_i leads to Wick's theorem

$$\langle x_{k_1} \cdots x_{k_n} \rangle = \frac{\partial}{\partial b_{k_1}} \cdots \frac{\partial}{\partial b_{k_n}} \exp\left[\sum_{i,j} \frac{1}{2} b_i A_{ij}^{-1} b_j \right]_{b=0}$$
$$= \sum_{\substack{\text{all possible pairings} \\ \text{of } k_1, \ldots, k_n}} K^{-1}_{k_{p_1} k_{p_2}} \cdots K^{-1}_{k_{p_{n-1}} k_{p_n}} : \tag{A.6}$$

Consider now a generic complex Gaussian integral of the form

$$I(A;b;b) = \int \prod_{i=1}^{n} dz_i dz_i \exp\left[-\sum_{i,j=1}^{n} \frac{1}{2} z_i A_{ij} z_j + \sum_{i=1}^{n} (b_i z_i + b_i z_i) \right] ; \tag{A.7}$$

in which A is a complex matrix, and z, z are complex valued n-component variables which are defined as:

$$z_i = (x_i + y_i)/\sqrt{2}; \qquad z_i = (x_i - y_i)/\sqrt{2}: \tag{A.8}$$

Following the same steps as before, we obtain

$$I(A;b;b) = (2\pi)^{n} (\det A)^{-1} \exp\left[\sum_{i,j} \frac{1}{2} b_i A_{ij}^{-1} b_j \right] : \tag{A.9}$$

The corresponding Wick's theorem reads

$$\langle z_{i_1} z_{j_1} \cdots z_{i_n} z_{j_n} \rangle = \sum_{\substack{\text{all permutations} \\ P \text{ of } j_1, \ldots, j_n}} A^{-1}_{j_{P_1} i_1} \cdots A^{-1}_{j_{P_n} i_n} : \tag{A.10}$$

A.2 Grassmann Algebra

The Green's functions one encounters in the study of fermion fields are antisymmetric with respect to the exchange of two of their arguments. Therefore, if we want to evaluate such Green's functions, we have to introduce anticommuting classical quantities, i.e. Grassmann variables, and define differentiation and integration operations on those variables.

The Grassmann algebra is constructed from a set of generators θ_i and their anticommuting products [6]:

$$\theta_i \theta_j + \theta_j \theta_i = 0 \qquad \forall i,j. \tag{A.11}$$

Note that, as a consequence, all elements in a Grassmann algebra are first degree polynomials in each generator (since $\theta_i^2 = 0$) and the vector space for an algebra of n generators is of dimension 2^n. For example, an algebra containing two generators θ_i, θ_j is a vector space of dimension 4, spanned by the quantities $\{1, \theta_i, \theta_j, \theta_i\theta_j\}$.

Differentiation operation

In Grassmann algebra the differentiation operation is defined by

$$\frac{\partial}{\partial \theta_i}\theta_j = \delta_{ij}. \tag{A.12}$$

Along with this rule we have the following property for differentiation over a product of Grassman variables

$$\frac{\partial}{\partial \theta_i}(\theta_j X) = \delta_{ij} X - \theta_j \frac{\partial}{\partial \theta_i} X. \tag{A.13}$$

In addition, one can easily verify that the following identities hold true:

$$\begin{aligned} \frac{\partial}{\partial \theta_i}\frac{\partial}{\partial \theta_j} + \frac{\partial}{\partial \theta_j}\frac{\partial}{\partial \theta_i} &= 0, \\ \theta_i \frac{\partial}{\partial \theta_j} + \frac{\partial}{\partial \theta_j}\theta_i &= \delta_{ij}. \end{aligned} \tag{A.14}$$

Integration operation

It can be shown that in a Grassmann algebra, the integration operation coincides with the differentiation. However, for convenience, we shall still denote it by an integral symbol:

$$\int d\theta_i A \equiv \frac{\partial}{\partial \theta_i} A. \tag{A.15}$$

Using the definition given in Eq. (A.15) we find that for a linear change of variables $\theta = a\theta' + b$, we have

$$\int d\,\theta\, f(\theta) = a^{-1} \int d\,\theta'\, f(a\,\theta' + b). \tag{A.16}$$

Notice that for Grassmann integrals the jacobian is a^{-1}, while in the case of commuting variables we would obtain a. In general, one can easily demonstrate that a change of variables

$$\theta_i = \theta_i(\theta');$$

for which the matrix $\partial\theta_i/\partial\theta'_j$ has an invertible part of degree zero, leads to

$$\int d\,\theta_1 \cdots d\,\theta_n = \int d\,\theta'_1 \cdots d\,\theta'_n\, J(\theta'); \tag{A.17}$$

with

$$J^{-1} = \det \frac{\partial\theta_i}{\partial\theta'_j}. \tag{A.18}$$

We can verify Eq. (A.17) with the following example:

$$1 = \int d\,\theta_1 \cdots d\,\theta_n\, \theta_n \cdots \theta_1. \tag{A.19}$$

With the linear change of variables

$$\theta_i = \sum_j a_{ij}\,\theta'_j; \tag{A.20}$$

the result depends on the identity

$$\theta_n \cdots \theta_1 = (\det a)\,\theta'_n \cdots \theta'_1. \tag{A.21}$$

and one concludes that $J = (\det a)^{-1}$.

Consider now the Gaussian integral which contains two sets of conjugate Grassmann variables, θ_i and η_i; $i = 1 \cdots n$:

$$I(a) = \int d\,\eta_1 d\,\theta_1 \cdots d\,\eta_n d\,\theta_n \exp \sum_{i,j} \eta_i a_{ij}\,\theta_j. \tag{A.22}$$

The result is simply the determinant of a:

$$I(a) = \det a. \tag{A.23}$$

Wick's theorem for fermions is given by

$$\langle\, \theta_{i_1}\,\eta_{j_1} \cdots \theta_{i_n}\,\eta_{j_n} \,\rangle = \sum_{\substack{\text{all permutations} \\ P \text{ of } j_1,\ldots,j_n}} \varepsilon(P)\, a^{-1}_{j_{P_1} i_1} \cdots a^{-1}_{j_{P_n} i_n}. \tag{A.24}$$

where $\varepsilon(P)$ is the sign of the permutation P. The result is similar to the expression (A.10) obtained for integrals over complex variables, except for the sign.

A.3 Dirac Matrices

One common convention for the Dirac matrices in four-dimensional Euclidean space is

$$
\begin{aligned}
{}_1 &= \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} & {}_2 &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \\
{}_3 &= \begin{pmatrix} 0 & 0 & i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix} & {}_4 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix};
\end{aligned}
\tag{A.25}
$$

with ${}_4$ being the time-like matrix. The de ning property of the Dirac matrices is the anticommutation relation

$$f \quad ; \quad g = 2 \quad : \tag{A.26}$$

It is convenient to de ne a fth Dirac matrix that is the product of the other four:

$$
{}_5 \quad {}_1 {}_2 {}_3 {}_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.
\tag{A.27}
$$

We can then write some useful identities:

(1) $f \; {}_5; \quad g = 0;$

(2) $\frac{2}{5} = 1;$

(3) $(\quad) = (\; {}_5) = 0;$

(4) $(\qquad) = 4 \quad ;$

(5) $(\; {}_1 \; : : : \; {}_n) = 0$ if n =odd,

(6) $(\; {}_5 \; {}_1 \; : : : \; {}_n) = 0$ if n = 2 or odd,

(7) $(\qquad\qquad) = 4 \qquad\quad 4 \qquad + 4 \qquad :$

(A.28)

A.4 SU(N) Algebra

The special unitary group, denoted $SU(N)$, is the Lie group that consists of $N \quad N$ unitary matrices with determinant 1.

$$\begin{aligned} U^{y}U &= 1; \\ \det U &= 1: \end{aligned} \tag{A.29}$$

In particular, the $SU(3)$ group is the symmetry group of Quantum Chromodynamics, with $N = 3$ being the number of colors.

One very useful concept is the algebra of $SU(N)$, denoted $su(N)$. General elements U of the the group correspond to an element T of the algebra through

$$U = e^{iT}: \tag{A.30}$$

The algebra is a linear vector space spanned by $N^2 - 1$ independent elements T^a, $a = 1;:::;N^2 - 1$, called *generators*. From the properties in Eq. (A.29) one can easily prove that the generators have to be hermitian matrices, with zero trace. We impose the following normalization condition:

$$Tr(T^a T^b) = \frac{1}{2}\ ^{ab}: \tag{A.31}$$

A very useful identity involving the generators in the fundamental representation is

$$\sum_{a=1}^{N^2-1} (T^a)_{ij} T^a_{kl} = \frac{1}{2}\ _{il}\ _{jk} - \frac{1}{2N}\ _{ij}\ _{kl}: \tag{A.32}$$

Many important identities can be obtained from (A.32), such as (a sum over repeated indices is implied):

$$\begin{aligned} Tr(XT^a)Tr(YT^a) &= \frac{1}{2}Tr(XY) - \frac{1}{2N}Tr(X)Tr(Y); \\ XT^aYT^aZ &= \frac{1}{2}Tr(Y)XZ - \frac{1}{2N}XYZ: \end{aligned} \tag{A.33}$$

Appendix B

Project Calculations

B.1 Tree Level

```
In[1]:= (* In this version of the tree-level diagram, we simply include an additional scale
factor,
           (m0)^(4e), as explained below. We do not integrate over x. The result will be
useful in
           constructing the MSbar-renormalized Green's functions in 4 dimensions.
*)
        (* Note that, in d-dimensions, the dimensionality of Green's functions in
coordinate space,
           unlike the typical behaviour in momentum space, depends on d.
           Typical momentum space Green's functions are of the type:
           < ψ_1(q_1)...ψ_m(q_m) A_1(p_1)...A_n(q_n) Operator[ψ_1(x_)...ψ_m(x)
A_1(x)...A_n(x)]>,
           and, thus, their dimensionality is: m*(-d-1)/2 + n*(-d-2)/2 + m*(d-1)/2 +
n*(d-2)/2 = -m -2n,
           (or 0, in the amputated case), irrespective of d.
           [Equivalently, if an integral over x is performed, the result is: (2π)^d
δ^d(q_1+...+p_m) * G(q,p),
            and again the dimensionality of G(q,p) is -m -2n (or 0, in the amputated case).]
           Thus, typically, momentum-space Green's functions at tree level are independent
of d.
           ----------------------------------------------------
           In coordinate space, a Green's function such as, e.g., <ψ(x)Γψ(x) ψ(y)Γ'ψ(y)>,
has dimensionality: 2d-2.
           A consequence of this is that, already at tree level, Green's functions in d-dim
differ from those in 4-dim.
           We may consider instead: <ψ(x)Γψ(x) ψ(y)Γ'ψ(y)> * (μ_0)^(4ε), which has fixed
dimension 6.
           Here μ_0 is some reference scale, not necessarily related to the one appearing
in:
           g_Lagrangian = g_dimensionless * μ^ε                    *)

In[2]:= (* Thus, the tree level Green's function, including the combinatorial factor of
-1, is:
        m0^(4e) Nc int^dp/(2pi)^d int^dk/(2pi)^d exp(i (-k+p)(x-y)) 1/p^2 1/k^2 tr(pslash
Y kslash X)
        Once we compute this, and renormalize it in MSbar, we will return to 4 dimensions,
        and then we will evaluate the integral of this renormalized Green's function over
d^3 x.
     To this end, we will first evaluate: int^dk/(2pi)^d exp(-i k x) 1/k^2 k_mu
     It is convenient to express 1/k^2 in terms of a Schwinger parameter:    *)

In[3]:= Integrate[Exp[-l k^2], {l,0,Infinity}, Assumptions -> k>0]

         -2
Out[3]= k

In[4]:= Integrate[Exp[-I k x - l k^2], {k, -Infinity, Infinity}]
```

```
                                    Sqrt[Pi]
Out[4]= ConditionalExpression[-----------------, Re[l] > 0]
                                      2
                                     x /(4 l)
                                    E         Sqrt[l]

In[5]:= %[[1]]^d // PowerExpand

                  d/2
                Pi
Out[5]= -------------------
                2
             (d x )/(4 l)   d/2
           E              l

In[6]:= % /. d x^2 -> x^2

                d/2
              Pi
Out[6]= ---------------
            2
           x /(4 l)   d/2
          E          l

In[7]:= Integrate[%, {l,0,Infinity}, Assumptions -> {x^2 >0}]

                                  -2 + d    d/2       2 - d             d
Out[7]= ConditionalExpression[2         Pi    Abs[x]      Gamma[-1 + -],
                                                                      2

>    Re[d] > 2]

In[8]:= %[[1]] /. d -> 4-2e /. Abs[x] -> x // ExpandAll

          2 - 2 e   2 - e  -2 + 2 e
Out[8]= 2         Pi      x         Gamma[1 - e]

In[9]:= % /. x^a_ :> xsq^Expand[a/2]

          2 - 2 e   2 - e    -1 + e
Out[9]= 2         Pi      xsq       Gamma[1 - e]

In[10]:= derivrules = {deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a,
deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b],
deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c],
deriv[a_?((FreeQ[#,xsq] && FreeQ[#,x])&),b_] ->0,
deriv[xsq, b_] :> 2 s2[2x,b],
deriv[s2[2x,a_],b_] :> delm[a,b]};
```

```
In[11]:= (* The expression: int^dk/(2pi)^d exp(-i k x) 1/k^2 k_mu   is thus equal to: *)

In[12]:= 1/(2 Pi)^(4-2e) im deriv[%9,mu];

In[13]:= % //. derivrules

                          -2 + e    -2 + e
         (-1 + e) im Pi       xsq       Gamma[1 - e] s2[2 x, mu]
Out[13]= ------------------------------------------------------
                                  2

In[14]:= (* Substituting this result into the tree-level expression, we obtain (below x
stands for x-y, and xsq = (x-y)^2: *)

In[15]:= m0^(4e) Nc (%13 /. mu -> rho[1]) (- %13 /. mu -> rho[2]) gtrace[rho[2],Y,rho[1],X]
/. im^2 -> -1 // FullSimplify

            4 e      -4 + 2 e    -4 + 2 e             2
Out[15]= (m0    Nc Pi         xsq         Gamma[2 - e]

>      gtrace[rho[2], Y, rho[1], X] s2[2 x, rho[1]] s2[2 x, rho[2]]) / 4

In[16]:= {{{},{}},
          {{},{gamma5}},
        {{},{nu[1]}},
        {{},{gamma5,nu[1]}},
        {{},{nu[1],nu[2]}},
        {{gamma5},{gamma5}},
        {{gamma5},{nu[1]}},
        {{gamma5},{gamma5,nu[1]}},
        {{gamma5},{nu[1],nu[2]}},
        {{nu[1]},{nu[1]}},
        {{nu[1]},{nu[2]}},
        {{nu[1]},{gamma5,nu[1]}},
        {{nu[1]},{gamma5,nu[2]}},
        {{nu[1]},{nu[1],nu[2]}},
        {{nu[1]},{nu[2],nu[3]}},
        {{gamma5,nu[1]},{gamma5,nu[1]}},
        {{gamma5,nu[1]},{gamma5,nu[2]}},
        {{gamma5,nu[1]},{nu[1],nu[2]}},
        {{gamma5,nu[1]},{nu[2],nu[3]}},
        {{nu[1],nu[2]},{nu[1],nu[2]}},
        {{nu[1],nu[2]},{nu[1],nu[3]}},
        {{nu[1],nu[2]},{nu[3],nu[4]}}};

In[17]:= Length[%]

Out[17]= 22
```

```
In[18]:= Sum[(tag @@ %%[[i]] /. nu -> NU) (%%% /. X -> Sequence @@ %%[[i,1]] /. Y -> Sequence
@@ %%[[i,2]]), {i, Length[%%]}] // Factor

            4 e      -4 + 2 e    -4 + 2 e             2
Out[18]= (m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, rho[1]]

>      s2[2 x, rho[2]] (gtrace[rho[2], rho[1]] tag[{}, {}] +

>        gtrace[rho[2], gamma5, rho[1]] tag[{}, {gamma5}] +

>        gtrace[rho[2], nu[1], rho[1]] tag[{}, {NU[1]}] +

>        gtrace[rho[2], gamma5, nu[1], rho[1]] tag[{}, {gamma5, NU[1]}] +

>        gtrace[rho[2], nu[1], nu[2], rho[1]] tag[{}, {NU[1], NU[2]}] +

>        gtrace[rho[2], gamma5, rho[1], gamma5] tag[{gamma5}, {gamma5}] +

>        gtrace[rho[2], nu[1], rho[1], gamma5] tag[{gamma5}, {NU[1]}] +

>        gtrace[rho[2], gamma5, nu[1], rho[1], gamma5]

>         tag[{gamma5}, {gamma5, NU[1]}] +

>        gtrace[rho[2], nu[1], nu[2], rho[1], gamma5]

>         tag[{gamma5}, {NU[1], NU[2]}] +

>        gtrace[rho[2], nu[1], rho[1], nu[1]] tag[{NU[1]}, {NU[1]}] +

>        gtrace[rho[2], nu[2], rho[1], nu[1]] tag[{NU[1]}, {NU[2]}] +

>        gtrace[rho[2], gamma5, nu[1], rho[1], nu[1]]

>         tag[{NU[1]}, {gamma5, NU[1]}] +

>        gtrace[rho[2], gamma5, nu[2], rho[1], nu[1]]

>         tag[{NU[1]}, {gamma5, NU[2]}] +

>        gtrace[rho[2], nu[1], nu[2], rho[1], nu[1]]

>         tag[{NU[1]}, {NU[1], NU[2]}] +

>        gtrace[rho[2], nu[2], nu[3], rho[1], nu[1]]

>         tag[{NU[1]}, {NU[2], NU[3]}] +

>        gtrace[rho[2], gamma5, nu[1], rho[1], gamma5, nu[1]]
```

```
>          tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +

>         gtrace[rho[2], gamma5, nu[2], rho[1], gamma5, nu[1]]

>          tag[{gamma5, NU[1]}, {gamma5, NU[2]}] +

>         gtrace[rho[2], nu[1], nu[2], rho[1], gamma5, nu[1]]

>          tag[{gamma5, NU[1]}, {NU[1], NU[2]}] +

>         gtrace[rho[2], nu[2], nu[3], rho[1], gamma5, nu[1]]

>          tag[{gamma5, NU[1]}, {NU[2], NU[3]}] +

>         gtrace[rho[2], nu[1], nu[2], rho[1], nu[1], nu[2]]

>          tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

>         gtrace[rho[2], nu[1], nu[3], rho[1], nu[1], nu[2]]

>          tag[{NU[1], NU[2]}, {NU[1], NU[3]}] +

>         gtrace[rho[2], nu[3], nu[4], rho[1], nu[1], nu[2]]

>          tag[{NU[1], NU[2]}, {NU[3], NU[4]}]) / 4

In[19]:= gtracerulesD :=
{gtrace[]:> 4,
gtrace[a__?(FreeQ[#,gamma5]&),gamma5,b___] :> gtrace[gamma5,b,a],
gtrace[a___,gamma5,gamma5,b___] :> gtrace[a,b],
gtrace[gamma5, nu[_]] :> 0,
gtrace[gamma5, a___, gamma5, nu[i_]] :> - gtrace[a,nu[i]],
gtrace[a___,gamma5, nu[i_], gamma5, b___] :> - gtrace[a,nu[i],b],
gtrace[a___,gamma5, nu[i_], nu[j_], gamma5, b___] :>  gtrace[a,nu[i],nu[j],b],
gtrace[a___,nu[i_],gamma5,nu[i_],b___] :> - gtrace[a,gamma5,b],
gtrace[a___,rho[i_],rho[j_],b___] s2[2x,rho[i_]] s2[2x,rho[j_]] :> gtrace[a,b] xsq,
gtrace[a___] :> 0 /; (FreeQ[{a},gamma5] && FreeQ[{a},X] && FreeQ[{a},Y] &&
OddQ[Length[{a}]]),
gtrace[c_?((FreeQ[#,X] && FreeQ[#,Y])&)] :> 0,
gtrace[a___,b_?(FreeQ[#,gamma5]&),b_,d___]:> gtrace[a,d] delm[b,b],
gtrace[a___, rho[i_], gamma5, rho[i_], c___] :> factor[-4 + 2 e - 4 e hv] gtrace[a, gamma5,
c],
gtrace[a___, rho[i_], gamma5, nu[j_], rho[i_], c___] :>  factor[2 - 2 e + 4 e hv] gtrace[a,
gamma5, nu[j], c],
gtrace[a___, rho[i_], nu[j_], gamma5, rho[i_], c___] :> - factor[2 - 2 e + 4 e hv] gtrace[a,
gamma5, nu[j], c],
gtrace[a___, rho[i_], gamma5, rho[j_], gamma5, rho[i_], c___] :> factor[-2 + 2 e] gtrace[a,
gamma5, rho[j], gamma5, c],
```

```
gtrace[gamma5,a___]:>0 /; (FreeQ[{a},gamma5] && FreeQ[{a},X] && FreeQ[{a},Y] &&
(Length[{a}]==2)),
gtrace[gamma5,a___,gamma5,b___] :> 0 /; (FreeQ[{a,b},gamma5] && FreeQ[{a,b},X] &&
FreeQ[{a,b},Y] && OddQ[Length[{a,b}]]),
gtrace[gamma5,a___,gamma5] :> gtrace[a],
gtrace[a_,b_]:> 4 delm[a,b] /; (FreeQ[{a,b},gamma5] && FreeQ[{a,b},X] && FreeQ[{a,b},Y])}

In[20]:= << ~/latt_pert/math/inputF.m
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[21]:= Expand[%%%] //. gtracerulesD;

In[22]:= % // simplifydelm // reducerho;

In[23]:= lb[%]

Out[23]= {16, 31504}

In[24]:= Select[Variables[%%], (!FreeQ[#, gtrace])&]

Out[24]= {gtrace[gamma5, rho , gamma5, rho ],
                          1            2

>    gtrace[gamma5, rho , nu , rho ], gtrace[rho , nu , nu , rho ],
                    2    1     1            2    1    2     1

>    gtrace[rho , nu , rho , nu ], gtrace[rho , nu , rho , nu ],
```

```
              2    1     1    1             2    2     1    1

>    gtrace[gamma5, nu , rho , nu , rho ],
                    1     1    1     2

>    gtrace[gamma5, nu , rho , nu , rho ],
                    2     1    1     2

>    gtrace[gamma5, rho , nu , nu , rho ],
                     2    1    2     1

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ],
                    1     1           1     2

>    gtrace[gamma5, nu , rho , nu , nu , rho ],
                    1     2    1    2     1

>    gtrace[gamma5, nu , rho , nu , nu , rho ],
                    1     2    2    3     1

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ],
                    2     1           1     2

>    gtrace[rho , nu , nu , rho , nu , nu ],
              2    1    2     1    1    2

>    gtrace[rho , nu , nu , rho , nu , nu ],
              2    1    3     1    1    2

>    gtrace[rho , nu , nu , rho , nu , nu ]}
              2    3    4     1    1    2

In[25]:= %%% //. gtrace[gamma5,a___,nu[i_],b___,gamma5,c___,d_,nu[i_],e___] :> -
gtrace[gamma5,a,nu[i],b,gamma5,c,nu[i],d,e] + 2 delm[d, nu[i]]
gtrace[gamma5,a,nu[i],b,gamma5,c,e];

In[26]:= % // Expand // simplifydelm // reducerho;

In[27]:= % //. gtrace[gamma5,a___,nu[i_],d_,b___,gamma5,nu[i_],e___] :> -
gtrace[gamma5,a,d,nu[i],b,gamma5,nu[i],e] + 2 delm[d, nu[i]]
gtrace[gamma5,a,b,gamma5,nu[i],e];

In[28]:= % // Expand // simplifydelm // reducerho;

In[29]:= % //. gtracerulesD;

In[30]:= % // simplifydelm // reducerho;

In[31]:= lb[%]
```

```
Out[31]= {17, 32832}

In[32]:= Select[Variables[%%], (!FreeQ[#, gtrace[a___, gamma5, b___, gamma5, c___]])&]

Out[32]= {gtrace[gamma5, rho , gamma5, rho ],
                          1            2

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ]}
                    2     1           1     2

In[33]:= Select[Variables[%%%], (!FreeQ[#, delm])&]

Out[33]= {}

In[34]:= Select[Variables[%%%%], (!FreeQ[#, gtrace])&]

Out[34]= {gtrace[gamma5, rho , gamma5, rho ],
                          1            2

>    gtrace[gamma5, rho , nu , rho ], gtrace[rho , nu , nu , rho ],
                     2    1     1            2    1    2     1

>    gtrace[rho , nu , rho , nu ], gtrace[rho , nu , rho , nu ],
               2    1     1    1            2    2     1    1

>    gtrace[gamma5, nu , rho , nu , rho ],
                    1     1    1     2

>    gtrace[gamma5, nu , rho , nu , rho ],
                    2     1    1     2

>    gtrace[gamma5, rho , nu , nu , rho ],
                     2    1    2     1

>    gtrace[gamma5, nu , rho , nu , nu , rho ],
                    1     2    1    2     1

>    gtrace[gamma5, nu , rho , nu , nu , rho ],
                    1     2    2    3     1

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ],
                    2     1           1     2

>    gtrace[rho , nu , nu , rho , nu , nu ],
               2    1    2     1    1    2

>    gtrace[rho , nu , nu , rho , nu , nu ],
               2    1    3     1    1    2
```

```
>    gtrace[rho , nu , nu , rho , nu , nu ]}
               2    3    4     1    1    2

In[35]:= %%%%% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);

In[36]:= % // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[37]:=
In[38]:=
In[39]:=
Out[39]= {23, 42552}

In[40]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[41]:=
In[42]:=
In[43]:=
In[44]:=
Out[44]= {28, 49920}

In[45]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[46]:=
In[47]:=
In[48]:=
In[49]:=
Out[49]= {36, 63800}

In[50]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
```

```
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[51]:=
In[52]:=
In[53]:=
In[54]:=
Out[54]= {35, 61456}

In[55]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[56]:=
In[57]:=
In[58]:=
In[59]:=
Out[59]= {31, 54080}

In[60]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[61]:=
In[62]:=
In[63]:=
In[64]:=
Out[64]= {29, 50448}

In[65]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[66]:=
In[67]:=
```

```
In[68]:=
In[69]:=
Out[69]= {29, 50448}

In[70]:= Variables[%%]

            e                e
Out[70]= {m0 , Nc, xsq, xsq , delm[nu , nu ], delm[nu , nu ], delm[nu , nu ],
                                       1    1          1    2          2    2

>    delm[nu , nu ], delm[nu , nu , nu ], Gamma[2 - e], gtrace[gamma5, rho ],
           2    3          1    2    3                                    1

>    gtrace[gamma5, nu , nu , nu ], gtrace[gamma5, nu , nu , rho ],
                    1    2    3                    1    2     1

>    gtrace[gamma5, nu , nu , rho ], gtrace[gamma5, rho , gamma5, rho ],
                    1    3     1                     1            2

>    gtrace[nu , nu , nu , nu ], gtrace[nu , nu , nu , rho ],
             1    2    3    4           1    2    3     1

>    gtrace[nu , nu , nu , rho ],
             1    2    4     1

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ], s2[2 x, nu ],
                    2     1           1     2            1

>    s2[2 x, nu ], s2[2 x, nu ], s2[2 x, nu ], s2[2 x, rho ], s2[2 x, rho ],
              2             3             4             1             2

>    tag[{}, {}], tag[{}, {NU[1], NU[2]}], tag[{gamma5}, {gamma5}],

>    tag[{gamma5}, {NU[1]}], tag[{NU[1]}, {NU[1]}], tag[{NU[1]}, {NU[2]}],

>    tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    tag[{gamma5, NU[1]}, {gamma5, NU[2]}],

>    tag[{gamma5, NU[1]}, {NU[1], NU[2]}],

>    tag[{gamma5, NU[1]}, {NU[2], NU[3]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[3]}], tag[{NU[1], NU[2]}, {NU[3], NU[4]}]}

In[71]:= %%% /. delm[a_,a_] :> 1;
```

```
In[72]:= % /. delm[a_,b__] :> 0 /. gtrace[gamma5,nu[i_],nu[j_],nu[k_]] :> 0 /.
gtrace[gamma5,nu[i_],nu[j_],rho[k_]] :> 0 /. gtrace[nu[i_],nu[j_],nu[k_],nu[l_]] :> 0 /.
gtrace[nu[i_],nu[j_],nu[k_],rho[l_]] :> 0;

In[73]:= % // reducerho;

In[74]:= lb[%]

Out[74]= {14, 22192}

In[75]:= Variables[%%]

            e                e
Out[75]= {m0 , Nc, xsq, xsq , Gamma[2 - e], gtrace[gamma5, rho ],
                                                                1

>    gtrace[gamma5, rho , gamma5, rho ],
                        1            2

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ], s2[2 x, nu ],
                       2     1           1     2            1

>    s2[2 x, nu ], s2[2 x, nu ], s2[2 x, rho ], s2[2 x, rho ], tag[{}, {}],
               2             3              1              2

>    tag[{gamma5}, {gamma5}], tag[{gamma5}, {NU[1]}], tag[{NU[1]}, {NU[1]}],

>    tag[{NU[1]}, {NU[2]}], tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    tag[{gamma5, NU[1]}, {gamma5, NU[2]}],

>    tag[{gamma5, NU[1]}, {NU[1], NU[2]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}], tag[{NU[1], NU[2]}, {NU[1], NU[3]}]}

In[76]:= (* in: gtrace[gamma5, nu[2],rho[1],gamma5,nu[1],rho[2]], the rho's must equal
either nu[1] or nu[2],
             since, otherwise, terms like: delm[nu[1],rho[1]]*delm[nu[2],rho[2]] will
vanish *)

In[77]:= %%%% /. gtrace[gamma5, nu[2], rho[1], gamma5, nu[1], rho[2]] :>
gtrace[nu[2],rho[1],nu[1],rho[2]];

In[78]:= % /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);

In[79]:= % // Expand // simplifydelm;
% //. gtracerulesD;
```

```
% // simplifydelm;
lb[%]

In[80]:=
In[81]:=
In[82]:=
Out[82]= {15, 23192}

In[83]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[84]:=
In[85]:=
In[86]:=
In[87]:=
Out[87]= {15, 23192}

In[88]:= Variables[%%]

            e              e
Out[88]= {m0 , Nc, xsq, xsq , delm[nu , nu ], Gamma[2 - e],
                                   1    2

>    gtrace[gamma5, rho ], gtrace[gamma5, rho , gamma5, rho ], s2[2 x, nu ],
                     1                    1            2            1

>    s2[2 x, nu ], s2[2 x, nu ], s2[2 x, rho ], s2[2 x, rho ], tag[{}, {}],
              2             3             1             2

>    tag[{gamma5}, {gamma5}], tag[{gamma5}, {NU[1]}], tag[{NU[1]}, {NU[1]}],

>    tag[{NU[1]}, {NU[2]}], tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    tag[{gamma5, NU[1]}, {gamma5, NU[2]}],

>    tag[{gamma5, NU[1]}, {NU[1], NU[2]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}], tag[{NU[1], NU[2]}, {NU[1], NU[3]}]}

In[89]:= (# / (# /. tag[__] -> 1 /. s2[__] -> 1 /. gtrace[___] -> 1))& /@ List @@ %%% //
Union

Out[89]= {tag[{}, {}], gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ]
                                      1            2            1
```

```
>     s2[2 x, rho ] tag[{gamma5}, {gamma5}],
               2

>    gtrace[gamma5, rho ] s2[2 x, nu ] s2[2 x, rho ] tag[{gamma5}, {NU[1]}],
                     1            1             1

                                        2
>    tag[{NU[1]}, {NU[1]}], s2[2 x, nu ]  tag[{NU[1]}, {NU[1]}],
                                      1

>    s2[2 x, nu ] s2[2 x, nu ] tag[{NU[1]}, {NU[2]}],
              1            2

                2
>    s2[2 x, nu ]  tag[{gamma5, NU[1]}, {gamma5, NU[1]}],
              1

>    gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                     1             2             1             2

>     tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    tag[{gamma5, NU[1]}, {gamma5, NU[2]}],

>    s2[2 x, nu ] s2[2 x, nu ] tag[{gamma5, NU[1]}, {gamma5, NU[2]}],
              1            2

>    gtrace[gamma5, rho ] s2[2 x, nu ] s2[2 x, rho ]
                     1            2             1

>     tag[{gamma5, NU[1]}, {NU[1], NU[2]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}],

                2
>    s2[2 x, nu ]  tag[{NU[1], NU[2]}, {NU[1], NU[2]}],
              1

                2
>    s2[2 x, nu ]  tag[{NU[1], NU[2]}, {NU[1], NU[2]}],
              2

>    s2[2 x, nu ] s2[2 x, nu ] tag[{NU[1], NU[2]}, {NU[1], NU[3]}]}
              2            3

In[90]:= %%%%

          4 e      -4 + 2 e    -3 + 2 e             2
```

```
Out[90]= m0    Nc Pi         xsq         Gamma[2 - e]  tag[{}, {}] +

        4 e      -4 + 2 e    -4 + 2 e             2
>    (m0    Nc Pi         xsq         Gamma[2 - e]

>       gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                           1             2             1             2

>       tag[{gamma5}, {gamma5}]) / 4 +

        4 e      -4 + 2 e    -4 + 2 e             2
>    (m0    Nc Pi         xsq         Gamma[2 - e]  gtrace[gamma5, rho ]
                                                                    1

>       s2[2 x, nu ] s2[2 x, rho ] tag[{gamma5}, {NU[1]}]) / 2 -
                 1             1

       4 e      -4 + 2 e    -3 + 2 e             2
>    m0    Nc Pi         xsq         Gamma[2 - e]  tag[{NU[1]}, {NU[1]}] +

         4 e      -4 + 2 e    -4 + 2 e             2             2
>    2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                 1

                                    4 e      -4 + 2 e    -4 + 2 e             2
>     tag[{NU[1]}, {NU[1]}] + 2 m0    Nc Pi         xsq         Gamma[2 - e]

>     s2[2 x, nu ] s2[2 x, nu ] tag[{NU[1]}, {NU[2]}] +
               1            2

         4 e      -4 + 2 e    -4 + 2 e             2             2
>    2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                 1

>     tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +

        4 e      -4 + 2 e    -4 + 2 e             2
>    (m0    Nc Pi         xsq         Gamma[2 - e]

>       gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                           1             2             1             2

>       tag[{gamma5, NU[1]}, {gamma5, NU[1]}]) / 4 -

       4 e      -4 + 2 e    -3 + 2 e                              2
>    m0    Nc Pi         xsq         delm[nu , nu ] Gamma[2 - e]
                                             1    2

>     tag[{gamma5, NU[1]}, {gamma5, NU[2]}] +
```

```
>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                  1

>      s2[2 x, nu ] tag[{gamma5, NU[1]}, {gamma5, NU[2]}] -
                2

>     (m0    Nc Pi         xsq         Gamma[2 - e]  gtrace[gamma5, rho ]
                                                                      1

>       s2[2 x, nu ] s2[2 x, rho ] tag[{gamma5, NU[1]}, {NU[1], NU[2]}]) / 2\
                  2             1

>      - m0    Nc Pi         xsq         Gamma[2 - e]

>      tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                  1

>      tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                  2

>      tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                  2

>      s2[2 x, nu ] tag[{NU[1], NU[2]}, {NU[1], NU[3]}]
                3

In[91]:= %%%%% /. delm[nu[1],nu[2]] -> 0;

In[92]:= lb[%]

Out[92]= {14, 21800}

In[93]:= (* Note also that: tr(g5,r1) = tr(g5,r1,r2,r2), where r2 != r1 is a g-matrix in
the d-4 directions
                                    = tr(r2,g5,r1,r2) = tr(g5,r2,r1,r2) =
-tr(g5,r1,r2,r2) = -tr(g5,r1) = 0  *)
```

```
In[97]:= %91 /. gtrace[gamma5, rho[_]] -> 0;

In[98]:= lb[%]

Out[98]= {12, 18264}

In[99]:= %%

            4 e      -4 + 2 e    -3 + 2 e             2
Out[99]= m0    Nc Pi         xsq         Gamma[2 - e]  tag[{}, {}] +

          4 e      -4 + 2 e    -4 + 2 e             2
>     (m0    Nc Pi         xsq         Gamma[2 - e]

>        gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                            1             2             1             2

>        tag[{gamma5}, {gamma5}]) / 4 -

         4 e      -4 + 2 e    -3 + 2 e             2
>     m0    Nc Pi         xsq         Gamma[2 - e]  tag[{NU[1]}, {NU[1]}] +

          4 e      -4 + 2 e    -4 + 2 e             2             2
>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                 1

                                  4 e      -4 + 2 e    -4 + 2 e             2
>      tag[{NU[1]}, {NU[1]}] + 2 m0    Nc Pi         xsq         Gamma[2 - e]

>      s2[2 x, nu ] s2[2 x, nu ] tag[{NU[1]}, {NU[2]}] +
                1             2

          4 e      -4 + 2 e    -4 + 2 e             2             2
>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                                 1

>      tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +

          4 e      -4 + 2 e    -4 + 2 e             2
>     (m0    Nc Pi         xsq         Gamma[2 - e]

>        gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                            1             2             1             2

>        tag[{gamma5, NU[1]}, {gamma5, NU[1]}]) / 4 +

          4 e      -4 + 2 e    -4 + 2 e             2
>     2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
```

```
>     s2[2 x, nu ] tag[{gamma5, NU[1]}, {gamma5, NU[2]}] -
               2

       4 e      -4 + 2 e    -3 + 2 e             2
>    m0    Nc Pi         xsq         Gamma[2 - e]

>     tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

         4 e      -4 + 2 e    -4 + 2 e             2             2
>    2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                     1

>     tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

         4 e      -4 + 2 e    -4 + 2 e             2             2
>    2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                     2

>     tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +

         4 e      -4 + 2 e    -4 + 2 e             2
>    2 m0    Nc Pi         xsq         Gamma[2 - e]  s2[2 x, nu ]
                                                     2

>     s2[2 x, nu ] tag[{NU[1], NU[2]}, {NU[1], NU[3]}]
               3

In[100]:= (* This is the tree level expression, for general values of the 4-vector x
(remember, x stands for x-y).
            We will use this later, in order to construct the renormalized Green's function
in 0+1 order *)

In[101]:= %% // InputForm

Out[101]//InputForm=
m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-3 + 2*e)*Gamma[2 - e]^2*tag[{}, {}] +
 (m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
   gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]]*
   tag[{gamma5}, {gamma5}])/4 - m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-3 + 2*e)*
  Gamma[2 - e]^2*tag[{NU[1]}, {NU[1]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*tag[{NU[1]}, {NU[1]}] +
 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]*
  s2[2*x, nu[2]]*tag[{NU[1]}, {NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*
  tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +
 (m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
   gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]]*
```

```
   tag[{gamma5, NU[1]}, {gamma5, NU[1]}])/4 + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]*s2[2*x, nu[2]]*
  tag[{gamma5, NU[1]}, {gamma5, NU[2]}] - m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-3 + 2*e)*Gamma[2 - e]^2*tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +
 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*
  tag[{NU[1], NU[2]}, {NU[1], NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[2]]^2*
  tag[{NU[1], NU[2]}, {NU[1], NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[2]]*s2[2*x, nu[3]]*
  tag[{NU[1], NU[2]}, {NU[1], NU[3]}]

In[102]:= Factor[%%%] // InputForm

Out[102]//InputForm=
(m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
  (4*xsq*tag[{}, {}] + gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*
    s2[2*x, rho[2]]*tag[{gamma5}, {gamma5}] - 4*xsq*tag[{NU[1]}, {NU[1]}] +
   8*s2[2*x, nu[1]]^2*tag[{NU[1]}, {NU[1]}] + 8*s2[2*x, nu[1]]*s2[2*x, nu[2]]*
    tag[{NU[1]}, {NU[2]}] + 8*s2[2*x, nu[1]]^2*tag[{gamma5, NU[1]},
     {gamma5, NU[1]}] + gtrace[gamma5, rho[1], gamma5, rho[2]]*
    s2[2*x, rho[1]]*s2[2*x, rho[2]]*tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +
   8*s2[2*x, nu[1]]*s2[2*x, nu[2]]*tag[{gamma5, NU[1]}, {gamma5, NU[2]}] -
   4*xsq*tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +
   8*s2[2*x, nu[1]]^2*tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +
   8*s2[2*x, nu[2]]^2*tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +
   8*s2[2*x, nu[2]]*s2[2*x, nu[3]]*tag[{NU[1], NU[2]}, {NU[1], NU[3]}]))/4
```

B.2 1-Loop Diagram 1

```
In[1]:= (* Let us find the determinant of the 7x7 coefficient matrix appearing
           in the scalar decomposition of the Tensor T_mnrs of diagram1  *)

In[2]:= {{1,  1,    1,    1,    2,     2,     2},
{1,  d^2,  d,    d,    2d,    2,     2},
{1,  d,    d^2,  d,    2,     2,     2d},
{1,  d,    d,    d^2,  2,     2d,    2},
{2,  2d,   2,    2,    2d+2,  4,     4},
{2,  2,    2,    2d,   4,     2d+2,  4},
{2,  2,    2d,   2,    4,     4,     2d+2}};

In[3]:= Det[%]

                          2       3      4       5       6       7      8
Out[3]= 8 (4 - 24 d + 57 d  - 61 d  + 9 d  + 51 d  - 61 d  + 33 d  - 9 d  +

        9
>      d )

In[4]:= Factor[%]

                  2         6
Out[4]= 8 (-2 + d)  (-1 + d)  (1 + d)

In[5]:= Inverse[%%%] // FullSimplify

          (2 + d) (4 + d)     1         1         1     2 + d   2 + d
Out[5]= {{---------------, -------, -------, -------, ------, ------,
                      2           2         2         2       2       2
              -1 + d      -1 + d    -1 + d    -1 + d    1 - d   1 - d

      2 + d        1             d                   1
>     ------}, {-------, -----------------, -------------------,
           2          2            2    3              2    3
      1 - d     -1 + d   2 - d - 2 d  + d   -2 + d + 2 d  - d

             1                     d                   1
>     -------------------, -(-----------------), -----------------,
                 2    3              2    3             2    3
      -2 + d + 2 d  - d     2 - d - 2 d  + d    2 - d - 2 d  + d

             1               1             1                   d
>     -----------------}, {-------, -----------------, -----------------,
                2    3          2              2    3             2    3
      2 - d - 2 d  + d     -1 + d   -2 + d + 2 d  - d   2 - d - 2 d  + d

             1                   1                 1
>     -----------------, -----------------, -----------------,
```

```
                2    3              2    3              2    3
      -2 + d + 2 d  - d    2 - d - 2 d  + d    2 - d - 2 d  + d

             d                  1              1
>     -(-----------------)}, {-------, ------------------,
                2    3             2               2    3
       2 - d - 2 d  + d       -1 + d    -2 + d + 2 d  - d

            1                    d                  1
>     ------------------, ------------------, ------------------,
                2    3              2    3              2    3
      -2 + d + 2 d  - d    2 - d - 2 d  + d    2 - d - 2 d  + d

             d                    1
>     -(-----------------), ------------------},
                2    3               2    3
       2 - d - 2 d  + d      2 - d - 2 d  + d

      2 + d              d                    1                  1
>     {------, -(------------------), ------------------, ------------------,
           2                2    3               2    3              2    3
      1 - d      2 - d - 2 d  + d     2 - d - 2 d  + d    2 - d - 2 d  + d

           2 + d                1                    1
>     -(--------------), -------------------, -------------------},
                  2                2    3               2    3
        4 + 2 d - 2 d     -2 + d + 2 d  - d    -2 + d + 2 d  - d

      2 + d            1                    1                    d
>     {------, ------------------, ------------------, -(------------------),
           2              2    3               2    3              2    3
      1 - d    2 - d - 2 d  + d     2 - d - 2 d  + d      2 - d - 2 d  + d

            1                  2 + d                1
>     ------------------, -(--------------), -------------------},
                2    3                 2               2    3
      -2 + d + 2 d  - d      4 + 2 d - 2 d     -2 + d + 2 d  - d

      2 + d            1                    d                    1
>     {------, ------------------, -(------------------), ------------------,
           2              2    3                2    3              2    3
      1 - d    2 - d - 2 d  + d      2 - d - 2 d  + d    2 - d - 2 d  + d

            1                    1                  2 + d
>     ------------------, ------------------, -(--------------)}}
                2    3               2    3                 2
      -2 + d + 2 d  - d    -2 + d + 2 d  - d      4 + 2 d - 2 d

In[6]:= Factor //@ %
```

```
                (2 + d) (4 + d)          1                  1
Out[6]= {{-----------------, -----------------, -----------------,
          (-1 + d) (1 + d)   (-1 + d) (1 + d)   (-1 + d) (1 + d)

>          1                  2 + d                2 + d
         -----------------, -(-----------------), -(-----------------),
         (-1 + d) (1 + d)     (-1 + d) (1 + d)     (-1 + d) (1 + d)

>          2 + d                  1                        d
         -(-----------------)}, {-----------------, -------------------------,
           (-1 + d) (1 + d)     (-1 + d) (1 + d)   (-2 + d) (-1 + d) (1 + d)

>                      1                          1
         -(-------------------------), -(-------------------------),
           (-2 + d) (-1 + d) (1 + d)    (-2 + d) (-1 + d) (1 + d)

>                      d                          1
         -(-------------------------), -------------------------,
           (-2 + d) (-1 + d) (1 + d)  (-2 + d) (-1 + d) (1 + d)

>                    1
         -------------------------},
         (-2 + d) (-1 + d) (1 + d)

>          1                          1
         {-----------------, -(-------------------------),
         (-1 + d) (1 + d)     (-2 + d) (-1 + d) (1 + d)

>                    d                          1
         -------------------------, -(-------------------------),
         (-2 + d) (-1 + d) (1 + d)    (-2 + d) (-1 + d) (1 + d)

>                    1                          1
         -------------------------, -------------------------,
         (-2 + d) (-1 + d) (1 + d)  (-2 + d) (-1 + d) (1 + d)

>                      d
         -(-------------------------)},
           (-2 + d) (-1 + d) (1 + d)

>          1                          1
         {-----------------, -(-------------------------),
         (-1 + d) (1 + d)     (-2 + d) (-1 + d) (1 + d)

>                      1                          d
         -(-------------------------), -------------------------,
           (-2 + d) (-1 + d) (1 + d)  (-2 + d) (-1 + d) (1 + d)
```

```
>      1                                  d
       ------------------------, -(------------------------),
       (-2 + d) (-1 + d) (1 + d)     (-2 + d) (-1 + d) (1 + d)

>      1
       ------------------------},
       (-2 + d) (-1 + d) (1 + d)

>          2 + d                          d
       {-(----------------), -(------------------------),
          (-1 + d) (1 + d)       (-2 + d) (-1 + d) (1 + d)

>      1                                  1
       ------------------------, ------------------------,
       (-2 + d) (-1 + d) (1 + d)  (-2 + d) (-1 + d) (1 + d)

>          2 + d                          1
       ------------------, -(------------------------),
       2 (-2 + d) (1 + d)     (-2 + d) (-1 + d) (1 + d)

>      1
       -(------------------------)},
          (-2 + d) (-1 + d) (1 + d)

>          2 + d                          1
       {-(----------------), ------------------------,
          (-1 + d) (1 + d)    (-2 + d) (-1 + d) (1 + d)

>      1                                  d
       ------------------------, -(------------------------),
       (-2 + d) (-1 + d) (1 + d)     (-2 + d) (-1 + d) (1 + d)

>      1                                  2 + d
       -(------------------------), ------------------,
          (-2 + d) (-1 + d) (1 + d)  2 (-2 + d) (1 + d)

>      1
       -(------------------------)},
          (-2 + d) (-1 + d) (1 + d)

>          2 + d                          1
       {-(----------------), ------------------------,
          (-1 + d) (1 + d)    (-2 + d) (-1 + d) (1 + d)

>      d                                  1
       -(------------------------), ------------------------,
          (-2 + d) (-1 + d) (1 + d)  (-2 + d) (-1 + d) (1 + d)

                   1                                  1
```

```
>      -(-------------------------), -(-------------------------),
         (-2 + d) (-1 + d) (1 + d)      (-2 + d) (-1 + d) (1 + d)

              2 + d
>      ------------------}}
       2 (-2 + d) (1 + d)
```

```
In[7]:= (* Let us introduce the expression for diagram 1 *)

In[8]:= << ~/latt_pert/math/inputF.m;
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prho0rder.m
Reading background-vertices.m
Reading contractF_fin.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m

In[9]:= (* denoms[a,b,l,m,n] stands for hat2[p]^a hat2[q]^m hat2[p-q]^l hat2[p-k]^b
hat2[q-k]^n *)

In[10]:= s2[2p, rho[1]] s2[2q, rho[2]] s2[2q -2k, rho[3]] s2[2p -2k, rho[4]]
denoms[1,1,1,1,1];

In[11]:= I1 * s2[2k,rho[1]] s2[2k,rho[2]] s2[2k, rho[3]] s2[2k,rho[4]] * hat2[k]^2 % +
I2 * delm[rho[1], rho[2]] delm[rho[3],rho[4]] * % +
I3 * delm[rho[1], rho[4]] delm[rho[3],rho[2]] * % +
I4 * delm[rho[1], rho[3]] delm[rho[2],rho[4]] * % +
I5 * (delm[rho[1],rho[2]] s2[2k,rho[3]] s2[2k,rho[4]] + delm[rho[3],rho[4]] s2[2k,rho[1]]
s2[2k,rho[2]]) hat2[k] * % +
I6 * (delm[rho[1],rho[3]] s2[2k,rho[2]] s2[2k,rho[4]] + delm[rho[2],rho[4]] s2[2k,rho[1]]
s2[2k,rho[3]]) hat2[k] * % +
I7 * (delm[rho[1],rho[4]] s2[2k,rho[3]] s2[2k,rho[2]] + delm[rho[3],rho[2]] s2[2k,rho[1]]
s2[2k,rho[4]]) hat2[k] * %;
```

```
In[12]:= % //. s2[a_+b_,c_] :> s2[a,c]+s2[b,c] // Expand // canonical;

In[13]:= simplifydelm[%] // reducerho;

In[14]:= % //. s2[2a_,_]^2 :> sisq[a] // reducerho;

In[15]:= Length[%]

Out[15]= 38

In[16]:= %% /. sisq[p]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a-i,b,c,d,e];

In[17]:= Variables[%]

Out[17]= {I3, denoms[0, 1, 1, 1, 1], s2[2 k, rho ], s2[2 q, rho ], I2,
                                                 1              1

>    denoms[1, 1, 1, 1, 1], s2[2 k, rho ], s2[2 p, rho ], s2[2 p, rho ],
                                       2              1              2

>    s2[2 q, rho ], I4, I5, hat2[k], s2[2 k, rho ], s2[2 p, rho ], I6, I7,
               2                                 3              3

>    s2[2 q, rho ], I1, s2[2 k, rho ], s2[2 p, rho ], sisq[k], sisq[q]}
               3                  4              4

In[18]:= diamonds = %% /. denoms[a___,0,b___] -> 0;

In[19]:= nondiamonds = %%% - %;

In[20]:= Length /@ {%%, %}

Out[20]= {34, 4}

In[21]:= diamonds /. sisq[q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c,d-i,e];

In[22]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[23]:= nondiamonds = nondiamonds + %% - %;

In[24]:= Variables[diamonds]

Out[24]= {I2, denoms[1, 1, 1, 1, 1], s2[2 k, rho ], s2[2 p, rho ],
                                                 2              1

>    s2[2 p, rho ], s2[2 q, rho ], I3, s2[2 k, rho ], s2[2 q, rho ], I4, I5,
               2              1                  1              2
```

```
>    hat2[k], s2[2 k, rho ], s2[2 p, rho ], I6, I7, s2[2 q, rho ], I1,
                          3              3                       3

>    s2[2 k, rho ], s2[2 p, rho ], sisq[k]}
               4              4

In[25]:= diamonds //. sisq[k]^i_. hat2[k]^j_. :> sisq[k]^(i-1) hat2[k]^(j-1);

In[26]:= % /. s2[2p,rho[i_]] s2[2k,rho[i_]] :> 1/2 (sisq[p]+sisq[k] - sisq[-p+k]) //
Expand;

In[27]:= % /. sisq[p]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a-i,b,c,d,e];

In[28]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[29]:= nondiamonds = nondiamonds + %% - %;

In[30]:= diamonds /. sisq[-p+k]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b-i,c,d,e];

In[31]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[32]:= nondiamonds = nondiamonds + %% - %;

In[33]:= Variables[diamonds]

Out[33]= {I2, denoms[1, 1, 1, 1, 1], s2[2 k, rho ], s2[2 p, rho ],
                                                2              1

>    s2[2 q, rho ], s2[2 q, rho ], I4, I5, I6, s2[2 p, rho ], sisq[k], I1,
               1              2                          2

>    I3, I7, hat2[k], s2[2 k, rho ], s2[2 p, rho ], s2[2 q, rho ],
                                 3              3              3

>    s2[2 k, rho ], s2[2 p, rho ]}
               4              4

In[34]:= diamonds /. s2[2p,rho[i_]] s2[2q,rho[i_]] :> 1/2 (sisq[p]+sisq[q] - sisq[-p+q])
// Expand;

In[35]:= % /. sisq[p]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a-i,b,c,d,e];

In[36]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[37]:= nondiamonds = nondiamonds + %% - %;

In[38]:= diamonds /. sisq[-p+q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c-i,d,e];

In[39]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;
```

```
In[40]:= nondiamonds = nondiamonds + %% - %;

In[41]:= diamonds /. sisq[q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c,d-i,e];

In[42]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[43]:= nondiamonds = nondiamonds + %% - %;

In[44]:= Variables[diamonds]

Out[44]= {I1, denoms[1, 1, 1, 1, 1], s2[2 k, rho ], s2[2 q, rho ], sisq[k],
                                                 2              2

>    I3, I4, I5, I6, I7, hat2[k], s2[2 k, rho ], s2[2 p, rho ],
                                               3              3

>    s2[2 q, rho ], s2[2 k, rho ], s2[2 p, rho ]}
               3              4              4

In[45]:= diamonds /. s2[2k,rho[i_]] s2[2q,rho[i_]] :> 1/2 (sisq[k]+sisq[q] - sisq[-k+q])
// Expand;

In[46]:= % /. sisq[q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c,d-i,e];

In[47]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[48]:= nondiamonds = nondiamonds + %% - %;

In[49]:= diamonds /. sisq[-k+q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c,d,e-i];

In[50]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[51]:= nondiamonds = nondiamonds + %% - %;

In[52]:= Variables[diamonds]

Out[52]= {I1, denoms[1, 1, 1, 1, 1], sisq[k], I3, I4, I5, I6, I7, hat2[k],

>    s2[2 k, rho ], s2[2 p, rho ], s2[2 q, rho ], s2[2 k, rho ],
               3              3              3              4

>    s2[2 p, rho ]}
               4

In[53]:= diamonds /. s2[2p,rho[i_]] s2[2k,rho[i_]] :> 1/2 (sisq[p]+sisq[k] - sisq[-p+k])
// Expand;

In[54]:= % /. sisq[p]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a-i,b,c,d,e];
```

```
In[55]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[56]:= nondiamonds = nondiamonds + %% - %;

In[57]:= diamonds /. sisq[-p+k]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b-i,c,d,e];

In[58]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[59]:= nondiamonds = nondiamonds + %% - %;

In[60]:= Variables[diamonds]

Out[60]= {I1, denoms[1, 1, 1, 1, 1], sisq[k], I3, I4, I5, I6, I7, hat2[k],

>    s2[2 k, rho ], s2[2 q, rho ]}
               3              3

In[61]:= diamonds /. s2[2k,rho[i_]] s2[2q,rho[i_]] :> 1/2 (sisq[k]+sisq[q] - sisq[-k+q])
// Expand;

In[69]:= %61 /. sisq[q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c,d-i,e];

In[70]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[71]:= nondiamonds = %59 + %% - %;

In[72]:= diamonds /. sisq[-k+q]^i_. denoms[a_,b_,c_,d_,e_] :> denoms[a,b,c,d,e-i];

In[73]:= diamonds = % /. denoms[a___,c_?((#<=0)&),b___] -> 0;

In[74]:= nondiamonds = nondiamonds + %% - %;

In[75]:= Variables[diamonds]

Out[75]= {I1, denoms[1, 1, 1, 1, 1], sisq[k], I3, I4, I5, I6, I7, hat2[k]}

In[76]:= diamonds = diamonds //. sisq[k]^i_. hat2[k]^j_. :> sisq[k]^(i-1) hat2[k]^(j-1);

In[77]:= Variables[diamonds]

Out[77]= {I1, denoms[1, 1, 1, 1, 1], sisq[k], I3, I4, I6, I7}

In[78]:= diamonds

                                        2
         I1 denoms[1, 1, 1, 1, 1] sisq[k]
Out[78]= --------------------------------- +
                      16
```

```
                                    2                                    2
     I3 denoms[1, 1, 1, 1, 1] sisq[k]    I4 denoms[1, 1, 1, 1, 1] sisq[k]
>    --------------------------------- + --------------------------------- +
                    4                                    4

                                    2                                    2
     I6 denoms[1, 1, 1, 1, 1] sisq[k]    I7 denoms[1, 1, 1, 1, 1] sisq[k]
>    --------------------------------- + ---------------------------------
                    4                                    4
```

```
In[79]:= Save["/home/haris/latt_pert/GIRS/diagram1/diagram1.m", diamonds, nondiamonds]

In[80]:= Length /@ {diamonds, nondiamonds}

Out[80]= {5, 129}
```

```
In[1]:= << ~/latt_pert/math/inputF.m;
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[2]:= << diagram1.m;

In[3]:= nondiamonds;

In[4]:= Variables[%]

Out[4]= {I3, denoms[1, 1, 1, 0, 1], s2[2 k, rho ], s2[2 p, rho ], I7,
                                                 1              1

>    hat2[k], s2[2 k, rho ], s2[2 p, rho ], denoms[0, 1, 1, 1, 1],
                         2              2

>    s2[2 q, rho ], I2, I5, denoms[1, 0, 1, 1, 1], I1, s2[2 q, rho ], I6, I4,
               1                                                 2

>    denoms[1, 1, 0, 1, 1], s2[2 k, rho ], s2[2 p, rho ], s2[2 q, rho ],
                                       3              3              3

>    s2[2 k, rho ], s2[2 p, rho ], sisq[k], denoms[1, 1, 1, 1, 0], sisq[q]}
               4              4

In[5]:= Sort[%]
```

```
Out[5]= {I1, I2, I3, I4, I5, I6, I7, denoms[0, 1, 1, 1, 1],

>    denoms[1, 0, 1, 1, 1], denoms[1, 1, 0, 1, 1], denoms[1, 1, 1, 0, 1],

>    denoms[1, 1, 1, 1, 0], hat2[k], s2[2 k, rho ], s2[2 k, rho ],
                                                    1             2

>    s2[2 k, rho ], s2[2 k, rho ], s2[2 p, rho ], s2[2 p, rho ],
               3             4             1             2

>    s2[2 p, rho ], s2[2 p, rho ], s2[2 q, rho ], s2[2 q, rho ],
               3             4             1             2

>    s2[2 q, rho ], sisq[k], sisq[q]}
               3

In[6]:= Select[%3,!FreeQ[#,sisq[q]]&]

Out[6]= I3 denoms[0, 1, 1, 1, 1] sisq[q]

In[7]:= (* This term leads to a scaleless expression, which therefore vanishes in
dimensional regularization *)

In[8]:= %3 - Select[%3,!FreeQ[#,sisq[q]]&];

In[9]:= Variables[%] // Sort

Out[9]= {I1, I2, I3, I4, I5, I6, I7, denoms[0, 1, 1, 1, 1],

>    denoms[1, 0, 1, 1, 1], denoms[1, 1, 0, 1, 1], denoms[1, 1, 1, 0, 1],

>    denoms[1, 1, 1, 1, 0], hat2[k], s2[2 k, rho ], s2[2 k, rho ],
                                                    1             2

>    s2[2 k, rho ], s2[2 k, rho ], s2[2 p, rho ], s2[2 p, rho ],
               3             4             1             2

>    s2[2 p, rho ], s2[2 p, rho ], s2[2 q, rho ], s2[2 q, rho ],
               3             4             1             2

>    s2[2 q, rho ], sisq[k]}
               3

In[10]:= (* Let us separate the above into "eye"- and "eyeglasses"-type integrals *)

In[11]:= eyeglasses = Select[%8, !FreeQ[#,denoms[1,1,0,1,1]]&];

In[12]:= eye = %8 - eyeglasses;
```

```
In[13]:= eyeglasses /. denoms[1,1,0,1,1] -> hat2[p] hat2[p-k] hat2[k-q] hat2[q];

In[14]:= % /. a_ hat2[p]^i_. hat2[p-k]^j_. :> (k^2)^(2-e-i-j)/(4Pi)^2 *
Sum[g[i,j,count[a,s2[2p,_]],s] (k^2)^s /s!/4^s Nest[box,a,s] ,{s,0,count[a,s2[2p,_]]/2}]
set[p,k];

In[15]:= Expand[%]; (* In an older version we had: Expand[mu^(2e) %]; we will
                       do this later, for the sake of uniformity *)

In[16]:= FreeQ[%,box]

Out[16]= True

In[17]:= %% /. a_. set[p,k] :> (a /. p -> k);

In[18]:= % /. a_ hat2[q]^i_. hat2[k-q]^j_. :> (k^2)^(2-e-i-j)/(4Pi)^2 *
Sum[g[i,j,count[a,s2[2q,_]],s] (k^2)^s /s!/4^s Nest[box,a,s] ,{s,0,count[a,s2[2q,_]]/2}]
set[q,k];

In[19]:= FreeQ[%,box]

Out[19]= True

In[20]:= %% /. a_. set[q,k] :> (a /. q -> k);

In[21]:= Variables[%]

                           2 e
Out[21]= {I2, I4, I5, I6, (k ) , g[1, 1, 0, 0], g[1, 1, 1, 0], hat2[k],

>    s2[2 k, rho ], s2[2 k, rho ], sisq[k]}
               2              3

In[22]:= %% //. s2[2k,a_]^2 :> sisq[k];

In[23]:= Variables[%]

                           2 e
Out[23]= {I2, I4, I5, I6, (k ) , g[1, 1, 0, 0], g[1, 1, 1, 0], hat2[k],

>    sisq[k]}

In[24]:= %% /. hat2[k]->1/k^2 /. sisq[k] -> k^2

               2 1 - 2 e              2         2 1 - 2 e              2
        -(I2 (k )        g[1, 1, 0, 0] )   I4 (k )        g[1, 1, 0, 0]
Out[24]= ------------------------------ + ------------------------------ -
                         4                                4
                  1024 Pi                          1024 Pi
```

```
     2 1 - 2 e              2        2 1 - 2 e              2
    I5 (k )        g[1, 1, 0, 0]    I6 (k )        g[1, 1, 0, 0]
>   ---------------------------- + ---------------------------- -
                    4                                 4
             1024 Pi                           1024 Pi

    4   2 -1 - 2 e
    I5 k  (k )         g[1, 1, 0, 0] g[1, 1, 1, 0]
>   ------------------------------------------- -
                         4
                   512 Pi

    4   2 -1 - 2 e
    I6 k  (k )         g[1, 1, 0, 0] g[1, 1, 1, 0]
>   ------------------------------------------- +
                         4
                   512 Pi

     2 1 - 2 e
    I2 (k )        g[1, 1, 0, 0] g[1, 1, 1, 0]
>   ---------------------------------------- +
                       4
                 512 Pi

     2 1 - 2 e
    I4 (k )        g[1, 1, 0, 0] g[1, 1, 1, 0]
>   ---------------------------------------- +
                       4
                 512 Pi

     2 1 - 2 e
    I5 (k )        g[1, 1, 0, 0] g[1, 1, 1, 0]
>   ---------------------------------------- +
                       4
                 512 Pi

     2 1 - 2 e
    I6 (k )        g[1, 1, 0, 0] g[1, 1, 1, 0]
>   ---------------------------------------- -
                       4
                 512 Pi

     2 1 - 2 e              2        2 1 - 2 e              2
    I2 (k )        g[1, 1, 1, 0]    I4 (k )        g[1, 1, 1, 0]
>   ---------------------------- - ----------------------------
                    4                                 4
             512 Pi                            512 Pi
```

```
In[25]:= eye /. a_. denoms[1,1,1,1,0] :> (a /. {p->q,q->p}) denoms[1,0,1,1,1];

In[26]:= Variables[%]

Out[26]= {I3, denoms[1, 1, 1, 0, 1], s2[2 k, rho ], s2[2 p, rho ], I7,
                                                  1             1

>    hat2[k], s2[2 k, rho ], s2[2 p, rho ], denoms[0, 1, 1, 1, 1],
                        2             2

>    s2[2 q, rho ], I2, I5, denoms[1, 0, 1, 1, 1], I1, s2[2 q, rho ], I6, I4,
               1                                                2

>    s2[2 k, rho ], s2[2 p, rho ], s2[2 q, rho ], s2[2 k, rho ],
               3             3             3             4

>    s2[2 p, rho ], sisq[k], s2[2 q, rho ]}
               4                       4

In[27]:= %% /. a_. denoms[1,1,1,0,1] :> (a /. {p->q,q->p}) denoms[0,1,1,1,1];

In[28]:= Variables[%]

Out[28]= {I2, denoms[0, 1, 1, 1, 1], s2[2 k, rho ], s2[2 p, rho ], I4, I5,
                                               2             2

>    I6, I3, s2[2 k, rho ], s2[2 q, rho ], s2[2 p, rho ],
                     1             1             1

>    denoms[1, 0, 1, 1, 1], I1, s2[2 q, rho ], I7, hat2[k], s2[2 k, rho ],
                                        2                           3

>    s2[2 p, rho ], s2[2 q, rho ], s2[2 k, rho ], s2[2 p, rho ], sisq[k],
               3             3             4             4

>    s2[2 q, rho ]}
               4

In[29]:= %% /. a_. denoms[0,1,1,1,1] :> (a /. {p->k-p,q->k-q}) denoms[1,0,1,1,1];

In[30]:= Variables[%]

Out[30]= {I2, denoms[1, 0, 1, 1, 1], s2[2 k, rho ], s2[2 (k - p), rho ], I4,
                                               2                     2

>    I5, I6, I3, s2[2 k, rho ], s2[2 (k - q), rho ], s2[2 (k - p), rho ], I1,
                           1                     1                     1

>    s2[2 (k - q), rho ], I7, hat2[k], s2[2 k, rho ], s2[2 (k - p), rho ],
```

```
                     2                           3                   3

>    s2[2 (k - q), rho ], s2[2 k, rho ], s2[2 (k - p), rho ], s2[2 p, rho ],
                    3             4                   4             1

>    s2[2 q, rho ], s2[2 q, rho ], s2[2 p, rho ], s2[2 p, rho ],
               1             2             2             3

>    s2[2 q, rho ], s2[2 p, rho ], sisq[k], s2[2 (k - q), rho ],
               3             4                            4

>    s2[2 q, rho ]}
               4

In[31]:= ExpandAll[%%] /. s2[a_ + b_, c_] :> s2[a,c]+s2[b,c] // Expand // canonical;

In[32]:= Variables[%]

Out[32]= {I2, denoms[1, 0, 1, 1, 1], s2[2 k, rho ], I3, I5, I1,
                                                1

>    s2[2 k, rho ], I6, I7, hat2[k], s2[2 k, rho ], s2[2 k, rho ],
               2                               3             4

>    s2[2 p, rho ], s2[2 p, rho ], s2[2 p, rho ], s2[2 p, rho ],
               1             2             3             4

>    s2[2 q, rho ], s2[2 q, rho ], I4, s2[2 q, rho ], sisq[k], s2[2 q, rho ]}
               1             2                 3                        4

In[33]:= eye = %%;

In[34]:= eyeglasses = %24;

In[35]:= eye /. denoms[1,0,1,1,1] -> hat2[p] hat2[q] hat2[k-q] hat2[p-q];

In[36]:= % /. a_ hat2[p]^i_. hat2[p-q]^j_. :> (q^2)^(2-e-i-j)/(4Pi)^2 *
Sum[g[i,j,count[a,s2[2p,_]],s] (q^2)^s /s!/4^s Nest[box,a,s] ,{s,0,count[a,s2[2p,_]]/2}]
set[p,q];

In[37]:= Expand[%]; (* In an older version we had: Expand[mu^(2e) %]; we will
                       do this later, for the sake of uniformity *)

In[38]:= FreeQ[%,box]

Out[38]= True

In[39]:= %% /. a_. set[p,q] :> (a /. p -> q);
```

```
In[40]:= Variables[%]

                                          2 e
Out[40]= {I1, I2, I3, I4, I5, I6, I7, (q ) , g[1, 1, 0, 0], g[1, 1, 1, 0],

>    hat2[k], hat2[k - q], hat2[q], s2[2 k, rho ], s2[2 k, rho ],
                                                 1             2

>    s2[2 k, rho ], s2[2 k, rho ], s2[2 q, rho ], s2[2 q, rho ],
               3             4             1             2

>    s2[2 q, rho ], s2[2 q, rho ], sisq[k]}
               3             4

In[41]:= %% /. q^2 -> 1/hat2[q];

In[42]:= Variables[%]

Out[42]= {I1, I2, I3, I4, I5, I6, I7, g[1, 1, 0, 0], g[1, 1, 1, 0], hat2[k],

                     1    e
>    hat2[k - q], (-------) , hat2[q], s2[2 k, rho ], s2[2 k, rho ],
                   hat2[q]                          1             2

>    s2[2 k, rho ], s2[2 k, rho ], s2[2 q, rho ], s2[2 q, rho ],
               3             4             1             2

>    s2[2 q, rho ], s2[2 q, rho ], sisq[k]}
               3             4

In[43]:= PowerExpand /@ %% ;

In[44]:= Variables[%]

Out[44]= {I1, I2, I3, I4, I5, I6, I7, g[1, 1, 0, 0], g[1, 1, 1, 0], hat2[k],

                                    e
>    hat2[k - q], hat2[q], hat2[q] , s2[2 k, rho ], s2[2 k, rho ],
                                                   1             2

>    s2[2 k, rho ], s2[2 k, rho ], s2[2 q, rho ], s2[2 q, rho ],
               3             4             1             2

>    s2[2 q, rho ], s2[2 q, rho ], sisq[k]}
               3             4

In[45]:= (# / (# /. hat2[q]->1))& /@ List @@ %% // Union

                 1 + e
```

```
Out[45]= {hat2[q]     }

In[46]:= %%% /. a_?(FreeQ[#,hat2]&) hat2[q]^i_. hat2[k-q]^j_. :> (k^2)^(2-e-i-j)/(4Pi)^2
* Sum[g[i,j,count[a,s2[2q,_]],s] (k^2)^s /s!/4^s
Nest[box,a,s] ,{s,0,count[a,s2[2q,_]]/2}] set[q,k];

In[47]:= FreeQ[%,box]

Out[47]= False

In[48]:= %% //. box[a_?(FreeQ[#,q]&) b_] :> a box[b];

In[49]:= % /. box[a_?(FreeQ[#,box]&)] :> deriv[deriv[a,rho[11]],rho[11]] // reducerho;

In[50]:= FreeQ[%,box]

Out[50]= True

In[51]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
lb[%]

In[52]:=
In[53]:=
In[54]:=
In[55]:=
In[56]:=
In[57]:=
Out[57]= {93, 148704}

In[58]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
lb[%]

In[59]:=
In[60]:=
In[61]:=
In[62]:=
In[63]:=
In[64]:=
Out[64]= {87, 132192}
```

```
In[65]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
lb[%]

In[66]:=
In[67]:=
In[68]:=
In[69]:=
In[70]:=
In[71]:=
Out[71]= {86, 129656}

In[72]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
lb[%]

In[73]:=
In[74]:=
In[75]:=
In[76]:=
In[77]:=
In[78]:=
Out[78]= {86, 129656}

In[79]:= Variables[%%]

                                          2 e
Out[79]= {I1, I2, I3, I4, I5, I6, I7, k, (k ) , delm[rho , rho ],
                                                    1     2

>    delm[rho , rho ], delm[rho , rho ], delm[rho , rho ], delm[rho , rho ],
           1     3           1     4           2     3           2     4

>    delm[rho , rho ], g[1, 1, 0, 0], g[1, 1, 1, 0], g[1 + e, 1, 0, 0],
           3     4

>    g[1 + e, 1, 1, 0], g[1 + e, 1, 2, 0], g[1 + e, 1, 2, 1],

>    g[1 + e, 1, 3, 0], g[1 + e, 1, 3, 1], hat2[k], s2[2 k, rho ],
                                                              1
```

```
>    s2[2 k, rho ], s2[2 k, rho ], s2[2 q, rho ], s2[2 q, rho ],
               2             3             1             2

>    s2[2 q, rho ], set[q, k], sisq[k]}
               3

In[80]:= %%% /. a_. set[q,k] :> (a /. q -> k);

In[81]:= Variables[%]

                                          2 e
Out[81]= {I1, I2, I3, I4, I5, I6, I7, k, (k ) , delm[rho , rho ],
                                                    1     2

>    delm[rho , rho ], delm[rho , rho ], delm[rho , rho ], delm[rho , rho ],
           1     3           1     4           2     3           2     4

>    delm[rho , rho ], g[1, 1, 0, 0], g[1, 1, 1, 0], g[1 + e, 1, 0, 0],
           3     4

>    g[1 + e, 1, 1, 0], g[1 + e, 1, 2, 0], g[1 + e, 1, 2, 1],

>    g[1 + e, 1, 3, 0], g[1 + e, 1, 3, 1], hat2[k], s2[2 k, rho ],
                                                          1

>    s2[2 k, rho ], s2[2 k, rho ], sisq[k]}
               2             3

In[82]:= %% // Expand // simplifydelm // reducerho ;

In[83]:= Variables[%]

                                          2 e
Out[83]= {I1, I2, I3, I4, I5, I6, I7, k, (k ) , nDim, g[1, 1, 0, 0],

>    g[1, 1, 1, 0], g[1 + e, 1, 0, 0], g[1 + e, 1, 1, 0], g[1 + e, 1, 2, 0],

>    g[1 + e, 1, 2, 1], g[1 + e, 1, 3, 0], g[1 + e, 1, 3, 1], hat2[k],

>    s2[2 k, rho ], s2[2 k, rho ], s2[2 k, rho ], sisq[k]}
               1             2             3

In[84]:= %% //. s2[2k,a_]^2 :> sisq[k];

In[85]:= Variables[%]

                                          2 e
Out[85]= {I1, I2, I3, I4, I5, I6, I7, k, (k ) , nDim, g[1, 1, 0, 0],
```

```
>    g[1, 1, 1, 0], g[1 + e, 1, 0, 0], g[1 + e, 1, 1, 0], g[1 + e, 1, 2, 0],

>    g[1 + e, 1, 2, 1], g[1 + e, 1, 3, 0], g[1 + e, 1, 3, 1], hat2[k],

>    sisq[k]}

In[86]:= %% /. hat2[k]->1/k^2 /. sisq[k] -> k^2;

In[87]:= eye = %;

In[88]:= diamonds

                                        2
         I1 denoms[1, 1, 1, 1, 1] sisq[k]
Out[88]= --------------------------------- +
                       16

                                      2                                    2
     I3 denoms[1, 1, 1, 1, 1] sisq[k]    I4 denoms[1, 1, 1, 1, 1] sisq[k]
>    --------------------------------- + --------------------------------- +
                    4                                    4

                                      2                                    2
     I6 denoms[1, 1, 1, 1, 1] sisq[k]    I7 denoms[1, 1, 1, 1, 1] sisq[k]
>    --------------------------------- + ---------------------------------
                    4                                    4

In[89]:= %/. denoms[1,1,1,1,1] -> (-1)/(2e) (k^2)^(-1-2e) /(4Pi)^4 (2 g[1,2,0,0] g[1,1,0,0]
-2 g[1,1,0,0]g[2,1+e,0,0]) /. sisq[k] -> k^2 // PowerExpand

                4 + 2 (-1 - 2 e)
Out[89]= -(I1 k

>         (2 g[1, 1, 0, 0] g[1, 2, 0, 0] - 2 g[1, 1, 0, 0] g[2, 1 + e, 0, 0]))\

                    4         4 + 2 (-1 - 2 e)
>      / (8192 e Pi ) - (I3 k

>        (2 g[1, 1, 0, 0] g[1, 2, 0, 0] - 2 g[1, 1, 0, 0] g[2, 1 + e, 0, 0]))\

                    4         4 + 2 (-1 - 2 e)
>      / (2048 e Pi ) - (I4 k

>        (2 g[1, 1, 0, 0] g[1, 2, 0, 0] - 2 g[1, 1, 0, 0] g[2, 1 + e, 0, 0]))\

                    4         4 + 2 (-1 - 2 e)
>      / (2048 e Pi ) - (I6 k
```

```
>       (2 g[1, 1, 0, 0] g[1, 2, 0, 0] - 2 g[1, 1, 0, 0] g[2, 1 + e, 0, 0]))\

                   4         4 + 2 (-1 - 2 e)
>      / (2048 e Pi ) - (I7 k

>       (2 g[1, 1, 0, 0] g[1, 2, 0, 0] - 2 g[1, 1, 0, 0] g[2, 1 + e, 0, 0]))\

                   4
>      / (2048 e Pi )

In[90]:= Simplify[%]

                                          2 - 4 e
Out[90]= -((I1 + 4 (I3 + I4 + I6 + I7)) k        g[1, 1, 0, 0]

                                                              4
>       (g[1, 2, 0, 0] - g[2, 1 + e, 0, 0])) / (4096 e Pi )

In[91]:= diamonds = %;

In[92]:= eye + eyeglasses + diamonds // Expand;
lb[%]

In[93]:=
Out[93]= {75, 62184}

In[94]:= Series[%%,{e,0,0}] // Normal

                2
Out[94]= (-(I1 k  g[1, 1, 0, 0] g[1, 2, 0, 0]) -

              2
>       4 I3 k  g[1, 1, 0, 0] g[1, 2, 0, 0] -

              2
>       4 I4 k  g[1, 1, 0, 0] g[1, 2, 0, 0] -

              2
>       4 I6 k  g[1, 1, 0, 0] g[1, 2, 0, 0] -

              2
>       4 I7 k  g[1, 1, 0, 0] g[1, 2, 0, 0] +

            2
>       I1 k  g[1, 1, 0, 0] g[2, 1, 0, 0] +

              2
>       4 I3 k  g[1, 1, 0, 0] g[2, 1, 0, 0] +
```

```
                2
>       4 I4 k  g[1, 1, 0, 0] g[2, 1, 0, 0] +

                2
>       4 I6 k  g[1, 1, 0, 0] g[2, 1, 0, 0] +

                2                                          4
>       4 I7 k  g[1, 1, 0, 0] g[2, 1, 0, 0]) / (4096 e Pi ) +

             2              2         2              2
>    (-2 I1 k  g[1, 1, 0, 0]  - 4 I2 k  g[1, 1, 0, 0]  -

              2              2         2              2
>       24 I3 k  g[1, 1, 0, 0]  - 4 I4 k  g[1, 1, 0, 0]  -

                2              2         2              2
>       4 I5 k  g[1, 1, 0, 0]  - 4 I6 k  g[1, 1, 0, 0]  -

                2              2         2
>       8 I7 k  g[1, 1, 0, 0]  - 8 I1 k  g[1, 1, 0, 0] g[1, 1, 1, 0] +

                2
>       8 I2 k  g[1, 1, 0, 0] g[1, 1, 1, 0] +

               2
>       16 I3 k  g[1, 1, 0, 0] g[1, 1, 1, 0] -

                2
>       8 I4 k  g[1, 1, 0, 0] g[1, 1, 1, 0] -

                2
>       8 I5 k  g[1, 1, 0, 0] g[1, 1, 1, 0] -

               2
>       24 I6 k  g[1, 1, 0, 0] g[1, 1, 1, 0] -

               2                                       2              2
>       32 I7 k  g[1, 1, 0, 0] g[1, 1, 1, 0] + 4 I1 k  g[1, 1, 1, 0]  -

                2              2         2              2
>       8 I2 k  g[1, 1, 1, 0]  - 8 I4 k  g[1, 1, 1, 0]  +

                2              2         2              2
>       8 I5 k  g[1, 1, 1, 0]  + 8 I6 k  g[1, 1, 1, 0]  +

               2              2         2
>       16 I7 k  g[1, 1, 1, 0]  + 8 I1 k  g[1, 1, 0, 0] g[1, 1, 2, 0] +

               2
```

```
>       16 I5 k  g[1, 1, 0, 0] g[1, 1, 2, 0] +

              2
>       16 I6 k  g[1, 1, 0, 0] g[1, 1, 2, 0] +

              2
>       32 I7 k  g[1, 1, 0, 0] g[1, 1, 2, 0] +

              2
>       16 I1 k  g[1, 1, 1, 0] g[1, 1, 2, 0] +

              2
>       16 I2 k  g[1, 1, 1, 0] g[1, 1, 2, 0] +

              2
>       16 I4 k  g[1, 1, 1, 0] g[1, 1, 2, 0] +

              2
>       32 I5 k  g[1, 1, 1, 0] g[1, 1, 2, 0] +

              2
>       32 I6 k  g[1, 1, 1, 0] g[1, 1, 2, 0] +

             2
>       4 I1 k  g[1, 1, 0, 0] g[1, 1, 2, 1] +

             2
>       8 I5 k  g[1, 1, 0, 0] g[1, 1, 2, 1] +

             2
>       8 I6 k  g[1, 1, 0, 0] g[1, 1, 2, 1] +

              2
>       16 I7 k  g[1, 1, 0, 0] g[1, 1, 2, 1] +

             2
>       8 I1 k  g[1, 1, 1, 0] g[1, 1, 2, 1] +

             2
>       8 I5 k  g[1, 1, 1, 0] g[1, 1, 2, 1] +

             2
>       8 I6 k  g[1, 1, 1, 0] g[1, 1, 2, 1] +

             2
>       8 I2 k  nDim g[1, 1, 1, 0] g[1, 1, 2, 1] +

             2
>       8 I4 k  nDim g[1, 1, 1, 0] g[1, 1, 2, 1] +
```

```
                2
>       8 I5 k  nDim g[1, 1, 1, 0] g[1, 1, 2, 1] +

                2
>       8 I6 k  nDim g[1, 1, 1, 0] g[1, 1, 2, 1] -

                 2
>       16 I1 k  g[1, 1, 1, 0] g[1, 1, 3, 0] -

                 2
>       32 I5 k  g[1, 1, 1, 0] g[1, 1, 3, 0] -

                 2
>       32 I6 k  g[1, 1, 1, 0] g[1, 1, 3, 0] -

                 2
>       24 I1 k  g[1, 1, 1, 0] g[1, 1, 3, 1] -

                 2
>       32 I5 k  g[1, 1, 1, 0] g[1, 1, 3, 1] -

                 2
>       32 I6 k  g[1, 1, 1, 0] g[1, 1, 3, 1] -

                 2
>       16 I5 k  nDim g[1, 1, 1, 0] g[1, 1, 3, 1] -

                 2
>       16 I6 k  nDim g[1, 1, 1, 0] g[1, 1, 3, 1] +

                2
>       4 I1 k  g[1, 1, 0, 0] g[1, 2, 0, 0] Log[k] +

                 2
>       16 I3 k  g[1, 1, 0, 0] g[1, 2, 0, 0] Log[k] +

                 2
>       16 I4 k  g[1, 1, 0, 0] g[1, 2, 0, 0] Log[k] +

                 2
>       16 I6 k  g[1, 1, 0, 0] g[1, 2, 0, 0] Log[k] +

                 2
>       16 I7 k  g[1, 1, 0, 0] g[1, 2, 0, 0] Log[k] -

                2
>       4 I1 k  g[1, 1, 0, 0] g[2, 1, 0, 0] Log[k] -
```

```
                2
>       16 I3 k  g[1, 1, 0, 0] g[2, 1, 0, 0] Log[k] -

                2
>       16 I4 k  g[1, 1, 0, 0] g[2, 1, 0, 0] Log[k] -

                2
>       16 I6 k  g[1, 1, 0, 0] g[2, 1, 0, 0] Log[k] -

                2
>       16 I7 k  g[1, 1, 0, 0] g[2, 1, 0, 0] Log[k] +

            2                (0,1,0,0)
>       I1 k  g[1, 1, 0, 0] g         [2, 1, 0, 0] +

                2                (0,1,0,0)
>       4 I3 k  g[1, 1, 0, 0] g         [2, 1, 0, 0] +

                2                (0,1,0,0)
>       4 I4 k  g[1, 1, 0, 0] g         [2, 1, 0, 0] +

                2                (0,1,0,0)
>       4 I6 k  g[1, 1, 0, 0] g         [2, 1, 0, 0] +

                2                (0,1,0,0)                          4
>       4 I7 k  g[1, 1, 0, 0] g         [2, 1, 0, 0]) / (4096 Pi )

In[95]:= {g[a_,b_,n_,s_] :> (4Pi)^e Gamma[a+b-s-2+e]/Gamma[a]/Gamma[b]
Beta[2-e-a+n-s,2-e-b+s]};

In[96]:= %%%% /. %;

In[97]:= Variables[%]

                                                 e    2 e
Out[97]= {e, I1, I2, I3, I4, I5, I6, I7, k, k , (k ) , nDim,

>    Beta[1 - 2 e, 1 - e], Beta[2 - 2 e, 1 - e], Beta[2 - 2 e, 2 - e],

>    Beta[3 - 2 e, 1 - e], Beta[3 - 2 e, 2 - e], Beta[4 - 2 e, 1 - e],

>    Beta[1 - e, 1 - e], Beta[1 - e, -e], Beta[2 - e, 1 - e],

>    Beta[-e, 1 - 2 e], Gamma[e], Gamma[2 e], Gamma[1 + e], Gamma[-1 + 2 e],

>    Gamma[1 + 2 e]}

In[98]:= (* Thus the result corresponding to In[11] of diagram1.doc is: *)
```

```
In[99]:= diamondsEyeEyeglasses = %92;

In[100]:= Save["diagram1.m3", diamondsEyeEyeglasses]
```

```
In[1]:= (*     I[mu,nu,rho,sigma] = f[1,k^2] * s2[2k,mu] s2[2k,nu] s2[2k, rho] s2[2k,sigma]
* hat2[k]^2  +
f[2,k^2] * delm[mu, nu] delm[rho,sigma]  +
f[3,k^2] * delm[mu, sigma] delm[rho,nu]  +
f[4,k^2] * delm[mu, rho] delm[nu,sigma]  +
f[5,k^2] * (delm[mu,nu] s2[2k,rho] s2[2k,sigma] + delm[rho,sigma] s2[2k,mu] s2[2k,nu])
hat2[k] +
f[6,k^2] * (delm[mu,rho] s2[2k,nu] s2[2k,sigma] + delm[nu,sigma] s2[2k,mu] s2[2k,rho])
hat2[k] +
f[7,k^2] * (delm[mu,sigma] s2[2k,rho] s2[2k,nu] + delm[rho,nu] s2[2k,mu] s2[2k,sigma])
hat2[k]    *)

In[2]:= (* The expression we need to integrate over k is:
          Exp[I k (x-y)] I[rho[1],rho[2],rho[3],rho[4]]
gtrace[X,rho[1],rho[5],rho[2],Y,rho[3],rho[5],rho[4]] / (2Pi)^d
          We will also make use of the fact that: f[i,k^2] = (k^2)^(1-2e) c[i], i=1,...,7,
where c[i] are constants  *)

In[3]:= f[1,k^2] * s2[2k,mu] s2[2k,nu] s2[2k, rho] s2[2k,sigma] * hat2[k]^2  +
f[2,k^2] * delm[mu, nu] delm[rho,sigma]  +
f[3,k^2] * delm[mu, sigma] delm[rho,nu]  +
f[4,k^2] * delm[mu, rho] delm[nu,sigma]  +
f[5,k^2] * (delm[mu,nu] s2[2k,rho] s2[2k,sigma] + delm[rho,sigma] s2[2k,mu] s2[2k,nu])
hat2[k] +
f[6,k^2] * (delm[mu,rho] s2[2k,nu] s2[2k,sigma] + delm[nu,sigma] s2[2k,mu] s2[2k,rho])
hat2[k] +
f[7,k^2] * (delm[mu,sigma] s2[2k,rho] s2[2k,nu] + delm[rho,nu] s2[2k,mu] s2[2k,sigma])
hat2[k]  /. f[i_,k^2] :> (k^2)^(1-2e) c[i] /. hat2[k] -> 1/k^2 // PowerExpand

         2 (1 - 2 e)
Out[3]= k            c[4] delm[mu, rho] delm[nu, sigma] +

      2 (1 - 2 e)
>    k           c[3] delm[mu, sigma] delm[rho, nu] +

      2 (1 - 2 e)
>    k           c[2] delm[mu, nu] delm[rho, sigma] +

      -4 + 2 (1 - 2 e)
>    k                c[1] s2[2 k, mu] s2[2 k, nu] s2[2 k, rho]

>     s2[2 k, sigma] + (c[7] (delm[mu, sigma] s2[2 k, nu] s2[2 k, rho] +

                                                    4 e
>         delm[rho, nu] s2[2 k, mu] s2[2 k, sigma])) / k    +

>    (c[6] (delm[nu, sigma] s2[2 k, mu] s2[2 k, rho] +
```

```
                                                          4 e
>         delm[mu, rho] s2[2 k, nu] s2[2 k, sigma])) / k    +

>    (c[5] (delm[rho, sigma] s2[2 k, mu] s2[2 k, nu] +

                                                          4 e
>         delm[mu, nu] s2[2 k, rho] s2[2 k, sigma])) / k

In[4]:= % /. k^a_ :> k^Expand[a]

         2 - 4 e
Out[4]= k        c[4] delm[mu, rho] delm[nu, sigma] +

      2 - 4 e
>    k        c[3] delm[mu, sigma] delm[rho, nu] +

      2 - 4 e
>    k        c[2] delm[mu, nu] delm[rho, sigma] +

      -2 - 4 e
>    k         c[1] s2[2 k, mu] s2[2 k, nu] s2[2 k, rho] s2[2 k, sigma] +

>    (c[7] (delm[mu, sigma] s2[2 k, nu] s2[2 k, rho] +

                                                          4 e
>         delm[rho, nu] s2[2 k, mu] s2[2 k, sigma])) / k    +

>    (c[6] (delm[nu, sigma] s2[2 k, mu] s2[2 k, rho] +

                                                          4 e
>         delm[mu, rho] s2[2 k, nu] s2[2 k, sigma])) / k    +

>    (c[5] (delm[rho, sigma] s2[2 k, mu] s2[2 k, nu] +

                                                          4 e
>         delm[mu, nu] s2[2 k, rho] s2[2 k, sigma])) / k

In[5]:= (* Let us compute the integral over k of: Exp[I k x] k^(-b) *)

In[6]:= Integrate[Exp[-l k^2] l^(b/2 -1) /Gamma[b/2], {l,0,Infinity}, Assumptions -> k>0]

                                -b
Out[6]= ConditionalExpression[k  , Re[b] > 0]

In[7]:= int[l] Exp[I k x] Exp[-l k^2] l^(b/2 -1) /Gamma[b/2]

            2
         -(k  l) + I k x  -1 + b/2
        E                l         int[l]
```

```
Out[7]= ---------------------------------
                          b
                  Gamma[-]
                          2

In[8]:= Integrate[Exp[I k x - l k^2] / (2Pi), {k, -Infinity, Infinity}]

                                                1
Out[8]= ConditionalExpression[----------------------------, Re[l] > 0]
                                     2
                                    x /(4 l)
                                2 E         Sqrt[l] Sqrt[Pi]

In[9]:= %[[1]]

                      1
Out[9]= ----------------------------
                 2
                x /(4 l)
             2 E         Sqrt[l] Sqrt[Pi]

In[10]:= %^d // PowerExpand

                       1
Out[10]= ----------------------------
                    2
           d   (d x )/(4 l)   d/2    d/2
          2  E              l     Pi

In[11]:= % /. d x^2 -> x^2

                     1
Out[11]= ------------------------
                 2
           d   x /(4 l)   d/2    d/2
          2  E          l     Pi

In[12]:= Integrate[% l^(b/2 -1) /Gamma[b/2], {l,0,Infinity}, Assumptions -> {x^2 >0}]

                                       b - d        -b + d
                                  Abs[x]       Gamma[------]
                                                      2
Out[12]= ConditionalExpression[-------------------------, Re[b] < Re[d]]
                                       b    d/2        b
                                    2  Pi     Gamma[-]
                                                      2

In[13]:= %[[1]] /. d -> 4-2e
```

```
                (-4 + 2 e)/2       -4 + b + 2 e       4 - b - 2 e
              Pi             Abs[x]             Gamma[-----------]
                                                          2
Out[13]= ------------------------------------------------------
                                  b       b
                                 2  Gamma[-]
                                         2

In[14]:= ExpandAll[% /. Abs[x] -> x]

                -2 + e  -4 + b + 2 e           b
              Pi       x             Gamma[2 - - - e]
                                           2
Out[14]= ---------------------------------------
                           b       b
                          2  Gamma[-]
                                  2

In[15]:= (% /. b -> #)& /@ {-2+4e, 4e, 2+4e}

                 2 - 4 e   -2 + e  -6 + 6 e           2 - 4 e
               2        Pi       x         Gamma[2 + ------- - e]
                                                        2
Out[15]= {-------------------------------------------------,
                                  -2 + 4 e
                              Gamma[--------]
                                     2

             -2 + e  -4 + 6 e
           Pi       x         Gamma[2 - 3 e]
>          ---------------------------------,
                    4 e
                   2    Gamma[2 e]

            -2 - 4 e   -2 + e  -2 + 6 e           -2 - 4 e
           2         Pi       x         Gamma[2 + -------- - e]
                                                     2
>          ------------------------------------------------------}
                                  2 + 4 e
                              Gamma[-------]
                                     2

In[16]:= ExpandAll[%]

                 2 - 4 e   -2 + e  -6 + 6 e
               2        Pi       x         Gamma[3 - 3 e]
Out[16]= {-----------------------------------------,
                           Gamma[-1 + 2 e]
```

```
       -2 + e  -4 + 6 e
     Pi       x         Gamma[2 - 3 e]
>    ---------------------------------,
               4 e
              2    Gamma[2 e]

      -2 - 4 e   -2 + e  -2 + 6 e
     2         Pi       x         Gamma[1 - 3 e]
>    ----------------------------------------}
                  Gamma[1 + 2 e]

In[17]:= % /. x^a_ :> xsq^Expand[a/2]

           2 - 4 e   -2 + e    -3 + 3 e
          2        Pi       xsq         Gamma[3 - 3 e]
Out[17]= {----------------------------------------,
                       Gamma[-1 + 2 e]

       -2 + e    -2 + 3 e
     Pi       xsq         Gamma[2 - 3 e]
>    ---------------------------------,
               4 e
              2    Gamma[2 e]

      -2 - 4 e   -2 + e    -1 + 3 e
     2         Pi       xsq         Gamma[1 - 3 e]
>    ----------------------------------------}
                  Gamma[1 + 2 e]

In[18]:= derivrules = {deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a,
deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b],
deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c],
deriv[a_?((FreeQ[#,xsq] && FreeQ[#,x])&),b_] ->0,
deriv[xsq, b_] :> 2 s2[2x,b],
deriv[s2[2x,a_],b_] :> delm[a,b]};

In[19]:= (* Let us compute s2[2 k, mu] s2[2 k, nu] Exp[I k (x-y)] k^(-4e) *)

In[20]:= deriv[deriv[%17[[2]],mu],nu] / im^2

                      -2 + e    -2 + 3 e
                    Pi       xsq         Gamma[2 - 3 e]
         deriv[deriv[---------------------------------, mu], nu]
                              4 e
                             2    Gamma[2 e]
Out[20]= ------------------------------------------------------
                                   2
                                  im
```

```
In[21]:= % //. derivrules

                        -2 + e
Out[21]= ((-2 + 3 e) Pi       Gamma[2 - 3 e]

             -3 + 3 e
>      (2 xsq         delm[mu, nu] +

                          -4 + 3 e
>        4 (-3 + 3 e) xsq         s2[2 x, mu] s2[2 x, nu])) /

       4 e   2
>    (2    im  Gamma[2 e])

In[22]:= (* Let us compute s2[2 k, mu] s2[2 k, nu] s2[2 k, rho] s2[2 k, sigma] Exp[I k (x-y)]
k^(-2-4e) *)

In[23]:= deriv[deriv[deriv[deriv[%17[[3]],mu],nu],rho],sigma] / im^4

Out[23]= deriv[deriv[deriv[deriv[

           -2 - 4 e   -2 + e    -1 + 3 e
          2         Pi       xsq         Gamma[1 - 3 e]
>         ----------------------------------------------, mu], nu], rho],
                       Gamma[1 + 2 e]

                 4
>     sigma] / im

In[24]:= % //. derivrules

           -2 - 4 e              -2 + e
Out[24]= (2         (-1 + 3 e) Pi       Gamma[1 - 3 e]

>      (2 (-2 + 3 e) delm[mu, sigma]

                -3 + 3 e
>         (2 xsq         delm[nu, rho] +

                            -4 + 3 e
>           4 (-3 + 3 e) xsq         s2[2 x, nu] s2[2 x, rho]) +

>        2 (-2 + 3 e) delm[mu, rho]

                -3 + 3 e
>         (2 xsq         delm[nu, sigma] +

                            -4 + 3 e
>           4 (-3 + 3 e) xsq         s2[2 x, nu] s2[2 x, sigma]) +
```

```
>        2 (-2 + 3 e) delm[mu, nu]

                -3 + 3 e
>         (2 xsq         delm[rho, sigma] +

                            -4 + 3 e
>           4 (-3 + 3 e) xsq         s2[2 x, rho] s2[2 x, sigma]) +

>        2 (-2 + 3 e) s2[2 x, mu]

                           -4 + 3 e
>         (4 (-3 + 3 e) xsq         delm[nu, sigma] s2[2 x, rho] +

                            -4 + 3 e
>           4 (-3 + 3 e) xsq         delm[nu, rho] s2[2 x, sigma] +

>           2 (-3 + 3 e) s2[2 x, nu]

                 -4 + 3 e
>            (2 xsq         delm[rho, sigma] +

                               -5 + 3 e
>              4 (-4 + 3 e) xsq         s2[2 x, rho] s2[2 x, sigma])))) /

        4
>    (im  Gamma[1 + 2 e])

In[25]:= Simplify[%]

                                    -2 + e    -5 + 3 e
Out[25]= ((-2 + 3 e) (-1 + 3 e) Pi       xsq         Gamma[1 - 3 e]

>      (xsq delm[mu, sigma] (xsq delm[nu, rho] +

>           6 (-1 + e) s2[2 x, nu] s2[2 x, rho]) +

>        xsq delm[mu, rho] (xsq delm[nu, sigma] +

>           6 (-1 + e) s2[2 x, nu] s2[2 x, sigma]) +

>        xsq delm[mu, nu] (xsq delm[rho, sigma] +

>           6 (-1 + e) s2[2 x, rho] s2[2 x, sigma]) +

>        6 (-1 + e) s2[2 x, mu]

>         (xsq delm[rho, sigma] s2[2 x, nu] +
```

```
>           xsq delm[nu, sigma] s2[2 x, rho] +

>           (xsq delm[nu, rho] + 2 (-4 + 3 e) s2[2 x, nu] s2[2 x, rho])

                                  2 e   4
>            s2[2 x, sigma])))) / (4    im  Gamma[1 + 2 e])

In[26]:= << ~/latt_pert/math/inputF.m
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[27]:= collect[#, delm]& /@ %%

                                    -2 + e    -5 + 3 e
Out[27]= ((-2 + 3 e) (-1 + 3 e) Pi       xsq         Gamma[1 - 3 e]

>       (6 (-1 + e) xsq delm[rho, sigma] s2[2 x, mu] s2[2 x, nu] +

>         6 (-1 + e) xsq delm[nu, sigma] s2[2 x, mu] s2[2 x, rho] +

                              2
>         delm[mu, sigma] (xsq  delm[nu, rho] +

>            6 (-1 + e) xsq s2[2 x, nu] s2[2 x, rho]) +

>         6 (-1 + e) xsq delm[nu, rho] s2[2 x, mu] s2[2 x, sigma] +

>         12 (-1 + e) (-4 + 3 e) s2[2 x, mu] s2[2 x, nu] s2[2 x, rho]
```

```
>         s2[2 x, sigma] + delm[mu, rho]

             2
>         (xsq  delm[nu, sigma] + 6 (-1 + e) xsq s2[2 x, nu] s2[2 x, sigma])\

                              2
>         + delm[mu, nu] (xsq  delm[rho, sigma] +

>           6 (-1 + e) xsq s2[2 x, rho] s2[2 x, sigma])) /

       2 e   4
>    (4    im  Gamma[1 + 2 e])

In[28]:= %4 /. s2[2 k, mu] s2[2 k, nu] s2[2 k, rho] s2[2 k, sigma] -> % * k^(2+4e);

In[29]:= {s2[2k, mu_] s2[2k, nu_] -> %21 k^(4e) }

Out[29]= {s2[2 k, mu_] s2[2 k, nu_] ->

                   4 e   -2 + e
>     ((-2 + 3 e) k    Pi       Gamma[2 - 3 e]

               -3 + 3 e
>        (2 xsq         delm[mu, nu] +

                           -4 + 3 e
>          4 (-3 + 3 e) xsq         s2[2 x, mu] s2[2 x, nu])) /

         4 e   2
>      (2    im  Gamma[2 e])}

In[30]:= %% /. % // Expand;

In[31]:= If[FreeQ[#,k], #, # %17[[1]] k^(-2+4e)]& /@ % // Simplify;

In[33]:= out31 = %31;

In[34]:= Save["diagram1.m5", out31]
```

```
In[1]:= << ~/latt_pert/math/inputF.m
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[2]:= << diagram1.m3;

In[3]:= diamondsEyeEyeglasses;

In[4]:= Variables[%]

                                          e    2 e
Out[4]= {e, I1, I2, I3, I4, I5, I6, I7, k, k , (k ) , nDim, g[1, 1, 0, 0],

>    g[1, 1, 1, 0], g[1, 2, 0, 0], g[2, 1 + e, 0, 0], g[1 + e, 1, 0, 0],

>    g[1 + e, 1, 1, 0], g[1 + e, 1, 2, 0], g[1 + e, 1, 2, 1],

>    g[1 + e, 1, 3, 0], g[1 + e, 1, 3, 1]}

In[5]:= Expand[%%];

In[6]:= ((% - (% /. # -> 0)) /. # -> 1)& /@ {I1, I2, I3, I4, I5, I6, I7};

In[7]:= SameQ[Expand[Inner[Times,%, {I1, I2, I3, I4, I5, I6, I7}, Plus]], %%]

Out[7]= True

In[8]:= {{1, 1,   1,   1,   2,    2,    2},
```

```
{1, d^2, d,   d,   2d,   2,    2},
{1, d,   d^2, d,   2,    2,    2d},
{1, d,   d,   d^2, 2,    2d,   2},
{2, 2d,  2,   2,   2d+2, 4,    4},
{2, 2,   2,   2d,  4,    2d+2, 4},
{2, 2,   2d,  2,   4,    4,    2d+2}};

In[9]:= Inverse[%8] . %6;

In[10]:= PowerExpand /@ %;

In[11]:= % /. k^a_ :> k^Expand[a];

In[12]:= % /. nDim -> d /. k^(2-4e) -> 1 ;

In[13]:= Length[%]

Out[13]= 7

In[14]:= Table[c[i] -> %%[[i]], {i, 7}];

In[15]:= listCs = %;

In[16]:= Save["diagram1.m6", listCs]

In[17]:= Simplify /@ %%%;

In[18]:= % /. d -> 4-2e /. {g[a_,b_,n_,s_] :> (4Pi)^e Gamma[a+b-s-2+e]/Gamma[a]/Gamma[b]
Beta[2-e-a+n-s,2-e-b+s]};

In[19]:= Simplify /@ %;

In[20]:= (#[[2]])& /@ % ;

In[21]:= Series[#,{e,0,0}]& /@ % ;

In[22]:= Normal /@ %;

In[23]:= Variables[%]

Out[23]= {e}

In[24]:= Simplify[%%];

In[25]:= FullSimplify /@ %;


In[26]:= % /. Log[a_ Pi]:> Log[a] + Log[Pi] /. Log[Pi] :> (log16pisq - 4Log[2])/2 //
          Simplify;
```

```
In[27]:= % //. Log[a_?((NumberQ[#] && (#>2))&)] :> Log[2]+ Log[a/2];

In[28]:= % /. EulerGamma -> EulerMinusLog4Pi + log16pisq / 2 // Simplify;

In[29]:= Table[c[i] -> %[[i]], {i, 7}]

                -11 + 48 Zeta[3]
Out[29]= {c  -> ----------------,
           1               4
                  61440 Pib

            -60 + e (-539 + 120 EulerMinusLog4Pi + 192 Zeta[3])
>    c  -> ---------------------------------------------------,
      2                                  4
                            1966080 e Pi

>    c  -> (-480 + 60 e (-51 + 16 EulerMinusLog4Pi) +
      3

          2                                                      2
>        e  (-12897 + 6120 EulerMinusLog4Pi - 960 EulerMinusLog4Pi  +

                2                                2   4
>           80 Pi  + 576 Zeta[3])) / (5898240 e  Pi ),

            -60 + e (-539 + 120 EulerMinusLog4Pi + 192 Zeta[3])
>    c  -> ---------------------------------------------------,
      4                                  4
                            1966080 e Pi

            140 + e (1137 - 280 EulerMinusLog4Pi - 576 Zeta[3])
>    c  -> ---------------------------------------------------,
      5                                 4
                            983040 e Pi

            -100 + e (-743 + 200 EulerMinusLog4Pi + 384 Zeta[3])
>    c  -> ----------------------------------------------------,
      6                                  4
                             983040 e Pi

>    c  -> (-240 + 60 e (-21 + 8 EulerMinusLog4Pi) +
      7

          2                                                    2        2
>        e  (-4989 + 2520 EulerMinusLog4Pi - 480 EulerMinusLog4Pi  + 40 Pi  +

                                     2   4
>           1152 Zeta[3])) / (2949120 e  Pi )}
```

```
In[30]:= listCsOrderE0 = %;

In[31]:= Save["diagram1.m6", listCsOrderE0]

In[32]:= << diagram1.m5;

In[34]:= Variables[out31]

                          e
Out[34]= {e, im, xsq, xsq , c , c , c , c , c , c , c , delm[mu, nu],
                            1   2   3   4   5   6   7

>    delm[mu, rho], delm[mu, sigma], delm[nu, rho], delm[nu, sigma],

>    delm[rho, sigma], Gamma[1 - 3 e], Gamma[2 - 3 e], Gamma[3 - 3 e],

>    Gamma[2 e], Gamma[-1 + 2 e], Gamma[1 + 2 e], s2[2 x, mu], s2[2 x, nu],

>    s2[2 x, rho], s2[2 x, sigma]}

In[35]:= out31 Mu^(2e) /. rho -> rho[3] /. mu -> rho[1] /. nu -> rho[2] /. sigma -> rho[4]
/. Mu -> mu;

In[36]:= d1 % gtrace[X, rho[1], rho[5], rho[2], Y, rho[3], rho[5], rho[4]] // Expand ;

In[37]:= Length[%]

Out[37]= 65

In[38]:= (# / (# /. im->1))& /@ List @@ %% // Union

               -4    -2
Out[38]= {1, im  , im  }

In[39]:= %%%  /. im^(-4) -> 1 /. im^(-2) -> -1 ;

In[40]:= (# / (# /. im->1))& /@ List @@ % // Union

Out[40]= {1}

In[41]:= << ../diagram2and3/diagram2_3.m ;

In[42]:= Expand[out152] /. mu^(2e) -> Mu^(2e);

In[43]:= Variables[%]

                             e          e
Out[43]= {d2, d3, e, im, Mu , xsq, xsq , delm[mu, nu], delm[mu, rho],
```

```
>    delm[mu, sigma], delm[nu, rho], delm[nu, sigma], delm[rho, sigma],

>    g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1], g[2 + e, 1, 4, 0],

>    g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2], Gamma[1 - 3 e], Gamma[2 - 3 e],

>    Gamma[3 - 3 e], Gamma[2 e], Gamma[-1 + 2 e], Gamma[1 + 2 e],

>    s2[2 x, mu], s2[2 x, nu], s2[2 x, rho], s2[2 x, sigma]}

In[44]:= %% /. rho -> rho[3] /. mu -> rho[1] /. nu -> rho[2] /. sigma -> rho[4] /. Mu ->
mu;

In[45]:= % /. d2 -> d2 gtrace[X, rho[1], rho[5], rho[2], rho[5], rho[3], Y, rho[4]] /.
             d3 -> d3 gtrace[X, rho[1], Y, rho[2], rho[5], rho[3], rho[5], rho[4]] /.
im^(-4) -> 1 /. im^(-2) -> -1 ;

In[47]:= (* We will now put all 3 diagrams together. Diagrams 2 & 3 are shown later in the
          appendices
         We must remember that d1 stands for:
             combinatorics:    (-1)
           2 vertices:     * (im g)^2
           4 propagators:  * (-im)^4
           color factor:   * (N^2-1)/2
           scale factor:   * (m0)^(4e)  [included so that the dimensionality of the
Green's function
                                        (before integration over spatial x) remains -6,
even in d-dimensions]
           (A factor of (2Pi)^(-d) for k-integration has already been included)
            We must also remember that both d2 and d3 stand for:
             combinatorics:    (-1)
             2 vertices:     * (im g)^2
           4 propagators:  * (-im)^4
           color factor:   * (N^2-1)/2
           scale factor:   * (m0)^(4e)  [included so that the dimensionality of the
Green's function
                                        (before integration over spatial x) remains -6,
even in d-dimensions]
           k-integration:  * (2Pi)^(-d),   *)

In[48]:= %39 + %45;

In[49]:= % // simplifydelm // reducerho;

In[51]:= %% /. gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] &&
FreeQ[#,Y])&),d_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),b_,e___]:> -
gtrace[a,d,e] delm[b,b] + 2 gtrace[a,b,e] delm[b,d] // Expand // simplifydelm // reducerho;
```

```
In[53]:= %% /. gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] &&
FreeQ[#,Y])&),d_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),b_,e___]:> -
gtrace[a,d,e] delm[b,b] + 2 gtrace[a,b,e] delm[b,d] // Expand // simplifydelm // reducerho;

In[55]:= %% /. gtrace[a___,b_?(FreeQ[#,gamma5]&),b_,d___]:> gtrace[a,d] delm[b,b] //
simplifydelm // reducerho;
{Length[%], Variables[%]}

In[56]:=
                                    e                 e
Out[56]= {399, {d1, d2, d3, e, mu , nDim, xsq, xsq , c , c , c , c , c , c ,
                                                    1   2   3   4   5   6

>     c , g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1],
       7

>     g[2 + e, 1, 4, 0], g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2],

>     Gamma[1 - 3 e], Gamma[2 - 3 e], Gamma[3 - 3 e], Gamma[2 e],

>     Gamma[-1 + 2 e], Gamma[1 + 2 e], gtrace[X, rho , Y, rho ],
                                                    1        1

>     gtrace[X, rho , Y, rho ], gtrace[X, rho , Y, rho , rho , rho ],
                 1        2                1        1     2     3

>     gtrace[X, rho , Y, rho , rho , rho ],
                 1        2     1     3

>     gtrace[X, rho , Y, rho , rho , rho ],
                 1        2     3     1

>     gtrace[X, rho , Y, rho , rho , rho ],
                 1        2     3     4

>     gtrace[X, rho , rho , rho , Y, rho ],
                 1     2     3        1

>     gtrace[X, rho , rho , rho , Y, rho ],
                 1     2     3        2

>     gtrace[X, rho , rho , rho , Y, rho ],
                 1     2     3        3

>     gtrace[X, rho , rho , rho , Y, rho ],
                 1     2     3        4

>     gtrace[X, rho , rho , rho , Y, rho ],
                 1     3     2        3
```

```
>     gtrace[X, rho , Y, rho , rho , rho ],
                3        1     3     2

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     3     2        1     3     2

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     3     2        2     3     1

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        1     4     3

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        2     4     3

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        3     4     1

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        3     4     2

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ], s2[2 x, rho ],
                1     5     2        3     5     4              1

>     s2[2 x, rho ], s2[2 x, rho ], s2[2 x, rho ]}}
               2              3              4

In[57]:= %% /. (c_ s2[2x, a_] s2[2x,b_] gtrace[d___, a_, b_, e___] :> c xsq gtrace[d,e]
/; (FreeQ[c,a] && FreeQ[c,b])) // reducerho;
{Length[%], Variables[%]}

In[58]:=
                                    e                 e
Out[58]= {347, {d1, d2, d3, e, mu , nDim, xsq, xsq , c , c , c , c , c , c ,
                                                      1   2   3   4   5   6

>     c , g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1],
      7

>     g[2 + e, 1, 4, 0], g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2],

>     Gamma[1 - 3 e], Gamma[2 - 3 e], Gamma[3 - 3 e], Gamma[2 e],

>     Gamma[-1 + 2 e], Gamma[1 + 2 e], gtrace[X, rho , Y, rho ],
                                                     1        1

>     gtrace[X, rho , Y, rho ], gtrace[X, rho , Y, rho , rho , rho ],
                1        2                1        2     1     3
```

```
>     gtrace[X, rho , rho , rho , Y, rho ],
                1     2     3        2

>     gtrace[X, rho , rho , rho , Y, rho ],
                1     3     2        3

>     gtrace[X, rho , Y, rho , rho , rho ],
                3        1     3     2

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     3     2        1     3     2

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     3     2        2     3     1

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        1     4     3

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        2     4     3

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        3     4     1

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ],
                1     4     2        3     4     2

>     gtrace[X, rho , rho , rho , Y, rho , rho , rho ], s2[2 x, rho ],
                1     5     2        3     5     4              1

>     s2[2 x, rho ], s2[2 x, rho ], s2[2 x, rho ]}}
               2              3              4

In[59]:= out57 = %57;

In[60]:= Save["diagram1.m6", out57]
```

```
In[1]:= << ~/latt_pert/math/inputF.m
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[2]:= << diagram1.m6;

In[3]:= out57;

In[5]:= lb[%3]

Out[5]= {347, 742760}

In[6]:= {{{},{}},
          {{},{gamma5}},
       {{},{nu[1]}},
       {{},{gamma5,nu[1]}},
       {{},{nu[1],nu[2]}},
       {{gamma5},{gamma5}},
       {{gamma5},{nu[1]}},
       {{gamma5},{gamma5,nu[1]}},
       {{gamma5},{nu[1],nu[2]}},
       {{nu[1]},{nu[1]}},
       {{nu[1]},{nu[2]}},
       {{nu[1]},{gamma5,nu[1]}},
       {{nu[1]},{gamma5,nu[2]}},
       {{nu[1]},{nu[1],nu[2]}},
       {{nu[1]},{nu[2],nu[3]}},
       {{gamma5,nu[1]},{gamma5,nu[1]}},
```

```
      {{gamma5,nu[1]},{gamma5,nu[2]}},
      {{gamma5,nu[1]},{nu[1],nu[2]}},
      {{gamma5,nu[1]},{nu[2],nu[3]}},
      {{nu[1],nu[2]},{nu[1],nu[2]}},
      {{nu[1],nu[2]},{nu[1],nu[3]}},
      {{nu[1],nu[2]},{nu[3],nu[4]}}};

In[7]:= (* We will use NU, rather than nu, for the tags, so that simplifydelm will not change
them *)

In[8]:= Sum[(tag @@ %%[[i]] /. nu -> NU) (%3 /. X -> Sequence @@ %%[[i,1]] /. Y -> Sequence
@@ %%[[i,2]]), {i, Length[%%]}];

In[9]:= Expand[%];

In[10]:= lb[%]

Out[10]= {7634, 19612712}

In[13]:= gtracerulesD :=
{gtrace[]:> 4,
gtrace[a__?(FreeQ[#,gamma5]&),gamma5,b___] :> gtrace[gamma5,b,a],
gtrace[a___,gamma5,gamma5,b___] :> gtrace[a,b],
gtrace[gamma5, nu[_]] :> 0,
gtrace[gamma5, a___, gamma5, nu[i_]] :> - gtrace[a,nu[i]],
gtrace[a___,gamma5, nu[i_], gamma5, b___] :> - gtrace[a,nu[i],b],
gtrace[a___,gamma5, nu[i_], nu[j_], gamma5, b___] :>  gtrace[a,nu[i],nu[j],b],
gtrace[a___,nu[i_],gamma5,nu[i_],b___] :> - gtrace[a,gamma5,b],
gtrace[a___,rho[i_],rho[j_],b___] s2[2x,rho[i_]] s2[2x,rho[j_]] :> gtrace[a,b] xsq,
gtrace[a___] :> 0 /; (FreeQ[{a},gamma5] && FreeQ[{a},X] && FreeQ[{a},Y] &&
OddQ[Length[{a}]]),
gtrace[c_?((FreeQ[#,X] && FreeQ[#,Y])&)] :> 0,
gtrace[a___,b_?(FreeQ[#,gamma5]&),b_,d___]:> gtrace[a,d] delm[b,b],
gtrace[a___, rho[i_], gamma5, rho[i_], c___] :> factor[-4 + 2 e - 4 e hv] gtrace[a, gamma5,
c],
gtrace[a___, rho[i_], gamma5, nu[j_], rho[i_], c___] :>   factor[2 - 2 e + 4 e hv] gtrace[a,
gamma5, nu[j], c],
gtrace[a___, rho[i_], nu[j_], gamma5, rho[i_], c___] :> - factor[2 - 2 e + 4 e hv] gtrace[a,
gamma5, nu[j], c],
gtrace[a___, rho[i_], gamma5, rho[j_], gamma5, rho[i_], c___] :> factor[-2 + 2 e] gtrace[a,
gamma5, rho[j], gamma5, c],
gtrace[gamma5,a___]:>0 /; (FreeQ[{a},gamma5] && FreeQ[{a},X] && FreeQ[{a},Y] &&
(Length[{a}]==2)),
gtrace[gamma5,a___,gamma5,b___] :> 0 /; (FreeQ[{a,b},gamma5] && FreeQ[{a,b},X] &&
FreeQ[{a,b},Y] && OddQ[Length[{a,b}]]),
gtrace[gamma5,a___,gamma5] :> gtrace[a],
gtrace[a_,b_]:> 4 delm[a,b] /; (FreeQ[{a,b},gamma5] && FreeQ[{a,b},X] && FreeQ[{a,b},Y])}

In[14]:= %9 //. gtracerulesD;
```

```
In[15]:= % // simplifydelm // reducerho;

In[16]:= lb[%]

Out[16]= {5709, 14755360}

In[18]:= %%% //. gtrace[gamma5,a___,rho[5],b___,gamma5,c___,d_,rho[5],e___] :> -
gtrace[gamma5,a,rho[5],b,gamma5,c,rho[5],d,e] + 2 delm[d, rho[5]]
gtrace[gamma5,a,rho[5],b,gamma5,c,e];

In[19]:= % // Expand // simplifydelm // reducerho;

In[20]:= % //. gtrace[gamma5,a___,rho[5],d_,b___,gamma5,rho[5],e___] :> -
gtrace[gamma5,a,d,rho[5],b,gamma5,rho[5],e] + 2 delm[d, rho[5]]
gtrace[gamma5,a,b,gamma5,rho[5],e];

In[21]:= % // Expand // simplifydelm // reducerho;

In[22]:= % //. gtracerulesD;

In[23]:= % // simplifydelm // reducerho;

In[24]:= lb[%]

Out[24]= {5739, 14813472}

In[26]:= %%% //. gtrace[gamma5,a___,rho[4],b___,gamma5,c___,d_,rho[4],e___] :> -
gtrace[gamma5,a,rho[4],b,gamma5,c,rho[4],d,e] + 2 delm[d, rho[4]]
gtrace[gamma5,a,rho[4],b,gamma5,c,e];

In[27]:= % // Expand // simplifydelm // reducerho;

In[28]:= % //. gtrace[gamma5,a___,rho[4],d_,b___,gamma5,rho[4],e___] :> -
gtrace[gamma5,a,d,rho[4],b,gamma5,rho[4],e] + 2 delm[d, rho[4]]
gtrace[gamma5,a,b,gamma5,rho[4],e];

In[29]:= % // Expand // simplifydelm // reducerho;

In[30]:= % //. gtracerulesD;

In[31]:= % // simplifydelm // reducerho;

In[32]:= lb[%]

Out[32]= {5875, 15147232}
```

```
In[34]:= %%% //. gtrace[gamma5,a___,rho[3],b___,gamma5,c___,d_,rho[3],e___] :> -
gtrace[gamma5,a,rho[3],b,gamma5,c,rho[3],d,e] + 2 delm[d, rho[3]]
gtrace[gamma5,a,rho[3],b,gamma5,c,e];

In[35]:= % // Expand // simplifydelm // reducerho;

In[36]:= % //. gtrace[gamma5,a___,rho[3],d_,b___,gamma5,rho[3],e___] :> -
gtrace[gamma5,a,d,rho[3],b,gamma5,rho[3],e] + 2 delm[d, rho[3]]
gtrace[gamma5,a,b,gamma5,rho[3],e];

In[37]:= % // Expand // simplifydelm // reducerho;

In[38]:= % //. gtracerulesD;

In[39]:= % // simplifydelm // reducerho;

In[40]:= lb[%]

Out[40]= {5974, 15341536}

In[42]:= %%% //. gtrace[gamma5,a___,rho[2],b___,gamma5,c___,d_,rho[2],e___] :> -
gtrace[gamma5,a,rho[2],b,gamma5,c,rho[2],d,e] + 2 delm[d, rho[2]]
gtrace[gamma5,a,rho[2],b,gamma5,c,e];

In[43]:= % // Expand // simplifydelm // reducerho;

In[44]:= % //. gtrace[gamma5,a___,rho[2],d_,b___,gamma5,rho[2],e___] :> -
gtrace[gamma5,a,d,rho[2],b,gamma5,rho[2],e] + 2 delm[d, rho[2]]
gtrace[gamma5,a,b,gamma5,rho[2],e];

In[45]:= % // Expand // simplifydelm // reducerho;

In[46]:= % //. gtracerulesD;

In[47]:= % // simplifydelm // reducerho;

In[48]:= lb[%]

Out[48]= {6065, 15558152}

In[50]:= %%% //. gtrace[gamma5,a___,rho[1],b___,gamma5,c___,d_,rho[1],e___] :> -
gtrace[gamma5,a,rho[1],b,gamma5,c,rho[1],d,e] + 2 delm[d, rho[1]]
gtrace[gamma5,a,rho[1],b,gamma5,c,e];

In[51]:= % // Expand // simplifydelm // reducerho;

In[52]:= % //. gtrace[gamma5,a___,rho[1],d_,b___,gamma5,rho[1],e___] :> -
gtrace[gamma5,a,d,rho[1],b,gamma5,rho[1],e] + 2 delm[d, rho[1]]
gtrace[gamma5,a,b,gamma5,rho[1],e];
```

```
In[53]:= % // Expand // simplifydelm // reducerho;

In[54]:= % //. gtracerulesD;

In[55]:= % // simplifydelm // reducerho;

In[56]:= lb[%]

Out[56]= {6118, 15614120}

In[58]:= %%% //. gtrace[gamma5,a___,nu[i_],b___,gamma5,c___,d_,nu[i_],e___] :> -
gtrace[gamma5,a,nu[i],b,gamma5,c,nu[i],d,e] + 2 delm[d, nu[i]]
gtrace[gamma5,a,nu[i],b,gamma5,c,e];

In[59]:= % // Expand // simplifydelm // reducerho;

In[60]:= % //. gtrace[gamma5,a___,nu[i_],d_,b___,gamma5,nu[i_],e___] :> -
gtrace[gamma5,a,d,nu[i],b,gamma5,nu[i],e] + 2 delm[d, nu[i]]
gtrace[gamma5,a,b,gamma5,nu[i],e];

In[61]:= % // Expand // simplifydelm // reducerho;

In[62]:= % //. gtracerulesD;

In[63]:= % // simplifydelm // reducerho;

In[64]:= lb[%]

Out[64]= {6272, 15925936}

In[66]:= %%% //. gtrace[gamma5,a___,nu[i_],b___,gamma5,c___,d_,nu[i_],e___] :> -
gtrace[gamma5,a,nu[i],b,gamma5,c,nu[i],d,e] + 2 delm[d, nu[i]]
gtrace[gamma5,a,nu[i],b,gamma5,c,e];

In[67]:= % // Expand // simplifydelm // reducerho;

In[68]:= % //. gtrace[gamma5,a___,nu[i_],d_,b___,gamma5,nu[i_],e___] :> -
gtrace[gamma5,a,d,nu[i],b,gamma5,nu[i],e] + 2 delm[d, nu[i]]
gtrace[gamma5,a,b,gamma5,nu[i],e];

In[69]:= % // Expand // simplifydelm // reducerho;

In[70]:= % //. gtracerulesD;

In[71]:= % // simplifydelm // reducerho;

In[72]:= lb[%]
```

```
Out[72]= {6264, 15906688}

In[73]:= Select[Variables[%%], (!FreeQ[#, gtrace[a___, gamma5, b___, gamma5, c___]])&]

Out[73]= {gtrace[gamma5, rho , gamma5, rho ],
                         1            2

>    gtrace[gamma5, nu , nu , rho , gamma5, rho ],
                    1    2     1            2

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ],
                    1     1           2     2

>    gtrace[gamma5, nu , rho , gamma5, rho , nu ],
                    1     1            2    2

>    gtrace[gamma5, nu , rho , nu , gamma5, rho ],
                    1     1    1            2

>    gtrace[gamma5, nu , rho , nu , gamma5, rho ],
                    1     1    2            2

>    gtrace[gamma5, rho , gamma5, rho , rho , rho ],
                     1            2     3     2

>    gtrace[gamma5, nu , rho , gamma5, rho , nu , rho , rho ]}
                    1     1            2    2     3     2

In[74]:= Select[Variables[%%%], (!FreeQ[#, delm])&]

Out[74]= {delm[nu , nu ], delm[nu , nu ]}
               1    1          1    2

In[75]:= %%%% /. delm[nu[1],nu[1]] -> 1 /. delm[nu[1],nu[2]] -> 0;

In[76]:= lb[%]

Out[76]= {5993, 15228928}

In[78]:= %%% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);

In[79]:= % // Expand // simplifydelm;

In[80]:= % //. gtracerulesD;

In[81]:= % // simplifydelm;
```

```
In[82]:= lb[%]

Out[82]= {8099, 20106152}

In[83]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[84]:=
In[85]:=
In[86]:=
In[87]:=
Out[87]= {8937, 22121720}

In[88]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[89]:=
In[90]:=
In[91]:=
In[92]:=
Out[92]= {9532, 23340256}

In[93]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[94]:=
In[95]:=
In[96]:=
In[97]:=
Out[97]= {9955, 24166000}

In[98]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
```

```
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[99]:=
In[100]:=
In[101]:=
In[102]:=
Out[102]= {10702, 25931416}

In[103]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[104]:=
In[105]:=
In[106]:=
In[107]:=
Out[107]= {10715, 25990792}

In[108]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[109]:=
In[110]:=
In[111]:=
In[112]:=
Out[112]= {10235, 24802088}

In[113]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[114]:=
In[115]:=
```

```
In[116]:=
In[117]:=
Out[117]= {9392, 22585152}

In[118]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[119]:=
In[120]:=
In[121]:=
In[122]:=
Out[122]= {8628, 20675904}

In[123]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[124]:=
In[125]:=
In[126]:=
In[127]:=
Out[127]= {8257, 19697368}

In[128]:= %% /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[129]:=
In[130]:=
In[131]:=
In[132]:=
Out[132]= {8257, 19697368}

In[134]:= %%% /. delm[a_,a_] -> 1 /. delm[a_,b__] :> 0 /.
gtrace[gamma5,nu[i_],nu[j_],nu[k_]] :> 0 /. gtrace[gamma5,nu[i_],nu[j_],rho[k_]] :> 0 /.
gtrace[nu[i_],nu[j_],nu[k_],nu[l_]] :> 0 /. gtrace[nu[i_],nu[j_],nu[k_],rho[l_]] :> 0;
```

```
In[135]:= % // reducerho;

In[136]:= lb[%]

Out[136]= {3254, 7200104}

In[137]:= Variables[%%]

                            e                 e
Out[137]= {d1, d2, d3, e, mu , nDim, xsq, xsq , c , c , c , c , c , c , c ,
                                             1   2   3   4   5   6   7

>    factor[-4 + 2 e - 4 e hv], factor[2 - 2 e + 4 e hv], g[1, 1, 1, 0],

>    g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1], g[2 + e, 1, 4, 0],

>    g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2], Gamma[1 - 3 e], Gamma[2 - 3 e],

>    Gamma[3 - 3 e], Gamma[2 e], Gamma[-1 + 2 e], Gamma[1 + 2 e],

>    gtrace[gamma5, rho ], gtrace[gamma5, rho , gamma5, rho ],
                      1                     1            2

>    gtrace[gamma5, nu , nu , rho , gamma5, rho ],
                    1    2     1             2

>    gtrace[gamma5, nu , rho , gamma5, nu , rho ], s2[2 x, nu ],
                    1     1           2     2             1

>    s2[2 x, nu ], s2[2 x, nu ], s2[2 x, rho ], s2[2 x, rho ], tag[{}, {}],
              2             3             1             2

>    tag[{gamma5}, {gamma5}], tag[{gamma5}, {NU[1]}], tag[{NU[1]}, {NU[1]}],

>    tag[{NU[1]}, {NU[2]}], tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    tag[{gamma5, NU[1]}, {gamma5, NU[2]}],

>    tag[{gamma5, NU[1]}, {NU[1], NU[2]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}], tag[{NU[1], NU[2]}, {NU[1], NU[3]}]}

In[138]:= (* in: gtrace[gamma5, nu[1],rho[1],gamma5,nu[2],rho[2]], the rho's must equal
either nu[1] or nu[2],
             since, otherwise, terms like: delm[nu[1],rho[1]]*delm[nu[2],rho[2]] will
vanish *)
```

```
In[139]:= %135 /. gtrace[gamma5, nu[1], rho[1], gamma5, nu[2], rho[2]] :>
gtrace[nu[1],rho[1],nu[2],rho[2]];

In[140]:= % /. gtrace[gamma5, nu[1], nu[2], rho[1], gamma5, rho[2]] :> -
gtrace[nu[1],nu[2],rho[1],rho[2]];

In[141]:= % /. (gtrace[a___,b_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),
c_?((FreeQ[#,gamma5] && FreeQ[#,X] && FreeQ[#,Y])&),d___] :>  -gtrace[a,c,b,d] + 2
delm[b,c] gtrace[a,d] /; !OrderedQ[{b,c}]);
% // Expand // simplifydelm;
% //. gtracerulesD;
% // simplifydelm;
lb[%]

In[142]:=
In[143]:=
In[144]:=
In[145]:=
Out[145]= {3417, 7486472}

In[146]:= Select[Variables[%%], MatchQ[#,delm[a_,b__]]&]

Out[146]= {delm[nu , nu ]}
                1    2

In[147]:= %%% /. delm[a_,b__] :> 0 ;

In[148]:= % // reducerho;

In[149]:= lb[%]

Out[149]= {3244, 7115264}

In[151]:= %%% /. nDim -> 4 - 2e /. factor[a_] :> a // Expand;

In[152]:= lb[%]

Out[152]= {3403, 7082768}

In[155]:= (# / (# /. tag[__] -> 1 /. s2[__] -> 1 /. gtrace[___] -> 1))& /@ List @@ %151
// Union

Out[155]= {tag[{}, {}], tag[{gamma5}, {gamma5}],

>    gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                      1            2            1            2

>     tag[{gamma5}, {gamma5}], gtrace[gamma5, rho ] s2[2 x, nu ]
                                              1           1
```

```
>     s2[2 x, rho ] tag[{gamma5}, {NU[1]}], tag[{NU[1]}, {NU[1]}],
               1

                2
>    s2[2 x, nu ]  tag[{NU[1]}, {NU[1]}],
              1

>    s2[2 x, nu ] s2[2 x, nu ] tag[{NU[1]}, {NU[2]}],
              1            2

>    tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

                2
>    s2[2 x, nu ]  tag[{gamma5, NU[1]}, {gamma5, NU[1]}],
              1

>    gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                     1             2             1             2

>     tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    s2[2 x, nu ] s2[2 x, nu ] tag[{gamma5, NU[1]}, {gamma5, NU[2]}],
              1            2

>    gtrace[gamma5, rho ] s2[2 x, nu ] s2[2 x, rho ]
                     1            2             1

>     tag[{gamma5, NU[1]}, {NU[1], NU[2]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}],

                2
>    s2[2 x, nu ]  tag[{NU[1], NU[2]}, {NU[1], NU[2]}],
              1

                2
>    s2[2 x, nu ]  tag[{NU[1], NU[2]}, {NU[1], NU[2]}],
              2

>    s2[2 x, nu ] s2[2 x, nu ] tag[{NU[1], NU[2]}, {NU[1], NU[3]}]}
              2            3

In[156]:= (* If we integrate over D-1 dimensions, then s2[2x,rho[1]] s2[2x,rho[2]] will
vanish unless rho[1]=rho[2], thus:
   gtrace[gamma5,rho[1],gamma5,rho[2]] s2[2x,rho[1]] s2[2x,rho[2]] = (HV)  -4
(x[1]^2+x[2]^2+x[3]^2+x[4]^2 - sum_rest x[i]^2),
                                                                    = (NDR) -4 xsq
```

which, given the upcoming integration over the D-1 "spatial" variables, is equivalent to (xt = temporal component) :
HV: -4 (xt^2 - (D-7) x[1]^2) = -4 (xt^2 (1+((D-7)/(D-1))) -(D-7)/(D-1) xsq) = - 4 (xt^2 (2D-8)/(D-1) -(D-7)/(D-1) xsq)
NDR: -4 xsq
Thus, altogether, we may replace: gtrace[gamma5,rho[1],gamma5,rho[2]] s2[2x,rho[1]] s2[2x,rho[2]]
by: -4 (xsq + hv (xt^2 - xsq)(2D-8)/(D-1)) = - 4 xsq + 16 e hv (xt^2 - xsq)/(D-1)
[Indeed, for D=4, the above just becomes -4 xsq]
The expression: gtrace[gamma5, rho[1]] s2[2x,nu[i_]] s2[2x,rho[1]] = 0 because rho[1]=nu[i]
The expression: s2[2x,nu[i_]] s2[2x,nu[j_]] = 0 because at least one of these odd directions will be integrated over. *)

```
In[157]:= %151 /. gtrace[gamma5, rho[1]] s2[2x,nu[i_]] s2[2x,rho[1]] :> 0 /. s2[2x,nu[i_]] s2[2x,nu[j_]] :> 0;

In[158]:= lb[%]

Out[158]= {2745, 5509056}

In[159]:= Variables[%%]

                                  e          e
Out[159]= {d1, d2, d3, e, hv, mu , xsq, xsq , c , c , c , c , c , c , c ,
                                              1   2   3   4   5   6   7

>    g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1], g[2 + e, 1, 4, 0],

>    g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2], Gamma[1 - 3 e], Gamma[2 - 3 e],

>    Gamma[3 - 3 e], Gamma[2 e], Gamma[-1 + 2 e], Gamma[1 + 2 e],

>    gtrace[gamma5, rho , gamma5, rho ], s2[2 x, nu ], s2[2 x, nu ],
                        1            2             1             2

>    s2[2 x, rho ], s2[2 x, rho ], tag[{}, {}], tag[{gamma5}, {gamma5}],
               1             2

>    tag[{NU[1]}, {NU[1]}], tag[{gamma5, NU[1]}, {gamma5, NU[1]}],

>    tag[{NU[1], NU[2]}, {NU[1], NU[2]}]}

In[163]:= out151 = %151 /. NU -> nu;

In[164]:= out157 = %157 /. NU -> nu;

In[165]:= Save["diagram1.m7", out151, out157]
```

```
In[1]:= << ~/latt_pert/math/inputF.m
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[2]:= << diagram1.m7;

In[3]:= out151;

In[4]:= lb[%]

Out[4]= {3382, 7056088}

In[5]:= Variables[%%]

                                 e          e
Out[5]= {d1, d2, d3, e, hv, mu , xsq, xsq , c , c , c , c , c , c , c ,
                                              1   2   3   4   5   6   7

>    g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1], g[2 + e, 1, 4, 0],

>    g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2], Gamma[1 - 3 e], Gamma[2 - 3 e],

>    Gamma[3 - 3 e], Gamma[2 e], Gamma[-1 + 2 e], Gamma[1 + 2 e],

>    gtrace[gamma5, rho ], gtrace[gamma5, rho , gamma5, rho ], s2[2 x, nu ],
                       1                     1            2            1

>    s2[2 x, nu ], s2[2 x, nu ], s2[2 x, rho ], s2[2 x, rho ], tag[{}, {}],
```

```
              2             3              1              2

>    tag[{gamma5}, {gamma5}], tag[{gamma5}, {nu }], tag[{nu }, {nu }],
                                                1           1      1

>    tag[{nu }, {nu }], tag[{gamma5, nu }, {gamma5, nu }],
           1      2                   1              1

>    tag[{gamma5, nu }, {gamma5, nu }], tag[{gamma5, nu }, {nu , nu }],
                  1              2                   1      1    2

>    tag[{nu , nu }, {nu , nu }], tag[{nu , nu }, {nu , nu }]}
           1    2      1    2           1    2      1    3

In[6]:= (* In this documentation we will not integrate over spatial x.
           Rather, we will first examine the poles of the 1-loop Green's function for
arbitrary values
           of the vector x_mu, in order to extract the Z^{DR,MSbar} factors.
           Once we do this, we will construct the MSbar-renormalized Green's functions,
for arbitrary
           x_mu, and take the limit e -> 0.
           Only at that point will we take the 3-dimensional (rather than (d-1)-dimensional)
integral over
           spatial x's, in order to find the GIRS conversion factors. *)

In[7]:= %3 /. gtrace[gamma5, rho[_]] -> 0;

In[8]:= lb[%]

Out[8]= {3128, 6435696}

In[10]:= Simplify[%%%] // Expand ;

In[11]:= lb[%]

Out[11]= {1077, 2206832}

In[13]:= (%10 /. d1 -> 0 /. d2 -> 0 /. d3 -> 1) - (%10 /. d1 -> 0 /. d2 -> 1 /. d3 -> 0) ;

In[14]:= lb[%]

Out[14]= {0, 16}

In[15]:= (* As expected, the contributions of d2 and d3 are identical *)

In[16]:= %10 /. d3 -> d2;

In[17]:= lb[%]
```

```
Out[17]= {740, 1466096}

In[19]:= (* Let us multiply by all factors needed for the complete Green's function.
            We must remember that d1 stands for:
              combinatorics:    (-1)
           2 vertices:     * (im g)^2
           4 propagators:  * (-im)^4
           color factor:   * (N^2-1)/2
           scale factor:   * (m0)^(4e)  [included so that the dimensionality of the
Green's function
                                        (before integration over spatial x) remains -6,
even in d-dimensions]
           (A factor of (2Pi)^(-d) for k-integration has already been included)
            We must also remember that both d2 and d3 stand for:
              combinatorics:    (-1)
              2 vertices:     * (im g)^2
           4 propagators:  * (-im)^4
           color factor:   * (N^2-1)/2
           scale factor:   * (m0)^(4e)  [included so that the dimensionality of the
Green's function
                                        (before integration over spatial x) remains -6,
even in d-dimensions]
           k-integration:  * (2Pi)^(-d)                      *)

In[20]:= %16 /. d1 -> (d1 g^2 n2m1 / 2) m0^(4e) /. d2 -> (d2 g^2 n2m1 / 2 m0^(4e) (2
Pi)^(-4+2e));

In[21]:= (* Let us make a Laurent series of the 1-loop result, %20, and concentrate on terms
of O(1/e), in order
            to derive the renormalization coefficients Z^{DR,MSbar} *)

In[22]:= << diagram1.m6;

In[23]:= listCsOrderE0

                 -11 + 48 Zeta[3]
Out[23]= {c  -> ----------------,
           1               4
                  61440 Pi

           -60 + e (-539 + 120 EulerMinusLog4Pi + 192 Zeta[3])
>    c  -> ---------------------------------------------------,
      2                                  4
                             1966080 e Pi

>    c  -> (-480 + 60 e (-51 + 16 EulerMinusLog4Pi) +
      3

          2                                                    2
```

```
>        e  (-12897 + 6120 EulerMinusLog4Pi - 960 EulerMinusLog4Pi  +

                2                              2   4
>          80 Pi  + 576 Zeta[3])) / (5898240 e  Pi ),

           -60 + e (-539 + 120 EulerMinusLog4Pi + 192 Zeta[3])
>    c  -> ---------------------------------------------------,
      4                                4
                            1966080 e Pi

           140 + e (1137 - 280 EulerMinusLog4Pi - 576 Zeta[3])
>    c  -> ---------------------------------------------------,
      5                                4
                            983040 e Pi

           -100 + e (-743 + 200 EulerMinusLog4Pi + 384 Zeta[3])
>    c  -> ----------------------------------------------------,
      6                                4
                             983040 e Pi

>    c  -> (-240 + 60 e (-21 + 8 EulerMinusLog4Pi) +
      7

          2                                                      2        2
>        e  (-4989 + 2520 EulerMinusLog4Pi - 480 EulerMinusLog4Pi  + 40 Pi  +

                                      2   4
>          1152 Zeta[3])) / (2949120 e  Pi )}

In[24]:= %20 /. % ;

In[26]:= %% /. {g[a_,b_,n_,s_] :> (4Pi)^e Gamma[a+b-s-2+e]/Gamma[a]/Gamma[b]
Beta[2-e-a+n-s,2-e-b+s]} // Expand ;

In[27]:= lb[%]

Out[27]= {1635, 3440744}

In[29]:= (# / (# /. mu -> 1 /. m0 -> 1 /. xsq -> 1 /. s2[__] -> 1 /. tag[_]->1))& /@ List
@@ %%% // Union

            4 e   2 e    -3 + 3 e    4 e   2 e    -4 + 3 e             2
Out[29]= {m0    mu    xsq        , m0    mu    xsq         s2[2 x, nu ] ,
                                                                    1

       4 e   2 e    -4 + 3 e
>    m0    mu    xsq         s2[2 x, nu ] s2[2 x, nu ],
                                  1            2
```

```
       4 e   2 e    -4 + 3 e             2
>    m0    mu    xsq          s2[2 x, nu ] ,
                                    2

       4 e   2 e    -4 + 3 e
>    m0    mu    xsq          s2[2 x, nu ] s2[2 x, nu ],
                                    2            3

       4 e   2 e    -4 + 3 e
>    m0    mu    xsq          s2[2 x, rho ] s2[2 x, rho ]}
                                     1             2

In[30]:= %26 /. m0^(4e) mu^(2e) -> m04mu2xsq3^e xsq^(-3e); (* m04mu2xsq3 = m0^4 mu^2 xsq^3
*)

In[31]:= Simplify[%] // Expand;

In[32]:= lb[%]

Out[32]= {1170, 2151160}

In[34]:= Series[#,{e,0,0}]& /@ %31;

In[35]:= Normal[%];

In[36]:= Expand[%];

In[37]:= lb[%]

Out[37]= {124, 125104}

In[41]:= (* Let us verify that the above is indeed a multiple of the tree level Green's
function,
            and extract Z^{DR,MRbar} for each operator.
            By the way, it is natural (though not necessary) to use a vector x_mu whose
only nonzero
            components are in 4 dimensions. This means that:
              gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]] --->
-4 xsq   *)

In[42]:= (* The tree level Green's function is (we only need it to O(e^0) for the present
purpose: *)

In[46]:= m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-3 + 2*e)*Gamma[2 - e]^2*tag[{}, {}] +
 (m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
   gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]]*
   tag[{gamma5}, {gamma5}])/4 - m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-3 + 2*e)*
  Gamma[2 - e]^2*tag[{NU[1]}, {NU[1]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*tag[{NU[1]}, {NU[1]}] +
```

```
 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]*
  s2[2*x, nu[2]]*tag[{NU[1]}, {NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*
  tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +
 (m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
   gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]]*
   tag[{gamma5, NU[1]}, {gamma5, NU[1]}])/4 + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]*s2[2*x, nu[2]]*
  tag[{gamma5, NU[1]}, {gamma5, NU[2]}] - m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-3 + 2*e)*Gamma[2 - e]^2*tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +
 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*
  tag[{NU[1], NU[2]}, {NU[1], NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[2]]^2*
  tag[{NU[1], NU[2]}, {NU[1], NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[2]]*s2[2*x, nu[3]]*
  tag[{NU[1], NU[2]}, {NU[1], NU[3]}] /. NU -> nu /. e -> 0

        Nc tag[{}, {}]
Out[46]= -------------- + (Nc gtrace[gamma5, rho , gamma5, rho ]
           4    3                              1            2
          Pi  xsq

                                                                    4    4
>       s2[2 x, rho ] s2[2 x, rho ] tag[{gamma5}, {gamma5}]) / (4 Pi  xsq ) -
                  1            2

                                        2
    Nc tag[{nu }, {nu }]   2 Nc s2[2 x, nu ]  tag[{nu }, {nu }]
             1      1                  1          1      1
>   -------------------- + ------------------------------------ +
           4    3                        4    4
         Pi  xsq                      Pi  xsq

    2 Nc s2[2 x, nu ] s2[2 x, nu ] tag[{nu }, {nu }]
                  1           2         1      2
>   ------------------------------------------------ +
                       4    4
                     Pi  xsq

                     2
    2 Nc s2[2 x, nu ]  tag[{gamma5, nu }, {gamma5, nu }]
                  1                 1              1
>   ------------------------------------------------- +
                         4    4
                       Pi  xsq

>   (Nc gtrace[gamma5, rho , gamma5, rho ] s2[2 x, rho ] s2[2 x, rho ]
                        1            2            1            2
```

```
                                                    4    4
>       tag[{gamma5, nu }, {gamma5, nu }]) / (4 Pi  xsq ) +
                      1             1

     2 Nc s2[2 x, nu ] s2[2 x, nu ] tag[{gamma5, nu }, {gamma5, nu }]
                   1           2                 1             2
>    ------------------------------------------------------------- -
                                4    4
                               Pi  xsq

     Nc tag[{nu , nu }, {nu , nu }]
              1    2      1    2
>    ------------------------------ +
                4    3
               Pi  xsq

                     2
     2 Nc s2[2 x, nu ]  tag[{nu , nu }, {nu , nu }]
                   1          1    2      1    2
>    ------------------------------------------- +
                       4    4
                      Pi  xsq

                     2
     2 Nc s2[2 x, nu ]  tag[{nu , nu }, {nu , nu }]
                   2          1    2      1    2
>    ------------------------------------------- +
                       4    4
                      Pi  xsq




  2 Nc s2[2 x, nu ] s2[2 x, nu ] tag[{nu , nu }, {nu , nu }]
                   2           3         1    2      1    3
>    ------------------------------------------------------
                               4    4
                              Pi  xsq
```

In[48]:= (* For each Green's function, here is the ratio: (1-loop order 1/e)/(tree level) : *)

In[49]:= Table[{%%[[i]], (%40 /. %%[[i]] -> 1 /. tag[__]->0)/(%46 /. %%[[i]] -> 1 /. tag[__] -> 0)}, {i, Length[%%]}]

```
                            2                                       2
Out[49]= {{tag[{}, {}], 3 g  n2m1}, {tag[{gamma5}, {gamma5}], 3 g  n2m1},
                           ----------                              ----------
```

```
                                2                                       2
                          16 e Nc Pi                              16 e Nc Pi

>    {tag[{nu }, {nu }], 0}, {tag[{nu }, {nu }], 0},
            1      1                1      2

>    {tag[{gamma5, nu }, {gamma5, nu }], 0},
                   1             1

>    {tag[{gamma5, nu }, {gamma5, nu }], 0},
                   1             2

>    {tag[{nu , nu }, {nu , nu }],
            1    2      1    2

                     2                  2    2                  2
         2          g  n2m1 s2[2 x, nu ]    g  n2m1 s2[2 x, nu ]
        g  n2m1                         1                       2
      ------------ - --------------------- - ---------------------
            6    3             6    4                6    4
      16 e Pi  xsq        8 e Pi  xsq            8 e Pi  xsq
>     ------------------------------------------------------------},
                                      2                   2
                        2 Nc s2[2 x, nu ]   2 Nc s2[2 x, nu ]
             Nc                       1                   2
         -(--------) + ------------------ + ------------------
            4    3           4    4              4    4
           Pi  xsq          Pi  xsq            Pi  xsq

                                    2
                                -(g  n2m1)
>    {tag[{nu , nu }, {nu , nu }], ----------}}
            1    2      1    3             2
                              16 e Nc Pi
```

In[50]:= Simplify[%]

```
                            2                                       2
                     3 g  n2m1                               3 g  n2m1
Out[50]= {{tag[{}, {}], ----------}, {tag[{gamma5}, {gamma5}], ----------},
                            2                                       2
                       16 e Nc Pi                              16 e Nc Pi

>    {tag[{nu }, {nu }], 0}, {tag[{nu }, {nu }], 0},
            1      1                1      2

>    {tag[{gamma5, nu }, {gamma5, nu }], 0},
                   1             1
```

```
>    {tag[{gamma5, nu }, {gamma5, nu }], 0},
                  1             2

                                    2
                                 -(g  n2m1)
>    {tag[{nu , nu }, {nu , nu }], ----------},
          1    2      1    2            2
                                 16 e Nc Pi

                                    2
                                 -(g  n2m1)
>    {tag[{nu , nu }, {nu , nu }], ----------}}
          1    2      1    3            2
                                 16 e Nc Pi

In[51]:= (* From the above, we conclude that Z^{DR,MSbar} equals:
           (Note that the non-diagonal pole in the last line above is compatible
                with the     diagonal pole in the previous line)                  *)

In[53]:= (ZDRMS[#[[1,1]]] -> 1 - (1/2) #[[2]])& /@ %%%

                              2                                  2
                           3 g  n2m1                          3 g  n2m1
Out[53]= {ZDRMS[{}] -> 1 - ----------, ZDRMS[{gamma5}] -> 1 - ----------,
                                 2                                  2
                          32 e Nc Pi                         32 e Nc Pi

>    ZDRMS[{nu }] -> 1, ZDRMS[{nu }] -> 1, ZDRMS[{gamma5, nu }] -> 1,
            1                  1                          1

                                                      2
                                                     g  n2m1
>    ZDRMS[{gamma5, nu }] -> 1, ZDRMS[{nu , nu }] -> 1 + ----------,
                     1                  1    2                  2
                                                   32 e Nc Pi

                                 2
                                g  n2m1
>    ZDRMS[{nu , nu }] -> 1 + ----------}
            1    2                   2
                              32 e Nc Pi

In[54]:= Union[%]

                              2                                  2
                           3 g  n2m1                          3 g  n2m1
Out[54]= {ZDRMS[{}] -> 1 - ----------, ZDRMS[{gamma5}] -> 1 - ----------,
                                 2                                  2
                          32 e Nc Pi                         32 e Nc Pi
```

```
>    ZDRMS[{nu }] -> 1, ZDRMS[{gamma5, nu }] -> 1,
              1                            1

                                  2
                                 g  n2m1
>    ZDRMS[{nu , nu }] -> 1 + -----------}
              1    2                    2
                               32 e Nc Pi

In[61]:= ZDRMSlist = %54;

In[62]:= Save["diagram1.m10", ZDRMSlist]

In[63]:= (* Let us now construct the MSbar-renormalized Green's function, and take the limit
e -> 0.
            Only after that step will we consider the integral d^3 x, which is relevant
for GIRS          *)

In[64]:= (* Here is the tree-level Green's function again, this time in d dimensions: *)

In[65]:= m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-3 + 2*e)*Gamma[2 - e]^2*tag[{}, {}] +
 (m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
   gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]]*
   tag[{gamma5}, {gamma5}])/4 - m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-3 + 2*e)*
  Gamma[2 - e]^2*tag[{NU[1]}, {NU[1]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*tag[{NU[1]}, {NU[1]}] +
 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]*
  s2[2*x, nu[2]]*tag[{NU[1]}, {NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*
  tag[{gamma5, NU[1]}, {gamma5, NU[1]}] +
 (m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*
   gtrace[gamma5, rho[1], gamma5, rho[2]]*s2[2*x, rho[1]]*s2[2*x, rho[2]]*
   tag[{gamma5, NU[1]}, {gamma5, NU[1]}])/4 + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]*s2[2*x, nu[2]]*
  tag[{gamma5, NU[1]}, {gamma5, NU[2]}] - m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-3 + 2*e)*Gamma[2 - e]^2*tag[{NU[1], NU[2]}, {NU[1], NU[2]}] +
 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[1]]^2*
  tag[{NU[1], NU[2]}, {NU[1], NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[2]]^2*
  tag[{NU[1], NU[2]}, {NU[1], NU[2]}] + 2*m0^(4*e)*Nc*Pi^(-4 + 2*e)*
  xsq^(-4 + 2*e)*Gamma[2 - e]^2*s2[2*x, nu[2]]*s2[2*x, nu[3]]*
  tag[{NU[1], NU[2]}, {NU[1], NU[3]}] /. NU -> nu;

In[66]:= % /. tag[a_,b_] :> ZDRMS[a]^2 tag[a,b] /. ZDRMSlist;
```

```
In[68]:= Series[%%,{g,0,2}] // Normal // Expand;

In[70]:= GreensFunction1loopBare = %36;

In[72]:= Save["diagram1.m10", GreensFunction1loopBare]

In[73]:= (* Now add up Z^2*Tree + 1loop, in order to be able to take the limit e -> 0 *)

In[74]:= %68 + GreensFunction1loopBare /. d1 -> 1 /. d2 -> 1 // Expand;

In[75]:= lb[%]

Out[75]= {86, 96552}

In[77]:= Series[%%%,{e,0,0}] // Normal ;

In[78]:= lb[%]

Out[78]= {4, 68976}

In[79]:= Expand[%%];

In[80]:= lb[%]

Out[80]= {86, 84304}

In[82]:= (* There is no e among the variables. We now have the 4-d renormalized Green's
function. *)

In[83]:= %79 /. gtrace[gamma5, rho[1], gamma5, rho[2]] s2[2x, rho[1]] s2[2x, rho[2]] ->
-4 xsq ;

In[84]:= Variables[%]

Out[84]= {EulerMinusLog4Pi, g, hv, n2m1, Nc, xsq, Log[m0], Log[m04mu2xsq3],

>    Log[xsq], s2[2 x, nu ], s2[2 x, nu ], s2[2 x, nu ], tag[{}, {}],
                        1             2             3

>    tag[{gamma5}, {gamma5}], tag[{nu }, {nu }], tag[{nu }, {nu }],
                                  1      1           1      2

>    tag[{gamma5, nu }, {gamma5, nu }], tag[{gamma5, nu }, {gamma5, nu }],
                   1              1                   1              2

>    tag[{nu , nu }, {nu , nu }], tag[{nu , nu }, {nu , nu }]}
          1    2      1    2           1    2      1    3

In[85]:= GreensFunctionTreePlus1loopRenormalized = %%;
```

```
In[86]:= Save["diagram1.m10", GreensFunctionTreePlus1loopRenormalized]

In[90]:= Factor /@ collect[%83, tag[__]];

In[91]:= % /. Log[xsq] -> Log[m02xsq] - 2 Log[m0] /. Log[m04mu2xsq3] -> 2 Log[m02xsq] +
Log[mu2xsq] ;

In[92]:= Factor /@ %; (* The scale m0 is now gone. *)

In[93]:= lb[Expand[%]]

Out[93]= {74, 62848}

In[94]:= %% /. EulerMinusLog4Pi -> EulerGamma - 2 Log[2] - Log[Pi];

In[95]:= Factor /@ %;

In[96]:= % /. Log[mu2xsq] -> Log[mubar2xsq] + EulerGamma - 2 Log[2] - Log[Pi];

In[97]:= Factor /@ %;

In[98]:= (* Log[2] and EulerGamma don't seem to go away, but no Log[Pi] is left ...??? *)

In[99]:= GreensFunctionTreePlus1loopRenormalized = %%;

In[100]:= Save["diagram1.m10", GreensFunctionTreePlus1loopRenormalized]

In[101]:= (* Now that we have the renormalized Green's function in 4 dimension, we turn
to GIRS.
            Quantities such as s2[2x, a_] s2[2x, b_] will vanish upon integration, since
they are odd *)

In[102]:= %%%%% /. s2[2x,a_] s2[2x,b_] :> 0

            2                      2               2      2
Out[102]= ((g  n2m1 + 6 EulerGamma g  n2m1 + 16 Nc Pi  - 6 g  n2m1 Log[2] +

            2                                          6    3
>         3 g  n2m1 Log[mubar2xsq]) tag[{}, {}]) / (16 Pi  xsq ) -

        2                     2           2                   2
>    ((g  n2m1 + 6 EulerGamma g  n2m1 + 8 g  hv n2m1 + 16 Nc Pi  -

            2                  2
>         6 g  n2m1 Log[2] + 3 g  n2m1 Log[mubar2xsq]) tag[{gamma5}, {gamma5}]

                  6    3
>       ) / (16 Pi  xsq ) -
```

```
          2               2                       2
      (3 g  n2m1 + 32 Nc Pi ) (xsq - 2 s2[2 x, nu ] ) tag[{nu }, {nu }]
                                                 1           1      1
>     ------------------------------------------------------------------ -
                                      6    4
                                  32 Pi  xsq

          2           2                   2                       2
>    ((3 g  n2m1 + 8 g  hv n2m1 + 32 Nc Pi ) (xsq - 2 s2[2 x, nu ] )
                                                                  1

                                                    6    4
>       tag[{gamma5, nu }, {gamma5, nu }]) / (32 Pi  xsq ) -
                       1             1

          2                      2                2      2
>    ((3 g  n2m1 - 2 EulerGamma g  n2m1 + 16 Nc Pi  + 2 g  n2m1 Log[2] -

           2                                            2                2
>         g  n2m1 Log[mubar2xsq]) (xsq - 2 s2[2 x, nu ]  - 2 s2[2 x, nu ] )
                                                       1                2

                                          6    4
>       tag[{nu , nu }, {nu , nu }]) / (16 Pi  xsq )
              1    2      1    2
```

```
In[103]:= (* Since the integration over spatial components will depend on whether s2[x,_]
is temporal or not, we will
          write the expression as a sum of 8 different cases: S, P, Vs, Vt, As, At, Ts,
Tt   *)

In[106]:= Expand[%102];

In[107]:= Select[Variables[%], MatchQ[#,tag[__]]&]

Out[107]= {tag[{}, {}], tag[{gamma5}, {gamma5}], tag[{nu }, {nu }],
                                                        1      1

>    tag[{gamma5, nu }, {gamma5, nu }], tag[{nu , nu }, {nu , nu }]}
                  1             1          1    2      1    2

In[108]:= (# -> 1)& /@ %

Out[108]= {tag[{}, {}] -> 1, tag[{gamma5}, {gamma5}] -> 1,

>    tag[{nu }, {nu }] -> 1, tag[{gamma5, nu }, {gamma5, nu }] -> 1,
           1      1                        1             1
```

```
>    tag[{nu , nu }, {nu , nu }] -> 1}
            1    2       1    2

In[109]:= Length /@ ((%106 /. # /. tag[__]->0)& /@ %)

Out[109]= {5, 6, 4, 6, 15}

In[110]:= Plus @@ %

Out[110]= 36

In[111]:= tag[S] (%106 /. %108[[1]] /. tag[__] -> 0) + tag[P] (%106 /. %108[[2]] /. tag[__]
-> 0) + tag[Vt] (%106 /. %108[[3]] /. tag[__] -> 0) + tag[Vs] (%106 /. %108[[3]] /. tag[__]
-> 0) + tag[At] (%106 /. %108[[4]] /. tag[__] -> 0) + tag[As] (%106 /. %108[[4]] /. tag[__]
-> 0) + tag[Ts] (%106 /. %108[[5]] /. tag[__] -> 0) + tag[Tt] (%106 /. %108[[5]] /. tag[__]
-> 0) // Expand;

In[112]:= Select[Variables[%], MatchQ[#,tag[__]]&]

Out[112]= {tag[As], tag[At], tag[P], tag[S], tag[Ts], tag[Tt], tag[Vs],

>    tag[Vt]}

In[113]:= (# -> 1)& /@ %

Out[113]= {tag[As] -> 1, tag[At] -> 1, tag[P] -> 1, tag[S] -> 1,

>    tag[Ts] -> 1, tag[Tt] -> 1, tag[Vs] -> 1, tag[Vt] -> 1}

In[114]:= Length /@ ((%%% /. # /. tag[__]->0)& /@ %)

Out[114]= {6, 6, 6, 5, 15, 15, 4, 4}

In[115]:= Plus @@ %

Out[115]= 61

In[116]:= (* Let us now prepare the whole expression for integration of the spatial
directions of x, by
            expressing everything in terms of xsq and the temporal component xt  *)

In[117]:= (# / (# /. tag[__] -> 1 /. s2[__] -> 1))& /@ List @@ Expand[%111] // Union

                                2                                 2
Out[117]= {tag[As], s2[2 x, nu ]  tag[As], tag[At], s2[2 x, nu ]  tag[At],
                               1                                 1

                                       2                       2
>    tag[P], tag[S], tag[Ts], s2[2 x, nu ]  tag[Ts], s2[2 x, nu ]  tag[Ts],
```

```
                                        1                       2

                          2                         2
>    tag[Tt], s2[2 x, nu ]  tag[Tt], s2[2 x, nu ]  tag[Tt], tag[Vs],
                         1                         2

                 2                                 2
>    s2[2 x, nu ]  tag[Vs], tag[Vt], s2[2 x, nu ]  tag[Vt]}
               1                                 1

In[118]:= %111 /. s2[2x,nu[1]]^2 tag[As] :> (xsq - xt^2) /3 tag[As] /. s2[2x,nu[1]]^2
tag[At] :> xt^2 tag[At] /.
                  s2[2x,nu[1]]^2 tag[Vs] :> (xsq - xt^2) /3 tag[Vs] /. s2[2x,nu[1]]^2
tag[Vt] :> xt^2 tag[Vt] /.
                  s2[2x,nu[_]]^2 tag[Ts] :> (xsq - xt^2) /3 tag[Ts] /.
                  s2[2x,nu[2]]^2 tag[Tt] :> (xsq - xt^2) /3 tag[Tt] /. s2[2x,nu[1]]^2
tag[Tt] :> xt^2 tag[Tt] // Expand;

In[119]:= lb[%]

Out[119]= {51, 22328}

In[120]:= Variables[%%]

Out[120]= {g, hv, n2m1, Nc, xsq, xt, Log[mubar2xsq], tag[As], tag[At],

>    tag[P], tag[S], tag[Ts], tag[Tt], tag[Vs], tag[Vt]}

In[121]:= %%% /. Log[mubar2xsq] -> 2 Log[mubar] + Log[xsq] // Expand;

In[122]:= lb[%]

Out[122]= {57, 25336}

In[123]:= (# / (# /. tag[__]->1 /. Log[xsq]->1 /. xsq -> 1 /. s2[__]->1))& /@ List @@ %%
// Union;

           tag[As]  tag[As]  tag[At]  tag[At]  tag[P]  Log[xsq] tag[P]
Out[123]= {-------, -------, -------, -------, ------, ---------------,
               4        3        4        3        3           3
            xsq      xsq      xsq      xsq      xsq         xsq

     tag[S]  Log[xsq] tag[S]  tag[Ts]  tag[Ts]  Log[xsq] tag[Ts]
>    ------, ---------------, -------, -------, -----------------,
         3           3            4        3             4
       xsq         xsq          xsq      xsq           xsq

     Log[xsq] tag[Ts]  tag[Tt]  tag[Tt]  Log[xsq] tag[Tt]  Log[xsq] tag[Tt]
>    ----------------, -------, -------, -----------------, ----------------,
```

```
             3            4        3             4                  3
           xsq          xsq      xsq           xsq                xsq

     tag[Vs]  tag[Vs]  tag[Vt]  tag[Vt]
>    -------, -------, -------, -------}
         4        3        4        3
       xsq      xsq      xsq      xsq

In[124]:= % /. tag[__] -> 1 // Union

              -4     -3  Log[xsq]  Log[xsq]
Out[124]= {xsq  , xsq  , --------, --------}
                            4         3
                          xsq       xsq

In[125]:= Integrate[4Pi r^2 (# /. xsq -> r^2+xt^2),{r,0,Infinity}, Assumptions -> xt > 0]&
/@ %

                            2    5
             2      2   Pi  (-(-) + Log[8] + 3 Log[xt])
           Pi     Pi           4
Out[125]= {-----, -----, -------------------------------,
             5      3                   5
           8 xt   4 xt               12 xt

       2
     Pi  (-1 + Log[16] + 4 Log[xt])
>    ------------------------------}
                    3
                8 xt

In[126]:= Table[%%[[i]] -> %[[i]], {i, Length[%]}]

                      2              2
              -4     Pi       -3     Pi
Out[126]= {xsq   -> -----, xsq   -> -----,
                     5              3
                  8 xt           4 xt

                   2    5
                 Pi  (-(-) + Log[8] + 3 Log[xt])
     Log[xsq]           4
>    -------- -> -------------------------------,
         4                       5
       xsq                   12 xt

                   2
     Log[xsq]    Pi  (-1 + Log[16] + 4 Log[xt])
>    -------- -> ------------------------------}
```

```
          3                       3
       xsq                   8 xt

In[127]:= %121 /. % // Expand;

In[128]:= Variables[%]

Out[128]= {g, hv, n2m1, Nc, xt, Log[mubar], Log[xt], tag[As], tag[P], tag[S],

>    tag[Ts], tag[Tt], tag[Vs]}

In[129]:= %% /. Log[xt] -> Log[xtTIMESmubar] - Log[mubar] // Expand;

In[130]:= Variables[%]

Out[130]= {g, hv, n2m1, Nc, xt, Log[xtTIMESmubar], tag[As], tag[P], tag[S],

>    tag[Ts], tag[Tt], tag[Vs]}

In[131]:= lb[%%]

Out[131]= {32, 13888}

In[132]:= collect[%%%, tag];

In[133]:= FullSimplify[%] // Expand // collect[#,tag]&;

In[134]:= (* To implement nonperturbative renormalization in GIRS, one must multiply each
bare operator O^{LATT} by
            a renormalization coefficient Z^{LATT,GIRS}, which is evaluated
nonperturbatively by requiring that:
                              <int dx1 dx2 dx3 O^{GIRS }(x1,x2,x3,0)
O^{GIRS }(0,0,0,xt)> = (the same at tree level) --->
            (Z^{LATT,GIRS})^2 <int dx1 dx2 dx3 O^{LATT }(x1,x2,x3,0) O^{LATT }(0,0,0,xt)>
= (the same at tree level)
            The expectation value in the line above is evaluated in simulations, and is
manifestly gauge invariant.
            Given that:       <int dx1 dx2 dx3 O^{MSbar}(x1,x2,x3,0)
O^{MSbar}(0,0,0,xt)> = %133,
            there follows that: Z^{LATT,MSbar} = Z^{LATT,GIRS} * (%133/treelevel)^(1/2)
            In other words: %133/treelevel = (Z^{DR,MSbar}/Z^{DR,GIRS})^2 =
(Z^{LATT,MSbar}/Z^{LATT,GIRS})^2
            The conversion factor from GIRS to MSbar: C^{GIRS,MSbar} =
Z^{X,MSbar}/Z^{X,GIRS} = (%133/treelevel)^(1/2)  *)

In[135]:= List @@ collect[%%, tag]

              2          2
           -(g  n2m1)    g  hv n2m1      Nc
```

```
Out[135]= {(---------- - ---------- - ---------) tag[As],
                4   3        4   3       2   3
            64 Pi  xt    24 Pi  xt    6 Pi  xt

         2                        2         2
        g  n2m1     3 EulerGamma g  n2m1   g  hv n2m1      Nc
>    (----------- - -------------------- - ---------- - --------- -
              4   3             4   3           4   3        2   3
       128 Pi  xt         32 Pi  xt         8 Pi  xt     4 Pi  xt

           2
        3 g  n2m1 Log[xtTIMESmubar]
>       --------------------------) tag[P],
                    4   3
                32 Pi  xt

         2                        2
       -(g  n2m1)    3 EulerGamma g  n2m1      Nc
>    (----------- + -------------------- + --------- +
              4   3             4   3           2   3
       128 Pi  xt         32 Pi  xt         4 Pi  xt

           2
        3 g  n2m1 Log[xtTIMESmubar]
>       --------------------------) tag[S],
                    4   3
                32 Pi  xt

           2                     2
       -25 g  n2m1    EulerGamma g  n2m1       Nc
>    (------------ + ------------------ - ---------- +
             4   3            4   3           2   3
       1152 Pi  xt        96 Pi  xt        12 Pi  xt

         2
        g  n2m1 Log[xtTIMESmubar]
>       -------------------------) tag[Ts],
                  4   3
              96 Pi  xt

           2                     2
        25 g  n2m1    EulerGamma g  n2m1       Nc
>    (------------ - ------------------ + ---------- -
             4   3            4   3           2   3
       1152 Pi  xt        96 Pi  xt        12 Pi  xt

         2                                    2
        g  n2m1 Log[xtTIMESmubar]           -(g  n2m1)      Nc
>       -------------------------) tag[Tt], (---------- - ---------) tag[Vs]}
```

```
                  4   3                          4   3       2   3
              96 Pi  xt                     64 Pi  xt    6 Pi  xt
```

```
In[136]:= (# / (# /. tag[_] -> 1))& /@ %
```

```
Out[136]= {tag[As], tag[P], tag[S], tag[Ts], tag[Tt], tag[Vs]}
```

```
In[137]:= Table[{%[[i]], %%[[i]] / (%%[[i]] /. g->0)}, {i, Length[%]}];
```

```
In[138]:= ExpandAll[%];
```

```
In[139]:= (* Take square root: *)
```

```
In[140]:= %% /. 1+a_ :> 1 + a/2 // ExpandAll;
```

```
In[141]:= convGIRSMSlist = (convGIRSMS[#[[1]]] -> #[[2]])& /@ %;
```

```
In[143]:= convGIRSMSlist = (%% /. n2m1 -> CF 2Nc /. 1/Pi^2 -> 16/pisq16);
```

```
In[145]:= Map[collect[#,{CF, g, pisq16}]&, %%, 2]
```

```
                                        2  3
                                     CF g  (- + 4 hv)
                                           2
Out[145]= {convGIRSMS[tag[As]] -> 1 + ----------------,
                                         pisq16
```

```
>    convGIRSMS[tag[P]] ->

                2    1
            CF g  (-(-) + 6 EulerGamma + 8 hv + 6 Log[xtTIMESmubar])
                   2
>        1 + ------------------------------------------------------,
                                  pisq16
```

```
>    convGIRSMS[tag[S]] ->

                2    1
            CF g  (-(-) + 6 EulerGamma + 6 Log[xtTIMESmubar])
                   2
>        1 + ---------------------------------------------,
                               pisq16
```

```
>    convGIRSMS[tag[Ts]] ->

                2  25
            CF g  (-- - 2 EulerGamma - 2 Log[xtTIMESmubar])
                   6
>        1 + ------------------------------------------,
```

```
                                pisq16

>    convGIRSMS[tag[Tt]] ->

                2  25
             CF g  (-- - 2 EulerGamma - 2 Log[xtTIMESmubar])
                   6
>     1 + ----------------------------------------------,
                                pisq16

                                      2
                                  3 CF g
>    convGIRSMS[tag[Vs]] -> 1 + --------}
                                  2 pisq16

In[146]:= convGIRSMSlist = % ;

In[147]:= Save["diagram1.m10", convGIRSMSlist]
```

B.3 1-Loop Diagrams 2 & 3

```
In[1]:= << ~/latt_pert/math/inputF.m;
Reading symmetrize.m
Reading tensor.m
Reading hattosine.m
Reading prhoOrder.m
Reading background-vertices.m
Reading contract_NoCollectCFirst.m
Reading reducerho.m
Reading simplifydelm.m
Reading expandInC.m
Reading applyperms.m
Reading collect.m
Reading replace.m
Reading replacec2.m
Reading replacec2p.m
Reading makes2sq.m
Reading matchindices.m
Reading bl.m
Reading overlap.m
Reading Iwasaki.m
Reading reducegamma.m
Reading replacec2exact.m
Reading makeindependent.m
Reading rorderall.m

In[2]:= (* Let us introduce the expression for diagram 2 and 3 *)

In[3]:= (* denoms[a,b,l,m,n] stands for hat2[p]^a hat2[q]^m hat2[p-q]^l hat2[p-k]^b
hat2[q-k]^n *)

In[4]:= d2 * denoms[2, 1, 1, 1, 0] s2[2p, mu] s2[2q, nu] s2[2p, rho]  s2[2p-2k, sigma] +
d3 * denoms[2, 1, 1, 1, 0] s2[2p - 2k, mu] s2[2p, nu] s2[2q, rho]  s2[2p, sigma] ;

In[5]:= % /. a_. denoms[2,1,1,1,0] :> (a /. {p->q,q->p}) denoms[1,0,1,2,1]

Out[5]= d3 denoms[1, 0, 1, 2, 1] s2[2 p, rho] s2[2 q, nu] s2[2 q, sigma]

>     s2[-2 k + 2 q, mu] + d2 denoms[1, 0, 1, 2, 1] s2[2 p, nu] s2[2 q, mu]

>     s2[2 q, rho] s2[-2 k + 2 q, sigma]

In[7]:= %% //. s2[a_+b_,c_] :> s2[a,c]+s2[b,c] // Expand // canonical;

In[8]:= % /. denoms[1,0,1,2,1] -> hat2[p] hat2[q]^2 hat2[k-q] hat2[p-q];

In[9]:= (* Let's perform the integration over p*)

In[10]:= Expand[mu^(2e) %%]; (* it has a factor of g^2 * (Nc^2 -1)/2 * exp[ik(x-y)]*)
```

```
In[11]:= % /. a_ hat2[p]^i_. hat2[p-q]^j_. :> (q^2)^(2-e-i-j)/(4Pi)^2 *
Sum[g[i,j,count[a,s2[2p,_]],s] (q^2)^s /s!/4^s Nest[box,a,s] ,{s,0,count[a,s2[2p,_]]/2}]
set[p,q];

In[12]:= FreeQ[%,box]

Out[12]= True

In[13]:= %% /. a_. set[p,q] :> (a /. p -> q);

In[14]:= % /. q^2 -> 1/hat2[q];

In[15]:= PowerExpand /@ %

                2 e                                    2 + e
Out[15]= -(d2 mu    g[1, 1, 1, 0] hat2[k - q] hat2[q]      s2[2 k, sigma]

                                                        2
>        s2[2 q, mu] s2[2 q, nu] s2[2 q, rho]) / (16 Pi ) -

           2 e                                    2 + e
>    (d3 mu    g[1, 1, 1, 0] hat2[k - q] hat2[q]      s2[2 k, mu] s2[2 q, nu]

                                                  2
>       s2[2 q, rho] s2[2 q, sigma]) / (16 Pi ) +

           2 e                                    2 + e
>    (d2 mu    g[1, 1, 1, 0] hat2[k - q] hat2[q]      s2[2 q, mu] s2[2 q, nu]

                                                  2
>       s2[2 q, rho] s2[2 q, sigma]) / (16 Pi ) +

           2 e                                    2 + e
>    (d3 mu    g[1, 1, 1, 0] hat2[k - q] hat2[q]      s2[2 q, mu] s2[2 q, nu]

                                                  2
>       s2[2 q, rho] s2[2 q, sigma]) / (16 Pi )

In[16]:= (* Let's perform the integration over q*)

In[17]:= %% /. a_ hat2[q]^i_. hat2[k-q]^j_. :> (k^2)^(2-e-i-j)/(4Pi)^2 *
Sum[g[i,j,count[a,s2[2q,_]],s] (k^2)^s /s!/4^s Nest[box,a,s] ,{s,0,count[a,s2[2q,_]]/2}]
set[q,k];

In[18]:= FreeQ[%,box]

Out[18]= False

In[19]:= %% //. box[a_?(FreeQ[#,q]&) b_] :> a box[b];
```

```
In[20]:= % /. box[a_?(FreeQ[#,box]&)] :> deriv[deriv[a,rho[11]],rho[11]] // reducerho;

In[21]:= FreeQ[%,box]

Out[21]= False

In[22]:= %% /. box[a_?(FreeQ[#,box]&)] :> deriv[deriv[a,rho[12]],rho[12]] // reducerho;

In[23]:= FreeQ[%,box]

Out[23]= True

In[24]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
{ByteCount[%],Length[%]}

In[25]:=
In[26]:=
In[27]:=
In[28]:=
In[29]:=
In[30]:=
Out[30]= {32416, 16}

In[31]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
{ByteCount[%],Length[%]}

In[32]:=
In[33]:=
In[34]:=
In[35]:=
In[36]:=
In[37]:=
Out[37]= {50032, 28}

In[38]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
```

```
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
{ByteCount[%],Length[%]}

In[39]:=
In[40]:=
In[41]:=
In[42]:=
In[43]:=
In[44]:=
Out[44]= {59824, 32}

In[45]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
{ByteCount[%],Length[%]}

In[46]:=
In[47]:=
In[48]:=
In[49]:=
In[50]:=
In[51]:=
Out[51]= {87920, 50}

In[52]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
{ByteCount[%],Length[%]}

In[53]:=
In[54]:=
In[55]:=
In[56]:=
In[57]:=
In[58]:=
Out[58]= {91392, 56}

In[59]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
```

```
Expand[%];
{ByteCount[%],Length[%]}

In[60]:=
In[61]:=
In[62]:=
In[63]:=
In[64]:=
In[65]:=
Out[65]= {52528, 34}

In[66]:= %% //. deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a;
% /. deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b];
% //. deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c];
% /. deriv[a_?(FreeQ[#,q]&),b_] ->0;
% /. deriv[s2[2q,a_],b_] :> delm[a,b];
Expand[%];
{ByteCount[%],Length[%]}

In[67]:=
In[68]:=
In[69]:=
In[70]:=
In[71]:=
In[72]:=
Out[72]= {52528, 34}

In[74]:= %%% /. a_. set[q,k] :> (a /. q -> k);

In[75]:= % // Expand // simplifydelm // reducerho ;
% // PowerExpand ;
% /. k^a_ :> k^Expand[a];

In[76]:=
In[77]:=

In[121]:= (* Let's perform the integration over k. We have three typologies, which are:
          k^(2-4e) 1,   k^(-4e) s2[2 k, a] s2[2 k, b] ,  k^(-2-4e) s2[2 k, a] s2[2 k, b]
s2[2 k, c] s2[2 k, d]*)

In[122]:= (* Consider the integral of the form: *)

In[123]:= (* Exp[I k x] k^(-b) *)

In[124]:= Integrate[Exp[-I k^2] I^(b/2 -1) /Gamma[b/2], {I,0,Infinity}, Assumptions ->
k>0]

                                  -b
Out[124]= ConditionalExpression[k  , Re[b] > 0]
```

```
In[125]:= int[l] Exp[I k x] Exp[-I k^2] l^(b/2 -1) /Gamma[b/2]

                2
           -(k  l) + I k x  -1 + b/2
          E                 l         int[l]
Out[125]= ---------------------------------
                           b
                      Gamma[-]
                           2

In[126]:= Integrate[Exp[I k x - I k^2], {k, -Infinity, Infinity}]
%[[1]]

                                    Sqrt[Pi]
Out[126]= ConditionalExpression[-----------------, Re[l] > 0]
                                  2
                                 x /(4 l)
                                E         Sqrt[l]

In[127]:= %[[1]]

              Sqrt[Pi]
Out[127]= -----------------
            2
           x /(4 l)
          E         Sqrt[l]

In[128]:= %^d // PowerExpand

                 d/2
                Pi
Out[128]= ------------------
               2
           (d x )/(4 l)  d/2
          E              l

In[129]:= % /. d x^2 -> x^2

               d/2
              Pi
Out[129]= --------------
            2
           x /(4 l)  d/2
          E          l

In[130]:= Integrate[% l^(b/2 -1) /Gamma[b/2], {l,0,Infinity}, Assumptions -> {x^2 >0}];

In[131]:= %[[1]] /. d -> 4-2e
```

```
          4 - b - 2 e   (4 - 2 e)/2      -4 + b + 2 e      4 - b - 2 e
         2           Pi           Abs[x]            Gamma[-----------]
                                                              2
Out[131]= ------------------------------------------------------------
                                     b
                                  Gamma[-]
                                     2

In[132]:= ExpandAll[% /. Abs[x] -> x]

          4 - b - 2 e   2 - e  -4 + b + 2 e          b
         2           Pi      x            Gamma[2 - - - e]
                                                   2
Out[132]= ------------------------------------------------
                                 b
                              Gamma[-]
                                 2

In[133]:= (% /. b -> #)& /@ {-2+4e, 4e, 2+4e}

           6 - 6 e   2 - e  -6 + 6 e          2 - 4 e
          2        Pi      x         Gamma[2 + ------- - e]
                                                2
Out[133]= {-----------------------------------------------,
                               -2 + 4 e
                           Gamma[--------]
                                  2

      4 - 6 e   2 - e  -4 + 6 e
     2        Pi      x         Gamma[2 - 3 e]
>    ---------------------------------------,
                   Gamma[2 e]

      2 - 6 e   2 - e  -2 + 6 e          -2 - 4 e
     2        Pi      x         Gamma[2 + -------- - e]
                                           2
>    -------------------------------------------------}
                          2 + 4 e
                       Gamma[-------]
                                  2

In[134]:= ExpandAll[%] ;

In[135]:= derivrules = {deriv[a_ b_, c_] :> deriv[a,c] b + deriv[b,c] a,
deriv[a_^i_,b_] :> i a^(i-1) deriv[a,b],
deriv[a_ + b_, c_] :> deriv[a,c] + deriv[b,c],
deriv[a_?((FreeQ[#,xsq] && FreeQ[#,x])&),b_] ->0,
deriv[xsq, b_] :> 2 s2[2x,b],
```

```
deriv[s2[2x,a_],b_] :> delm[a,b]};

In[136]:= listk = %% /. x^a_ :> xsq^Expand[a/2]

           6 - 6 e   2 - e    -3 + 3 e
          2        Pi      xsq         Gamma[3 - 3 e]
Out[136]= {----------------------------------------,
                       Gamma[-1 + 2 e]

      4 - 6 e   2 - e    -2 + 3 e
     2        Pi      xsq         Gamma[2 - 3 e]
>    ----------------------------------------,
                    Gamma[2 e]

      2 - 6 e   2 - e    -1 + 3 e
     2        Pi      xsq         Gamma[1 - 3 e]
>    ----------------------------------------}
                  Gamma[1 + 2 e]

In[138]:= (* Let us compute s2[2 k, mu] s2[2 k, nu] Exp[I k (x-y)] k^(-4e) *)

In[139]:= deriv[deriv[listk[[2]],mu],nu] / im^2

                       4 - 6 e   2 - e    -2 + 3 e
                      2        Pi      xsq         Gamma[2 - 3 e]
          deriv[deriv[----------------------------------------, mu], nu]
                                    Gamma[2 e]
Out[139]= ------------------------------------------------------------
                                      2
                                    im

In[140]:= % //. derivrules

            4 - 6 e             2 - e
Out[140]= (2        (-2 + 3 e) Pi      Gamma[2 - 3 e]

              -3 + 3 e
>      (2 xsq         delm[mu, nu] +

                         -4 + 3 e                                2
>        4 (-3 + 3 e) xsq         s2[2 x, mu] s2[2 x, nu])) / (im  Gamma[2 e])

In[141]:= (* Let us compute s2[2 k, mu] s2[2 k, nu] s2[2 k, rho] s2[2 k, sigma] Exp[I k
(x-y)] k^(-2-4e) *)

In[142]:= deriv[deriv[deriv[deriv[listk[[3]],mu],nu],rho],sigma] / im^4

Out[142]= deriv[deriv[deriv[deriv[
```

```
          2 - 6 e   2 - e    -1 + 3 e
         2        Pi      xsq         Gamma[1 - 3 e]
>        ----------------------------------------, mu], nu], rho], sigma]\
                     Gamma[1 + 2 e]

         4
>     / im

In[143]:= % //. derivrules;

In[144]:= Simplify[%]

           2 - 3 e                          2 - e    -5 + 3 e
Out[144]= (4        (-2 + 3 e) (-1 + 3 e) Pi      xsq         Gamma[1 - 3 e]

>      (xsq delm[mu, sigma] (xsq delm[nu, rho] +

>         6 (-1 + e) s2[2 x, nu] s2[2 x, rho]) +

>        xsq delm[mu, rho] (xsq delm[nu, sigma] +

>         6 (-1 + e) s2[2 x, nu] s2[2 x, sigma]) +

>        xsq delm[mu, nu] (xsq delm[rho, sigma] +

>         6 (-1 + e) s2[2 x, rho] s2[2 x, sigma]) +

>        6 (-1 + e) s2[2 x, mu]

>         (xsq delm[rho, sigma] s2[2 x, nu] +

>         xsq delm[nu, sigma] s2[2 x, rho] +

>         (xsq delm[nu, rho] + 2 (-4 + 3 e) s2[2 x, nu] s2[2 x, rho])

                                  4
>          s2[2 x, sigma])) / (im  Gamma[1 + 2 e])

In[145]:= list2sin = {s2[2k, mu_] s2[2k, nu_] -> %140 k^(4e) }

Out[145]= {s2[2 k, mu_] s2[2 k, nu_] ->

       4 - 6 e            4 e   2 - e
>     (2        (-2 + 3 e) k    Pi      Gamma[2 - 3 e]

              -3 + 3 e
>        (2 xsq        delm[mu, nu] +

                          -4 + 3 e
```

```
>          4 (-3 + 3 e) xsq         s2[2 x, mu] s2[2 x, nu])) /

          2
>      (im  Gamma[2 e])}

In[146]:= list4sin = {s2[2k, mu_] s2[2k, nu_] s2[2k, rho_] s2[2k, sigma_] -> %144 k^(2+4e) }

Out[146]= {s2[2 k, mu_] s2[2 k, nu_] s2[2 k, rho_] s2[2 k, sigma_] ->

        2 - 3 e                       2 + 4 e   2 - e    -5 + 3 e
>     (4        (-2 + 3 e) (-1 + 3 e) k        Pi      xsq

>        Gamma[1 - 3 e] (xsq delm[mu, sigma]

>           (xsq delm[nu, rho] + 6 (-1 + e) s2[2 x, nu] s2[2 x, rho]) +

>          xsq delm[mu, rho] (xsq delm[nu, sigma] +

>             6 (-1 + e) s2[2 x, nu] s2[2 x, sigma]) +

>          xsq delm[mu, nu] (xsq delm[rho, sigma] +

>             6 (-1 + e) s2[2 x, rho] s2[2 x, sigma]) +

>          6 (-1 + e) s2[2 x, mu]

>           (xsq delm[rho, sigma] s2[2 x, nu] +

>             xsq delm[nu, sigma] s2[2 x, rho] +

>             (xsq delm[nu, rho] + 2 (-4 + 3 e) s2[2 x, nu] s2[2 x, rho])

                                      4
>              s2[2 x, sigma]))) / (im  Gamma[1 + 2 e])}

In[147]:= list0sin = k^(-2+4e) * listk[[1]]

           6 - 6 e  -2 + 4 e   2 - e    -3 + 3 e
          2        k         Pi      xsq         Gamma[3 - 3 e]
Out[147]= ----------------------------------------------------
                              Gamma[-1 + 2 e]


In[150]:= ExpandAll[%] ;
{Variables[%],Length[%]}

In[151]:=
                                e    e          e
Out[151]= {{d2, d3, e, im, k, k , mu , xsq, xsq , delm[mu, nu],
```

```
>     delm[mu, rho], delm[mu, sigma], delm[nu, rho], delm[nu, sigma],

>     delm[rho, sigma], g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1],

>     g[2 + e, 1, 4, 0], g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2],

>     Gamma[1 - 3 e], Gamma[2 - 3 e], Gamma[2 e], Gamma[1 + 2 e],

>     s2[2 x, mu], s2[2 x, nu], s2[2 x, rho], s2[2 x, sigma]}, 236}

In[152]:= If[FreeQ[#,k], #, # listOsin]& /@ %% // Simplify ;
{Variables[%],Length[%]}

In[153]:=
                              e          e
Out[153]= {{d2, d3, e, im, mu , xsq, xsq , delm[mu, nu], delm[mu, rho],

>     delm[mu, sigma], delm[nu, rho], delm[nu, sigma], delm[rho, sigma],

>     g[1, 1, 1, 0], g[2 + e, 1, 3, 0], g[2 + e, 1, 3, 1], g[2 + e, 1, 4, 0],

>     g[2 + e, 1, 4, 1], g[2 + e, 1, 4, 2], Gamma[1 - 3 e], Gamma[2 - 3 e],

>     Gamma[3 - 3 e], Gamma[2 e], Gamma[-1 + 2 e], Gamma[1 + 2 e],

>     s2[2 x, mu], s2[2 x, nu], s2[2 x, rho], s2[2 x, sigma]}, 10}

In[154]:= out152 = %%;
```

Bibliography

[1] K.G. Chetyrkin and F.V. Tkachov. Integration by parts: the algorithm to calculate -functions in 4 loops. *Nuclear Physics B*, 192:159, 1981.

[2] J.C. Collins. *Renormalization.* Cambridge University Press, 1984.

[3] Michael Peskin. *An introduction to quantum eld theory*. CRC press, 2018.

[4] Heinz J Rothe. *Lattice Gauge Theories: An Introduction Third Edition*, volume 74. World Scienti c Publishing Company, 2005.

[5] Gerardus t Hooft and Martinus JG Veltman. Diagrammar. Technical report, CERN, 1973.

[6] Jean Zinn-Justin. *Quantum eld theory and critical phenomena*. Clarendon Press, 1996.

[7] A. Skouroupathis and H. Panagopoulos, *Two-loop renormalization of scalar and pseudoscalar fermion bilinears on the lattice*, Phys. Rev. D76, 094514 (2007).

[8] A. Skouroupathis and H. Panagopoulos, *Two-loop renormalization of vector, axial-vector and tensor fermion bilinears on the lattice*, Phys. Rev. D79, 094508 (2009).

[9] J.A. Gracey, *Three loop anomalous dimension of non-singlet quark currents in the RI' scheme*, Nucl. Phys. B662, 247 (2003).

[10] J.A. Gracey, *Three loop $\overline{MS}$ operator correlation functions for deep inelastic scattering in the chiral limit*, JHEP 0904:127,2009.

[11] J.A.K. G. Chetyrkin, A. Maier, *Massless correlators of vector, scalar and tensor currents in position space at orders $\frac{3}{s}$ and $\frac{4}{s}$: explicit analytical results*, Nucl.Phys.B844:266-288,2011.

[12] V.Gimenez, L.Giusti, S.Guerriero, V.Lubicz, G. Martinelli, S.Petrarca, J.Reyes, B.Taglienti, E.Trevigne, *Non-perturbative renormalization of lattice operators in coordinate space*, Phys.Lett. B598 (2004) 227-236.