**Perturbative Expansion of GF**

**GContour starting in |Ω0> at t = t0 with V­time-dependent(t) whenever**

So in the identical particles file, we develop the GF expansion from the equations of motion, at least in the simple single particle potential case. Such an approach doesn’t seem so useful here – the expansion doesn’t seem to bear much resemblance as judging from the Wick expansion. So I’ll just jump straight into the Wick approach. There are four GF that will naturally go together within the expansion:



(GC,AC will have that convergence factor, e-η|t-t´| too) I’ll refer to these GF’s collectively as GContour. We will consider the operators to evolve according to the Hamiltonian,



where H0 is a time-independent bilinear Hamiltonian,



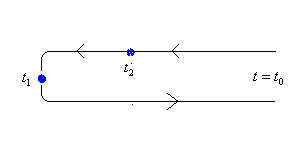
and V(t) is some possibly time-dependent interaction. It may well contain time-independent and time-dependent parts, like, well V(t) = (1/3!)λx3 + h(t)x, etc. In order to use Wick’s theorem, we’ll presume |Ω0> to be an eigenstate of H0 or some linear combination thereof. Also, |Ω0> is the state we start in at time t = t0 (we’ll allow t0 to be non-zero for the sake of discussion), and our time development operator will therefore be U(t,t0) = U0(t,t0)S(t,t0). We need this time development operator, instead of the mixed phase approach, because otherwise it would seem we can’t properly extract the averages we expected to get from the GF. And so the interaction picture operator will be given by AI(t,t0) = U0†(t,t0)AU0(t,t0), and I’ll just call this A(t).

**Perturbative expansion for G>**

Let’s form G>.



Let’s assume that t1 > t2. Then we could write the contents of the expectation as being ordered along the contour,



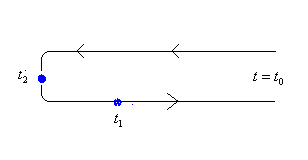
And we could then write our GF as:



where TC refers to the contour above. And S would look like this:



where the contour runs from t0 → t1 and then back from t1 → t0. What this means in practice will be made clearer in an example later. But what if t2 is greater than t1? Then we have the contour,



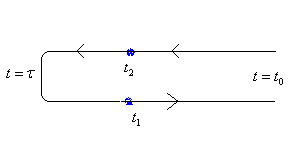
and,



In the general case, we could do the first with a θ(t­1 – t2) added to the second with a θ(t2 – t1) tacked on. This simplifies into an expression involving GR, etc., but…we can avoid these separate cases if we write out the contour as in the following way. Let’s introduce some τ > t1, t2 and write out the S-matrix expansion as…



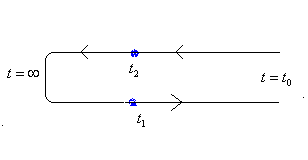
and so we have,



and,



Note how this contour ordering would still be the same in either case t­1 > t2 or t2 > t1. All that would change is that t1 and t2 would slide up or down the top and bottom contour respectively, but the form of the contour wouldn’t change. And we could – often do – take τ → ∞, and so have,



and,



**Perturbative expansion for G<**

We start with,



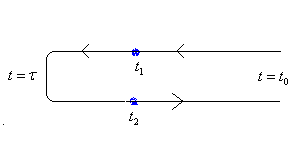
We will assume now that , and also write this in a convenient way …,



and so we have,



Where the contour orders the times according to



Note again how this contour ordering would still be the same in either case t­1 > t2 or t2 > t1. All that would change is that t1 and t2 would slide up or down the top and bottom contour respectively, but the form of the contour wouldn’t change.

**Perturbative expansion for GC**

We start with



Again, we will introduce . And we have, if t1 > t2:



Now let’s suppose t2­ > t1. Then we’d have,



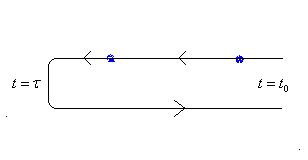
We could cover both cases by writing,



which we’ll write more succinctly as:



Where the contour orders the times according to:



which automatically time orders the operators – note that the first line time orders and the second line anti-time orders.

**Perturbative expansion for GAC**

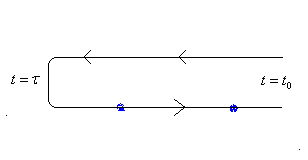
We start with



which we can write as:



with the contour…



**General Diagrammatic Rules**

So let’s say we have:



for example, and we start off in |Ω0> at time t = t0. And we’re interested in any one of the GF functions in GContour (it’s more intuitive to state the rules in terms of the contractions, rather than the GF’s themselves). I’ll specialize to GF’s of the type where A and B are just , though the procedure would work just as well for any bilinear in the creation / annihilation operators – for instance for too.



We can write all of these as:



where the times t and t´ are on the contours discussed above. **We only need these four G’s because in the out and back contour, any two times can only have these relative orientations: both on top, both on bottom, first on top/second on bottom, or first on bottom/second on top.** Anyway, then we just expand,



(V is in interaction picture) in a power series, where each term involves an integration that runs from t0 🡪 τ and back, and the two x’s in the GF are placed along the contour as shown above. Then we use Wick’s theorem to pick out contractions between the x(tA) and x(tB) terms. In general we get:



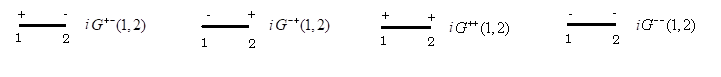
There is a shorthand notation that can be developed that’s very useful though. Can motivate it a little by considering the time-ordered correlation function.



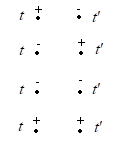
Might see that can write the expansion in terms of two vertices: one that time orders and another that anti-time orders. Anyway, we’ll just have:

**Real Space Rules**

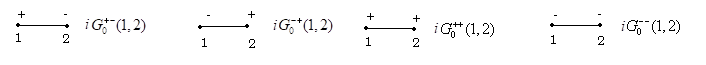
Our four contractions are represented with bold lines:



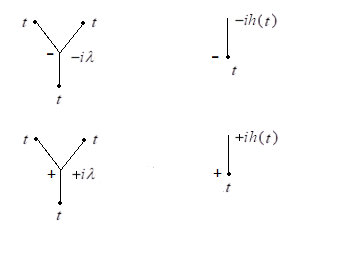
And the perturbative series works as follows. So we start with the external points. We represent the two arguments/operators in the GF we’re trying to get with external points. We use the first if we want G+-, the next for G-+, the next for G—and the last for G++.



Then at each order of PT, we bring down the same number of vertices (discussed in a second) and connect with bare iGF’s. The bare (unperturbed) iGF’s are represented with non-bold lines:



The two S’s give rise to two vertices. The (-) vertex comes from the time ordered S, and the (+) vertex comes from the anti-time ordered vertex. Note there’s nothing per se´ that seems to prevent λ from being time-dependent. But it basically never is, so I’ll just leave any such dependence off.



So the general procedure is as follows…a more sufficient motivation is given in a special case in the next file.

**Topology**

Connect all topologically distinct, connected (meaning no bubbles, but end points can be disconnected from each other), diagrams together, associating each element in diagram with the indicated term. See GS Perturbation file for some justification.

**Equal time issues**

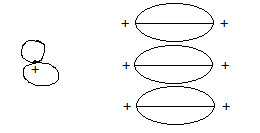
There are no equal time issues to be conscerned about to my knowledge.

**Signs/Numerical Factors**

The topologically distinct diagrams show up in many different equivalent ways in the direct S-matrix expansion. And so we have to multiply each by the number of times it would appear in the expansion, by the 1/n! coming from the S-matrix expansion, and by the λ/3!,4!, etc. coming from the interaction prefactor. The rules for calculating this number, sans the coupling constant, are, I think:



For instance,



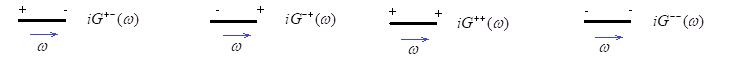
the cuartic vacuum bubble has a factor of (1/2)(1/2)(1/2) = 1/8 because of two coincident vacuum propagators, and two indistinguishable propagators. The triple cubic vacuum bubble would have a factor of [(1/3!)(1/2)]3(1/3!) due to each single bubble having 3 indistinguishable propagators, two indistinguishable vertices, and the overall triple bubble having a set of 3 pairs of indistinguishable vertices. Can look to next file for more on this. But might have to pay particular attention to the last point here, as we wouldn’t expect (-) (+) vertices to be indistinguishable.

**Sum**

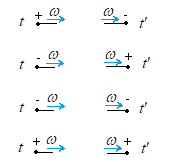
Then sum/integrate over all internal positions/times.

**Fourier Space Rules**

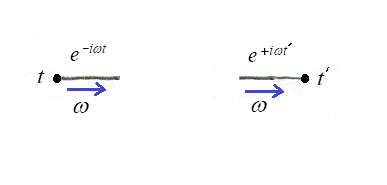
We can express the diagram in terms of Fourier components instead. Though, for finite time integrations, we’d have to put step functions on V(t) to allow integration over the entire real line. There is basically no point in doing this unless λ is position-independent. So for the sake of discussion I’ll presume so. Most of the following is justified in the Fourier transforms file. As for the GF’s, bold lines represent the exact GF:



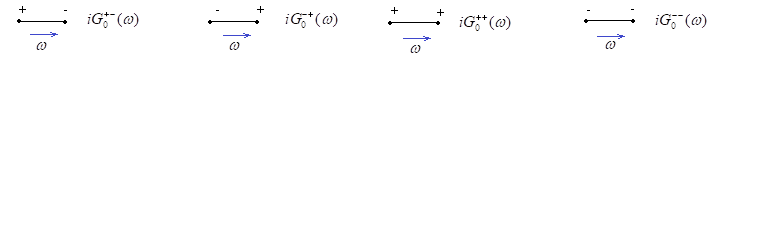
and then external points are same as before,



These carry the Fourier transform variable. And if we’re ultimately interested in the real time GF, then we must add the following factors to each:



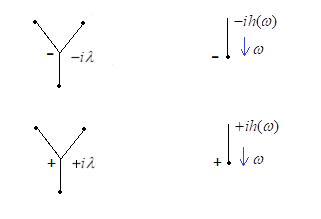
We connect these with the unperturbed GF’s:



where these G’s are the Fourier transforms of the real space guys,



and the vertices:



**Topology**

Connect all topologically distinct, connected (meaning no bubbles, but end points can be disconnected from each other), diagrams together, associating each element in diagram with the indicated term. See GS Perturbation file for some justification. And then we conserve energy/frequency at each vertex.

**Equal time issues**

None, again.

**Signs/Numerical Factors**

The same as above.

**Sum**

Then sum/integrate over all independent internal energies (frequencies). If want the real space GF then include the external frequency too.

