**Perturbative Expansion of Green’s Function**

So the main quantity of interest vis a vis thermally averaged GF, and even the other ones, is the retarded GF because this is the one which shows up in transport properties, as we’ll see in NESM. So one possibility is to develop an expansion based off of the differential equation satisfied by the GF. This would be done the same as how we did it in the regular QM case. But I’ll skip that and just go into the Wick approach.

**Thermal averaged real time Green’s functions**

Now let’s examine a perturbative scheme to get the thermal averaged GF from the Wick expansion. We’ll go the historical route and consider real time GF’s first. Let’s consider G<(t1,t2) for example. Let H = H0 + V1 + V2 perhaps. We’ll denote the net perturbation by simply V, and recall that it’s time-independent. Then,



[We’ll start time development at t = 0 I guess, but it shouldn’t matter because with time-independent interactions, the GF should be a function only of the time difference – a fact we’ll utilize in next line, and note 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)]. We can write this as (the τ here is not complex time, just some reference time):



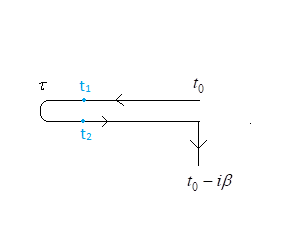
But problem with this is that the thermal averaging is being done against the interacting K. And Wick’s theorem only applies to non-interacting thermal averaging. So we have to go back and expand e-βK too. So we’ll write:



And so, compactly, we’d have:



where TCS(t0-iβ,t0) is short hand for S evaluating along the contour below, upon which the time-ordering takes place:

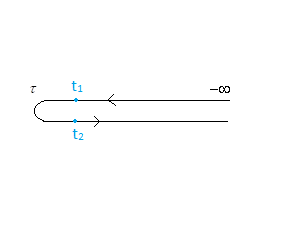


We can expand our GF according to this contour. And using Wick’s theorem, as we’ve already established we can for traces such as these, we would separate the integration into 3 parts: (t0, τ), (τ, t0), (t0, t0 – iβ). And we would have 4 GF: G>, G<, G--, G++. But we would have to evaluate some GF’s for complex times, since ‘s’ can now be on any of the three curves. This would be okay if say s and s´ were both on negative imaginary line since we’ve got that case covered with the complex time GF. We haven’t calculated GF where s is real, and s´ is on imaginary line though, so this would take a little more investigation…

It seems to me the result shouldn’t depend on what t0 is. But it’s alleged that if we send t0 → -∞, we can then neglect the t0 – iβ part of the contour. Hmmmm….I suppose this has to do with the exponential convergence factors ~ e-η(t-t´) that would/should be on all the real time GF’s. This would make all contributions at t0 = -∞ go to zero. And so then we could naturally neglect the tail end of the contour. Then have:



with the simple contour:



Seems kind of awkward to have a reference point t0 → -∞. But this won’t actually affect the GF’s, since our H’s are time-independent, and so G(t,t´) = G(t-t´) = G((t-t0) – (t´-t0)), and so the reference time will just cancel out. So the only place it will show up is in the lower limit of integration. So our calculation should resemble the real-time ‘state-averaged’ GF’s that we calculated in the QM folder, except that now we would replace the state average with a thermal average. And like with the ground state GF’s, I expect the denominator to simply cancel vacuum bubbles (well, wouldn’t they cancel out anyway, due to the *purely* out/back contour?). So I’d think we’ll only have to calculate fully connected diagrams. Finally, once we have these four GF’s G++, G+-, G-+, G-- (or any one of them), we can calculate the retarded GF.

Oh, well, I guess we would have to divide top/bottom of our GF expression by Tre-βK\_0, technically, to do the Wick expansion in terms of non-interacting GF’s.

**Feynman Rules for Interacting Distinct Particles**

I’ll be more or less copying the rules from Quantum Mechanics/Multiple Particles file. So for the sake of discussion say we have a classical field Hamiltonian, describing an elastic solid:



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 α(R), though the procedure would work just as well for any bilinear in the creation / annihilation operators.



We can write all of these as:



where the times t and t´ are on the contours below, respectively.

A picture containing line, diagram, text, screenshot

Description automatically generated

A picture containing line, diagram, screenshot, plot

Description automatically generated

A picture containing line, diagram, plot, screenshot

Description automatically generated

A picture containing diagram, line, plot, screenshot

Description automatically generated

Then we just expand,



where:



(V is in interaction picture) in a power series, where each term involves an integration that runs from -∞ 🡪 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:



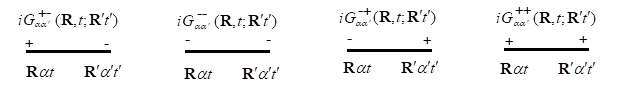
There is a shorthand notation that can be developed that’s very useful though. Can motivate it a little by considering the time-ordered correlation function.



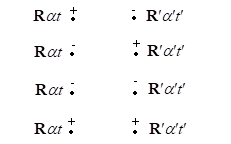
Might see that can write the expansion in terms of two vertices: one that time orders and another that anti-time orders. Anyway, we’ll just have:

**Real Space Rules**

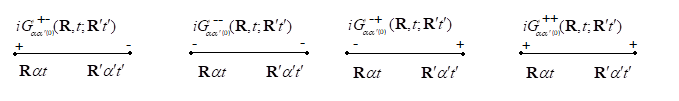
Our four contractions are represented with bold lines:



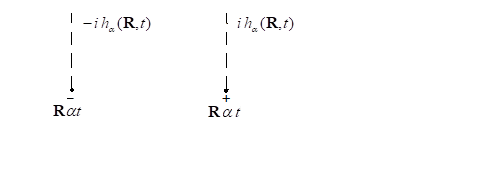
And the perturbative series works as follows. So we start with the external points. We represent the two arguments/operators in the GF we’re trying to get with external points. We use the first if we want G+-, the next for G-+, the next for G—and the last for G++.



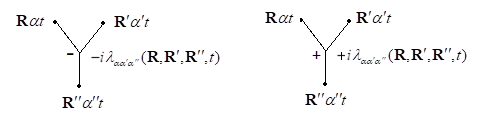
Then at each order of PT, we bring down the same number of vertices (discussed in a second) and connect with bare iGF’s. The bare (unperturbed) iGF’s are represented with non-bold lines:



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. The single particle potential would be:



and the interaction vertex, cubic in this example, would be:



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 concerned 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, divided by the 1/n! coming from the S-matrix expansion. The rules for calculating this number are, I think as follows. And it is probably necessary to stipulate the whatever λmm´m´´(R,R´,R´´) is, it is symmetric w/r to interachange of any two of its coordinate+index pairs, say for instance (m,R) with (m´´,R´´). This would probably make λ a scalar.



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, though this distinction wouldn’t apply in the next file.

**Sum**

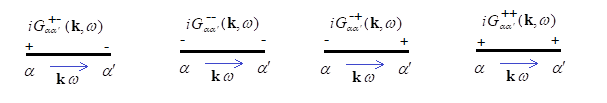
Then sum/integrate over all internal positions/times/indices. There is a simplification to be had if we have a homogeneous isotropic medium (then K is diagonal I think as only isotropic N×N matrix is diagonal one, and interaction must be cuartic as there is no isotropic 3D tensor). Then GF0 and GF are diagonal in their indices αα´. In that case, if we were to be interested in calculating G11, say, then the δαα´ attached to the GF would, along with the requisite index sum, end up fixing all GF’s in the diagram to that index G(0)11. So then we wouldn’t have to worry about any index sum per se’. Except in the case that we have closed loops. Then it seems to me to be similar to the fermion loop discussed in the identical particles file, and we’d have to multiply such loops by a factor of 3 (presuming GF11 = GF22 = GF33 which seems to be necessary in such case).

**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 time/position-independent and completely local so that it takes the form:



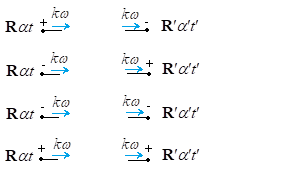
So for the sake of discussion I’ll presume so. Most of the following is justified in the Fourier transforms file. So the Green’s functions, as usual:



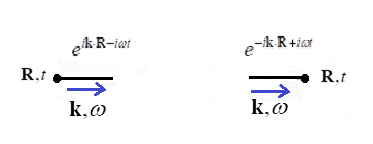
where these G’s are the Fourier transforms of the real space guys,



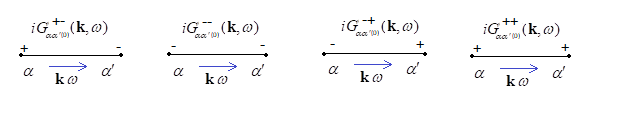
and then external points are same as before,



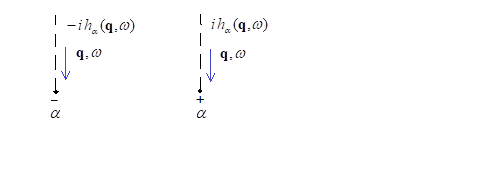
These carry the Fourier transform variable. And if we’re ultimately interested in the real space, time GF, then we must add the following factors to each:



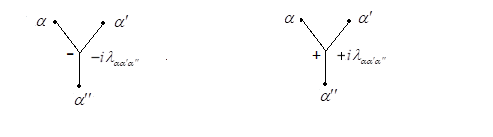
We connect these with the unperturbed GF’s:



Then the single particle potential is:



and the interaction vertex,



**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/momentum at each vertex.

**Equal time issues**

None, again.

**Signs/Numerical Factors**

The same as above.

**Sum**

Then sum/integrate over all independent momenta (wavenumbers)/energies (frequencies), indices, including the external one if you want the real space GF.



**Feynman Rules for Interacting Indistinct Particles**

So let’s say we have:



where,



And we’re interested in the GF’s in GContour (it’s more intuitive to state the rules in terms of the contractions, rather than the G’s themselves). I’ll specialize to GF’s of the type where A = ψ(x) and B = ψ†(x´):



though the procedure would work just as well for any bilinear in the creation/annihilation operators. We can write these as:



where the times t and t´ are on the contours below, respectively.

A picture containing line, diagram, text, screenshot

Description automatically generated

A picture containing line, diagram, screenshot, plot

Description automatically generated

A picture containing line, diagram, plot, screenshot

Description automatically generated

A picture containing diagram, line, plot, screenshot

Description automatically generated

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 ψ’s in the GF are placed along the contour as shown above. Then we use Wick’s theorem to pick out contractions between the ψσA(xA,tA) and ψσB†(xB,tB) terms, being sure to include a -ε factor for every transposition required to bring the two operators together. In general the contractions will equate to:



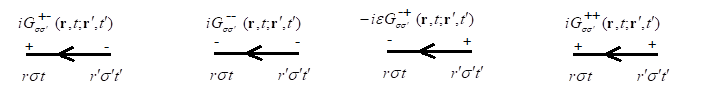
There is a shorthand notation that can be developed that’s very useful though. Can motivate it a little by considering the time-ordered correlation function.



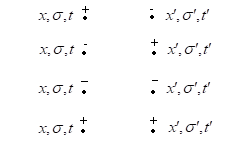
Might see that can write the expansion in terms of two vertices: one that time orders and another that anti-time orders. The strict justification for proceeding to certain combinations of GF will be skipped and instead, the heuristic Feynman method will be discussed.

**Real Space Rules**

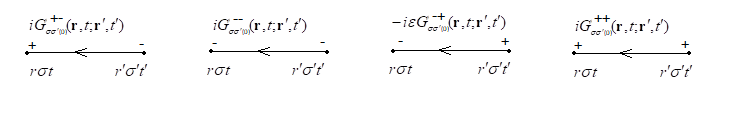
Our four contractions are represented with bold lines:



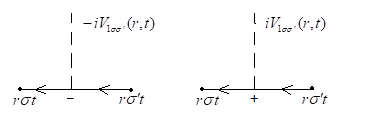
And the perturbative series works as follows. So we start with the external points. We represent the two arguments/operators in the GF we’re trying to get with external points. We use the first if we want G+-, the next for G-+, the next for G—and the last for G++.



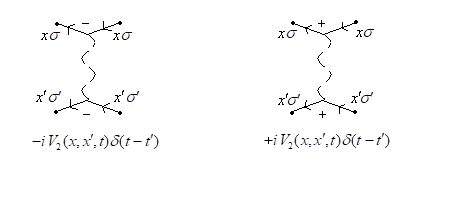
Then at each order of PT, we bring down the same number of vertices (discussed in a second) and connect with bare iGF’s. The bare (unperturbed) iGF’s are represented with non-bold lines:



and the single-particle potential we represent with a vertex as before. Now we have two, corresponding to the two S-operators in the expansion:



and the two particle potential also gets two versions. Presuming it’s a spin-independent interaction, we have:



Note that arrows pointing away from vertex represent the ‘dagger’ part of the operator, while arrows pointing towards the vertex represent the other guy.

One other note. The diagrams are written flipped about the vertical from the convention some other authors use. Time flows roughly in the direction of the arrows and so some others prefer that time flow left to right. Written this way, one would read the arguments of the GF’s from left to right, so it’s a little easier to translate into equations. So the general procedure is as follows…a more sufficient motivation is given in a special case in the next file.

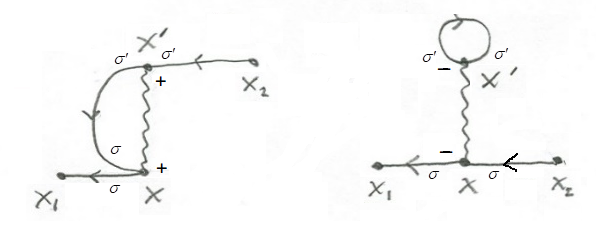
**Topology**

Connect all topologically distinct, fully connected (meaning no bubbles), diagrams together, associating each element in diagram with the indicated term (see GS Perturbative Expansion file for justification of some of this topology). Note the order of indices is important, Gσσ´(xk,xj) ≠ Gσ´σ(xj,xk), so always order the positions against the directions of the arrows.

**Equal time issues**

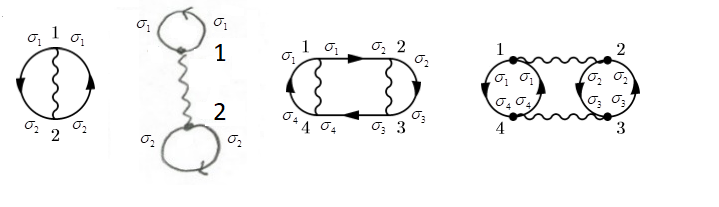
There are two equal time issues, depicted below. In the former, G(0)σσ´++(x,t;x´,t´) occurs at at same time thanks to the δ(t – t´) the potential line carries. And the latter,

G(0)σ´σ´--(x´,t´;x´,t´) obviously occurs at same time. This makes the time-ordering operator ambiguous. But because each of these occurs due to contractions of operators within V2, we should interpret the GF as occuring in that same order, whereby ψ† appears before ψ. So we should interpret them as G(0)σσ´++(x,t;x´,t-) and G(0)σ´σ´--(x´,t´;x´,t´+) respectively. The latter would be equal to G(0)σ´σ´--(x´,t´;x´,t´+) = -i<Tψ(x´,t´)ψ†(x´,t´+)> = -i(-ε)< ψ†(x´,t ´+)ψ(x´,t´)> = iεnΩ0(x´,σ´) [where nΩ0 is the occupation number of the state against which we’re doing the expectation].

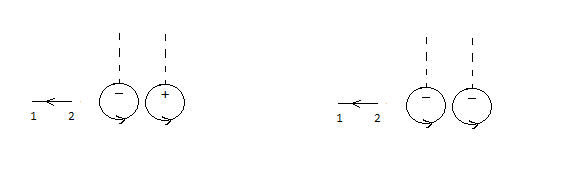


**Signs/Numerical Factors**

Each diagram is associated with a numerical factor, because each topologically distinct diagram comes from a variety of contractions and because we must factor in the 1/n! term coming from the S-matrix expansion. Basically, diagrams with external legs seem to have a simple net factor of unity. Bubbles have a symmetry factor of 1/p, where p is the number of permutations (including the identity permutation) of points that result in the same diagram [note discussion below presumes all vertices are *same* sign]. For instance,



We can do the permutation (12) and so the first guy gets a factor of ½. Same with the second guy. The third gets a factor of ½ as well because all we can do is the permutation (13)(24). Can verify that flipping diagram about horizontal and vertical would result in this permutation and so it works. Or another way, both 1 and 3 have interaction vertex, vertical propagator going into it, and horizontal propagator going out of it. Not sure why (14)(23) wouldn’t also be acceptable, but for the possible fact that there doesn’t seem to be a corresponding set of reflections that would result in this permutation, like there was for the previous. And maybe also because in present state, we have propagator going from 1 → 4, and 3 → 2, and if you switch the numbers, then the propagators would be going 4 → 1 and 2 → 3. So that’s different. Then the last guy has the permutations, (12)(34), (13)(24), and (14)(23). So it gets a factor of ¼. This also applies to disjoint diagrams. For instance, the diagram on the left has a factor of 1, but the diagram on the right has a factor of ½.



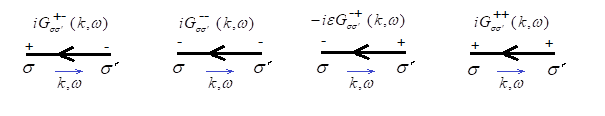
(Fermion) loops have there own special sign factors. Each fermion loop get a factor of (-ε), which would be just 1 for a boson loop – this has to do with intracacies of transposing the operators to put them in proper order. The four diagrams have 1, 2, 1, 2 Fermion loops respectively.

**Sum**

Then integrate/sum over all internal vertices, including spin ones. Must sum over spin indices if fermion too. Apropos the spin sum, if spin is conserved by H, and presuming G0σσ´ to be diagonal in spin index as ours is, then we’ll note that if we start off trying to calculate G↑↑, then the δσσ´ attached to our GF, coupled with our spin sum, will simply have the effect of enforcing all other GF’s in the diagram to also be G0↑↑. Likewise if we tried to calculate the spin down GF. So basically we can ignore the spin sum part. Except in one instance: fermion loops. In these parts of the diagram, the GF’s are in some sense disconnected from the external points, and since they circle back on themselves we end up with a term like Σσ1,σ2,σ3G(0)σ1σ2G(0)σ2σ3G(0)σ3σ1. And this reduces to Σσ3G(0)σ3σ3G(0)σ3σ3G(0)σ3σ3 = G(0)↑↑G(0)↑↑G(0)↑↑ + G(0)↓↓G(0)↓↓G(0)↓↓ = 2[G(0)↑↑G(0)↑↑G(0)↑↑], say. And so we just need to multiply our diagrams by 2 (or 2s+1 if higher spin) to account for these loops. The simple factor of two *does* presume that G(0)↑↑ = G(0)↓↓, however. This could feasibly not be so if we included a Bz in our unperturbed H0. So if not then we’d just replace the trace by whatever that works out to be.

**Fourier Space Rules**

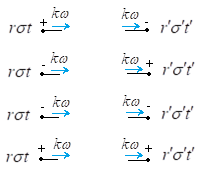
We can express the diagram in terms of Fourier components instead. Though, for finite time integrations, we’d have to put step functions on V(t) to allow integration over the entire real line. Then if V1 = 0, and V2 = V2(x-x´) with no time dependence, then the Fourier transform of the diagram will just be a product of Fourier transforms of each individual piece in the diagram, which is nice. But even if not, most typically because V1 ≠ 0, we can still represent everything in terms of its Fourier components. Most of the following is justified in the Fourier transforms file. So for GF’s, bold lines represent the exact GF:



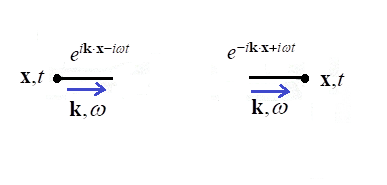
where these G’s are the Fourier transforms of the real space guys,



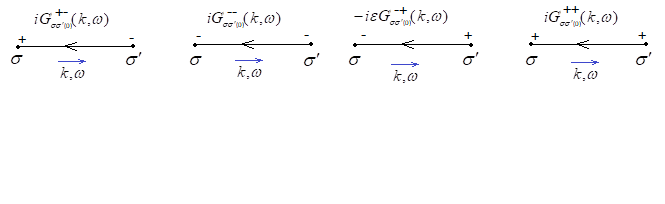
and then external points are same as before,



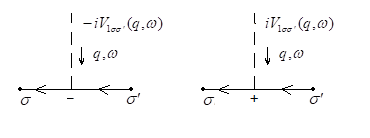
These carry the Fourier transform variable. And if we’re ultimately interested in the real space, time GF, then we must add the following factors to each:



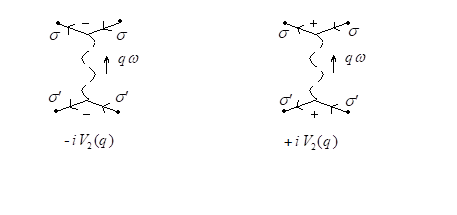
We connect these with the unperturbed GF’s:



And we associate single particle potential with:



[if V1 is time-independent, then there is no ω] and the two particle potential with (assuming translationally invariant and no time-dependence):



[note we’d have V2(q,ω), but that’s just V2(q) as Fourrier transform of delta function δ(t-t´) is 1] and we also have external legs if want the real space GF out of this:

**Topology**

Connect all topologically distinct, fully connected (meaning no bubbles), diagrams together, associating each element in diagram with the indicated term (see GS Perturbative Expansion file for justification of some of this topology). Note Remember that all energy-momentum labels **k**, ω must be going in the same way w/r to the GF arrow. They must all be going against the arrows (or with them I suppose). And then we conserve energy/momentum at each vertex.

**Equal time issues**

Apropos the equal time issue, in Fourier space this would be handled as:



So we need to include a factor of exp(±iω0+), as the case may be, for each bare Green’s evaluated at same time. Fundamentally, the problem occurs when the temporal argument of the bare Green’s function is zero.

**Signs/Numerical Factors**

The same as above.

**Sum**

Then sum/integrate over all independent internal momenta/energies/spins. If finite spatial transform (finite volume), then the guy on the left. If continuous spatial transform (infinite volume), then the guy on the right:



and include the external momentum/energy if want the real space GF.

**Feynman Rules for Photons**

I’ll just say it’s basically the same as the rules for phonons.