myLPT

(Written by Maximilian Ammer, 2023)
This package contains definitions and functions to do calculations in lattice perturbation theory.

```
(* Save as .m *)
(* Use with Get["~path~/myLPT.m"] *)
```

Generating Feynman rules

```
myExpandU[expr_]:=(a1=expr/.U[n_,-mu_,x_]:> Ud[n,mu,x-e[mu]]/.e[-mu_]:> -e[mu];
(* Insert Expansions of U and Udagger *)
a2=a1/.U[n_,mu_,x_]:>(1+I g0 A[n,mu,x ]- g0^2/2  A[n,mu,x]×A[n+1/10,mu,x ]-I g0^3/6A[n,
a3=a2/.Ud[n_,mu_,x_]:>(1-I g0 A[n,mu,x ]-g0^2/2  A[n,mu,x]×A[n+1/10,mu,x ]+I g0^3/6A[n,
(* Truncate at order g0^3 *)
a4=Normal[a3+O[g0]^4]//ExpandAll;
(* Sort As *)
a5=a4/.A[n1_,mu_,x_]×A[n2_,nu_,y_]×A[n3_,rho_,z_]/;{n1,n2,n3}∈Rationals &&n1<n2&&n2<n3
a6=a5/.A[n1_,mu_,x_]×A[n2_,nu_,y_]/;{n1,n2}∈Rationals &&n1<n2:>T[a,b]×A[a,mu,x]×A[b,nu,
a7=a6/.A[n1_,mu_,x_]/;n1∈Rationals:>T[a]×A[a,mu,x])
```

```
(* Fourier Transform fields and deltas *)
myFourierTransform[expr_]:=(a1=expr/.{ψ[x_]:>Int[p] Exp[I x p]Ψ[p],ψbar[x_]:>Int[q] Exp
 A[a_,mu_,x_]:> Int[k[a]]Exp[I(x+e[mu]/2)k[a]]A[a,mu,k[a]],
 δ[x_,y_]:> Int[s]Exp[I(x-y)s]};
a2=a1/.{k[a]→k1,k[b]→k2,k[c]→k3})
```

```
myFRintegrate[expr_]:=
(iv1=expr /. Int[p]×Int[q] ×Ψ[p]×Ψbar[q] → 1/.Int[k2]→1/.Int[k3]→ 1;
iv2=D[iv1/.x→x/I, x]/.x→0;
iv3=(iv2/.(s+p__):> I delta[s+p])/.delta[s+p_]×Int[s] ex_:> (ex/.s→ -p);
iv4=D[iv3/.y→y/I, y]/.y→0/.(k1+p__):> I delta[k1+p];
iv5=iv4/.delta[k1+p_]×Int[k1] ex_:> (ex/.k1→ -p)//ExpandAll;
iv6=iv5/.Exp[x_]:> Cos[x/I]+I Sin[x/I]//ExpandAll;
iv7=iv6/.k_ e[mu_]/;MemberQ[{p,q,k1,k2,k3},k]:>k[mu])
```

```
myTrigExpand[expr_]:=expr//.{Cos[a_+b_]:> Cos[a]Cos[b]-Sin[a]Sin[b],Sin[a_+b_]:>Sin[a]Co
myTrigReduce[expr_,set_]:=FixedPoint[ExpandAll[ReplaceRepeated[#,{Cos[a_. f_[mu_]+x_.]
1/2(Cos[a f[mu]-b g[mu]+x-y]+Cos[a f[mu]+b g[mu]+x+y]),Sin[a_. f_[mu_]+x_.]Sin[b_. g_[
1/2(Cos[a f[mu]-b g[mu]+x-y]-Cos[a f[mu]+b g[mu]+x+y]),Sin[a_. f_[mu_]+x_.]Cos[b_. g_[
1/2(Sin[a f[mu]-b g[mu]+x-y]+Sin[a f[mu]+b g[mu]+x+y])}]]&,expr]
```

```
myFRbreakdown[ex_]:=Module[{x,y},
 x=ex/.A[a_,mu_,-k3-k2-p+q]:→ A[a,mu,k1]/.A[a_,mu_,-k2-p+q]:→ A[a,mu,k1]/.A[a_,mu_,-p+q]:
 y=x/.expr_ SUM[nu_]/;MemberQ[{nu1,nu2,nu3},nu]:→ (expr -1/4(expr/.nu→mu))SUM[nu]//Expan
 x=y/.A[a,nui_,k1]×SUM[nui_]expr_/;MemberQ[{nu0,nu1,nu2,nu3},nui]:→(expr/.nui→munew/.mu-
 y=x/.A[b,nui_,k2]×SUM[nui_]expr_/;MemberQ[{nu0,nu1,nu2,nu3},nui]:→(expr/.nui→nu)A[b,nu,
 x=y/.A[c,nui_,k3]×SUM[nui_]expr_/;MemberQ[{nu0,nu1,nu2,nu3},nui]:→(expr/.nui→rho)A[c,rh
 y=x/.A[a,mu,k1]×A[b,mu,k2]×A[c,mu,k3]→KroneckerDelta[mu,nu]KroneckerDelta[mu,rho]A[a,m
 x=y/.A[a,mu,k1]×A[b,mu,k2]→KroneckerDelta[mu,nu]A[a,mu,k1]×A[b,nu,k2];
 y=x/.A[a,mu,k1]×A[c,mu,k3]→KroneckerDelta[mu,rho]A[a,mu,k1]×A[c,rho,k3];
 x=y/.A[b,nu_,k2]×A[c,nu_,k3]→KroneckerDelta[nu,rho]A[b,nu,k2]×A[c,rho,k3];
 y=x/.A[a,mu,k1]×A[b,nu,k2]×A[c,rho,k3]→1/.A[a,mu,k1]×A[b,nu,k2]→1/.A[a,mu,k1]→1;
 x=y/.σ[a_,rho]:→-σ[rho,a]/.σ[a_,nu]:→-σ[nu,a]/.σ[a_,mu]:→-σ[mu,a];
 y=myTrigExpand[x];
 x=myTrigReduce[y,{p,q,k1,k2,k3}];
 y=x//.expr_ SUM[nu_]:→ Sum[expr,{nu,1,4}];
 x=y/.σ[a_,b_]/;a>b:→-σ[b,a]/.σ[a_,a_]:→0]
```

Calculating diagrams/integrals

Definitions

Momenta

In[•]:=
```
pv={p1,p2,p3,p4};
qv={q1,q2,q3,q4};
kv={k1,k2,k3,k4};
p[mu_]:=pv[[mu]]
q[mu_]:=qv[[mu]]
k[mu_]:=kv[[mu]]
```

Gamma matrices

In[•]:=
```
𝕀={{1,0,0,0},{0,1,0,0},{0,0,1,0},{0,0,0,1}};
γ[i_]:=If[i==4,Internal`DiracGammaMatrix[1,"Basis"→"Chiral"],-I Internal`DiracGammaMatr
σ[i_,j_]:=I/2(γ[i].γ[j]-γ[j].γ[i])
```

Feynman rules

In[•]:=
```
myVecExpand[expr_]:=expr//.(x_. f_+z_.)[m_]/;MemberQ[{p,q,k},f]:→ x f[m]+(z)[m]/. 0[m_]
```

Gluon Propagator

```
(* hat{k} *)
hat[k_,mu_]:=2Sin[1/2 k[mu]]//myVecExpand
(* hat{k}^2 *)
DB[k_]:= Sum[hat[k,mu]^2,{mu,4}]//myVecExpand
(* Delta_4 *)
Delta4[k_]:=(DB[k]-c1 Sum[hat[k,rho]^4,{rho,4}])(
 DB[k]-c1(
 DB[k]^2+Sum[hat[k,tau]^4,{tau,4}])+
 1/2 c1^2 (
 DB[k]^3+
 2Sum[hat[k,tau]^6,{tau,4}]-DB[k]×Sum[hat[k,tau]^4,{tau,4}]
 )
)-4c1^3Sum[hat[k,rho]^4×Product[If[tau==rho,1,hat[k,tau]^2],{tau,4}],{rho,4}]
(* A_{mu,nu}(k) (Lüscher Weisz)*)
ALW[mu_,nu_,k_]:=(1-KroneckerDelta[mu,nu])1/Delta4[k](
 DB[k]^2-c1 DB[k](
 2Sum[hat[k,rho]^4,{rho,4}]+
 DB[k]×Sum[Boole[rho≠mu &&rho≠nu]hat[k,rho]^2,{rho,4}]
 )+c1^2(
 (Sum[hat[k,rho]^4,{rho,4}])^2+
 DB[k]×Sum[hat[k,rho]^4,{rho,4}]×Sum[Boole[rho≠mu &&rho≠nu]hat[k,rho]^2,{rho,4}]+
 DB[k]^2×Product[If[rho≠mu &&rho≠nu,hat[k,rho]^2,1],{rho,4}]
 )
)
(* Lüscher Weisz gluon propagator *)
GLW[mu_,nu_,k_]:=a^2 𝕀/DB[k]^2( hat[k,mu]×hat[k,nu] +
 Sum[(hat[k,sigma]KroneckerDelta[mu,nu]-hat[k,nu]KroneckerDelta[mu,sigma])hat[k,sigma]×
ALW[sigma,nu,k],{sigma,4}])//myVecExpand
```

Fermion Propagator

```
(* Wilson fermion propagator *)
DFWil[k_]:=1/4( 4 Sum[Sin[k[i]]^2,{i,4}]+r^2DB[k]^2)//myVecExpand
SWil[k_]:=a (- I Sum[ γ[i]Sin[ k[i]],{i,4}]+r/2 DB[k]𝕀 )/DFWil[k]//myVecExpand
```

```
(* Brillouin fermion propagator *)
Δbri[k_]:=4((Cos[k[1]/2]Cos[k[2]/2]Cos[k[3]/2]Cos[k[4]/2])^2-1)
Diso[k_,mu_]:=I/27 Sin[k[mu]]Product[If[nu==mu,1,(Cos[k[nu]]+2)],{nu,4}]
DFBri[k_]:=1/4 r^2 Δbri[k]^2-Sum[Diso[k,mu]^2,{mu,4}]
SBri[k_]:=a (- Sum[ γ[mu]×Diso[k,mu],{mu,4}]-1/2 r Δbri[k]𝕀 )/DFBri[k]//myVecExpand
```

ggg Vertex

In[ ]:=
```
Vg30[mu_,nu_,rho_,k1_,k2_,k3_]:=
KroneckerDelta[mu,nu]hat[k1-k2,rho]Cos[1/2 k3[mu]]
Vg31[mu_,nu_,rho_,k1_,k2_,k3_]:=
8Vg30[mu,nu,rho,k1,k2,k3]+
KroneckerDelta[mu,nu](
 Cos[1/2 k3[mu]](
  hat[k1-k2,mu](KroneckerDelta[mu,rho]DB[k3]-hat[k3,mu]×hat[k3,rho])-
  hat[k1-k2,rho](hat[k1,rho]^2+hat[k2,rho]^2)
 )+
 hat[k1-k2,rho](
  hat[k1,mu]×hat[k2,mu]-2Cos[1/2 k1[mu]]Cos[1/2 k2[mu]]hat[k3,mu]^2))
(* Lüscher-Weisz ggg-vertex *)
(* f[A,B,C] Vg3[mu_,nu_,rho_,k1_,k2_,k3_] *)
Vg3LW[mu_,nu_,rho_,k1_,k2_,k3_]:=
- 1/6I g0 /a(
 c0 (Vg30[mu,nu,rho,k1,k2,k3]+
  Vg30[rho,mu,nu,k3,k1,k2]+
  Vg30[nu,rho,mu,k2,k3,k1])+
 c1  (Vg31[mu,nu,rho,k1,k2,k3]+
  Vg31[rho,mu,nu,k3,k1,k2]+
  Vg31[nu,rho,mu,k2,k3,k1]))I//myVecExpand
```

Color factors

```
myColor[expr_]:=expr/.{T[a_,b_,a_]:>-1/(2Nc) T[b],T[a_,a_,b_]:>CF T[b],T[b_,a_,a_]:>CF T
```

Self-Energy

Extracting Σ_0, Σ_1

```
Sigma0[expr_]:=1/4Tr[expr]/.p1→0/.p2→0/.p3→0/.p4→0;
```

$c_{SW}^{(1)}$

Extracting G_1

```
D1[Λ_,mu_]:= D[Tr[Λ[mu]],pv〚mu〛]+ D[Tr[Λ[mu]],qv〚mu〛]/.pi_/;MemberQ[Join[pv,qv],pi]→ 0
D2[Λ_,mu_,nu_]:= D[Tr[Λ[mu].γ[nu].γ[mu]],pv〚nu〛]-D[Tr[Λ[mu].γ[nu].γ[mu]],qv〚nu〛]/.pi_
G1[Λ_,mu_,nu_]:=-1/8(D1[Λ,mu]-D2[Λ,mu,nu]);
```

Numerical Integration

In[∘]:=

```
myNumInt[expr_,p_]:=Flatten[{Chop[NIntegrate[expr, {k1,-Pi,Pi}, {k2,-Pi,Pi}, {k3,-Pi,P
 Method→{"LocalAdaptive","SymbolicProcessing"→0},PrecisionGoal→p,
 IntegrationMonitor:→((error=Through[#1@"Error"])&) ]],error}/(2 Pi)^4 ]
```

```
myrloop[expr_,n1_,n2_]:={out={0,0,0,0,0,0,0,0,0,0,0,0};,
 For[i=0,i<11,i++,{
 Int1=AbsoluteTiming[FullForm[myNumInt[expr/.r→(5/10+i/10),n1]]],
Int2=AbsoluteTiming[FullForm[myNumInt[expr/.r→(5/10+i/10),n2]]],
 ee=Int2[[2]][[1,1]],
 ii=Int2[[2]][[1,1]]-Int1[[2]][[1,1]],
 nn=Abs[Ceiling[Log[10,Abs[ee]]]],
 ff=Abs[Ceiling[Log[10,Abs[ii]]]],
 tt=Int1[[1]]+Int2[[1]];
 out[[i+1]]={ee,ff},
 If[tt>3600,{tt=tt/3600,hms="h"},If[tt>60,{tt=tt/60,hms="m"},hms="s"]],
 Print["time: ",NumberForm[tt//N,{4,2}],hms,"  $",NumberForm[(5/10+i/10)//N,{2,1}],"$ &
NumberForm[ee,{nn+ff,ff}],"(1)$ &"]}],Clear[i];};
```