Self-Energy

(Written by Maximilian Ammer, 2023)
This notebook calculates the fermion self-energy at one loop in lattice perturbation theory.
By choosing c1=0 or c1=-1/12 one can choose between the plaquette-(Wilson) gluon action or the Lüscher-Weisz gluon action.
By loading the appropriate files for the definition of the vertices V1, V2, V3 one can choose between the Wilson and the Brillouin fermion actions.

```
SetDirectory["~path~"];
```

In[ ]:=
```
Get["myLPT.m"]
```

Definitions

Parameters

In[ ]:=
```
(* Wilson parameter*)
```

In[ ]:=
```
r = 1;
```

In[ ]:=
```
(* Parameters of the Lüscher-Weisz gluon action*)
```

In[ ]:=
```
c1 = 0;(*-1/12;*)
c0 = 1 - 8 c1;
```

In[ ]:=
```
(* Parameters of the Brillouin action *)
λ0 = -240 / 64; λ1 =  8 / 64; λ2 =  4 / 64; λ3 = 2 / 64; λ4 =  1 / 64;
ϱ1 = 64 / 432;  ϱ2 =  16 / 432;  ϱ3 = 4 / 432;  ϱ4 = 1 / 432;
```

Feynman Rules

Propagators

```
(* Gluon Propagator*)
```

In[ ]:=
```
G[mu_, nu_, k_] := GLW[mu, nu, k]
```

```
(* Fermion Propagator (Wilson or Brillouin)*)
S[k_] := SWil[k](*SBri[k]*)
```

Vertices

In[ ]:=
```
(* ggg-vertex *)
Vg3[mu_, nu_, rho_, k1_, k2_, k3_] := Vg3LW[mu, nu, rho, k1, k2, k3]
```

In[ ]:=
```
(* qqg-vertex *)
```

```
In[ ]:= DumpGet["V1_Wilson.mx"]
        DumpGet["V1_Brillouin.mx"]
        DumpGet["V1_Clover.mx"];
```

```
In[ ]:= V1[mu_, p_, q_] :=
         (V1Wilson(*V1Brillouin*)[mu, p, q] + V1Clover[mu, p, q]) // myVecExpand
```

```
In[ ]:= (* qqgg-vertex *)
```

```
In[ ]:= DumpGet["V2_Wilson.mx"]
        DumpGet["V2_Brillouin.mx"]
        DumpGet["V2_Clover.mx"]
```

```
In[ ]:= V2[mu_, nu_, p_, q_, k1_, k2_] :=
         a (V2Wilson[mu, nu, p, q] (*V2Brillouin[mu,nu,p,q,k1,k2]*) +
             V2Clover[mu, nu, p, q, k1, k2]) // myVecExpand
```

Calculation

Self-energy tadpole

```
(* Color factor *)
SEtadpoleColor = T[a, a] // myColor
Out[ ]=
CF
```

```
In[ ]:= (* Diagram *)
        SEtadpole = Sum[ G[mu, nu, k].V2[mu, nu, p, p, k, -k] / a^3, {mu, 4}, {nu, 4}];
```

Σ_0

```
In[ ]:= S0tadpole = Sigma0[SEtadpole];
```

```
In[ ]:= S0tadpoleN = myNumInt[16 Pi^2 S0tadpole / g0^2, 6]
Out[ ]=
        {-48.9321, 0.0000110734}
```

Self-energy sunset

```
(* Color factor *)
SEsunsetColor = T[a, a] // myColor
Out[ ]=
CF
```

```
In[ ]:= (* Diagram *)
        SEsunset =
          Sum[G[mu, nu, p - k].V1[mu, k, p].S[k].V1[nu, p, k], {mu, 4}, {nu, 4}] / a^3;
```

Σ_0

In[ ]:=
```
S0sunset = Sigma0[SEsunset];
```

In[ ]:=
```
S0sunsetN0 = myNumInt[16 Pi^2 S0sunset / g0^2 /. csw → 0, 4]
S0sunsetN1 = myNumInt[16 Pi^2 Coefficient[S0sunset, csw] / g0^2, 4]
S0sunsetN2 = myNumInt[16 Pi^2 Coefficient[S0sunset, csw^2] / g0^2 /. csw → 0, 4]
S0sunsetN = myNumInt[16 Pi^2 S0sunset / g0^2 /. csw → 1, 4]
```

Out[ ]=

$\{-2.50254, 1.35978 \times 10^{-6}\}$

Out[ ]=

$\{13.7331, 7.52485 \times 10^{-9}\}$

Out[ ]=

$\{5.71497, 9.18194 \times 10^{-6}\}$

Out[ ]=

$\{16.946, 0.0000242284\}$

In[ ]:=
```
S0sunsetN0[[1]] + csw0 S0sunsetN1[[1]] + csw0^2 S0sunsetN2[[1]]
```

Out[ ]=

$-2.50254 + 13.7331\,\mathrm{csw0} + 5.71497\,\mathrm{csw0}^2$

Sum

Σ_0

In[ ]:=
```
HoldForm[g0^2 / (16 Pi^2)]
  (SEtadpoleColor S0tadpoleN[[1]] + SEsunsetColor S0sunsetN[[1]])
```

Out[ ]=

$$-31.9861\,\mathrm{CF}\,\frac{\mathrm{g0}^2}{16\,\pi^2}$$