Pensieve header: A fresh implementation of baby DoPeGDO - first working version. Continues pensieve://2020-09/, pensieve://2020-03/Testing123.nb, and pensieve://People/VanDerVeen/TimidHeisenbergRGeneralForm@.nb.

$\mathbb{E}[\omega, Q, P]$ represents ωe^{Q+P}, where ω is a scalar, Q is an ϵ-free quadratic, and P is a perturbation (it is ill-advised to include ω in P because then it will have log terms).

Scheme: 𝔼_[_] // 𝔼_[_] calls FZip or Zip, which are functionally the same. Zip works by handling the quadratic part and calling PZip for the perturbation-only part. PZip works by iteratively solving the synthesis equation. FZip works by encapsulating coefficients, calling Zip, and back-substituting.

Initialization and “Define” Code

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\BabyDoPeGDO"];
Once[<< KnotTheory`];
Once[Get@"../Profile/Profile.m"];
```

```
Loading KnotTheory` version of February 2, 2020, 10:53:45.2097.
Read more at http://katlas.org/wiki/KnotTheory.
This is Profile.m of http://www.drorbn.net/AcademicPensieve/Projects/Profile/.
This version: April 2020. Original version: July 1994.
```

```
In[∘]:= $k=1;
```

```
In[∘]:= CCF[ℰ_] := ExpandDenominator@ExpandNumerator@Together[ℰ];
CF[ℰ_List] := CF /@ ℰ;
CF[sd_SeriesData] := MapAt[CF, sd, 3];
CF[ℰ_] := PP_CF@Module[
    {vs = Cases[ℰ, (y | x | η | ξ)_, ∞] ⋃ {y | x | η | ξ}},
    Total[CoefficientRules[Expand[ℰ], vs] /. (ps_ → c_) ⧴ CCF[c] (Times @@ vs^ps)]
   ];
CF[ℰ_𝔼] := CF /@ ℰ;
CF[𝔼_sp___[ℰs___]] := CF /@ 𝔼_sp[ℰs];
```

Define[lhs = rhs, ...] defines the lhs to be rhs, except that rhs is computed only once for each value of $k. Fancy Mathematica not for the faint of heart. Most readers should ignore.

In[•]:=
```
SetAttributes[Define, HoldAll];
Define[def_, defs__] := (Define[def]; Define[defs];);
Define[op_is__ = ℰ_] := Module[{SD, ii, jj, kk, isp, nis, nisp, sis}, Block[{i, j, k},
  ReleaseHold[Hold[
     SD[op_{nisp,$k_Integer}, Block[{i, j, k}, op_{isp,$k} = ℰ; op_{nis,$k}]];
     SD[op_isp, op_{{is},$k}]; SD[op_sis__, op_{sis}];
    ] /. {SD → SetDelayed,
     isp → {is} /. {i → i_, j → j_, k → k_},
     nis → {is} /. {i → ii, j → jj, k → kk},
     nisp → {is} /. {i → ii_, j → jj_, k → kk_}
    }] ]]
```

The Basic Tensors

In[•]:=
```
Define[m_{i,j→k} = 𝔼_{{i,j}→{k}}[1, -ξ_i η_j + (η_i + η_j) y_k + (ξ_i + ξ_j) x_k, 0]]
```

In[•]:=
```
Define[
 R_{i,j} = 𝔼_{{}→{i,j}}[1, (-1 + T) x_j (y_i - y_j),
   ϵ (-1/2 (1 - T) x_j^2 y_i^2 + x_i x_j y_i y_j + 1/2 (1 - 3 T) x_j^2 y_i y_j) + O[ϵ]^2],
 R̄_{i,j} = 𝔼_{{}→{i,j}}[1, (-1 + 1/T) x_j (y_i - y_j),
   ϵ (-((-1 + T) x_i x_j y_i^2)/T^2 - ((1 - T) x_j^2 y_i^2)/(2 T^3) - (x_i x_j y_i y_j)/T^2 - ((-1 - T) x_j^2 y_i y_j)/(2 T^3)) + O[ϵ]^2],
 CC_i = 𝔼_{{}→{i}}[√T, 0, -(ϵ x_i y_i)/T + O[ϵ]^2],
 C̄C̄_i = 𝔼_{{}→{i}}[1/√T, 0, (ϵ x_i y_i)/T + O[ϵ]^2]
]
```

In[•]:=
```
Define[Kink_i = CC_3 R_{1,2} // m_{2,3→2} // m_{2,1→i}, K̄ink_i = CC_3 R̄_{1,2} // m_{1,3→1} // m_{1,2→i}]
```

The Main Program

Variables and their duals:

In[•]:=
```
{y*, x*, η*, ξ*} = {η, ξ, y, x};
(vs_List)* := (v ↦ v*) /@ vs;
(u_i_)* := (u*)_i;
```

𝔼 operations:

In[∘]:=
```
𝔼 /: 𝔼[ω1_, Q1_, P1_] ≡ 𝔼[ω2_, Q2_, P2_] :=
  CF[ω1 == ω2] ∧ CF[Q1 == Q2] ∧ CF[Normal[P1 - P2] == 0];
𝔼 /: 𝔼[ω1_, Q1_, P1_] 𝔼[ω2_, Q2_, P2_] := 𝔼[ω1 ω2, Q1 + Q2, P1 + P2];
𝔼_{d1_→r1_}[ℰ1s___] ≡ 𝔼_{d2_→r2_}[ℰ2s___] ^:= (d1 == d2) ∧ (r1 == r2) ∧ (𝔼[ℰ1s] ≡ 𝔼[ℰ2s]);
𝔼_{d1_→r1_}[ℰ1s___] 𝔼_{d2_→r2_}[ℰ2s___] ^:= 𝔼_{(d1⋃d2)→(r1⋃r2)} @@ (𝔼[ℰ1s] 𝔼[ℰ2s]);
𝔼_{dr_}[ℰs___]_{$k_} := 𝔼_{dr} @@ 𝔼[ℰs]_{$k};
```

In[∘]:=
```
𝔼_{d1_→r1_}[ℰ1s___] // 𝔼_{d2_→r2_}[ℰ2s___] := Module[{is = r1 ⋂ d2, lvs},
  lvs = Flatten@Table[{x_{$@i}, y_{$@i}}, {i, is}];
  𝔼_{(d1⋃Complement[d2,is])→(r2⋃Complement[r1,is])} @@ (Zip_{lvs⋃lvs*}[lvs*.lvs, Times[
      𝔼[ℰ1s] /. Table[(v : x | y)_i → v_{$@i}, {i, is}],
      𝔼[ℰ2s] /. Table[(v : ξ | η)_i → v_{$@i}, {i, is}]
     ]])
 ]
```

$$[F:\mathcal{E}]_B := \mathbb{e}^{\frac{1}{2}\sum_{i,j\in B}F_{ij}\partial_{z_i}\partial_{z_j}}\mathcal{E} \quad\text{and}\quad \langle F:\mathcal{E}\rangle_B := [F:\mathcal{E}]_B|_{z_B\to 0},$$

where $\mathcal{E}$ is a docile perturbed Gaussian. The following lemma allows us to restrict to the case where $\mathcal{E}$ has no B-B quadratic part:

Lemma 1. With convergences left to the reader,

$$\left\langle F:\mathcal{E}\mathbb{e}^{\frac{1}{2}\sum_{i,j\in B}G_{ij}z_iz_j}\right\rangle_B = \det(1-GF)^{-1/2}\left\langle F(1-GF)^{-1}:\mathcal{E}\right\rangle_B.$$

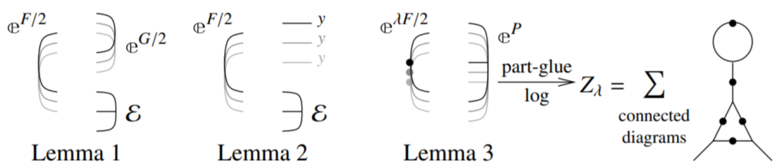
The next lemma dispatches the case where $\mathcal{E}$ has a B-linear part:

Lemma 2. $\left\langle F:\mathcal{E}\mathbb{e}^{\sum_{i\in B}y_iz_i}\right\rangle_B = \mathbb{e}^{\frac{1}{2}\sum_{i,j\in B}F_{ij}y_iy_j}\left\langle F:\ \mathcal{E}|_{z_B\to z_B+Fy_B}\right\rangle_B$.

Finally, we deal with the docile perturbation case:

Lemma 3. With an extra variable λ, $Z_\lambda := \log[\lambda F:\mathbb{e}^P]_B$ satisfies and is determined by the following PDE / IVP:

$$Z_0 = P \quad\text{and}\quad \partial_\lambda Z_\lambda = \frac{1}{2}\sum_{i,j\in B}F_{ij}\left(\partial_{z_i}\partial_{z_j}Z_\lambda + (\partial_{z_i}Z_\lambda)(\partial_{z_j}Z_\lambda)\right).$$



In[∘]:=
```
Zip_{vs_}[ℱ_, ℰ_] := ⟨ℱ, ℰ⟩ // Zip1_{vs} // Zip2_{vs} // Zip3_{vs}
```

Getting rid of the quadratic.

Lemma 1. With convergences left to the reader,

$$\left\langle F:\mathcal{E}\mathbb{e}^{\frac{1}{2}\sum_{i,j\in B}G_{ij}z_iz_j}\right\rangle_B = \det(1-GF)^{-1/2}\left\langle F(1-GF)^{-1}:\mathcal{E}\right\rangle_B$$

In[ ]:=
```
Zip1_vs_ @⟨𝓕_, 𝔼[ω_, Q_, P_]⟩ := PP_Zip1@Module[{𝓘, F, G, u, v},
  𝓘 = IdentityMatrix@Length@vs;
  F = Table[∂_{u,v}𝓕, {u, vs*}, {v, vs*}];
  G = Table[∂_{u,v}Q, {u, vs}, {v, vs}];
  CF /@ ⟨vs*.F.Inverse[𝓘 - G.F].vs* / 2, 𝔼[ω Det[𝓘 - G.F]^(-1/2), Q - vs.G.vs / 2, P]⟩
]
```

Getting rid of linear terms.

Lemma 2. $\left\langle F:\mathcal{E}\,\mathbb{e}^{\sum_{i\in B}y_iz_i}\right\rangle_B=\mathbb{e}^{\frac{1}{2}\sum_{i,j\in B}F_{ij}y_iy_j}\left\langle F:\ \mathcal{E}|_{z_B\to z_B+Fy_B}\right\rangle_B$.

In[ ]:=
```
Zip2_vs_ @⟨𝓕_, 𝔼[ω_, Q_, P_]⟩ := PP_Zip2@Module[{F, Y, u, v},
  F = Table[∂_{u,v}𝓕, {u, vs*}, {v, vs*}];
  Y = Table[∂_v Q, {v, vs}];
  CF /@ ⟨𝓕, 𝔼[ω, Q - Y.vs + Y.F.Y / 2, P /. Thread[vs → vs + F.Y]]⟩
]
```

Dealing with Feynman diagrams.

Lemma 3. With an extra variable λ, $Z_\lambda:=\log[\lambda F:\mathbb{e}^P]_B$ satisfies and is determined by the following PDE / IVP:

$$Z_0=P\quad\text{and}\quad\partial_\lambda Z_\lambda=\frac{1}{2}\sum_{i,j\in B}F_{ij}\left(\partial_{z_i}\partial_{z_j}Z_\lambda+(\partial_{z_i}Z_\lambda)(\partial_{z_j}Z_\lambda)\right).$$

Note that the power m of λ is at most $k-1+\frac{2k+2}{2}=2k$.

In[ ]:=
```
Zip3_vs_ @⟨𝓕_, 𝔼[ω_, Q_, P_]⟩ := PP_Zip3@Module[{u, v, m, j},
  Z[0] = P;
  For[m = 0, m < 2 $k, ++m,
   Z[m + 1] = CF[1/(2 (m + 1))
      Sum[∂_{u*,v*}𝓕 (∂_{u,v}Z[m] + Sum[(∂_u Z[j]) (∂_v Z[m - j]), {j, 0, m}]), {u, vs}, {v, vs}]]
  ];
  𝔼[ω, Q, CF[Sum[Z[m], {m, 0, 2 $k}] /. Table[v → 0, {v, vs}]]]
]
```

Some Knot Theory

In[∘]:=

```
RVK[pd_PD] := Module[{n, xs, x, rots, front = {0}, k},
   n = Length@pd; rots = Table[0, {2 n}];
   xs = Cases[pd, x_X :> { Xp[x[[4]], x[[1]]]  PositiveQ@x
                           Xm[x[[2]], x[[1]]]  True        ];
   For[k = 0, k < 2 n, ++k, If[k == 0 ∨ FreeQ[front, -k],
     front = Flatten[front /. k → (xs /. {
            Xp[k + 1, l_] | Xm[l_, k + 1] :> {l, k + 1, 1 - l},
            Xp[l_, k + 1] | Xm[k + 1, l_] :> (++rots[[l]]; {1 - l, k + 1, l})
           })],
     Cases[front, k | -k] /. {k, -k} :> --rots[[k + 1]];
    ]];
   RVK[xs, rots] ];
RVK[K_] := RVK[PD[K]];
```

In[∘]:=

```
rot[i_, 0] := 𝔼_{{}→{i}}[1, 0, 0];
rot[i_, n_] := Module[{j},
   rot[i, n] = If[n > 0, rot[i, n - 1] CC_j, rot[i, n + 1] CC̄_j] // m_{i,j→i}];
```

In[•]:=
```
Z[K_] := Z[RVK@K];
Z[rvk_RVK] := (*Z[rvk] =*)
 Module[{todo, n, rots, ζ, done, st, cx, ζ1, i, j, k, k1, k2, k3},
  {todo, rots} = List @@ rvk;
  AppendTo[rots, 0];
  n = Length[todo];
  ζ = 𝔼_{}→{0}[1, 0, 0];
  done = {0};
  st = Range[0, 2 n + 1];
  While[{} =!= ($M = todo),
   {cx} = MaximalBy[todo, Length[done ⋂ {#[[1]], #[[2]], #[[1]] - 1, #[[2]] - 1}] &, 1];
   {i, j} = List @@ cx;
   ζ1 = Switch[Head[cx],
     Xp, (R_{i,j} OverBar[Kink_k]) // m_{j,k→j},
     Xm, (OverBar[R]_{i,j} Kink_k) // m_{j,k→j}
    ];
   ζ1 = (rot[k, rots[[i]]] ζ1) // m_{k,i→i}; rots[[i]] = 0;
   ζ1 = (ζ1 rot[k, rots[[i + 1]]]) // m_{i,k→i}; rots[[i + 1]] = 0;
   ζ1 = (rot[k, rots[[j]]] ζ1) // m_{k,j→j}; rots[[j]] = 0;
   ζ1 = (ζ1 rot[k, rots[[j + 1]]]) // m_{j,k→j}; rots[[j + 1]] = 0;
   ζ *= ζ1;
   If[MemberQ[done, i], ζ = ζ // m_{i,i+1→i}; st = st /. st[[i + 2]] → st[[i + 1]]];
   If[MemberQ[done, i - 1], ζ = ζ // m_{st[[i]],i→st[[i]]}; st = st /. st[[i + 1]] → st[[i]]];
   If[MemberQ[done, j], ζ = ζ // m_{j,j+1→j}; st = st /. st[[j + 2]] → st[[j + 1]]];
   If[MemberQ[done, j - 1], ζ = ζ // m_{st[[j]],j→st[[j]]}; st = st /. st[[j + 1]] → st[[j]]];
   done = done ⋃ {i - 1, i, j - 1, j};
   todo = DeleteCases[todo, cx]
  ];
  CF /@ (ζ (*/. {x_0→x,y_0→y,a_0→a}*))
 ]
```

In[•]:=
```
BeginProfile[];
PopupWindow[Button["Show Profile Monitor"],
 Dynamic[PrintProfile[], UpdateInterval → 3, TrackedSymbols → {}]]
```

Out[•]= Show Profile Monitor

In[•]:=
```
NewBit[K_] := Module[{Alex = Alexander[K][T]},
  T^3 Alex^2/(T - 1) Coefficient[Z[K][[3]], ϵ] // Factor]
```

In[•]:=
```
NewBit /@ AllKnots[{3, 5}]
```

KnotTheory: Loading precomputed data in PD4Knots`.

Out[•]= $\left\{2-T+T^2,\ (1+T)\ \left(1-3\,T+T^2\right),\ \frac{4-3\,T+5\,T^2-3\,T^3+3\,T^4-T^5+T^6}{T^2},\ 9-11\,T+7\,T^2-T^3\right\}$

In[∘]:=
```
(*Two knots with equal Alexander, new bit does not agree*)
Alexander[Knot[6, 1]] == Alexander[Knot[9, 46]]
Timing[NewBit[Knot[6, 1]] == NewBit[Knot[9, 46]]]
```

Out[∘]= True

Out[∘]= $\{88.4688, 5 - 11\,T - T^2 + 3\,T^3 == 7 - 21\,T + 9\,T^2 + T^3\}$

In[∘]:=
```
PrintProfile[]
```

Out[∘]=
```
ProfileRoot is root. Profiled time: 145.863
  (  18)   0.078/   0.078 above CF
  ( 237)   1.390/   6.161 above Zip1
  ( 237)   0.877/  32.253 above Zip2
  ( 237)  87.547/ 107.370 above Zip3
Zip3: called 237 times, time in 87.547/107.371
  ( 237)  87.547/ 107.370 under ProfileRoot
  ( 710)  19.824/  19.824 above CF
CF: called 2624 times, time in 56.049/56.049
  (  18)   0.078/   0.078 under ProfileRoot
  ( 948)   4.771/   4.771 under Zip1
  ( 948)  31.376/  31.376 under Zip2
  ( 710)  19.824/  19.824 under Zip3
Zip1: called 237 times, time in 1.39/6.161
  ( 237)   1.390/   6.161 under ProfileRoot
  ( 948)   4.771/   4.771 above CF
Zip2: called 237 times, time in 0.877/32.253
  ( 237)   0.877/  32.253 under ProfileRoot
  ( 948)  31.376/  31.376 above CF
```