In this worksheet show how to derive the differential operator for the Cross Witten diagram of section 4.2 of the paper
Algorithm for differential equations for Feynman integrals in general dimensions https://arxiv.org/abs/2401.09908
by
Leonardo de la Cruz and Pierre Vanhove

```
SetOptions[$FrontEnd, EvaluationCompletionAction → "ShowTiming"]
```

This worksheet needs FiniteFlow https://arxiv.org/abs/1905.08019

```
<< FiniteFlow`
```

Routines for the Griffiths-Dwork reduction

The routines follow the step of the reduction given in sections 3.1.1, 3.1.2 and 3.1.3 of the paper

```
SaveFile = False;
(* Change this to True if you want to save intermediate results*)
```

```
JacIdeal[pol_, vars__] := D[pol, #] & /@ vars;
DegreeHomogeneity[f_, xxx__] :=
  Exponent[Simplify[(f /. Table[xxtmp → xxtmp λ, {xxtmp, xxx}]) / f], λ];
```

```
allMonoms[n_, deg_, x_] :=
 DeleteCases[Coefficient[(List @@ Expand[(1 + Total[Array[x, n]]) ^ deg] /.
      j_Integer * monom : _ :> monom) /. {x[i_] :> λ x[i]}, λ ^ deg], 0]
```

```
homPols[n_, deg_, sym_] :=
  Module[{mons}, If[deg == 0, sym, mons = allMonoms[n, deg, x];
    Return[(Plus @@ (sym /@ ((Exponent[#, Array[x, n]] & /@ mons)) mons)) /.
      {sym[A__] :> sym[Sequence @@ A]}];]];
```

```
getEquations[pol_, vars__] :=
  (CoefficientArrays[{pol}, vars]["NonzeroValues"]) // Through // Flatten;
```

Reduction with respect to F

```
ReductionF[M_, degreeP_, P_, coeffPtmp_, varunknown_, Fpol_, var_, varlong_] :=
 Block[{degtmp, poltoreduce, Chattmp, systmp, varsystmp,
   solfiniteflowtmp, Chattmpresult, coeffPresult, JacFtmp, Filenametmp},
  Print["Reduction with respect to Jac(", Fpol, ")"];
  JacFtmp = JacIdeal[Fpol, var];
  degtmp = degreeP - DegreeHomogeneity[JacFtmp, var][[1]];
  Print["degree coefficient ", degtmp];
  Chattmp =
   Table[homPols[Length[var], degtmp, varunknown[r]], {r, Length[var]}];
  If[ListQ[P], poltoreduce = M + coeffPtmp.P - Chattmp.JacFtmp,
   poltoreduce = M + coeffPtmp * P - Chattmp.JacFtmp];
  systmp = getEquations[poltoreduce, var];
  varsystmp = Complement[Variables[systmp], varlong];
  Print["Number of equations ", Length[systmp],
   ", number of variables ", Length[varsystmp]];
  If[SaveFile, Filenametmp =
     "Reduction-F-Case-" <> ToString[degreeP] <> "-" <> DateString[
       {"ISODate", "-", "Hour", ":", "Minute", ":", "Second"}] <> ".txt";
    Print["system saved in ", Filenametmp];
    Save[Filenametmp, {systmp, varsystmp}];] × Print["Calling Finite Flow "];
  solfiniteflowtmp = FFDenseSolve[Equal[#, 0] & /@ systmp,
    varsystmp, "ApplyFunction" → Together, MaxPrimes → 20];
  If[solfiniteflowtmp == FFImpossible, Echo["System cannot be solved"]];
  Print["finite flow done -- length result ", Length[solfiniteflowtmp]];
  Chattmpresult = Chattmp /. solfiniteflowtmp // Expand;
  coeffPresult = coeffPtmp /. solfiniteflowtmp // Expand;
  {Chattmpresult, degtmp, coeffPresult, solfiniteflowtmp}]
```

Reduction with respect to U

```
ReductionU[M_, degree_, Qnametmp_, Upol_, var_, varlong_] :=
 Block[{degtmp, poltoreduce, Qtmp, systmp, varsystmp,
   solfiniteflowtmp, polU, Mresult, Qtmpresult, JacUtmp, Filenametmp},
  Print["Reduction with respect to Jac(", Upol, ")"];
  JacUtmp = JacIdeal[Upol, var];
  polU = M.JacUtmp;
  degtmp = degree - 1;
  Print["Degree quotient ", degtmp];
  Qtmp = homPols[Length[var], degtmp, Qnametmp];
  poltoreduce = polU - Qtmp * Upol;
  systmp = DeleteCases[Flatten[CoefficientList[poltoreduce, var]], 0];
  varsystmp = Complement[Variables[systmp], varlong];
  Print["Number of equations ", Length[systmp],
   ", number of variables ", Length[varsystmp]];
  If[SaveFile,
    Filenametmp = "Reduction-U-Case-" <> ToString[degree] <> "-" <> DateString[
       {"ISODate", "-", "Hour", ":", "Minute", ":", "Second"}] <> ".txt";
    Print["system saved in ", Filenametmp];
    Save[Filenametmp, {systmp, varsystmp}];] × Print["Calling Finite Flow "];
  solfiniteflowtmp = FFDenseSolve[Equal[#, 0] & /@ systmp,
    varsystmp, "ApplyFunction" → Together, MaxPrimes → 20];
  If[solfiniteflowtmp == FFImpossible, Echo["System cannot be solved"]];
  Print["finite flow done -- length result ", Length[solfiniteflowtmp]];
  Mresult = M /. solfiniteflowtmp;
  Qtmpresult = Qtmp /. solfiniteflowtmp;
  {Mresult, degtmp, Qtmpresult, solfiniteflowtmp}]
```

The boundary term using equation 3.39 without including the factor of Ω_Γ

```
Bvec[Chat_, derivativeorder_, Fpol_, powerF_] :=
  Chat / (Fpol ^ (derivativeorder - 1)) / (derivativeorder - 1 + powerF);
```

Routine for the parametric representation from the propagator representation of a Feynman graph

```
PropagatorToParametric3[listprop_, varloop_, rules_] :=
 Block[{listtmp, listtmp2, powertmp, Qtmp, Qtmp2, preftmp,
   mattmp, Utmp, Jtmp, ptmp, Ftmp, omegatmp}, {listtmp = listprop;
   listtmp2 = Table[List @@ listtmp[[i]], {i, Length[listtmp]}];
   powertmp = Table[-listtmp2[[i, 2]] // Simplify, {i, Length[listtmp2]}];
   Print[listtmp2];
   Qtmp = Sum[x[i] × listtmp2[[i, 1]], {i, Length[listtmp2]}] /. xx_ . yy_ → xx yy;
   preftmp = Product[
     x[i] ^ (powertmp[[i]] - 1) / Gamma[powertmp[[i]]], {i, Length[listtmp2]}];
   mattmp = DiagonalMatrix[Table[Coefficient[Qtmp, varloop[[i]], 2], {i,
        Length[varloop]}]] + Table[Coefficient[Coefficient[Qtmp, varloop[[i]]],
        varloop[[j]]] / 2, {i, 1, Length[varloop]}, {j, 1, Length[varloop]}];
   Utmp = Det[mattmp];
   Qtmp2 = Qtmp - varloop.(mattmp.varloop) - ptmp.varloop // FullSimplify;
   ptmp =
    Table[Coefficient[Qtmp2, varloop[[i]]] // Simplify, {i, Length[varloop]}];
   Jtmp = Qtmp - varloop.(mattmp.varloop) - ptmp.varloop // FullSimplify;
   Ftmp = Simplify[Expand[Together[
        Det[mattmp] (Jtmp - 1 / 4 ptmp.(Inverse[mattmp].ptmp))]] //. rules];
   omegatmp = Plus @@ powertmp - Length[varloop] * dim / 2;
   {preftmp * Pi ^ (dim / 2 * Length[varloop]) * Gamma[omegatmp] *
     Utmp ^ (omegatmp - dim / 2) / Ftmp ^ (omegatmp), Utmp, Ftmp, (preftmp * Pi ^
        (dim / 2 * Length[varloop]) * Gamma[omegatmp]) /. x[i_] :> 1, omegatmp}}]
```

Cross Witten diagram from equation (4.9) [arXiv:2201.09626]

Initialisation

Derive the parametric representation from the propagator representation

```
WittenCross =
  PropagatorToParametric3[{1 / (X.X), 1 / ((X - u1).(X - u1)) ^ ((1 - 4 ϵ)),
      1 / ((X - uz).(X - uz))}, {X}, {u u1 → 0, u uz → 0, u1^2 → 1, uz^2 → ξ ξb,
      u1 uz → (ξ + ξb) / 2, u^2 → 1}][[1]] /. dim → 4 - 4 ϵ // Simplify;
```

```
{{X.X, -1}, {(-u1 + X).(-u1 + X), -1 + 4 ϵ}, {(-uz + X).(-uz + X), -1}}
```

```
intcross = WittenCross[[1]]
```

$$\left(\pi^{2-2\epsilon}\,\text{Gamma}[1-2\epsilon]\,x[2]^{-4\epsilon}\left((-1+\xi)\,(-1+\xi b)\,x[2]\times x[3]+x[1]\,(x[2]+\xi\,\xi b\,x[3])\right)^{-1+2\epsilon}\right)\Big/\left(\text{Gamma}[1-4\epsilon]\,(x[1]+x[2]+x[3])\right)$$

```
ucross = WittenCross[[2]]
```

```
x[1] + x[2] + x[3]
```

```
fcross = WittenCross[[3]]
```

```
(-1 + ξ) (-1 + ξb) x[2] × x[3] + x[1] (x[2] + ξ ξb x[3])
```

```
var = Variables[ucross]
```

```
{x[1], x[2], x[3]}
```

```
Kinematics = Complement[Variables[fcross], Variables[ucross]]
```

```
{ξ, ξb}
```

```
varlong = Join[{ξ, ξb, ϵ}, var]
```

```
{ξ, ξb, ϵ, x[1], x[2], x[3]}
```

```
(*checking homogeneoity*)
```

```
DegreeHomogeneity[intcross, var]
```

```
-3
```

```
powerU =
  Coefficient[Simplify[PowerExpand[Log[intcross]]], Simplify[Log[ucross]]]
```

```
-1
```

```
powerF =
  -Coefficient[Simplify[PowerExpand[Log[intcross]]], Simplify[Log[fcross]]]
```

```
1 - 2 ∊
```

```
powerQ =
  Coefficient[Simplify[PowerExpand[Log[intcross]]], Simplify[Log[x[2]]]]
```

```
-4 ∊
```

```
Q = x[2]
```

Derivative w.r.t. ζ

We derive the partial differential equation of equation (4.16)

```
listderivative =
  Reverse[Table[Simplify[D[intcross, {ζ, i}] / intcross] // Numerator, {i, 2}]]
```

```
{2 (-1 + ∊) (-1 + 2 ∊) (x[2] - ζb (x[1] + x[2]))^2 x[3]^2,
  (-1 + 2 ∊) (-x[2] + ζb (x[1] + x[2])) x[3]}
```

```
listdegree = DegreeHomogeneity[listderivative, var]
```

```
{4, 2}
```

We start at second derivative order in ζ

```
derivativeorder = 2;
RedFStep1 = ReductionF[listderivative[[1]],
   listdegree[[1]], 0, 0, λ1, fcross, var, varlong];
RedUStep1 = ReductionU[RedFStep1[[1]], RedFStep1[[2]], μ3, ucross, var, varlong];
solnRstep1 = ReductionU[RedUStep1[[1]], RedFStep1[[2]], μ31, Q, var, varlong];
(*Building M using equation 3.31*)
Mstep1 =
  (Sum[D[RedUStep1[[1]][[i]], var[[i]]], {i, Length[var]}] + powerU * RedUStep1[[3]] +
     powerQ * solnRstep1[[3]] ) / (derivativeorder - 1 + powerF) /. solnRstep1[[4]];
```

```
Reduction with respect to Jac((-1 + ζ) (-1 + ζb) x[2] × x[3] + x[1] (x[2] + ζ ζb x[3]))
degree coefficient 3
Number of equations 15, number of variables 30
Calling Finite Flow
finite flow done -- length result 15
Reduction with respect to Jac(x[1] + x[2] + x[3])
Degree quotient 2
Number of equations 10, number of variables 21
Calling Finite Flow
finite flow done -- length result 10
Reduction with respect to Jac(x[2])
Degree quotient 2
Number of equations 10, number of variables 17
Calling Finite Flow
finite flow done -- length result 10
```

second step

```
derivativeorder = 1;
RedFStep2 = ReductionF[Mstep1, listdegree[[2]],
   listderivative[[2]], c1, λ2, fcross, var, varlong];
RedUStep2 = ReductionU[RedFStep2[[1]], RedFStep2[[2]], μ2, ucross, var, varlong];
solnRstep2 = ReductionU[RedUStep2[[1]], RedFStep2[[2]], μ22, Q, var, varlong];
(*Building M using equation 3.31*)
Mstep2 =
  (Sum[D[solnRstep2[[1]][[i]], var[[i]]], {i, Length[var]}] + powerU * RedUStep2[[3]] +
     powerQ * solnRstep2[[3]]) / (derivativeorder - 1 + powerF) /. solnRstep2[[4]];
```

```
listcoPF1 =
 {-Mstep2, c1, 1} //. RedFStep1⟦4⟧ //. solnRstep1⟦4⟧ //. RedFStep2⟦4⟧ //.
   RedUStep2⟦4⟧ //. solnRstep2⟦4⟧ // Simplify
```

$$\left\{\frac{(-1+2\epsilon)\left(-\zeta^2+2\epsilon(\zeta+\zeta b)\right)}{(-1+\zeta)\zeta^2(\zeta-\zeta b)},\frac{(-1+\zeta b)\zeta b\left(-1+2\epsilon-2\zeta\lambda_1[3][0,0,1]+2\zeta^2\lambda_1[3][0,0,1]\right)}{(-1+2\epsilon)(-1+\zeta)\zeta(-1+2\zeta b)},1\right\}$$

```
Btotal =
  ( Bvec[solnRstep1⟦1⟧, 2, fcross, 1 - 2 ϵ] +  Bvec[solnRstep2⟦1⟧, 1, fcross,
          1 - 2 ϵ]) intcross //. RedFStep1⟦4⟧ //. solnRstep1⟦4⟧ //.
     RedFStep2⟦4⟧ //. RedUStep2⟦4⟧ //. solnRstep2⟦4⟧ // Expand;
```

```
BtotalVar = {λ1[3][1, 1, 1], λ1[3][1, 2, 0], λ1[3][2, 1, 0],
   λ2[1][1, 0, 0], λ2[3][0, 0, 1], λ2[3][0, 1, 0], λ2[3][1, 0, 0]};
```

```
dBtotal = Sum[D[Btotal⟦i⟧, var⟦i⟧], {i, Length[var]}];
```

```
dBtotalsimp = Collect[dBtotal, BtotalVar, Simplify]
```

$$\Big(\pi^{2-2\epsilon}(-1+2\epsilon)\,\mathrm{Gamma}[1-2\epsilon]\,x[2]^{-4\epsilon}\left((-1+\zeta)(-1+\zeta b)\,x[2]\times x[3]+x[1]\,(x[2]+\zeta\,\zeta b\,x[3])\right)^{-3+2\epsilon}$$
$$\Big(\zeta^2\left((-1+\zeta+\zeta b-\zeta\,\zeta b)\,x[2]\times x[3]+x[1]\,(x[2]-(-2+\zeta)\,\zeta b\,x[3])\right)\left((-1+\zeta b)^2\,x[2]\times x[3]+x[1]\left(x[2]+\zeta b^2\,x[3]\right)\right)+$$
$$2\epsilon\Big(-\zeta b\,x[2]^2\,(x[1]+x[3]-\zeta b\,x[3])^2+\zeta^3\,(-x[2]+\zeta b\,(x[1]+x[2]))\,x[3]\left(\left(1-3\zeta b+2\zeta b^2\right)x[2]\times x[3]+x[1]\left(x[2]+2\zeta b^2\,x[3]\right)\right)-$$
$$\zeta\,x[2]\,(x[1]+x[3]-\zeta b\,x[3])\left(\left(1-5\zeta b+4\zeta b^2\right)x[2]\times x[3]+x[1]\left(x[2]+4\zeta b^2\,x[3]\right)\right)-\zeta^2\,(-x[2]+\zeta b\,(x[1]+x[2]))\,x[3]$$
$$\left(\left(2-7\zeta b+5\zeta b^2\right)x[2]\times x[3]+x[1]\left(-((-2+\zeta b)\,x[2])+4\zeta b^2\,x[3]\right)\right)\Big)\Big)\Big)\Big/$$
$$\left((-1+\zeta)\,\zeta^2\,(\zeta-\zeta b)\,\mathrm{Gamma}[1-4\epsilon]\,(x[1]+x[2]+x[3])\right)$$

```
ApplyPFcross = Sum[D[intcross, {ζ, i}] × listcoPF1⟦i + 1⟧, {i, 0, 2}] // Simplify;
```

We satisfy the identity in eq 3.40

```
checkpf = dBtotalsimp + ApplyPFcross // Simplify
```

```
0
```

We now evaluate the boundary contribution

```
B1simp = Collect[Btotal⟦1⟧, BtotalVar, Simplify];
```

```
B2simp = Collect[Btotal⟦2⟧, BtotalVar, Simplify];
```

```
B3simp = Collect[Btotal[[3]], BtotalVar, Simplify];
```

We evaluate the limit of the various component for x_1, x_2 and x_3 to 0 for ϵ>0

```
B1x10 = Series[B1simp, {x[1], 0, 0}, Assumptions →
    x[2] > 0 && x[3] > 0 && ϵ > 0 && ϵ < 1] // Normal // PowerExpand // Simplify;
```

$$\frac{1}{2\,\text{Gamma}[1-4\epsilon]}\pi^{2-2\epsilon}(-1+\zeta)^{-2+2\epsilon}(-1+\zeta b)^{-1+2\epsilon}\,\text{Gamma}[1-2\epsilon]\,x[2]^{-1-2\epsilon}\,x[3]^{-2+2\epsilon}$$
$$\Bigg(-\Big(\Big(2(-1+2\epsilon)\Big((1-4\epsilon)^2 x[2]^2+\Big(2(3-8\epsilon)\epsilon\zeta^2+\big(1-10\epsilon+20\epsilon^2\big)\zeta b-$$
$$\zeta\big(-1+\epsilon(10-8\zeta b)+\zeta b+4\epsilon^2(-5+4\zeta b)\big)\Big)x[2]\times x[3]-$$
$$2\epsilon\big(\zeta-2\epsilon\zeta+(-3+8\epsilon)\zeta^2+\zeta b-2\epsilon\zeta b\big)x[3]^2\Big)\Big)\Big/$$
$$\Big((\zeta-4\epsilon\zeta)^2(\zeta-\zeta b)(x[2]+x[3])\Big)\Big)+\frac{4\epsilon(-1+\zeta)x[3]\lambda_1[3][1,1,1]}{(-1+\epsilon)(-1+4\epsilon)}-$$
$$\frac{(-1+\zeta)(x[2]-4\epsilon x[2]+4\epsilon\zeta\zeta b\,x[3])\lambda_1[3][1,2,0]}{(-1+\epsilon)(-1+4\epsilon)}+$$
$$\frac{(-1+\zeta)((1-4\epsilon)x[2]+(-1-4\epsilon+\zeta+\zeta b)x[3])\lambda_1[3][2,1,0]}{(-1+\epsilon)(-1+4\epsilon)}-$$
$$\frac{4\epsilon(-1+\zeta)x[3]\lambda_2[1][1,0,0]}{1-6\epsilon+8\epsilon^2}+$$
$$\frac{4\epsilon(-3+8\epsilon)(-1+\zeta)x[3]\lambda_2[3][0,0,1]}{(1-4\epsilon)^2(-1+2\epsilon)}-$$
$$\frac{2(-1+\zeta)(x[2]-4\epsilon x[2]+2\epsilon\zeta\zeta b\,x[3])\lambda_2[3][0,1,0]}{1-6\epsilon+8\epsilon^2}+$$
$$\frac{1}{(1-4\epsilon)^2(-1+2\epsilon)}$$
$$2(-1+\zeta)\Big(-(1-4\epsilon)^2x[2]+\Big(1-\zeta+8\epsilon^2(-1+\zeta(-1+\zeta b)-\zeta b)-$$
$$\zeta b+\epsilon(-2\zeta(-3+\zeta b)+6\zeta b)\Big)x[3]\Big)\lambda_2[3][1,0,0]\Bigg)$$

```
B2x20 =
  Series[B2simp, {x[2], 0, 0}, Assumptions → x[1] > 0 && x[2] > 0 && x[3] > 0 &&
     ϵ > 0 && ϵ < 1] // Normal // PowerExpand // Simplify;
```

```
0
```

```
B3x30 = Assuming[
    x[1] > 0 && x[2] > 0 && x[3] > 0 && ϵ > 0 && ϵ < 1, Series[B3simp, {x[3], 0, 0},
        Assumptions → x[1] > 0 && x[2] > 0 && x[3] > 0 && ϵ > 0 && ϵ < 1] // Normal //
      PowerExpand // FullSimplify // PowerExpand // Simplify];
```

$$\frac{1}{2\,\text{Gamma}[1-4\epsilon]}\pi^{2-2\epsilon}\,\text{Gamma}[1-2\epsilon]\,x[1]^{2(-1+\epsilon)}\,x[2]^{-1-2\epsilon}\left(\frac{2(-1+2\epsilon)\,x[2]^2}{(-1+\xi)\,\xi^2\,(\xi-\xi b)\,(x[1]+x[2])}-\frac{x[2]\,\lambda_1[3][1,2,0]}{-1+\epsilon}+\frac{(-x[1]+x[2])\,\lambda_1[3][2,1,0]}{-1+\epsilon}+\frac{2\,x[2]\,\lambda_2[3][0,1,0]}{1-2\epsilon}+\frac{2\,(-x[1]+x[2])\,\lambda_2[3][1,0,0]}{-1+2\epsilon}\right)$$

We evaluate the limit of the various component for x_1, x_2 and x_3 to ∞ for ϵ>0
The limit x_1 and x_3 to ∞ for ϵ>0 exists only if impose these constraints on the free parameters

```
subλ =
```

$$\left\{\lambda_2[3][1,0,0]\to-\frac{(-1+4\epsilon)\,\lambda_1[3][2,1,0]}{2\,(-1+\epsilon)},\ \lambda_2[3][0,1,0]\to\frac{1}{4\,\xi^2\,\xi b^2}(-1+2\epsilon)\left(\frac{4\,(-1+2\epsilon)\,\xi b\,\left(\xi-2\epsilon\,\xi+(-3+8\epsilon)\,\xi^2+\xi b-2\epsilon\,\xi b\right)}{(-1+4\epsilon)\,(-1+\xi)\,\xi\,(\xi-\xi b)}+\frac{4\,\xi\,\xi b\,\lambda_1[3][1,1,1]}{-1+\epsilon}-\frac{4\,\xi^2\,\xi b^2\,\lambda_1[3][1,2,0]}{-1+\epsilon}+\frac{\xi\,\xi b\,(-1-4\epsilon+\xi+\xi b)\,\lambda_1[3][2,1,0]}{(-1+\epsilon)\,\epsilon}-(\xi\,\xi b\,(-((-1+2\epsilon)\,(1+(-1+4\epsilon)\,\xi))+(-1+4\epsilon)\,(1+2\epsilon\,(-1+\xi))\,\xi b)\,\lambda_1[3][2,1,0])/((-1+\epsilon)\,\epsilon\,(-1+2\epsilon))-\frac{8\,\xi\,\xi b\,\lambda_2[3][0,0,1]}{1-4\epsilon}\right)\right\};$$

```
B1x1∞ = 0;
```

```
B2x2∞ = 0
```

```
0
```

```
B3x3∞ = 0;
```

```
Source1 = Assuming[x[1] > 0 && x[2] > 0 && x[3] > 0,
    B1x1∞ - B1x10 //. subλ // PowerExpand // Simplify];
```

```
Source2 = B2x2∞ - B2x20
```

```
0
```

```
Source3 = Assuming[x[1] > 0 && x[2] > 0 && x[3] > 0,
    B3x3∞ - B3x30 //. subλ // PowerExpand // Simplify];
```

```
Source1proj = Source1 /. x[3] → 1 /. subλ // Expand // Together // Simplify;
```

The integral of the contributions of the remaining free coefficients vanish

```
Source1projλ =
  Table[Simplify[Coefficient[Source1proj, xxx]], {xxx, BtotalVar}];
Table[Integrate[xxx, {x[2], 0, ∞}, GenerateConditions → False],
 {xxx, Source1projλ}]
```

$$\left\{\frac{(-1+\zeta)^{-1+2\epsilon}(-1+\zeta b)^{-1+2\epsilon}x[2]^{-2\epsilon}}{\zeta\,\zeta b-\epsilon\,\zeta\,\zeta b}, \frac{(-1+\zeta)^{-1+2\epsilon}(-1+\zeta b)^{-1+2\epsilon}x[2]^{-2\epsilon}}{2(-1+\epsilon)},\right.$$
$$\frac{1}{2(-1+\epsilon)(-1+2\epsilon)\zeta\,\zeta b}(-1+\zeta)^{-1+2\epsilon}(-1+\zeta b)^{-1+2\epsilon}x[2]^{-1-2\epsilon}$$
$$((-2+2\zeta+2\zeta b-\zeta\,\zeta b)\,x[2]+2\epsilon(\zeta\,\zeta b(-1+x[2])-2\zeta\,x[2]-2(-1+\zeta b)\,x[2])),$$
$$\frac{2\epsilon(-1+\zeta)^{-1+2\epsilon}(-1+\zeta b)^{-1+2\epsilon}x[2]^{-1-2\epsilon}}{1-6\epsilon+8\epsilon^2},$$
$$\left.-\frac{2(-1+\zeta)^{-1+2\epsilon}(-1+\zeta b)^{-1+2\epsilon}x[2]^{-1-2\epsilon}(\epsilon\,\zeta\,\zeta b-x[2]+2\epsilon\,x[2])}{(-1+2\epsilon)(-1+4\epsilon)\zeta\,\zeta b}, 0, 0\right\}$$

```
Source1proj0 = Source1proj /. Table[xxx → 0, {xxx, BtotalVar}] // Simplify;
S1 = Integrate[Source1proj0, {x[2], 0, ∞}, GenerateConditions → False]
```

The integral of the contributions of the remaining free coefficients vanish

```
Source3proj = Source3 /. x[1] → 1 /. subλ // Simplify;
S3 = Integrate[Source3proj, {x[2], 0, ∞}, GenerateConditions → False]
```

```
Stotal = (-1 + ξ) ξ^2 (ξ - ξb) (S1 + S3) // FullSimplify
```

$$-\pi(-1+2\epsilon)\left(-1+(-1+\zeta)^{2\epsilon}(-1+\zeta b)^{2\epsilon}\right)\mathrm{Csc}[2\pi\epsilon]$$

Checking that the evaluation given in [arXiv:2201.09626] at leading order in ϵ satisfies the differential equation

```
DB[ξ_, ξb_] = (PolyLog[2, ξ] - PolyLog[2, ξb] -
      1 / 2 Log[ξ ξb] (PolyLog[1, ξ] - PolyLog[1, ξb])) / (2 I);
W[ξ_, ξb_] = 4 I DB[ξ, ξb] / (ξ - ξb);
```

$$2\left(-\frac{1}{2}(-\mathrm{Log}[1-\zeta]+\mathrm{Log}[1-\zeta b])\,\mathrm{Log}[\zeta\,\zeta b]+\mathrm{PolyLog}[2,\zeta]-\mathrm{PolyLog}[2,\zeta b]\right)$$

```
pf1 =
 ((-1 + ξ) ξ (ξ - ξb) D[W[ξ, ξb], {ξ, 2}] + (3 ξ^2 - ξ (2 + ξb)) D[W[ξ, ξb], {ξ, 1}] + ξ
          W[ξ, ξb] // Simplify // PowerExpand // FullSimplify) /.
    Log[1 - ξ] → Log[-1] + Log[-1 + ξ] /. Log[1 - ξb] →
    -Log[-1] + Log[-1 + ξb] // Simplify
```

$$-\frac{\text{Log}[-1+\xi]+\text{Log}[-1+\xi b]}{\xi}$$

```
Stotal0 = (-1 + ξ) ξ (ξ - ξb) Normal[Series[Stotal, {ϵ, 0, 0}]] // Simplify
```

$$\frac{\text{Log}[-1+\xi]+\text{Log}[-1+\xi b]}{\xi}$$

```
pf1 + Stotal0 // FullSimplify
```

```
0
```

Derivative w.r.t. ζ, ζb

We derive the partial differential equation of equation (4.17)

```
listderivative = {Simplify[D[intcross, {ξ, 1}] / intcross] // Numerator,
  Simplify[D[intcross, {ξb, 1}] / intcross] // Numerator}
```

```
{(-1 + 2 ϵ) (-x[2] + ξb (x[1] + x[2])) x[3], (-1 + 2 ϵ) (-x[2] + ξ (x[1] + x[2])) x[3]}
```

```
listdegree = DegreeHomogeneity[listderivative, var]
```

```
{2, 2}
```

```
derivativeorder = 1;
RedFStep1 = ReductionF[listderivative[[1]], listdegree[[1]],
   listderivative[[2]], c₁, Subscript[λ, 1], fcross, var, varlong];
RedUStep1 = ReductionU[RedFStep1[[1]], RedFStep1[[2]], μ₁, ucross, var, varlong];
solnRstep1 = ReductionU[RedUStep1[[1]], RedFStep1[[2]], μ₁₁, Q, var, varlong];
(*Building M using equation 3.31*)
Mstep1 = (Sum[D[RedUStep1[[1]][[i]], var[[i]]], {i, Length[var]}] - RedUStep1[[3]] -
      4 ϵ solnRstep1[[3]] ) / (derivativeorder - 1 + 1 - 2 ϵ) /. solnRstep1[[4]];
```

```
Reduction with respect to Jac((-1 + ζ) (-1 + ζb) x[2] × x[3] + x[1] (x[2] + ζ ζb x[3]))
degree coefficient 1
Number of equations 6, number of variables 10
Calling Finite Flow
finite flow done -- length result 6
Reduction with respect to Jac(x[1] + x[2] + x[3])
Degree quotient 0
Number of equations 3, number of variables 5
Calling Finite Flow
finite flow done -- length result 3
Reduction with respect to Jac(x[2])
Degree quotient 0
Number of equations 3, number of variables 3
Calling Finite Flow
finite flow done -- length result 3
```

```
listcoPF2 =
 {-Mstep1, c1, 1} //. RedFStep1[[4]] //. solnRstep1[[4]] //. RedUStep1[[4]] //.
   solnRstep1[[4]] // FullSimplify
```

$$\left\{\frac{\zeta\,\zeta b-2\,\epsilon\,(\zeta+\zeta b)}{(-1+\zeta)\,\zeta\,\zeta b},\ \frac{-1+\zeta b}{-1+\zeta},\ 1\right\}$$

```
Btotal = ( Bvec[solnRstep1[[1]], 1, fcross, 1 - 2 ϵ]) intcross //. RedFStep1[[4]] //.
     solnRstep1[[4]] // Expand // Simplify;
```

```
BtotalVar = {λ1[3][1, 1, 1], λ1[3][1, 2, 0], λ1[3][2, 1, 0],
   λ2[1][1, 0, 0], λ2[3][0, 0, 1], λ2[3][0, 1, 0], λ2[3][1, 0, 0]};
```

```
dBtotal = Sum[D[Btotal[[i]], var[[i]]], {i, Length[var]}] // Simplify;
```

```
ApplyPFcross =
  (listcoPF2[[1]] intcross + listcoPF2[[2]] × D[intcross, ζb] + D[intcross, ζ]) //
   Simplify;
```

```
checkpf = dBtotal  + ApplyPFcross;
checkpf // Simplify
```

```
0
```