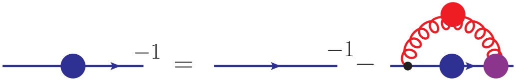
Derivation of Dyson-Schwinger equations

Markus Q. Huber

$$\mathcal{L}_{\mathrm{QCD}} = \overline{\psi}(-\not{D} + m)\psi + \mathcal{L}_{YM} \qquad \rightarrow$$

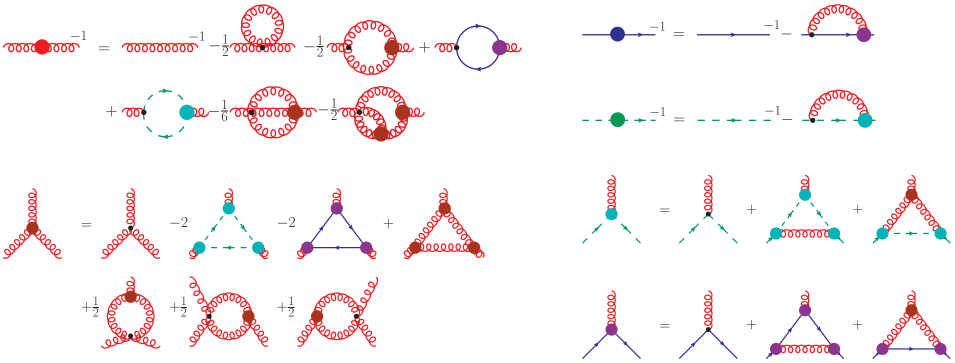


ECT* Doctoral Training Program
“Hadron physics with functional methods”

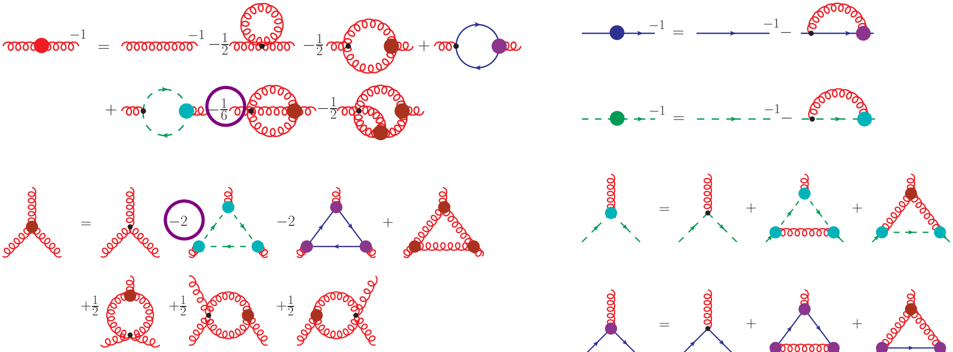
Trento, Italy

May 9, 2022

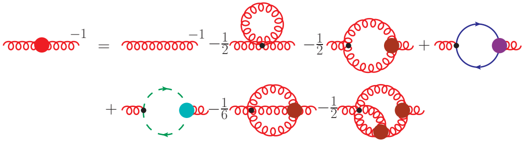
Dyson-Schwinger equations



Dyson-Schwinger equations



Dyson-Schwinger equations

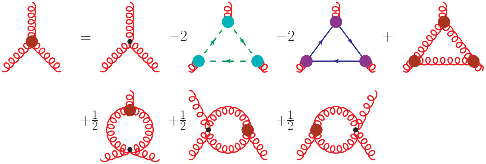


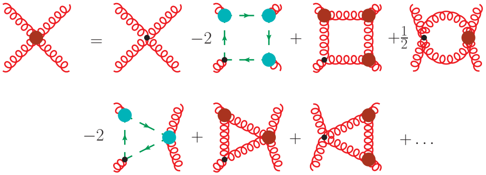
$$K_G(p,q) = \frac{\left(x^2+(y-z)^2-2x(y+z)\right)}{4xy^2z}$$

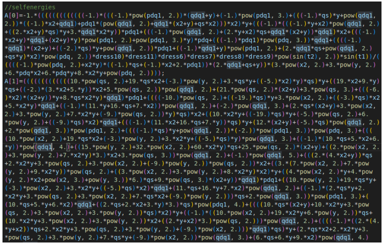
$$K_Z^{gh,\zeta}(p,q) = \frac{x^2\left(\zeta-2\right)+2x(y+z)-\zeta(y-z)^2}{12x^2yz}$$

$$K_Z^{gl,\zeta}(p,q) = \frac{z^2\zeta}{24x^2y^2}+\frac{z(5x-x\zeta+4y\zeta)}{12x^2y^2}+\frac{x^2(-19+\zeta)+2xy(-17+\zeta)-18y^2\zeta}{24x^2y^2}$$
$$+\frac{(x-y)^2\left(x^2+10xy+y^2\zeta\right)}{24x^2y^2z^2}+\frac{4x^3+xy^2(-17+\zeta)+4y^3\zeta-x^2y(15+\zeta)}{12x^2y^2z}$$

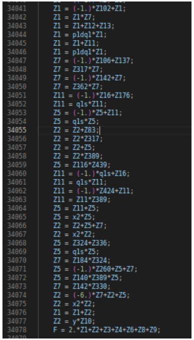
Dyson-Schwinger equations







```
//selfenergies
A[0]=-1.*((((((((((((((-1.)*(((-1.)*pow(pdq1, 2.))*(qdq1+y)+(-1.)*pow(pdq1, 3.)+(((-1.)*qs)*y+pow(qdq1,
2.))*((-1.)*x2+qdq1)+pdq1*(pow(qdq1, 2.)+qdq1*(x2+y)+qs*x2)))*x2)*y+(((-1.)*(((-1.)*y+x2)*pow(qdq1, 2.)
+((2.*x2+y)*qs)*y+3.*qdq1*x2*y))*pdq1+(((-1.)*pow(qdq1, 2.)+(2.*y+x2)*qs+qdq1*(x2+y))*qdq1)*x2+(((-1.)
*x2+y)*qdq1+(x2+y)*y)*pow(pdq1, 2.)+pow(pdq1, 3.)*y)*pdq+(((-1.)*pdq1)*pow(pdq, 3.))*qdq1+((((-1.)
*qdq1)*(x2+y)+((-2.)*qs)*y+pow(qdq1, 2.))*pdq1+((-1.)*qdq1+y)*pow(pdq1, 2.)+(2.*qdq1*qs+pow(qdq1, 2.)
+qs*y)*x2)*pow(pdq, 2.))*dress10)*dress11)*dress6)*dress7)*dress8)*dress9)*pow(sin(t2), 2.))*sin(t1))/(
((((-1.)*pow(pdq, 2.)+x2*y)*((-1.)*qs+(-1.)*x2+2.*pdq1))*(2.*qdq1+qs+y))*(3.*pow(x2, 2.)+3.*pow(y, 2.)
+6.*pdq*x2+6.*pdq*y+8.*x2*y+pow(pdq, 2.))));
A[1]=((((((((((((10.*pow(qs, 2.)+19.*qs*x2+(-3.)*pow(y, 2.)+3.*qs*y+((-5.)*x2)*y)*qs)*y+((19.*x2+9.*y)
*qs+((-2.)*(3.*x2+5.*y))*x2+5.*pow(qs, 2.))*pow(qdq1, 2.)+(21.*pow(qs, 2.)*(x2+y)+3.*pow(qs, 3.)+(((-6.
)*x2)*(x2+y))*y+8.*qs*x2*y)*qdq1)*pdq1+((((-10.)*pow(qs, 2.)+((-19.)*qs)*y+3.*pow(x2, 2.)+((-3.)*qs)*x2
+5.*x2*y)*qdq1+((-1.)*(11.*y+16.*qs+7.*x2))*pow(qdq1, 2.)+(-2.)*pow(qdq1, 3.)+(2.*qs*(x2+y)+3.*pow(x2,
2.)+3.*pow(y, 2.)+7.*x2*y+(-9.)*pow(qs, 2.))*y)*qs)*x2+((10.*x2*y+((-19.)*qs)*y+(-5.)*pow(qs, 2.)+6.
*pow(y, 2.)+((-9.)*qs)*x2)*qdq1+(((-1.)*(11.*x2+16.*qs+7.*y))*qs)*y+(12.*(x2+y)+(-5.)*qs)*pow(qdq1, 2.)
+2.*pow(qdq1, 3.))*pow(pdq1, 2.)+((((-1.)*qs)*y+pow(qdq1, 2.))*(-2.))*pow(pdq1, 3.))*pow(pdq, 3.)+(((
(10.*pow(x2, 2.)+19.*qs*x2+(-3.)*pow(y, 2.)+3.*x2*y+((-5.)*qs)*y)*pow(qdq1, 3.)+((-1.)*(10.*qs+5.*x2+6.
*y))*pow(qdq1, 4.)+((15.*pow(y, 2.)+32.*pow(x2, 2.)+60.*x2*y)*qs+25.*pow(qs, 2.)*(x2+y)+((2.*pow(x2, 2.
)+3.*pow(y, 2.)+7.*x2*y)*3.)*x2+3.*pow(qs, 3.))*pow(qdq1, 2.)+(-1.)*pow(qdq1, 5.)+(((2.*(4.*x2+y))*qs
+2.*x2*y+3.*pow(qs, 2.)+3.*pow(x2, 2.)+(-9.)*pow(y, 2.))*pow(qs, 2.))*x2+((3.*(7.*pow(x2, 2.)+7.*pow
(y, 2.)+9.*x2*y))*pow(qs, 2.)+((3.*pow(x2, 2.)+3.*pow(y, 2.)+8.*x2*y)*x2)*y+((4.*pow(x2, 2.)*y+4.*pow
(y, 2.)*x2+pow(x2, 3.)+pow(y, 3.))*6.)*qs+9.*pow(qs, 3.)*(x2+y))*qdq1)*pdq1+((10.*pow(y, 2.)+19.*qs*y+
(-3.)*pow(x2, 2.)+3.*x2*y+((-5.)*qs)*x2)*qdq1+(11.*qs+16.*y+7.*x2)*pow(qdq1, 2.)+((-1.)*(2.*qs*y+2.
*x2*y+3.*pow(qs, 2.)+3.*pow(x2, 2.)+7.*qs*x2+(-9.)*pow(y, 2.)))*qs+2.*pow(qdq1, 3.))*pow(pdq1, 3.)+(
(10.*qs+5.*y+6.*x2)*qdq1+((2.*qs+2.*x2+3.*y)*3.)*qs)*pow(pdq1, 4.)+((((10.*qs*(x2+y)+10.*x2*y+3.*pow
(qs, 2.)+3.*pow(x2, 2.)+3.*pow(y, 2.))*qs)*x2)*y+((-1.)*((10.*pow(x2, 2.)+19.*x2*y+6.*pow(y, 2.))*qs+
(10.*x2*y+3.*pow(x2, 2.)+3.*pow(y, 2.))*x2+((2.*y+x2)*3.)*pow(qs, 2.)))*pow(qdq1, 2.)+((((-1.)*((2.*(4.
*y+x2))*qs+2.*x2*y+3.*pow(qs, 2.)+3.*pow(y, 2.)+(-9.)*pow(x2, 2.)))*qdq1)*qs)*y+(2.*qs*x2+2.*x2*y+3.
*pow(qs, 2.)+3.*pow(y, 2.)+7.*qs*y+(-9.)*pow(x2, 2.))*pow(qdq1, 3.)+(6.*qs+6.*y+9.*x2)*pow(qdq1, 4.)
```

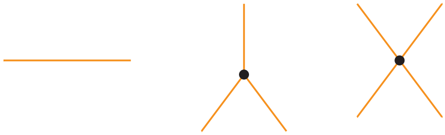


```
Z1 = (-1.)*Z102+Z1;
Z1 = Z1*Z7;
Z1 = Z1+Z12+Z13;
Z1 = pldql*Z1;
Z1 = Z1+Z11;
Z1 = pldql*Z1;
Z7 = (-1.)*Z106+Z137;
Z7 = Z317*Z7;
Z7 = (-1.)*Z142+Z7;
Z7 = Z362*Z7;
Z11 = (-1.)*Z16+Z176;
Z11 = qls*Z11;
Z5 = (-1.)*Z5+Z11;
Z5 = qls*Z5;
Z2 = Z2+Z83;
Z2 = Z2*Z317;
Z2 = Z2+Z5;
Z2 = Z2*Z389;
Z5 = Z116*Z439;
Z11 = (-1.)*qls+Z16;
Z11 = qls*Z11;
Z11 = (-1.)*Z424+Z11;
Z11 = Z11*Z389;
Z5 = Z11+Z5;
Z5 = x2*Z5;
Z2 = Z2+Z5+Z7;
Z2 = x2*Z2;
Z5 = Z324+Z336;
Z5 = qls*Z5;
Z7 = Z184*Z324;
Z5 = (-1.)*Z260+Z5+Z7;
Z5 = Z140*Z389*Z5;
Z7 = Z142*Z330;
Z2 = (-6.)*Z7+Z2+Z5;
Z2 = x2*Z2;
Z1 = Z1+Z2;
Z2 = y*Z10;
F = 2.*Z1+Z2+Z3+Z4+Z6+Z8+Z9;
```

Generating functionals

Example: Scalar theory

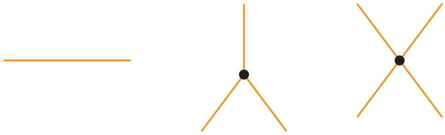
$$S[\phi] = \int dx \left(\phi(-\partial^2 + m^2)\phi + \frac{\lambda_3}{3!}\phi^3 + \frac{\lambda_4}{4!}\phi^4 \right)$$



Generating functionals

Example: Scalar theory

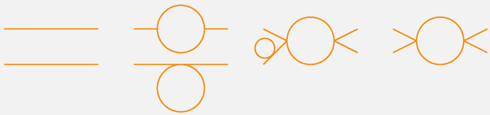
$$S[\phi] = \int dx \left(\phi(-\partial^2 + m^2)\phi + \frac{\lambda_3}{3!}\phi^3 + \frac{\lambda_4}{4!}\phi^4 \right)$$



Path integral:

$$Z[J] = \int D[\phi] e^{-S[\phi] + \int dx \phi(x) J(x)} = e^{W[J]}$$

$Z[J] \to$ Generating functional for full correlation functions



$W[J] \to$ Generating functional for connected correlation functions

1PI effective action

Legendre transform: New variable $\Phi(x)$ (averaged field Φ in presence of external source J)

$$\Phi(\boldsymbol{x}) := \langle\phi(x)\rangle_J = \frac{\delta W[J]}{\delta J(x)} = Z[J]^{-1}\int D[\phi]\phi(\boldsymbol{x})e^{-S[\phi]+\int dy\phi(y)J(y)} \qquad \left(J(x) = \frac{\delta\Gamma[\Phi]}{\delta\Phi(x)}\right)$$

$$\Gamma[\Phi] = -W[J] + \int dx\Phi(\boldsymbol{x})J(x)$$

1PI effective action

Legendre transform: New variable $\Phi(x)$ (averaged field Φ in presence of external source J)

$$\Phi(x) := \langle\phi(x)\rangle_J = \frac{\delta W[J]}{\delta J(x)} = Z[J]^{-1}\int D[\phi]\phi(x)e^{-S[\phi]+\int dy\phi(y)J(y)} \qquad \left(J(x) = \frac{\delta\Gamma[\Phi]}{\delta\Phi(x)}\right)$$

$$\Gamma[\Phi] = -W[J] + \int dx\Phi(x)J(x)$$

$\Gamma[\Phi] \to$ 1PI effective action, generating functional of one-particle irreducible correlation functions

(All correlation functions can be constructed from 1PI correlation functions.)

Propagators and vertices

Propagator:

$$D(x,y) = D(x-y) = \left.\frac{\delta^2 W[J]}{\delta J(x)\delta J(y)}\right|_{J=0} = \langle\phi(x)\phi(y)\rangle - \langle\phi(x)\rangle\langle\phi(y)\rangle$$

$$D(x,y)^J := \frac{\delta^2 W[J]}{\delta J(x)\delta J(y)} = \left(\frac{\delta^2\Gamma[\Phi]}{\delta\Phi(x)\delta\Phi(y)}\right)^{-1}$$

Propagators and vertices

Propagator:

$$D(x,y) = D(x-y) = \left.\frac{\delta^2 W[J]}{\delta J(x)\delta J(y)}\right|_{J=0} = \langle\phi(x)\phi(y)\rangle - \langle\phi(x)\rangle\langle\phi(y)\rangle$$

$$D(x,y)^J := \frac{\delta^2 W[J]}{\delta J(x)\delta J(y)} = \left(\frac{\delta^2\Gamma[\Phi]}{\delta\Phi(x)\delta\Phi(y)}\right)^{-1}$$

Derivatives of 1PI effective action: (Note $J \neq 0$ and "$-$" by convention.)

$$\Gamma(x_1,\ldots,x_n)^J := -\frac{\delta\Gamma[\Phi]}{\delta\Phi(x_1)\cdots\delta\Phi(x_n)}$$

Physical vertices

$$\Gamma(x_1,\ldots,x_n) := \Gamma(x_1,\ldots,x_n)^{J=0}, \quad n>2$$

Derivation of DSEs I

Integral of a total derivative vanishes:

$$0 = \int D[\phi] \frac{\delta}{\delta\phi} e^{-S+\int dy \phi(y) J(y)} = \int D[\phi] \left(-\frac{\delta S}{\delta\phi(x)} + J(x) \right) e^{-S+\int dy \phi(y) J(y)}$$

Derivation of DSEs I

Integral of a total derivative vanishes:

$$0 = \int D[\phi] \frac{\delta}{\delta\phi} e^{-S+\int dy \phi(y) J(y)} = \int D[\phi] \left(-\frac{\delta S}{\delta\phi(x)} + J(x) \right) e^{-S+\int dy \phi(y) J(y)}$$

Pull in front of integral $\rightarrow$ Master DSE for full correlation functions

$$0 = \left(-\frac{\delta S}{\delta\phi(x)}\bigg|_{\phi(x')=\delta/\delta J(x')} + J(x) \right) \underbrace{Z[J]}_{e^{W[J]}} = 0$$

Derivation of DSEs I

Integral of a total derivative vanishes:

$$0 = \int D[\phi] \frac{\delta}{\delta\phi} e^{-S+\int dy \phi(y) J(y)} = \int D[\phi] \left(-\frac{\delta S}{\delta\phi(x)} + J(x) \right) e^{-S+\int dy \phi(y) J(y)}$$

Pull in front of integral $\rightarrow$ Master DSE for full correlation functions

$$0 = \left(-\frac{\delta S}{\delta\phi(x)}\bigg|_{\phi(x')=\delta/\delta J(x')} + J(x) \right) \underbrace{Z[J]}_{e^{W[J]}} = 0$$

$$e^{-W[J]} \left(\frac{\delta}{\delta J(x)} \right) e^{W[J]} = \frac{\delta W[J]}{\delta J(x)} + \frac{\delta}{\delta J(x)}$$

$\rightarrow$ Master DSE for connected correlation functions

$$-\frac{\delta S}{\delta\phi(x)}\bigg|_{\phi(x')=\frac{\delta W[J]}{\delta J(x')}+\frac{\delta}{\delta J(x')}} + J(x) = 0.$$

Derivation of DSEs II

Legendre transformation:

$$\frac{\delta W[J]}{\delta J(x)} \to \Phi(x)$$

$$\frac{\delta}{\delta J(x)} \to \int dz\, D(x,z)^J \frac{\delta}{\delta \Phi(z)}$$

$$\left(\frac{\delta}{\delta J(x)} = \int dz \frac{\delta \Phi(z)}{\delta J(x)} \frac{\delta}{\delta \Phi(z)} = \int dz \frac{\delta}{\delta J(x)} \frac{\delta W[J]}{\delta J(z)} \frac{\delta}{\delta \Phi(z)} = \int dz \frac{\delta^2 W[J]}{\delta J(x) \delta J(z)} \frac{\delta}{\delta \Phi(z)} \right)$$

Derivation of DSEs II

Legendre transformation:

$$\frac{\delta W[J]}{\delta J(x)} \to \Phi(x)$$

$$\frac{\delta}{\delta J(x)} \to \int dz\, D(x,z)^J \frac{\delta}{\delta \Phi(z)}$$

$$\left(\frac{\delta}{\delta J(x)} = \int dz \frac{\delta \Phi(z)}{\delta J(x)} \frac{\delta}{\delta \Phi(z)} = \int dz \frac{\delta}{\delta J(x)} \frac{\delta W[J]}{\delta J(z)} \frac{\delta}{\delta \Phi(z)} = \int dz \frac{\delta^2 W[J]}{\delta J(x) \delta J(z)} \frac{\delta}{\delta \Phi(z)} \right)$$

Master DSE for 1PI correlation functions

$$-\frac{\delta S}{\delta \phi(x)}\bigg|_{\phi(x')=\Phi(x')+\int dz\, D(x',z)^J\, \delta/\delta\Phi(z)} + \frac{\delta \Gamma[\Phi]}{\delta \Phi(x)} = 0$$

Get DSE for n-point function by applying $n-1$ derivatives.

Master equation for scalar theory I

$$S[\phi] = \int dx \left(\phi(-\partial^2 + m^2)\phi + \frac{\lambda_3}{3!}\phi^3 + \frac{\lambda_4}{4!}\phi^4\right) \to \frac{1}{2}S_{ij}\phi_i\phi_j - \frac{1}{3!}S_{ijk}\phi_i\phi_j\phi_k - \frac{1}{4!}S_{ijkl}\phi_i\phi_j\phi_k\phi_l$$

Integration over indices as position arguments.

Bare vertices:

$$S_{ijk} = \left.\frac{\delta^3 S[\phi]}{\delta\phi_i\delta\phi_j\delta\phi_k}\right|_{\phi=0} = \lambda_3\delta(x_i - x_j)\delta(x_i - x_k), \qquad S_{ijkl} = \left.\frac{\delta^4 S[\phi]}{\delta\phi_i\delta\phi_j\delta\phi_k\delta\phi_l}\right|_{\phi=0} = \lambda_4\delta(x_i - x_j)\delta(x_i - x_k)\delta(x_i - x_l)$$

Master equation for scalar theory I

$$S[\phi] = \int dx \left(\phi(-\partial^2 + m^2)\phi + \frac{\lambda_3}{3!}\phi^3 + \frac{\lambda_4}{4!}\phi^4\right) \to \frac{1}{2} S_{ij}\phi_i\phi_j - \frac{1}{3!} S_{ijk}\phi_i\phi_j\phi_k - \frac{1}{4!} S_{ijkl}\phi_i\phi_j\phi_k\phi_l$$

Integration over indices as position arguments.

We need derivative:

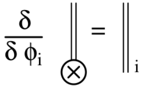
$$\frac{\delta S}{\delta \phi} = S_{ij}\phi_i - \frac{1}{2!} S_{ijk}\phi_i\phi_j - \frac{1}{3!} S_{ijkl}\phi_i\phi_j\phi_k$$

Example for replacement:

$$\frac{1}{2!} S_{ijk}\phi_i\phi_j\Big|_{\phi_{i'} = \Phi_{i'} + D^J_{i'm}\,\delta/\delta\Phi_m} = \frac{1}{2} S_{ijk}\left(\Phi_i + D^J_{im}\frac{\delta}{\delta\Phi_m}\right)\Phi_j = \frac{1}{2} S_{ijk}\Phi_i\Phi_j + \frac{1}{2} S_{ijk} D^J_{ij}$$

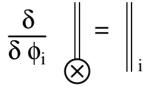
Derivative rules

$$\frac{\delta}{\delta \Phi_i} \Phi_j = \delta_{ij}$$

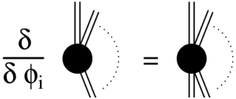


Derivative rules

$$\frac{\delta}{\delta \Phi_i}\Phi_j = \delta_{ij}$$



$$\frac{\delta}{\delta \Phi_i}\Gamma^J_{j_1\ldots j_n} = \Gamma^J_{ij_1\ldots j_n}$$

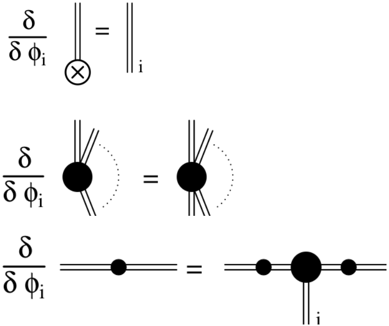


Derivative rules

$$\frac{\delta}{\delta\Phi_i}\Phi_j = \delta_{ij}$$

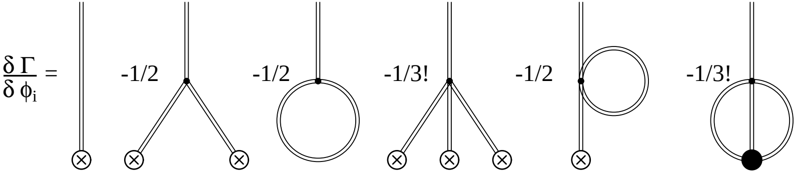
$$\frac{\delta}{\delta\Phi_i}\Gamma^J_{j_1\dots j_n} = \Gamma^J_{ij_1\dots j_n}$$

$$\frac{\delta}{\delta\Phi_i}D^J_{jk} = -\boldsymbol{\epsilon^{i}_{jm}} D^J_{jm}\Gamma^J_{imn}D^J_{nk}$$



Signs for fermions!

Master equation for scalar theory II

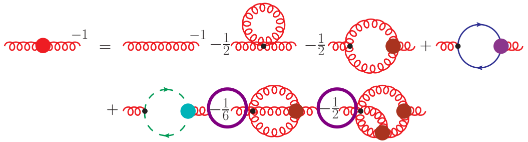


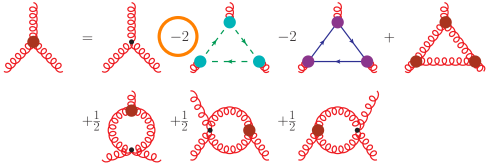
DSEs for n-point function by applying $n-1$ further derivatives.

Exercise: Derive that equation.

Symmetry factors and signs

Symmetry factors:
As for perturbation theory. If there are various realizations, divide by that number.





Fermions:

Sign changes emerge automatically for anticommuting fields.

Shortcut when at most two fermions: Closed fermion loops have the opposite sign.

Fermion loops can go round in two directions. $\rightarrow$ $\times 2$

More complicated theories

If a theory has more than one field, intermediate expressions can by “unphysical” since $J \neq 0$.

For example:
D^J_{im} can be a “mixed” propagator, e.g., $\boldsymbol{D^{A\psi,J}_{im}}$, as long as $J \neq 0$. Relevant for further derivatives.

$\rightarrow$ Use superfields.

More complicated theories

If a theory has more than one field, intermediate expressions can by "unphysical" since $J \neq 0$.

For example:
D^J_{im} can be a "mixed" propagator, e.g., $\boldsymbol{D^{A\psi,J}_{im}}$, as long as $J \neq 0$. Relevant for further derivatives.

$\rightarrow$ Use superfields.

Example: QCD

Fields $\overline{\psi}^c_{df}$, ψ^c_{df}, A^c_l, $\overline{c}^c$, $c^c \rightarrow \Phi_i$

$\rightarrow$ Sum includes then field types:

$$\frac{\delta}{\delta J^A_i} \rightarrow D^{A\Phi,J}_{\boldsymbol{im}} \frac{\delta}{\delta \Phi_{\boldsymbol{m}}} = D^{AA,J}_{im} \frac{\delta}{\delta A_m} + D^{Ac,J}_{im} \frac{\delta}{\delta c_m} + D^{A\overline{c},J}_{im} \frac{\delta}{\delta \overline{c}_m} + D^{A\psi,J}_{im} \frac{\delta}{\delta \psi_m} + D^{A\overline{\psi},J}_{im} \frac{\delta}{\delta \overline{\psi}_m}$$

See example.

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.
3. Set sources to zero. $\rightarrow$ Physical quantities

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.
3. Set sources to zero. $\rightarrow$ Physical quantities
4. (Apply truncation.)

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.
3. Set sources to zero. $\rightarrow$ Physical quantities
4. (Apply truncation.)
5. Plug in Feynman rules.

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.
3. Set sources to zero. $\rightarrow$ Physical quantities
4. (Apply truncation.)
5. Plug in Feynman rules.
6. (Project and) perform contractions (Lorentz, color, Dirac, flavor, ...)

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.
3. Set sources to zero. $\rightarrow$ Physical quantities
4. (Apply truncation.)
5. Plug in Feynman rules.
6. (Project and) perform contractions (Lorentz, color, Dirac, flavor, ...)
   - Calculate analytically.

Recipe for deriving a DSE

1. Write down master equation (dropping terms irrelevant for target DSE).
2. Perform derivatives.
3. Set sources to zero. $\to$ Physical quantities
4. (Apply truncation.)
5. Plug in Feynman rules.
6. (Project and) perform contractions (Lorentz, color, Dirac, flavor, ...)
   - Calculate analytically.
   - Put into code form and calculate numerically.

Automatized derivation with *DoFun*

Derivation of functional equations [Alkofer, MQH, Schwenzer, ’08; MQH, Braun, ’11; MQH, Cyrol, Pawlowski, ’19]

→ `https://github.com/markusqh/DoFun/`

Works in two steps:

- Symbolic derivation (no Feynman rules, just types of fields)
- Algebraic: Plug in Feynman rules

See also *QMeS-Derivation*
[Pawlowski, Schneider, Wink, ’21]

doDSE

doDSE[*ac*, *flis*, [*opts*]] derives the DSE from the action *ac* for the fields contained in *flis*.

doDSE[*ac*, *flis*, *props*, [*opts*]] derives the DSE only with propagators contained in prop

doDSE[*ac*, *flist*, *vtest*, [*opts*]] derives the DSE only with vertices allowed by *vtest*.
Allowed propagators will be taken from *ac* if the *props* argument is not given.

▾ Details

- The following options can be given:

sourcesZero	True

DoFun example: Three-gluon vertex DSE

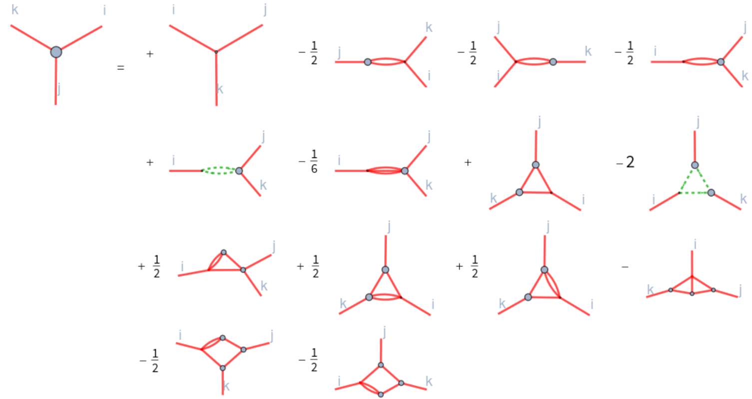
```
setFields[{A}, {{c, cb}}]; (* defines A as boson and c, cb as fermions *)
dse = doDSE[{{A, A}, {c, cb}, {A, cb, c}, {A, A, A},
  {A, A, A, A}}, {A, A, A}]; (* derives AAA DSE *)

op[S[{A, i1}, {A, i2}, {A, i3}]] -
1/2 op[P[{A, r1}, {A, s1}],
 P[{A, t1}, {A, u1}],
 S[{A, i1}, {A, i3}, {A, r1}, {A, u1}],
 V[{A, i2}, {A, s1}, {A, t1}]] + ...
```

DoFun example: Three-gluon vertex DSE

Plotting possible:

```
DSEPlot[dse, {{A, Thick, Red}, {c, Darker@Green, Dashed, Thick}}]
```



Further uses of computers

- Tracing: *FORM*, *FormTracer*, *FeynCalc*, your own code, . . .
- Cross checks, e.g., analytic calculations for certain limits in *Mathematica*
- Transforming to code, esp. when expressions are long
- Optimizing code, e.g., with *FORM* [Kuipers, Ueda, Vermaseren, '13]

Further uses of computers

- Tracing: *FORM*, *FormTracer*, *FeynCalc*, your own code, ...
- Cross checks, e.g., analytic calculations for certain limits in *Mathematica*
- Transforming to code, esp. when expressions are long
- Optimizing code, e.g., with *FORM* [Kuipers, Ueda, Vermaseren, '13]

  Example:

  $y - 3x + 5xz + 2x^2 y\, z - 3x^2 y^2 z + 5x^2 y^2 z^2$ has 18 multiplications and 5 additions.

  $Z_1 = -3 + 5y$, $y + x(Z_1 + x(y(2\,z + y\,z\,Z_1)))$ has 7 multiplications and 4 additions.

1PI effective action

1PI effective action

Depends on expectation value of field: $\Gamma[\Phi]$

$$\Gamma[\Phi] = -W[J] + \int dx \Phi(x) J(x)$$

$$\Phi(x) := \langle \phi(x) \rangle_J = \frac{\delta W[J]}{\delta J(x)}$$

Equations of motion: Exact equations in terms of the moments of the 1PI effective action (=n-point functions)

Infinite tower:

Equations of motion with finite number of terms for infinitely many correlation functions.

*n*PI effective actions

*n*PI effective actions include
sources for up to *n*-point functions, e.g., 3PI:
[Cornwall, Jackiw, Tomboulis, '74; Berges, '04; Carrington, Guo, '10]

$$e^{W[J,R^{(2)},R^{(3)}]} = Z[J, R^{(2)}, R^{(3)}] = \int D[\phi]e^{-S+\phi_i J_i+\frac{1}{2}R^{(2)}_{ij}\phi_i\phi_j+\frac{1}{3!}R^{(3)}_{ijk}\phi_i\phi_j\phi_k}$$

nPI effective actions

nPI effective actions include sources for up to n-point functions, e.g., 3PI: [Cornwall, Jackiw, Tomboulis, '74; Berges, '04; Carrington, Guo, '10]

$$e^{W[J,R^{(2)},R^{(3)}]} = Z[J,R^{(2)},R^{(3)}] = \int D[\phi] e^{-S+\phi_i J_i + \frac{1}{2} R^{(2)}_{ij}\phi_i\phi_j + \frac{1}{3!} R^{(3)}_{ijk}\phi_i\phi_j\phi_k}$$

$R^{(n)}_{i\ldots}$: source terms for propagator D_{ij} and vertices $\Gamma^{(n)}_{i_1 i_2 \ldots i_n} = \Gamma_{i_1 i_2 \ldots i_n}$

$$\frac{\delta W}{\delta J_i} = \langle \phi_i \rangle_J = \Phi_i, \qquad \frac{\delta W}{\delta R^{(2)}_{ij}} = \frac{1}{2}\left(D^J_{ij} + \Phi_i \Phi_j\right),$$

$$\frac{\delta W}{\delta R^{(3)}_{ijk}} = \frac{1}{6}\left(D^{(3),J}_{ijk} + D^J_{ij}\Phi_k + D^J_{ik}\Phi_j + D^J_{jk}\Phi_i + \Phi_i\Phi_j\Phi_k\right) \qquad D^{(3),J}_{ijk} = \frac{\delta^3 W}{\delta J_i \delta J_j \delta J_k}$$

*n*PI effective actions

*n*PI effective actions include [Cornwall, Jackiw, Tomboulis, ’74; Berges, ’04; Carrington, Guo, ’10]
sources for up to *n*-point functions, e.g., 3PI:

$$e^{W[J,R^{(2)},R^{(3)}]} = Z[J,R^{(2)},R^{(3)}] = \int D[\phi] e^{-S+\phi_i J_i + \frac{1}{2} R^{(2)}_{ij}\phi_i\phi_j + \frac{1}{3!} R^{(3)}_{ijk}\phi_i\phi_j\phi_k}$$

$R^{(n)}_{i\ldots}$: source terms for propagator D_{ij} and vertices $\Gamma^{(n)}_{i_1 i_2 \ldots i_n} = \Gamma_{i_1 i_2 \ldots i_n}$

$$\frac{\delta W}{\delta J_i} = \langle \phi_i \rangle_J = \Phi_i, \qquad \frac{\delta W}{\delta R^{(2)}_{ij}} = \frac{1}{2}\left(D^J_{ij} + \Phi_i \Phi_j\right),$$

$$\frac{\delta W}{\delta R^{(3)}_{ijk}} = \frac{1}{6}\left(D^{(3),J}_{ijk} + D^J_{ij}\Phi_k + D^J_{ik}\Phi_j + D^J_{jk}\Phi_i + \Phi_i\Phi_j\Phi_k\right) \qquad D^{(3),J}_{ijk} = \frac{\delta^3 W}{\delta J_i \delta J_j \delta J_k}$$

Legendre transform:

$$\Gamma = -W + \frac{\delta W}{\delta J_i} J_i + \frac{\delta W}{\delta R^{(2)}_{ij}} R^{(2)}_{ij} + \frac{\delta W}{\delta R^{(3)}_{ijk}} R^{(3)}_{ijk} + \ldots.$$

Equations of motion

Vanishing sources $J_i = R^{(2)}_{ij} = R^{(3)}_{ijk} = 0 \to$ stationarity conditions:

$$\frac{\delta\Gamma}{\delta\Phi_i} = 0, \qquad \frac{\delta\Gamma}{\delta D_{ij}} = 0, \qquad \frac{\delta\Gamma}{\delta\Gamma^{(3)}_{ijk}} = 0,$$

Lead to equations of motion.

Hierarchy of *n*PI effective actions

All effective actions are equivalent:

$$\Gamma[\Phi] = \Gamma[\Phi, D] = \ldots = \Gamma[\Phi, D, \Gamma^{(3)}, \ldots, \Gamma^{(n)}]$$

However, only correlation functions up to n legs are treated self-consistently (m-point functions not dressed for nPI, $n < m$).

Hierarchy of *n*PI effective actions

All effective actions are equivalent:

$$\Gamma[\Phi] = \Gamma[\Phi, D] = \ldots = \Gamma[\Phi, D, \Gamma^{(3)}, \ldots, \Gamma^{(n)}]$$

However, only correlation functions up to n legs are treated self-consistently (m-point functions not dressed for nPI, $n < m$).

$\rightarrow$ Loop expansions

Higher *n*PI effective actions are equivalent at the same expansion order:

$$\Gamma^{\text{1-loop}}[\Phi] = \Gamma^{\text{1-loop}}[\Phi, ...]$$
$$\Gamma^{\text{2-loop}}[\Phi, D] = \Gamma^{\text{2-loop}}[\Phi, D, ...]$$
$$\Gamma^{\text{n-loop}}[\Phi, D, \ldots, \Gamma^{(n)}] = \Gamma^{\text{n-loop}}[\Phi, D, ..., \Gamma^{(m)}] \quad \forall n \leq m$$

Hierarchy of *n*PI effective actions

All effective actions are equivalent:

$$\Gamma[\Phi] = \Gamma[\Phi, D] = \ldots = \Gamma[\Phi, D, \Gamma^{(3)}, \ldots, \Gamma^{(n)}]$$

However, only correlation functions up to n legs are treated self-consistently (m-point functions not dressed for nPI, $n < m$).

$\rightarrow$ Loop expansions

Higher nPI effective actions are equivalent at the same expansion order:

$$\Gamma^{\text{1-loop}}[\Phi] = \Gamma^{\text{1-loop}}[\Phi, ...]$$
$$\Gamma^{\text{2-loop}}[\Phi, D] = \Gamma^{\text{2-loop}}[\Phi, D, ...]$$
$$\Gamma^{\text{n-loop}}[\Phi, D, \ldots, \Gamma^{(n)}] = \Gamma^{\text{n-loop}}[\Phi, D, ..., \Gamma^{(m)}] \quad \forall n \leq m$$

Note: nPI effective actions are n-particle irreducible (n cuts do not lead to separate *loop* terms) up to $n = 4$ [Carrington, Guo, '10]

3PI effective action: 3-loop expansion

Infinite tower:

Equations of motions for finite number of correlations functions with infinitely many terms.

Truncation at the level of the action: loop expansion $\rightarrow$ finite number of terms

$$\Gamma[\Phi, D, \Gamma^{(3)}] = S[\Phi] + \frac{1}{2}\ln D_{ii}^{-1} + \frac{1}{2} S_{ij}[\Phi] D_{ij} - \Gamma_2[\Phi, D, \Gamma^{(3)}]$$
$$\Gamma_2[\Phi, D, \Gamma^{(3)}] = \Gamma_2^0[\Phi, D, \Gamma^{(3)}] + \Gamma_2^{\text{int}}[D, \Gamma^{(3)}]$$

3PI effective action: 3-loop expansion

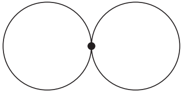
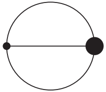
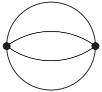
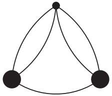
Infinite tower:

Equations of motions for finite number of correlations functions with infinitely many terms.

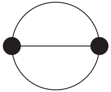
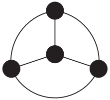
Truncation at the level of the action: loop expansion $\rightarrow$ finite number of terms

$$\Gamma[\Phi, D, \Gamma^{(3)}] = S[\Phi] + \frac{1}{2}\ln D^{-1}_{ii} + \frac{1}{2}S_{ij}[\Phi]D_{ij} - \Gamma_2[\Phi, D, \Gamma^{(3)}]$$

$$\Gamma_2[\Phi, D, \Gamma^{(3)}] = \Gamma_2^0[\Phi, D, \Gamma^{(3)}] + \Gamma_2^{\text{int}}[D, \Gamma^{(3)}]$$

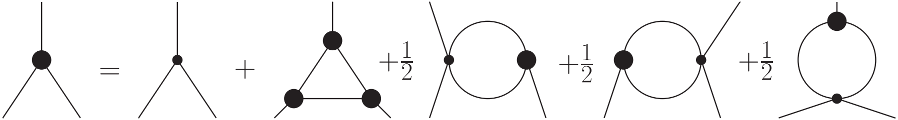
$\Gamma_2^0[\Phi, D, \Gamma^{(3)}] = \frac{1}{8}$  $+\frac{1}{6}$  $+\frac{1}{48}$  $+\frac{1}{8}$ 

[Berges, '04]

$\Gamma_2^{\text{int}}[D, \Gamma^{(3)}] = -\frac{1}{12}$  $+\frac{1}{24}$ 

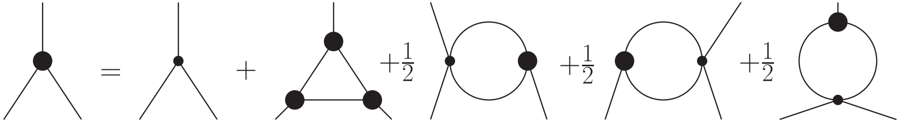
3PI equations

Vertex:

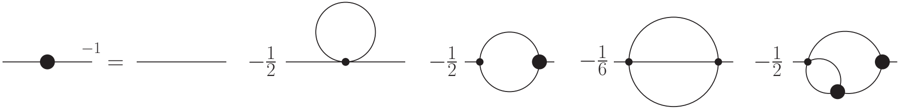


3PI equations

Vertex:



Propagator (after using vertex equation): $D^{-1} = D_0^{-1} - 2\frac{\delta\Gamma_2}{\delta D}$ Exercise: Show that.



Equal to DSE except for dressed 4-point function. $\rightarrow$ Identical for 4PI.

Summary

- Derivation of DSEs is a straightforward (tedious) process

Summary

- Derivation of DSEs is a straightforward (tedious) process
- Use help of computer: derivation, tracing, code writing, tests, optimization

Summary

- Derivation of DSEs is a straightforward (tedious) process
- Use help of computer: derivation, tracing, code writing, tests, optimization
- Checks: comparison with literature, perturbative checks, ...

Summary

- Derivation of DSEs is a straightforward (tedious) process
- Use help of computer: derivation, tracing, code writing, tests, optimization
- Checks: comparison with literature, perturbative checks, ...

- DSEs:
  Infinitely many coupled equations

Summary

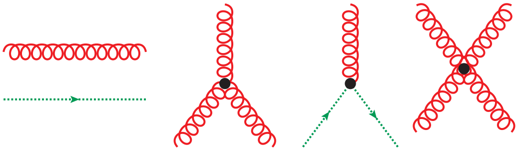
- Derivation of DSEs is a straightforward (tedious) process
- Use help of computer: derivation, tracing, code writing, tests, optimization
- Checks: comparison with literature, perturbative checks, . . .

- DSEs:
  Infinitely many coupled equations
- Equations of motion of *n*PI effective actions:
  Finite number of equations with infinitely many terms (in loop expansion)

Hands-on: Derivation of the ghost-gluon vertex DSEs

[Schleifenbaum, Maas, Wambach, Alkofer, ’04; Alkofer, MQH, Schwenzer, ’08]

$$S[A,\bar{c},c] = \frac{1}{2} S^{AA}_{ij} A_i A_j + S^{\bar{c}c}_{ij} \bar{c}_i c_j - \frac{1}{3!} S^{AAA}_{ijk} A_i A_j A_k - \frac{1}{4!} S^{AAAA}_{ijkl} A_i A_j A_k A_l - S^{A\bar{c}c}_{ijk} A_i \bar{c}_j c_k$$

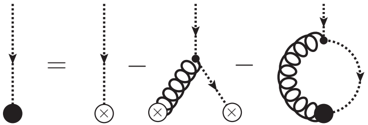


Quark-gluon vertex DSE

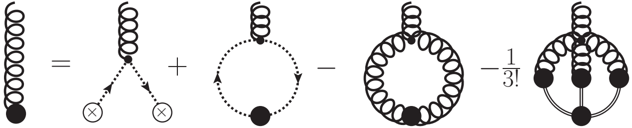
Same diagrammatic structure!

Master equations

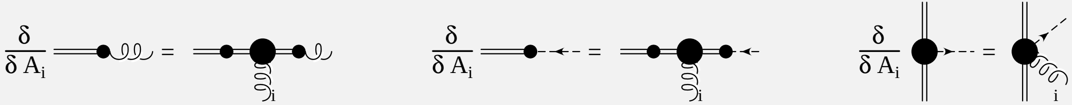
Ghost derivative first:



Gluon derivative first:

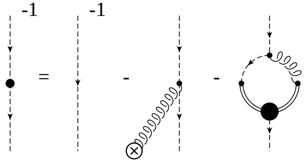


Some differentiation rules



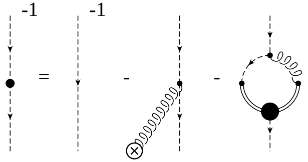
Ghost derivative first $\rightarrow$ c-DSE

$\dfrac{\delta^2\Gamma}{\delta\bar{c}\delta c}$:

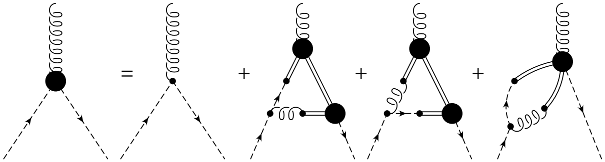


Ghost derivative first $\rightarrow$ c-DSE

$\frac{\delta^2\Gamma}{\delta\bar{c}\delta c}$:



$\frac{\delta^2\Gamma}{\delta A\delta\bar{c}\delta c}$:

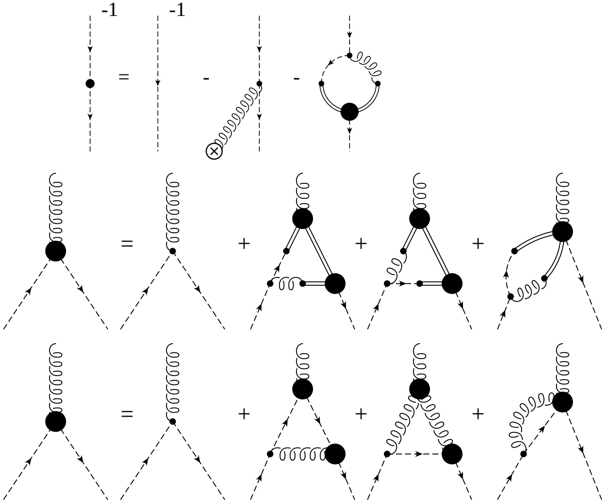


Ghost derivative first $\rightarrow$ c-DSE

$\frac{\delta^2\Gamma}{\delta\bar{c}\delta c}$:

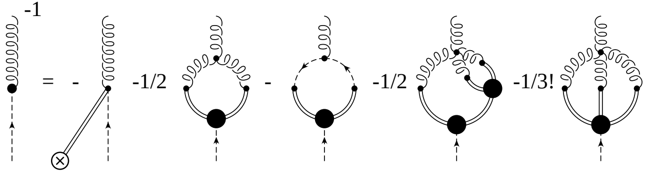
$\frac{\delta^2\Gamma}{\delta A\delta\bar{c}\delta c}$:

$\left.\frac{\delta^2\Gamma}{\delta A\delta c\delta\bar{c}}\right|_{J=0}$:



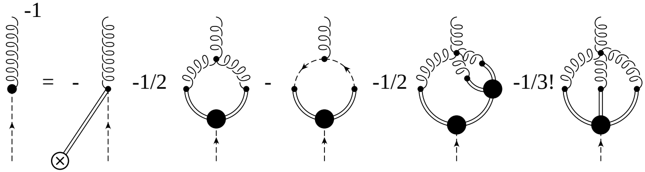
Gluon derivative first → A-DSE

$\frac{\delta^2\Gamma}{\delta c \delta A}$:

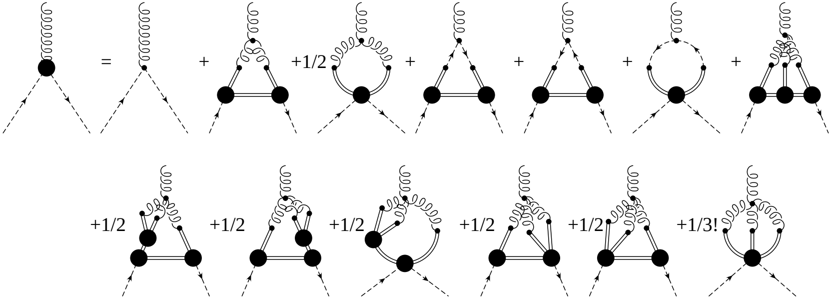


Gluon derivative first → A-DSE

$\frac{\delta^2 \Gamma}{\delta c \delta A}$:



$\frac{\delta^2 \Gamma}{\delta \bar{c} \delta c \delta A}$:



Gluon derivative first $\rightarrow$ A-DSE

$$\left.\frac{\delta^2\Gamma}{\delta\bar{c}\delta c\delta A}\right|_{J=0}:$$