DoFun - Derivation of functional equations with Mathematica

Jens Braun, Markus Q. Huber

Institute of Theoretical Physics, Friedrich-Schiller-University Jena

Feb. 28, 2011

49. Internationale Universitätswochen für Theoretische Physik, Schladming

More information: arXiv:1102.5307

http://www.tpi.uni-jena.de/qfphysics/homepage/mhub/DoFun

Applications of functional equations

Dyson-Schwinger equations (DSEs) and functional renormalization group equations (RGEs) for description of non-perturbative phenomena in many fields:

- condensed matter
- Yang-Mills theories
- QCD and QCD-like theories
- standard model physics
- supersymmetry
- gravity
- ...

$\rightarrow$ Wide variety of theories.

Derivation of functional equations

Derivation of functional equations is based on a few basic steps, but tedious by hand, esp. for several fields and involved tensor structures.

Derivation of functional equations

Derivation of functional equations is based on a few basic steps, but tedious by hand, esp. for several fields and involved tensor structures.

Using a computer algebra system,

e.g. [Fister, Haas, Pawlowski; Benedetti, Groh, Machado, Saueressig, 1012.3081]:

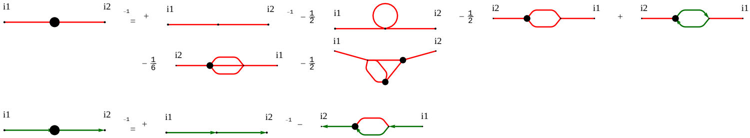
- High number of terms no problem.
- High n-point functions.
- Handling of tensor structures
  $\rightarrow$ several packages/programs readily available.
- Direct link to numerical programs.
- Testing of next higher level of truncation easy.

$\rightarrow$ Play around with equations without spending too much time on derivations.

Mathematica used

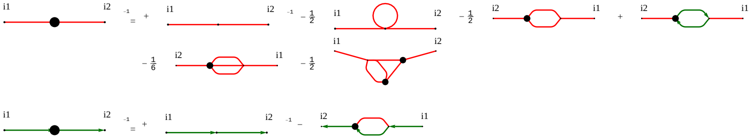
An example: The maximally Abelian gauge

Yang-Mills theory is typically done in the Landau gauge:



An example: The maximally Abelian gauge

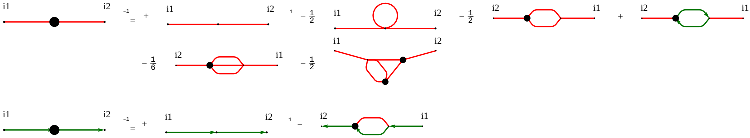
Yang-Mills theory is typically done in the Landau gauge:



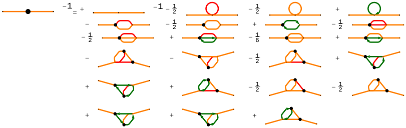
Another interesting gauge: the maximally Abelian gauge
$\longrightarrow$ dual superconductor picture of confinement

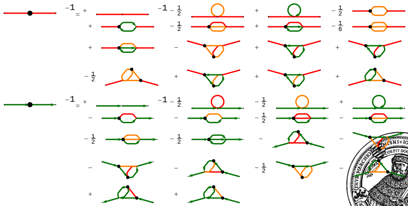
An example: The maximally Abelian gauge

Yang-Mills theory is typically done in the Landau gauge:



Another interesting gauge: the maximally Abelian gauge
$\longrightarrow$ dual superconductor picture of confinement





Identification of two-loop diagrams
as IR leading diagrams.

Algorithm for the derivation

Starting points

DSEs:

$$\frac{\delta\Gamma}{\delta\Phi_i} = \left.\frac{\delta S}{\delta\phi_i}\right|_{\phi_i=\Phi_i+D_J^{ij}\,\delta/\delta\Phi_j}$$

with $\Phi = \langle\phi\rangle_J$

RGEs:

$$\partial_k\Gamma_k[\Phi] = \frac{1}{2}\mathrm{Str}\left(\Gamma_k[\Phi]+R_k\right)^{-1}\partial_k R_k$$

Obtain DSEs/RGEs by differentiations with respect to fields.

$$\frac{\delta}{\delta\Phi_i}D_J^{jk} = D_J^{jm}\Gamma_J^{min}D_J^{nk},$$

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

$$\frac{\delta}{\delta\Phi_i}\Gamma_J^{j_1\ldots j_n} = -\frac{\delta\Gamma}{\delta\Phi_i\delta\Phi_{j_1}\ldots\delta\Phi_{j_n}} = \Gamma_J^{ij_1\ldots j_n}$$

Concepts of *DoFun*

Derivation is a two-step process!

1.
   - Reduce action to combinatoric information:
     Which interactions do we have?
   - Put all indices and momentum dependence into one index only.
   - Express everything in generic propagators and vertices.
     E.g. from Yang-Mills theory:

$$\int_{q_i} g^2 f^{eab} f^{ecd} A^a_\mu(q_1) A^b_\nu(q_2) A^c_\mu(q_3) A^d_\nu(-q_1-q_2-q_3) \longrightarrow S^{AAAA}_{ijlm} A_i A_j A_l A_m$$

$$\longrightarrow \texttt{\{A,A,A,A\}}$$

2.
   - Define propagators and vertices specifically,
     i.e. Feynman rules required ($\longrightarrow$ *DoFR*)
   - Replace generic by specific quantities.

Actions

DSEs: classical action

RGEs: ansatz for the effective average action

Actions

DSEs: classical action

RGEs: ansatz for the effective average action

Symbolic action:
contains only types of interactions,
distinguishes between bosons and Grassmann-valued fields,
interaction given by list of fields, e.g. `{A,A,A,A}`

Physical action:
contains all details like indices and momentum dependence
$\longrightarrow$ Feynman rules (possible with package *DoFR*)

Symbolic actions

Contain types of interactions, but no specific details.

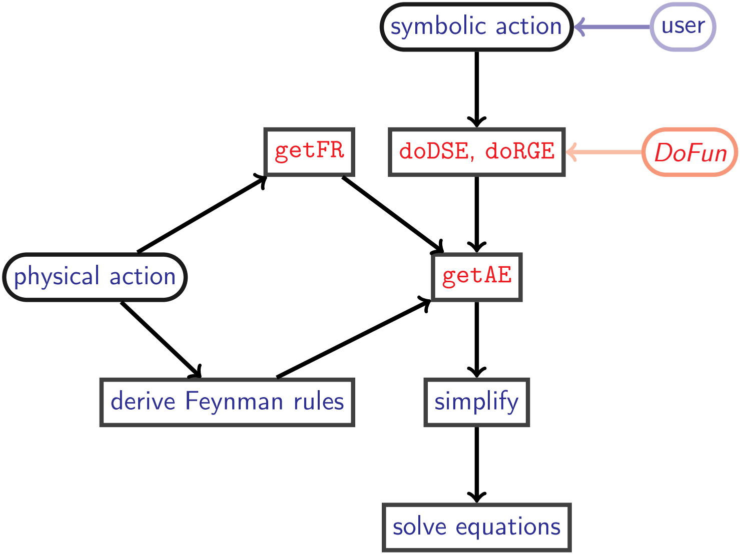
Example: $\{\{\varphi,\varphi\},\{\varphi,\varphi,\varphi,\varphi\}\}$

$$S[\varphi] = \frac{1}{2}\int \frac{d^d q}{(2\pi)^d}\varphi(q)q^2\varphi(-q) \qquad \text{(DSEs)}$$
$$+\frac{1}{4!}\int \frac{d^d q}{(2\pi)^d}\frac{d^d r}{(2\pi)^d}\frac{d^d s}{(2\pi)^d}\lambda\,\varphi(q)\varphi(r)\varphi(s)\varphi(-q-r-s)$$

or

$$\Gamma_k[\varphi] = \int \frac{1}{2}\frac{d^d q}{(2\pi)^d}\varphi^i(q)Z_k(q^2)q^2\varphi^i(-q) \qquad \text{(RGEs)}$$
$$+\frac{1}{4!}\int \frac{d^d q}{(2\pi)^d}\frac{d^d r}{(2\pi)^d}\frac{d^d s}{(2\pi)^d}\lambda_k(q,r,s)\varphi^i(q)\varphi^i(r)\varphi^j(s)\varphi^j(-q-r-s)$$

Workflow with *DoFun*



Building blocks

- Fields: `{phi, i}`
  pair of field name and (symbolic) index
- Propagator: `P[{phi,i}, {phi,j}]`
- Dressed vertex: `V[{phi,i}, {phi,j}, {phi, l}, {phi, m}]`
- Bare vertex: `S[{phi,i}, {phi,j}, {phi, l}, {phi, m}]`
- Regulator insertion: `dR[{phi,i}, {phi,j}]`

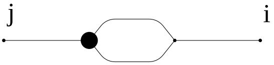
Building blocks

- Fields: `{phi, i}`
  pair of field name and (symbolic) index
- Propagator: `P[{phi,i}, {phi,j}]`
- Dressed vertex: `V[{phi,i}, {phi,j}, {phi, l}, {phi, m}]`
- Bare vertex: `S[{phi,i}, {phi,j}, {phi, l}, {phi, m}]`
- Regulator insertion: `dR[{phi,i}, {phi,j}]`

Group everything together: `op[...]` $\sim$ diagram

Example:

```
op[S[{W,i},{W,k},{W,l}], P[{W,k},{W,ks}], P[{W,l},{W,ls}],
 V[{W,ks},{W,ls},{W,j}]]
```

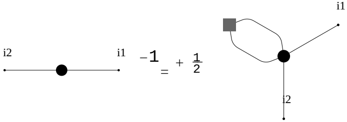


Example: First Step

Derivation of DSEs/RGEs with `doDSE`/`doRGE`,
e.g. scalar $O(N)$ theory (in symmetric phase):

```
doRGE[{{phi, phi}, {phi, phi, phi, phi}}, {phi, phi}]

--> 1/2 op[dR[{phi, r1}, {phi, s1}], P[{phi, t1}, {phi, r1}],
  P[{phi, s1}, {phi, v1}],
  V[{phi, i2}, {phi, i1}, {phi, v1}, {phi, t1}]]
```



Feynman rules

For replacing the generic quantities by specific ones their Feynman rules are required:

```
defineFieldsSpecific[{phi[momentum, type]}];

P[phi[p1_, i_], phi[p2_, j_], explicit -> True] :=
 delta[type, i, j]/(p1^2+R[k,p1^2])

dR[phi[p1_, i_], phi[p2_, j_], explicit -> True] :=
 delta[type, i, j] dR[k, p1^2]

V[phi[p1_, i_], phi[p2_, j_], phi[p3_, k_], phi[p4_, l_],
  explicit -> True] :=
 -lambda delta[type, i, l] delta[type, j, k] -
  lambda delta[type, i, k] delta[type, j, l] -
  lambda delta[type, i, j] delta[type, k, l]
```

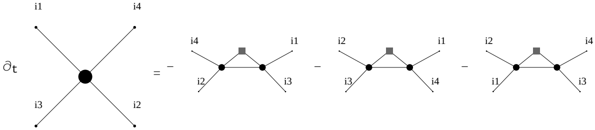
Example: Second step

```
fourPointAlg=Plus@@getAE[fourPointSymbolic, {{phi, a, 0, i},
 {phi, b, 0, j}, {phi, c, 0, l}, {phi, d, 0, m}}];

integrateDeltas[
 delta[type, i, 1] delta[type, j, 1] delta[type, l, 1]
 delta[type, m, 1] fourPointAlg
] /. dim[type] :> N;
```

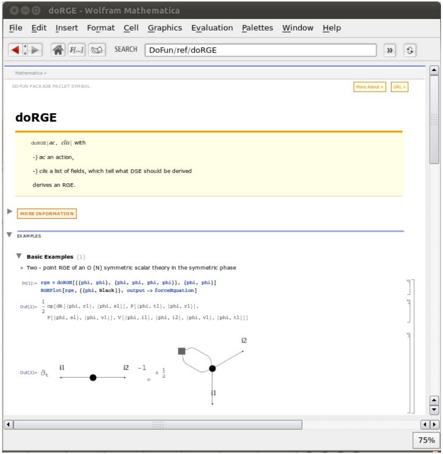
This yields the right-hand side of the flow of the four-point function:

$$-\lambda^2(24+3N)\int\frac{d^dq}{(2\pi)^d}\frac{\partial_t R_k}{(q^2+R_k)^3}$$



Documentation

- Basic usage information via *Mathematica*'s frontend, e.g. `?doRGE`.
- Help in html on homepage.
- Documentation available in the Documentation Center:
  basic information implemented, updates planned.



Applications of *DoFun*

Results obtained with support of *DoFun*

- Yang-Mills theory in the maximally Abelian gauge [Huber, Schwenzer, Alkofer, EPJC68]
- Gribov-Zwanziger action in the Landau gauge [Huber, Alkofer, Sorella, PRD81]
- Yang-Mills theory + fundamental/adjoint scalar [Fister, Alkofer, Schwenzer, PLB688; Alkofer, Fister, Macher, Maas, 1011.5831]
- Bound state equations [Alkofer, Alkofer, 1102.2753]

Tests by comparing with known results [Huber, Braun, 1102.5307]:

- $O(N)$ symmetric scalar theory [Berges, Tetradis, Wetterich, PR363]
- Gross-Neveu model in $3d$ [Braun, Gies, Scherer, 1011.1456] (Dirac structure only requires one set of rules)

Summary

What *DoFun* does not do:

- Solve DSEs or RGEs.
- Simplify color, Dirac, ... algebra.
  $\longrightarrow$ Use existing packages or own solutions.

Summary

Input for *DoFun*:

- Symbolic action (incl. truncation)
- Feynman rules

What *DoFun* can do:

- Derive DSEs and RGEs for bosons and fermions.
- Assist in deriving Feynman rules ($\to$ *DoFR*).
- Easy use of user's preferred syntax for Feynman rules.
- Plot simple graphical representations of the equations.
- Alleviate testing of new truncations.
- Handling of many diagrams and involved tensor structures as required, for example, in gauge theories.

The end

Thank you very much for your attention.