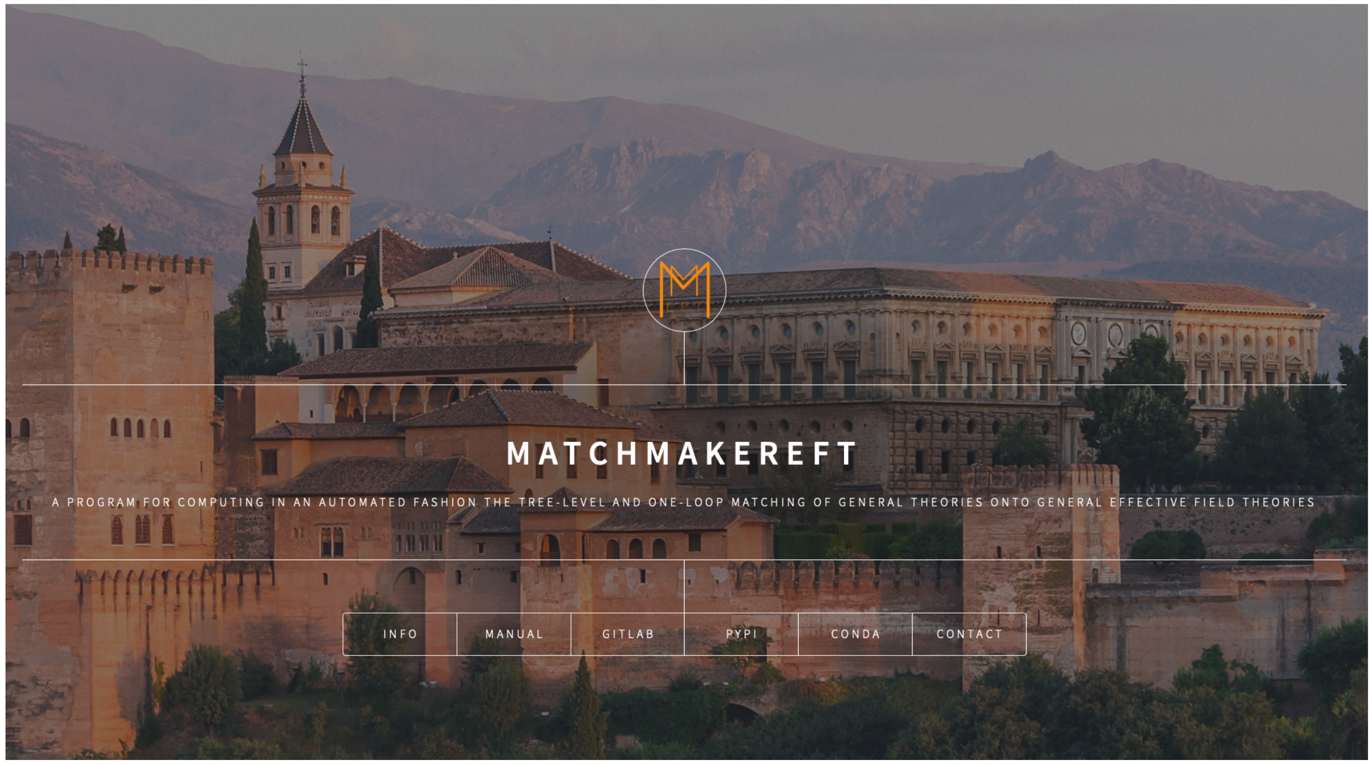
Matchmakereft

And its future developments

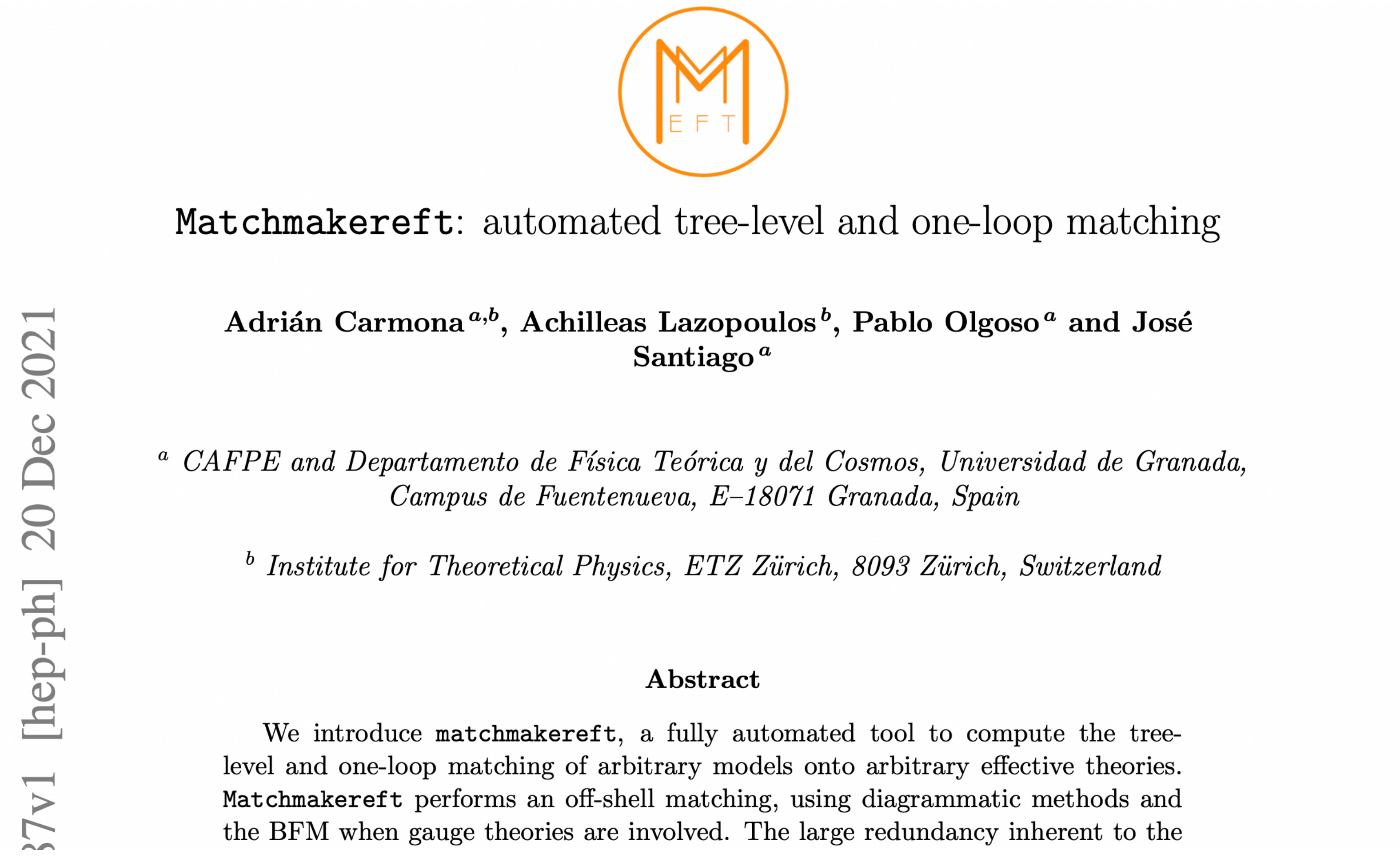


MATCHMAKEREFT

A PROGRAM FOR COMPUTING IN AN AUTOMATED FASHION THE TREE-LEVEL AND ONE-LOOP MATCHING OF GENERAL THEORIES ONTO GENERAL EFFECTIVE FIELD THEORIES

Achilleas Lazopoulos | 15 Sep. 2022 | SMEFT TOOLS

A VERY LONG PROJECT FINALLY REACHED PUBLICATION, SEE e-Print: 2112.10787 [hep-ph]



`Matchmakereft`: automated tree-level and one-loop matching

Adrián Carmona[a,b], **Achilleas Lazopoulos**[b], **Pablo Olgoso**[a] **and José Santiago**[a]

[a] *CAFPE and Departamento de Física Teórica y del Cosmos, Universidad de Granada, Campus de Fuentenueva, E–18071 Granada, Spain*

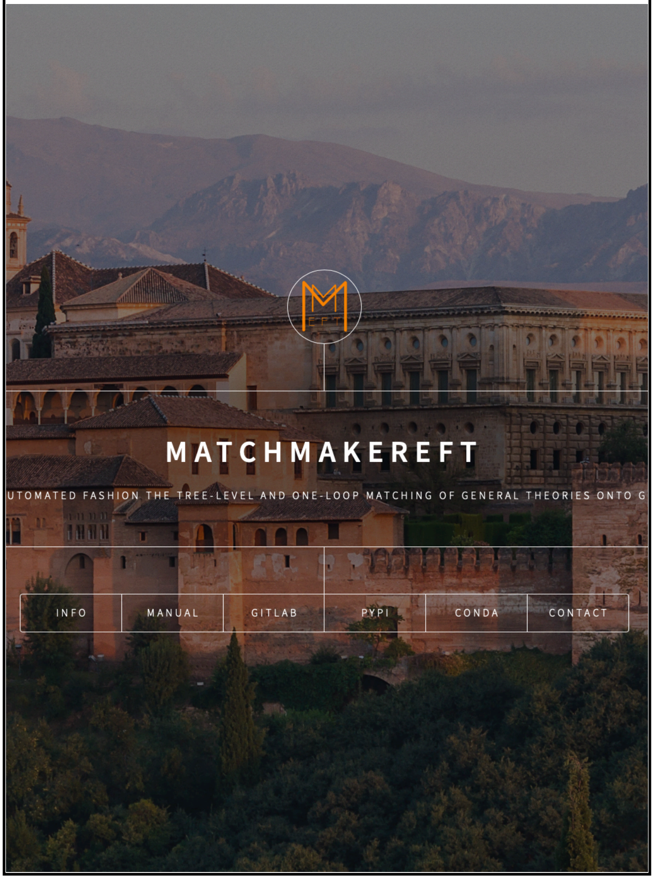
[b] *Institute for Theoretical Physics, ETZ Zürich, 8093 Zürich, Switzerland*

Abstract

We introduce `matchmakereft`, a fully automated tool to compute the tree-level and one-loop matching of arbitrary models onto arbitrary effective theories. `Matchmakereft` performs an off-shell matching, using diagrammatic methods and the BFM when gauge theories are involved. The large redundancy inherent to the

Leonard Cohen was still alive when we started (I checked)

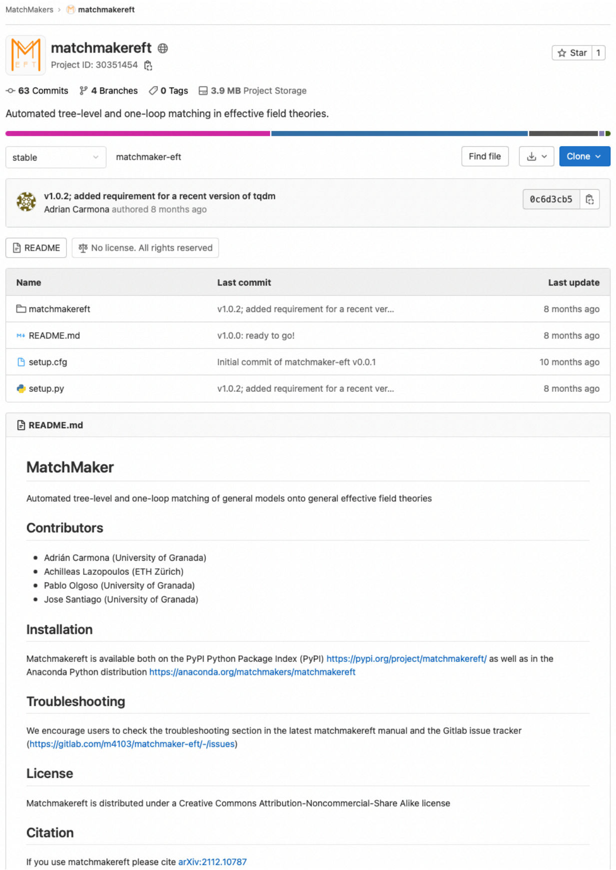
WEBSITE OF THE PROJECT
https://ftae.ugr.es/matchmakereft/



MATCHMAKEREFT

UTOMATED FASHION THE TREE-LEVEL AND ONE-LOOP MATCHING OF GENERAL THEORIES ONTO G

THERE IS A GITLAB PAGE, GOOD FOR CLONING AND RAISING ISSUES.



matchmakereft

Project ID: 30351454

63 Commits · 4 Branches · 0 Tags · 3.9 MB Project Storage

Automated tree-level and one-loop matching in effective field theories.

v1.0.2; added requirement for a recent version of tqdm
Adrian Carmona authored 8 months ago · 0c6d3cb5

Name	Last commit	Last update
matchmakereft	v1.0.2; added requirement for a recent ver...	8 months ago
README.md	v1.0.0: ready to go!	8 months ago
setup.cfg	Initial commit of matchmaker-eft v0.0.1	10 months ago
setup.py	v1.0.2; added requirement for a recent ver...	8 months ago

README.md

MatchMaker

Automated tree-level and one-loop matching of general models onto general effective field theories

Contributors

- Adrián Carmona (University of Granada)
- Achilleas Lazopoulos (ETH Zürich)
- Pablo Olgoso (University of Granada)
- Jose Santiago (University of Granada)

Installation

Matchmakereft is available both on the PyPI Python Package Index (PyPI) https://pypi.org/project/matchmakereft/ as well as in the Anaconda Python distribution https://anaconda.org/matchmakers/matchmakereft

Troubleshooting

We encourage users to check the troubleshooting section in the latest matchmakereft manual and the Gitlab issue tracker (https://gitlab.com/m4103/matchmaker-eft/-/issues)

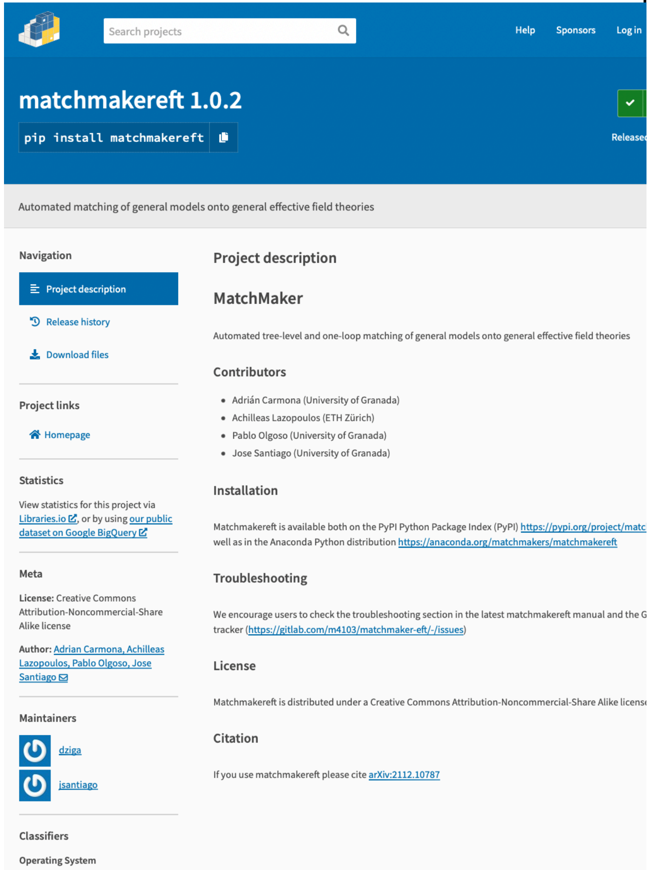
License

Matchmakereft is distributed under a Creative Commons Attribution-Noncommercial-Share Alike license

Citation

If you use matchmakereft please cite arXiv:2112.10787

YOU CAN INSTALL IT WITH PIP…



matchmakereft 1.0.2

```
pip install matchmakereft
```

Automated matching of general models onto general effective field theories

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Meta

License: Creative Commons Attribution-Noncommercial-Share Alike license

Author: Adrian Carmona, Achilleas Lazopoulos, Pablo Olgoso, Jose Santiago

Maintainers

dziga
jsantiago

Classifiers

Operating System

Project description

MatchMaker

Automated tree-level and one-loop matching of general models onto general effective field theories

Contributors

- Adrián Carmona (University of Granada)
- Achilleas Lazopoulos (ETH Zürich)
- Pablo Olgoso (University of Granada)
- Jose Santiago (University of Granada)

Installation

Matchmakereft is available both on the PyPI Python Package Index (PyPI) https://pypi.org/project/matc
well as in the Anaconda Python distribution https://anaconda.org/matchmakers/matchmakereft

Troubleshooting

We encourage users to check the troubleshooting section in the latest matchmakereft manual and the G
tracker (https://gitlab.com/m4103/matchmaker-eft/-/issues)

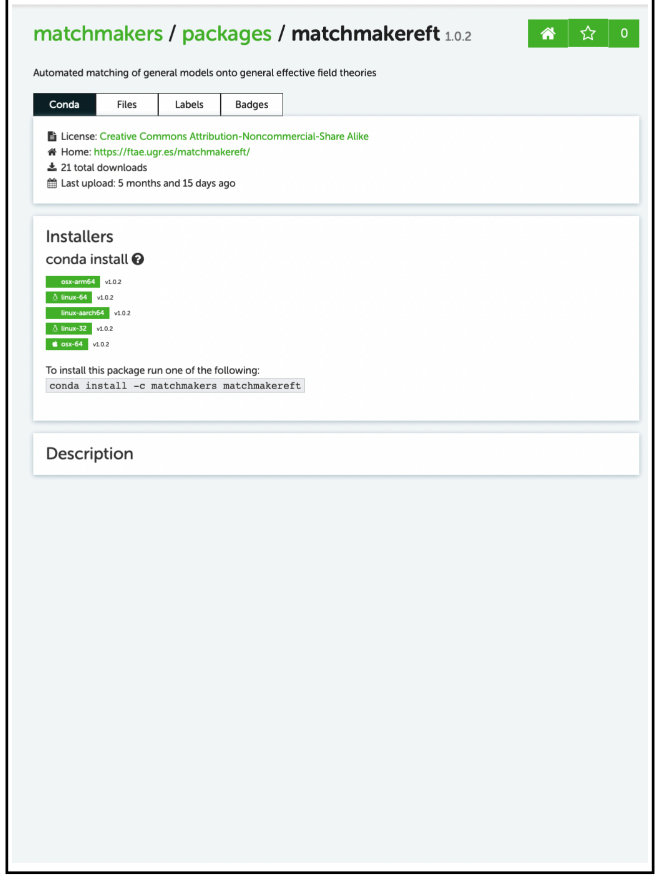
License

Matchmakereft is distributed under a Creative Commons Attribution-Noncommercial-Share Alike licens

Citation

If you use matchmakereft please cite arXiv:2112.10787

…OR CONDA



matchmakers / packages / **matchmakereft** 1.0.2

Automated matching of general models onto general effective field theories

License: Creative Commons Attribution-Noncommercial-Share Alike
Home: https://ftae.ugr.es/matchmakereft/
21 total downloads
Last upload: 5 months and 15 days ago

Installers

conda install

osx-arm64	v1.0.2
linux-64	v1.0.2
linux-aarch64	v1.0.2
linux-32	v1.0.2
osx-64	v1.0.2

To install this package run one of the following:

```
conda install -c matchmakers matchmakereft
```

Description

1. Matching a UV model to an EFT model at tree-level and one loop

Off-shell matching

Background field gauge

Reduction to physical basis

γ_5 and its treatment

limitations

2. Computing the one loop RGEs of a model

RGEs, a special kind of matching

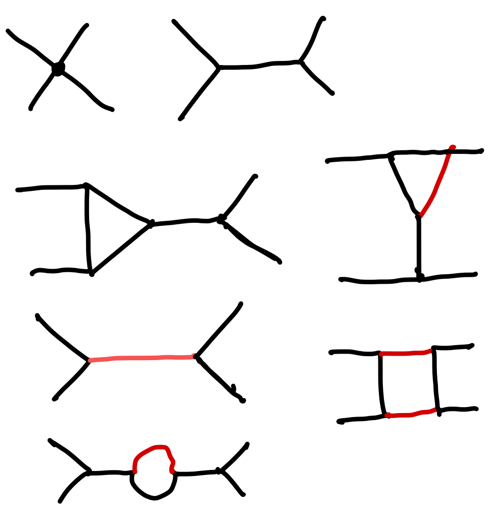
Redundant operators treated the same as in matching

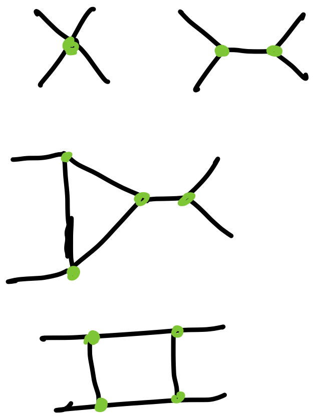
Tadpoles and their treatment

Off-shell matching

ON-SHELL MATCHING: AMPLITUDES ARE MATCHED, ALL DIAGRAMS ARE PRESENT, ALL MOMENTA ARE ON-SHELL

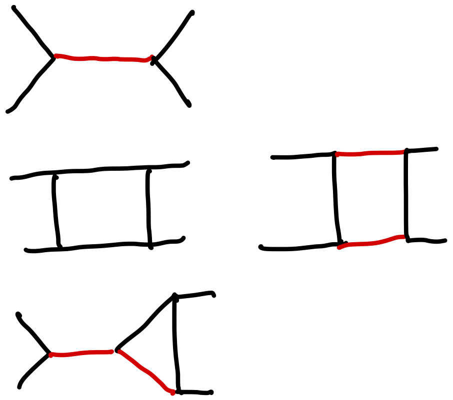
$$\left[\mathcal{M}_{UV}\right]^{\text{REN}}_{\text{EXPANDED}\ \mathcal{O}\left(\left(\frac{p}{\Lambda}\right)^{k}\right)} = \left[\mathcal{M}_{EFT}\right]^{\text{REN}}_{\text{EXPANDED}\ \mathcal{O}\left(\left(\frac{p}{\Lambda}\right)^{k}\right)}$$

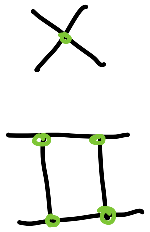




OFF-SHELL MATCHING: 1LPI GREEN'S FUNCTIONS ARE MATCHED, ALL MOMENTA ARE OFF-SHELL

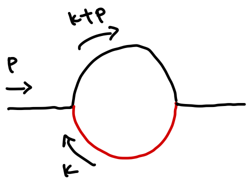
$$\left[\Gamma^{(n)}_{\substack{UV\\ 1LPI}}\right]^{\text{REN}}_{\text{EXPANDED}\ \mathcal{O}\left(\left(\frac{p}{\Lambda}\right)^{k}\right)} = \left[\Gamma^{(n)}_{EFT}\right]^{\text{REN}}_{\text{EXPANDED}\ \mathcal{O}\left(\left(\frac{p}{\Lambda}\right)^{k}\right)}$$





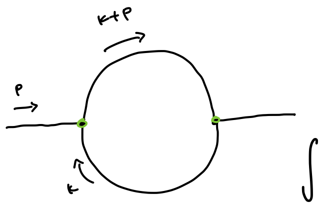
Off-shell matching

HARD REGION EXPANSION: a diagram of the UV theory



$$\sim \int [dk] \overbrace{\underbrace{\frac{i}{k^2-M^2}\,\frac{i}{(k+p)^2}}_{\text{IR FINITE}}}^{\text{UV SINGULAR}}$$

On the other hand the EFT diagrams have no heavy scale: they vanish upon expansion



$$\int [dk] \frac{1}{k^2 (k+p)^2} = \underbrace{\int [dk] \frac{1}{k^4} - \frac{p^2}{k^6} + \frac{(4\,p\cdot k)^2}{k^8}}_{\text{scale-less integrals}} = 0$$

…the UV singularities cancel the IR singularities

Its hard region has new singularities

$$p \ll k \; : \quad \frac{1}{(k+p)^2} = \frac{1}{k^2\left(1+\frac{2k\cdot p+p^2}{k^2}\right)}$$

$$= \frac{1}{k^2}\left(1 - \frac{2k\cdot p + p^2}{k^2} + \left(\frac{2k\cdot p}{k^2}\right)^2 + O(p^3)\right)$$

$$\int [dk] \overbrace{\underbrace{\frac{1}{(k^2-M^2)\,k^2}}_{\text{IR FINITE}}}^{\text{UV SINGULAR}} \underbrace{\overbrace{- \frac{p^2}{(k^2-M^2)\,k^4} + \frac{4(k\cdot p)^2}{(k^2-M^2)\,k^6}}^{\text{UV FINITE}}}_{\text{IR SINGULAR}} + O(p^3)$$

…so schematically, the IR structure of the hard region of the UV diagrams is in fact the UV structure of the EFT diagrams. The same applies for any finite term that originates from poles times epsilon

$$\left[\Gamma^{(n)}_{\substack{\text{UV}\\ \text{1LPI}}}\right]^{\text{REN}}_{\substack{\text{EXPANDED}\\ O\left(\left(\frac{p}{\Lambda}\right)^k\right)}} = \left[\Gamma^{(n)}_{\text{EFT}}\right]^{\text{REN}}_{\substack{\text{EXPANDED}\\ O\left(\left(\frac{p}{\Lambda}\right)^k\right)}}$$

$$\frac{a}{\epsilon_{UV}} - \frac{b}{\epsilon_{IR}} \qquad\qquad \frac{b}{\epsilon_{UV}} - \frac{b}{\epsilon_{IR}}$$

same IR structure

$$\frac{a}{\epsilon_{UV}} - \frac{b}{\epsilon_{UV,EFT}} \qquad\qquad 0$$

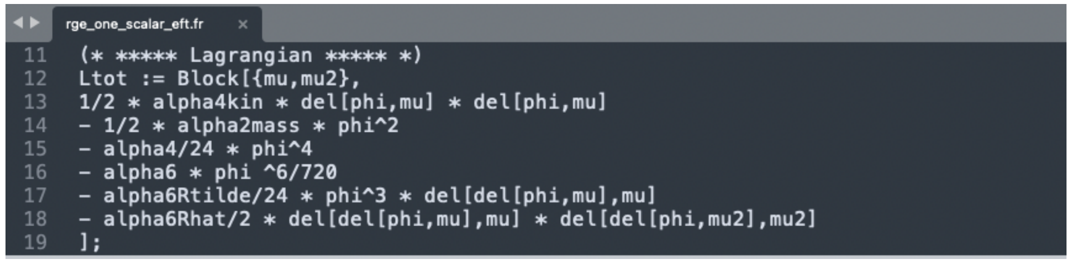
Off-shell matching

OFF-SHELL MATCHING, the price to pay: redundant operators in the EFT model are necessary

$$\mathcal{L} = \mathcal{L}^{(4)} + \underbrace{\sum_i \underset{\substack{\downarrow \\ \text{physical}}}{\frac{c_i}{\Lambda^2}} O_i + \sum_i \underset{\substack{\downarrow \\ \text{redundant by e.o.m}}}{\frac{b_i}{\Lambda^2}} R_i}_{\text{complete Green basis}}$$

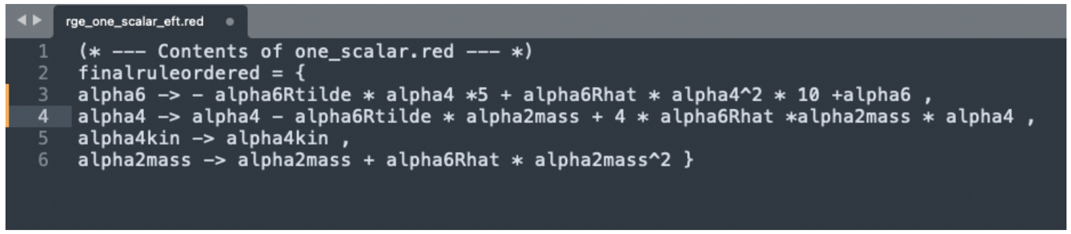
Using e.o.m $\{b_i\} \longrightarrow \{C_i\}$ [user supplied info]

This means that the EFT Lagrangian, as specified in the corresponding Feynrules file has redundant operators, here alphaRtilde and alphaRhat…



```
(* ***** Lagrangian ***** *)
Ltot := Block[{mu,mu2},
1/2 * alpha4kin * del[phi,mu] * del[phi,mu]
- 1/2 * alpha2mass * phi^2
- alpha4/24 * phi^4
- alpha6 * phi ^6/720
- alpha6Rtilde/24 * phi^3 * del[del[phi,mu],mu]
- alpha6Rhat/2 * del[del[phi,mu],mu] * del[del[phi,mu2],mu2]
];
```

…and the reduction relations must be declared in a .red file



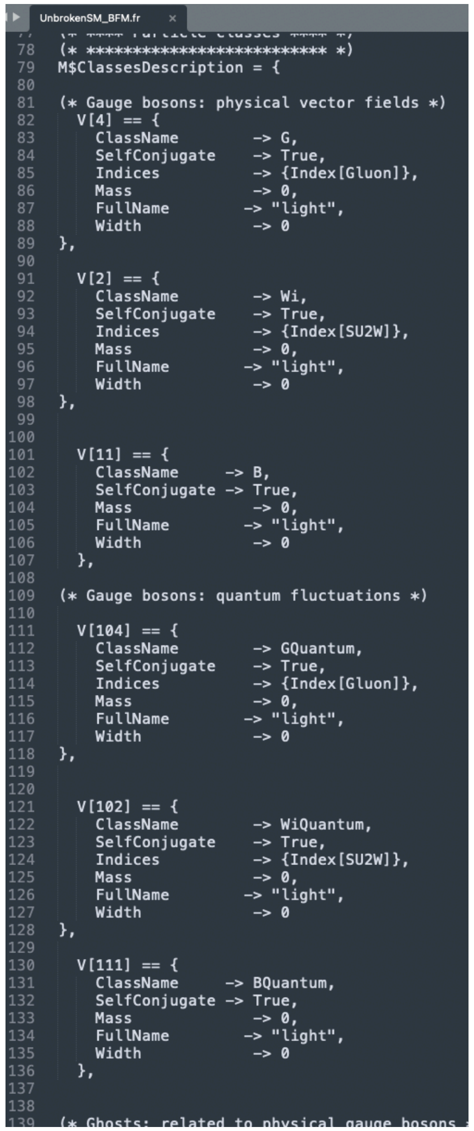
```
(* --- Contents of one_scalar.red --- *)
finalruleordered = {
alpha6 -> - alpha6Rtilde * alpha4 *5 + alpha6Rhat * alpha4^2 * 10 +alpha6 ,
alpha4 -> alpha4 - alpha6Rtilde * alpha2mass + 4 * alpha6Rhat *alpha2mass * alpha4 ,
alpha4kin -> alpha4kin ,
alpha2mass -> alpha2mass + alpha6Rhat * alpha2mass^2 }
```

Background Field gauge

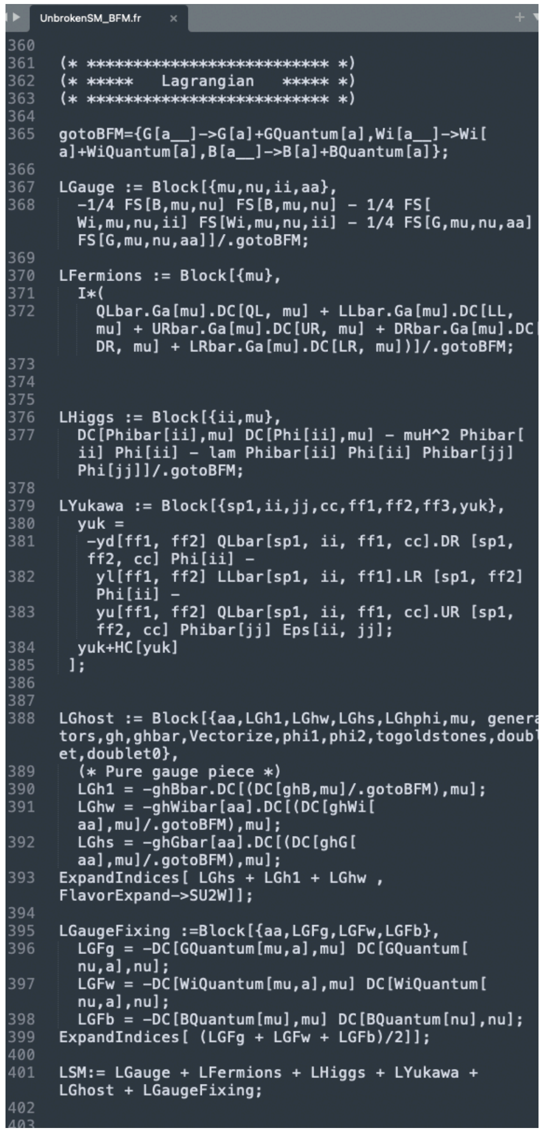
The background field gauge is employed for all gauge fields

- Gauge fields are split in background fields (that do not propagate in the loop) and quantum fields (that appear in the loop).
- The gauge is chosen such that the terms in L that contain only background fields are gauge invariant.
- Vertices with gauge bosons and ghosts are modified
- The great advantage: the gauge invariance of the results in intermediate steps and at the final result is built in

The corresponding fields are declared in .fr files



```
(* ************************* *)
M$ClassesDescription = {

(* Gauge bosons: physical vector fields *)
  V[4] == {
    ClassName        -> G,
    SelfConjugate    -> True,
    Indices          -> {Index[Gluon]},
    Mass             -> 0,
    FullName        -> "light",
    Width            -> 0
},

  V[2] == {
    ClassName        -> Wi,
    SelfConjugate    -> True,
    Indices          -> {Index[SU2W]},
    Mass             -> 0,
    FullName        -> "light",
    Width            -> 0
},


  V[11] == {
    ClassName     -> B,
    SelfConjugate -> True,
    Mass             -> 0,
    FullName        -> "light",
    Width            -> 0
  },

(* Gauge bosons: quantum fluctuations *)

  V[104] == {
    ClassName        -> GQuantum,
    SelfConjugate    -> True,
    Indices          -> {Index[Gluon]},
    Mass             -> 0,
    FullName        -> "light",
    Width            -> 0
},


  V[102] == {
    ClassName        -> WiQuantum,
    SelfConjugate    -> True,
    Indices          -> {Index[SU2W]},
    Mass             -> 0,
    FullName        -> "light",
    Width            -> 0
},

  V[111] == {
    ClassName     -> BQuantum,
    SelfConjugate -> True,
    Mass             -> 0,
    FullName        -> "light",
    Width            -> 0
  },


(* Ghosts: related to physical gauge bosons
```



```

(* ************************* *)
(* *****   Lagrangian   ***** *)
(* ************************* *)

gotoBFM={G[a__]->G[a]+GQuantum[a],Wi[a__]->Wi[
a]+WiQuantum[a],B[a__]->B[a]+BQuantum[a]};

LGauge := Block[{mu,nu,ii,aa},
  -1/4 FS[B,mu,nu] FS[B,mu,nu] - 1/4 FS[
  Wi,mu,nu,ii] FS[Wi,mu,nu,ii] - 1/4 FS[G,mu,nu,aa]
  FS[G,mu,nu,aa]]/.gotoBFM;

LFermions := Block[{mu},
  I*(
    QLbar.Ga[mu].DC[QL, mu] + LLbar.Ga[mu].DC[LL,
    mu] + URbar.Ga[mu].DC[UR, mu] + DRbar.Ga[mu].DC
    DR, mu] + LRbar.Ga[mu].DC[LR, mu])]/.gotoBFM;



LHiggs := Block[{ii,mu},
  DC[Phibar[ii],mu] DC[Phi[ii],mu] - muH^2 Phibar[
  ii] Phi[ii] - lam Phibar[ii] Phi[ii] Phibar[jj]
  Phi[jj]]/.gotoBFM;

LYukawa := Block[{sp1,ii,jj,cc,ff1,ff2,ff3,yuk},
  yuk =
   -yd[ff1, ff2] QLbar[sp1, ii, ff1, cc].DR [sp1,
   ff2, cc] Phi[ii] -
    yl[ff1, ff2] LLbar[sp1, ii, ff1].LR [sp1, ff2]
    Phi[ii] -
    yu[ff1, ff2] QLbar[sp1, ii, ff1, cc].UR [sp1,
    ff2, cc] Phibar[jj] Eps[ii, jj];
  yuk+HC[yuk]
 ];


LGhost := Block[{aa,LGh1,LGhw,LGhs,LGhphi,mu, genera
tors,gh,ghbar,Vectorize,phi1,phi2,togoldstones,doubl
et,doublet0},
  (* Pure gauge piece *)
  LGh1 = -ghBbar.DC[(DC[ghB,mu]/.gotoBFM),mu];
  LGhw = -ghWibar[aa].DC[(DC[ghWi[
  aa],mu]/.gotoBFM),mu];
  LGhs = -ghGbar[aa].DC[(DC[ghG[
  aa],mu]/.gotoBFM),mu];
ExpandIndices[ LGhs + LGh1 + LGhw ,
FlavorExpand->SU2W]];

LGaugeFixing :=Block[{aa,LGFg,LGFw,LGFb},
  LGFg = -DC[GQuantum[mu,a],mu] DC[GQuantum[
  nu,a],nu];
  LGFw = -DC[WiQuantum[mu,a],mu] DC[WiQuantum[
  nu,a],nu];
  LGFb = -DC[BQuantum[mu],mu] DC[BQuantum[nu],nu];
ExpandIndices[ (LGFg + LGFw + LGFb)/2]];

LSM:= LGauge + LFermions + LHiggs + LYukawa +
LGhost + LGaugeFixing;

```

γ_5 and its treatment

Usual problems with the definition of gamma matrices in DimReg.

$$\{\gamma^\mu, \gamma^\nu\} \overset{?}{=} 2g_{\mu\nu} \cdot \mathbf{1}\,, \quad g^{\mu}{}_{\mu} = D\,,$$

$$\mathrm{Tr}(\gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma \gamma_5) = 4\mathrm{i}\varepsilon^{\mu\nu\rho\sigma}$$

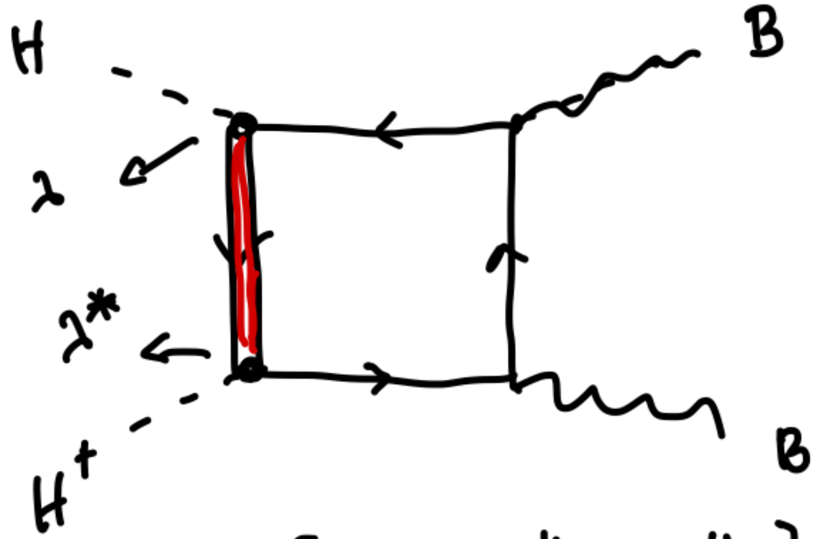
☠ ☣ ☢

Particularly annoying

- Worse than normal: when computing cross-sections one has to use a consistent scheme himself.
- Here, we compute Wilson coefficients to be used in one-loop computations by someone else.

γ_5 and its treatment

γ5 - ambiguities are present in closed fermion loops.



$\mathrm{Tr}[\gamma_5 \gamma^{\mu_1} \gamma^{\mu_2} .. \gamma^{\mu_6}]$

has ambiguous $\sim D-4$ term

$\mathrm{Tr}[\ ] \sim i\,\varepsilon^{\mu_1 \dots}$

At one loop and with a renormalizable UV model, one can get away with fixing a posteriori the ambiguity, using hermitianity of the operators.

necessary for matching

$H^\dagger H\, X_{\mu\nu} \tilde{X}^{\mu\nu}$

$\downarrow$

hermitian

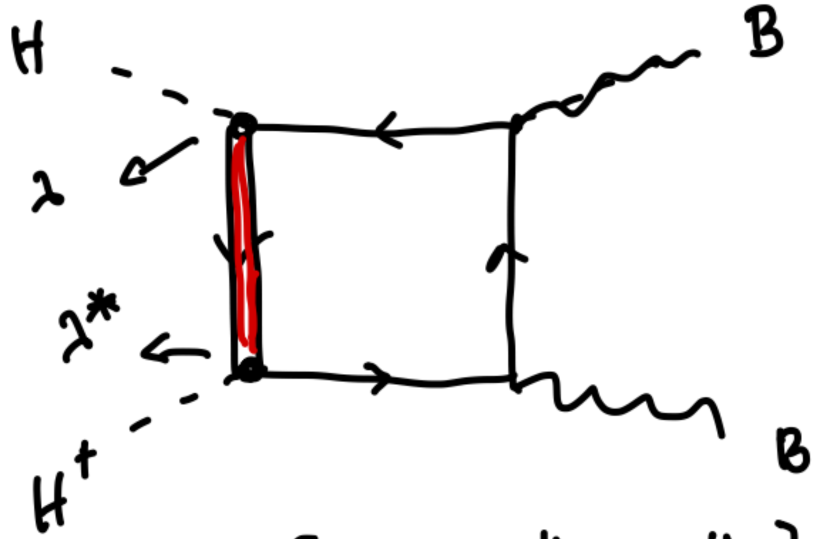
$\downarrow$

WC is real

↳ can be set to zero by hermitianity

γ_5 and its treatment

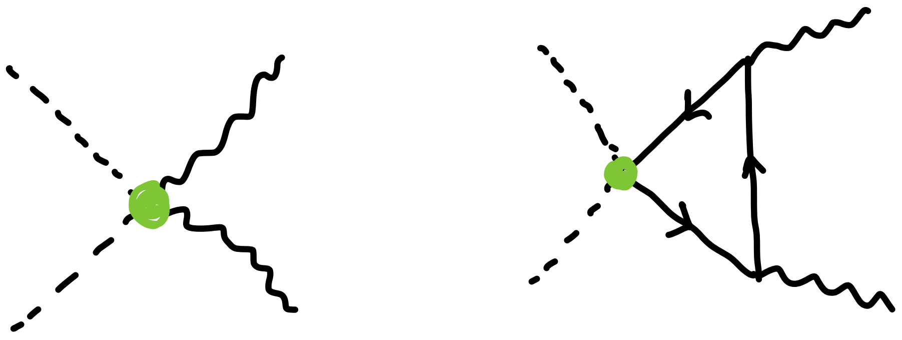
γ5 - ambiguity when specifying the Wilson coefficient



$$\mathrm{Tr}[\gamma_5 \gamma^{\mu_1} \gamma^{\mu_2} .. \gamma^{\mu_6}]$$

$$H^\dagger H \, X_{\mu\nu} \tilde{X}^{\mu\nu}$$

When performing one loop computations with the EFT with this coefficient, there is an identical ambiguity from the triangle loop (not fixed by anomaly cancelation)



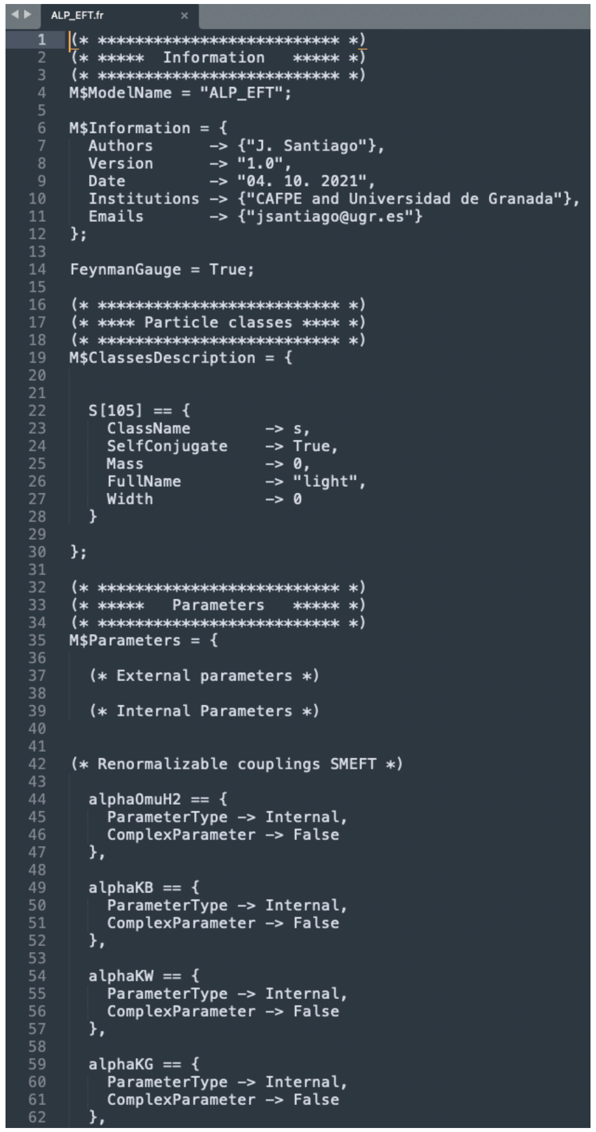
same trace

same ambiguity

One can see the ambiguity as a scheme dependence. We would like to avoid the cancelation and we set the ambiguous part to zero by fiat. It would be nice to have a γ5-scheme that does the job. ‘Naive’ γ5 prescription depends on wisely selecting a starting point for the trace, tHV has complications in implementation.

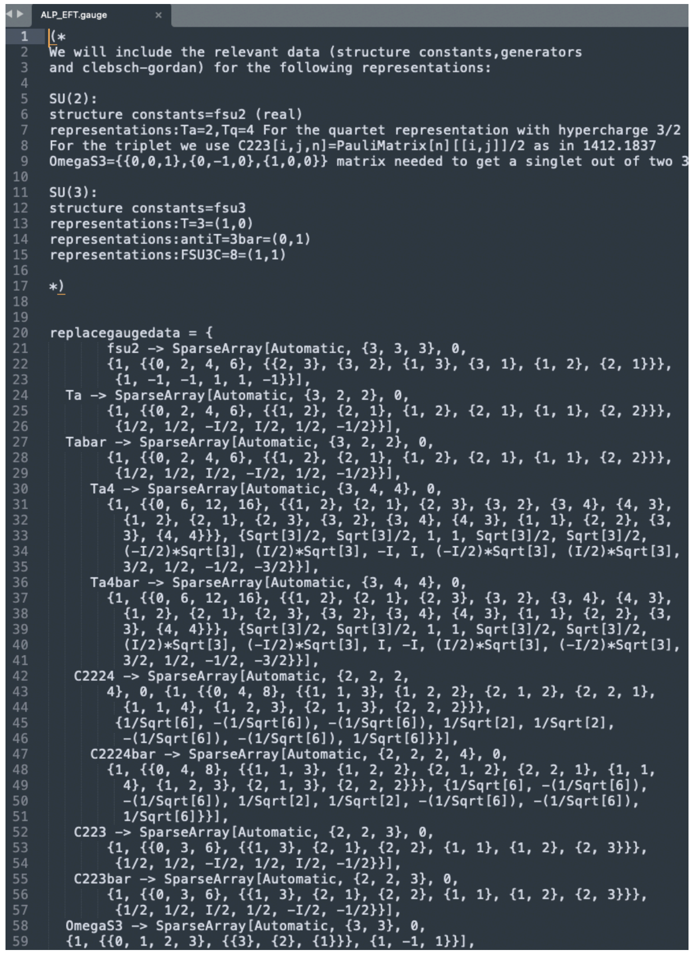
Input for constructing a model

A feynrules file **.fr



```
(* ************************** *)
(* *****  Information   ***** *)
(* ************************** *)
M$ModelName = "ALP_EFT";

M$Information = {
  Authors      -> {"J. Santiago"},
  Version      -> "1.0",
  Date         -> "04. 10. 2021",
  Institutions -> {"CAFPE and Universidad de Granada"},
  Emails       -> {"jsantiago@ugr.es"}
};

FeynmanGauge = True;

(* ************************** *)
(* **** Particle classes **** *)
(* ************************** *)
M$ClassesDescription = {


  S[105] == {
    ClassName        -> s,
    SelfConjugate    -> True,
    Mass             -> 0,
    FullName         -> "light",
    Width            -> 0
  }

};

(* ************************** *)
(* *****   Parameters   ***** *)
(* ************************** *)
M$Parameters = {

  (* External parameters *)

  (* Internal Parameters *)


(* Renormalizable couplings SMEFT *)

  alphaOmuH2 == {
    ParameterType -> Internal,
    ComplexParameter -> False
  },

  alphaKB == {
    ParameterType -> Internal,
    ComplexParameter -> False
  },

  alphaKW == {
    ParameterType -> Internal,
    ComplexParameter -> False
  },

  alphaKG == {
    ParameterType -> Internal,
    ComplexParameter -> False
  },
```

A group theory info file **.gauge

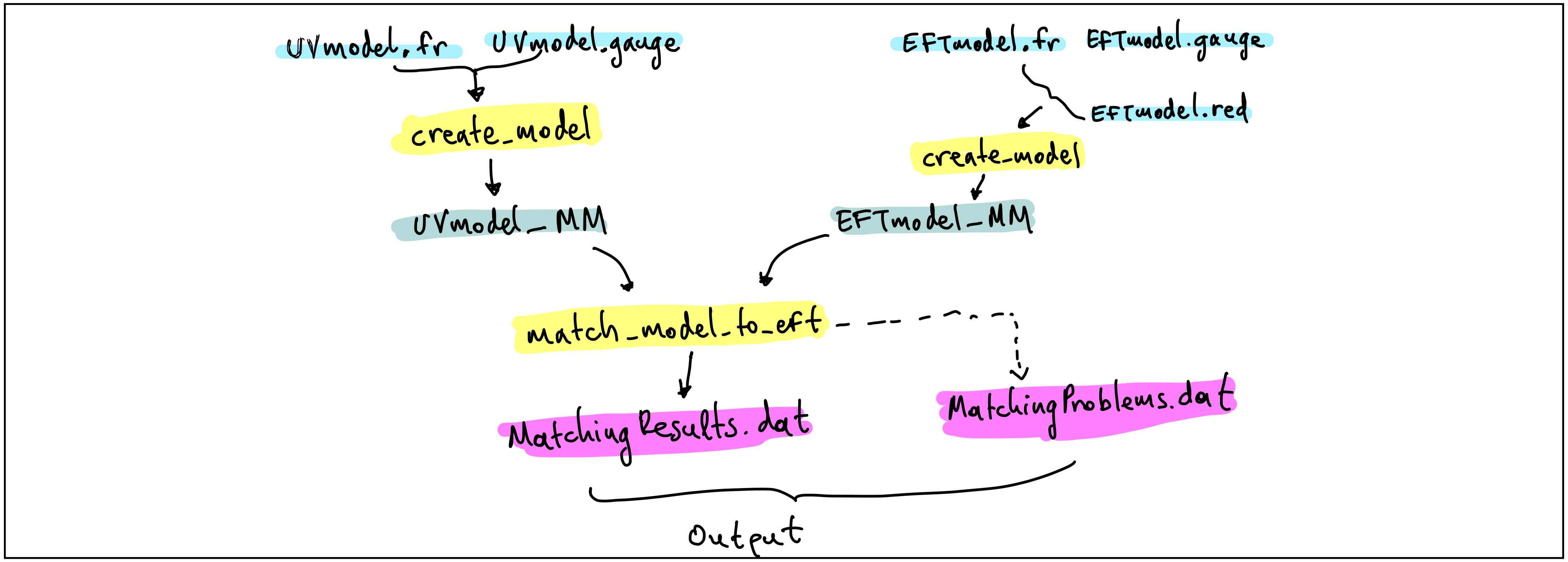


```
(*
We will include the relevant data (structure constants,generators
and clebsch-gordan) for the following representations:

SU(2):
structure constants=fsu2 (real)
representations:Ta=2,Tq=4 For the quartet representation with hypercharge 3/2
For the triplet we use C223[i,j,n]=PauliMatrix[n][[i,j]]/2 as in 1412.1837
OmegaS3={{0,0,1},{0,-1,0},{1,0,0}} matrix needed to get a singlet out of two 3

SU(3):
structure constants=fsu3
representations:T=3=(1,0)
representations:antiT=3bar=(0,1)
representations:FSU3C=8=(1,1)

*)


replacegaugedata = {
        fsu2 -> SparseArray[Automatic, {3, 3, 3}, 0,
        {1, {{0, 2, 4, 6}, {{2, 3}, {3, 2}, {1, 3}, {3, 1}, {1, 2}, {2, 1}}},
         {1, -1, -1, 1, 1, -1}}],
  Ta -> SparseArray[Automatic, {3, 2, 2}, 0,
        {1, {{0, 2, 4, 6}, {{1, 2}, {2, 1}, {1, 2}, {2, 1}, {1, 1}, {2, 2}}},
         {1/2, 1/2, -I/2, I/2, 1/2, -1/2}}],
  Tabar -> SparseArray[Automatic, {3, 2, 2}, 0,
        {1, {{0, 2, 4, 6}, {{1, 2}, {2, 1}, {1, 2}, {2, 1}, {1, 1}, {2, 2}}},
         {1/2, 1/2, I/2, -I/2, 1/2, -1/2}}],
     Ta4 -> SparseArray[Automatic, {3, 4, 4}, 0,
        {1, {{0, 6, 12, 16}, {{1, 2}, {2, 1}, {2, 3}, {3, 2}, {3, 4}, {4, 3},
          {1, 2}, {2, 1}, {2, 3}, {3, 2}, {3, 4}, {4, 3}, {1, 1}, {2, 2}, {3,
          3}, {4, 4}}}, {Sqrt[3]/2, Sqrt[3]/2, 1, 1, Sqrt[3]/2, Sqrt[3]/2,
          (-I/2)*Sqrt[3], (I/2)*Sqrt[3], -I, I, (-I/2)*Sqrt[3], (I/2)*Sqrt[3],
          3/2, 1/2, -1/2, -3/2}}],
     Ta4bar -> SparseArray[Automatic, {3, 4, 4}, 0,
        {1, {{0, 6, 12, 16}, {{1, 2}, {2, 1}, {2, 3}, {3, 2}, {3, 4}, {4, 3},
          {1, 2}, {2, 1}, {2, 3}, {3, 2}, {3, 4}, {4, 3}, {1, 1}, {2, 2}, {3,
          3}, {4, 4}}}, {Sqrt[3]/2, Sqrt[3]/2, 1, 1, Sqrt[3]/2, Sqrt[3]/2,
          (I/2)*Sqrt[3], (-I/2)*Sqrt[3], I, -I, (I/2)*Sqrt[3], (-I/2)*Sqrt[3],
          3/2, 1/2, -1/2, -3/2}}],
   C2224 -> SparseArray[Automatic, {2, 2, 2,
        4}, 0, {1, {{0, 4, 8}, {{1, 1, 3}, {1, 2, 2}, {2, 1, 2}, {2, 2, 1},
          {1, 1, 4}, {1, 2, 3}, {2, 1, 3}, {2, 2, 2}}},
         {1/Sqrt[6], -(1/Sqrt[6]), -(1/Sqrt[6]), 1/Sqrt[2], 1/Sqrt[2],
          -(1/Sqrt[6]), -(1/Sqrt[6]), 1/Sqrt[6]}}],
     C2224bar -> SparseArray[Automatic, {2, 2, 2, 4}, 0,
        {1, {{0, 4, 8}, {{1, 1, 3}, {1, 2, 2}, {2, 1, 2}, {2, 2, 1}, {1, 1,
          4}, {1, 2, 3}, {2, 1, 3}, {2, 2, 2}}}, {1/Sqrt[6], -(1/Sqrt[6]),
          -(1/Sqrt[6]), 1/Sqrt[2], 1/Sqrt[2], -(1/Sqrt[6]), -(1/Sqrt[6]),
          1/Sqrt[6]}}],
   C223 -> SparseArray[Automatic, {2, 2, 3}, 0,
        {1, {{0, 3, 6}, {{1, 3}, {2, 1}, {2, 2}, {1, 1}, {1, 2}, {2, 3}}},
         {1/2, 1/2, -I/2, 1/2, I/2, -1/2}}],
   C223bar -> SparseArray[Automatic, {2, 2, 3}, 0,
        {1, {{0, 3, 6}, {{1, 3}, {2, 1}, {2, 2}, {1, 1}, {1, 2}, {2, 3}}},
         {1/2, 1/2, I/2, 1/2, -I/2, -1/2}}],
  OmegaS3 -> SparseArray[Automatic, {3, 3}, 0,
  {1, {{0, 1, 2, 3}, {{3}, {2}, {1}}}, {1, -1, 1}}],
```

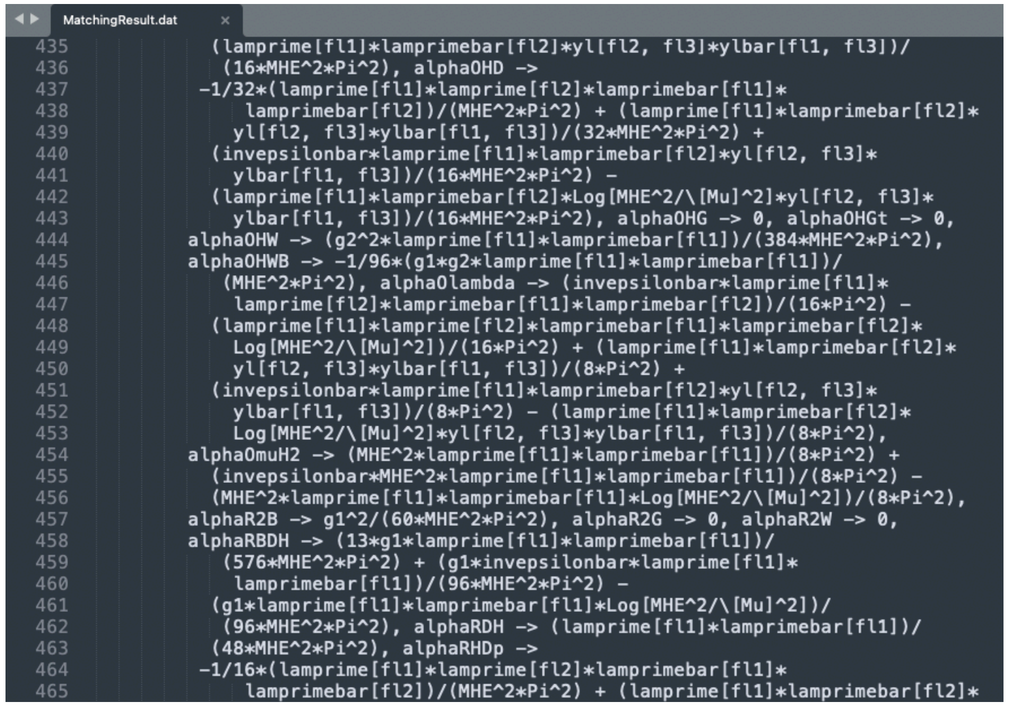
If it's an EFT, we also need a .red file

```
finalruleordered = {alphaOkappas -> alphaOkappas + 6*alphaOms2*alphaRsbox,
        alphaOkappasphi -> alphaOkappasphi + alphaOms2*alphaRphisbox +
          alphaOmuH2*(alphaRsphibox + alphaRsphiboxbar),
        alphaOlambda -> alphaOlambda + alphaOkappasphi*alphaRphisbox,
        alphaOlambdas -> alphaOlambdas + 12*alphaOkappas*alphaRsbox,
        alphaOlambdasphi -> alphaOlambdasphi + alphaOkappas*alphaRphisbox +
          2*alphaOkappasphi*(alphaRsbox + alphaRsphibox + alphaRsphiboxbar),
        alphaOms2 -> alphaOms2, alphaOmuH2 -> alphaOmuH2,
        alphaOs -> alphaOs - alphaOlambdasphi*alphaRphisbox -
          2*alphaOlambda*(alphaRsphibox + alphaRsphiboxbar),
        alphaOs3 -> alphaOs3 - (alphaOlambdas*alphaRphisbox)/6 -
          alphaOlambdasphi*(alphaRsbox + (alphaRsphibox + alphaRsphiboxbar)/2),
        alphaOsB -> alphaOsB, alphaOsBt -> alphaOsBt, alphaOsG -> alphaOsG,
        alphaOsGt -> alphaOsGt, alphaOsW -> alphaOsW, alphaOsWt -> alphaOsWt,
        alphaOlambdau[fl97_, fl99_] -> alphaOlambdau[fl97, fl99],
        alphaOlambdaubar[fl97_, fl99_] -> alphaOlambdaubar[fl97, fl99],
        alphaOlambdad[fl97_, fl99_] -> alphaOlambdad[fl97, fl99],
        alphaOlambdadbar[fl97_, fl99_] -> alphaOlambdadbar[fl97, fl99],
        alphaOlambdae[fl97_, fl99_] -> alphaOlambdae[fl97, fl99],
        alphaOlambdaebar[fl97_, fl99_] -> alphaOlambdaebar[fl97, fl99],
        alphaOsuphi[fl97_, fl99_] ->
         -(alphaRsphibox*alphaOlambdau[fl97, fl99]) + alphaOsuphi[fl97, fl99] +
          alphaOlambdau[miF1, fl99]*alphaRsq[fl97, miF1] +
          alphaOlambdau[fl97, miF1]*alphaRsubar[fl99, miF1],
        alphaOsuphibar[fl97_, fl99_] ->
         -(alphaRsphiboxbar*alphaOlambdaubar[fl97, fl99]) +
          alphaOsuphibar[fl97, fl99] + alphaOlambdaubar[miF1, fl99]*
           alphaRsqbar[fl97, miF1] + alphaOlambdaubar[fl97, miF1]*
           alphaRsu[fl99, miF1], alphaOsdphi[fl97_, fl99_] ->
         -(alphaRsphiboxbar*alphaOlambdad[fl97, fl99]) +
          alphaOsdphi[fl97, fl99] + alphaOlambdad[fl97, miF1]*
           alphaRsdbar[fl99, miF1] + alphaOlambdad[miF1, fl99]*
           alphaRsq[fl97, miF1], alphaOsdphibar[fl97_, fl99_] ->
         -(alphaRsphibox*alphaOlambdadbar[fl97, fl99]) +
          alphaOsdphibar[fl97, fl99] + alphaOlambdadbar[fl97, miF1]*
           alphaRsd[fl99, miF1] + alphaOlambdadbar[miF1, fl99]*
           alphaRsqbar[fl97, miF1], alphaOsephi[fl97_, fl99_] ->
         -(alphaRsphiboxbar*alphaOlambdae[fl97, fl99]) +
          alphaOsephi[fl97, fl99] + alphaOlambdae[fl97, miF1]*
           alphaRsebar[fl99, miF1] + alphaOlambdae[miF1, fl99]*
           alphaRsl[fl97, miF1], alphaOsephibar[fl97_, fl99_] ->
         -(alphaRsphibox*alphaOlambdaebar[fl97, fl99]) +
          alphaOsephibar[fl97, fl99] + alphaOlambdaebar[fl97, miF1]*
           alphaRse[fl99, miF1] + alphaOlambdaebar[miF1, fl99]*
           alphaRslbar[fl97, miF1]}
```

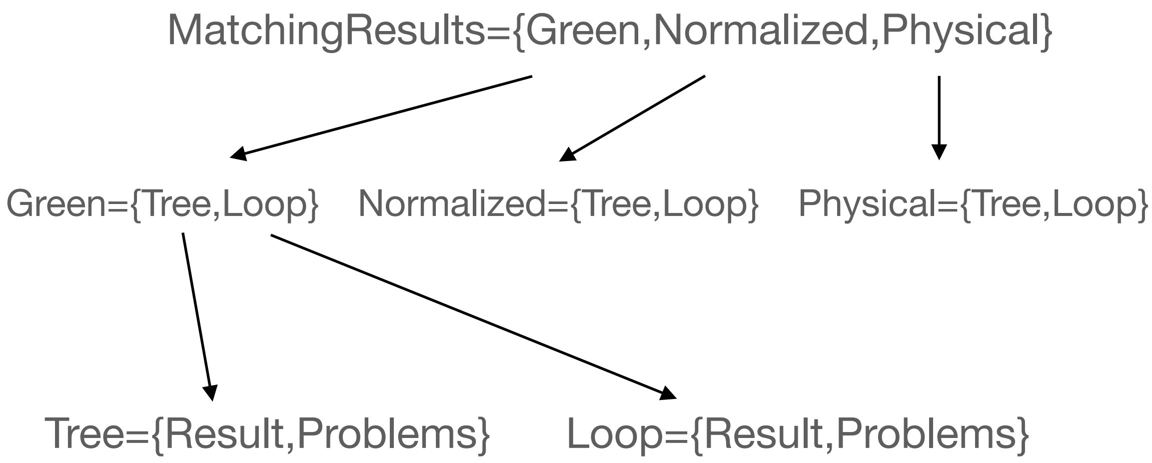
Workflow and results



MatchingResults.dat



```
  (lamprime[fl1]*lamprimebar[fl2]*yl[fl2, fl3]*ylbar[fl1, fl3])/
   (16*MHE^2*Pi^2), alphaOHD ->
 -1/32*(lamprime[fl1]*lamprime[fl2]*lamprimebar[fl1]*
     lamprimebar[fl2])/(MHE^2*Pi^2) + (lamprime[fl1]*lamprimebar[fl2]*
    yl[fl2, fl3]*ylbar[fl1, fl3])/(32*MHE^2*Pi^2) +
  (invepsilonbar*lamprime[fl1]*lamprimebar[fl2]*yl[fl2, fl3]*
    ylbar[fl1, fl3])/(16*MHE^2*Pi^2) -
  (lamprime[fl1]*lamprimebar[fl2]*Log[MHE^2/\[Mu]^2]*yl[fl2, fl3]*
    ylbar[fl1, fl3])/(16*MHE^2*Pi^2), alphaOHG -> 0, alphaOHGt -> 0,
alphaOHW -> (g2^2*lamprime[fl1]*lamprimebar[fl1])/(384*MHE^2*Pi^2),
alphaOHWB -> -1/96*(g1*g2*lamprime[fl1]*lamprimebar[fl1])/
   (MHE^2*Pi^2), alphaOlambda -> (invepsilonbar*lamprime[fl1]*
    lamprime[fl2]*lamprimebar[fl1]*lamprimebar[fl2])/(16*Pi^2) -
  (lamprime[fl1]*lamprime[fl2]*lamprimebar[fl1]*lamprimebar[fl2]*
    Log[MHE^2/\[Mu]^2])/(16*Pi^2) + (lamprime[fl1]*lamprimebar[fl2]*
    yl[fl2, fl3]*ylbar[fl1, fl3])/(8*Pi^2) +
  (invepsilonbar*lamprime[fl1]*lamprimebar[fl2]*yl[fl2, fl3]*
    ylbar[fl1, fl3])/(8*Pi^2) - (lamprime[fl1]*lamprimebar[fl2]*
    Log[MHE^2/\[Mu]^2]*yl[fl2, fl3]*ylbar[fl1, fl3])/(8*Pi^2),
alphaOmuH2 -> (MHE^2*lamprime[fl1]*lamprimebar[fl1])/(8*Pi^2) +
  (invepsilonbar*MHE^2*lamprime[fl1]*lamprimebar[fl1])/(8*Pi^2) -
  (MHE^2*lamprime[fl1]*lamprimebar[fl1]*Log[MHE^2/\[Mu]^2])/(8*Pi^2),
alphaR2B -> g1^2/(60*MHE^2*Pi^2), alphaR2G -> 0, alphaR2W -> 0,
alphaRBDH -> (13*g1*lamprime[fl1]*lamprimebar[fl1])/
   (576*MHE^2*Pi^2) + (g1*invepsilonbar*lamprime[fl1]*
    lamprimebar[fl1])/(96*MHE^2*Pi^2) -
  (g1*lamprime[fl1]*lamprimebar[fl1]*Log[MHE^2/\[Mu]^2])/
   (96*MHE^2*Pi^2), alphaRDH -> (lamprime[fl1]*lamprimebar[fl1])/
  (48*MHE^2*Pi^2), alphaRHDp ->
 -1/16*(lamprime[fl1]*lamprime[fl2]*lamprimebar[fl1]*
     lamprimebar[fl2])/(MHE^2*Pi^2) + (lamprime[fl1]*lamprimebar[fl2]*
```



MatchingResults={Green,Normalized,Physical}

Green={Tree,Loop} Normalized={Tree,Loop} Physical={Tree,Loop}

Tree={Result,Problems} Loop={Result,Problems}

Matchmaker Demo

UV theory: SM + a vector-like, EW singlet, lepton

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \bar{E}(\mathrm{i}\not{D} - M_E)E - \left[\tilde{\lambda}_i \bar{\ell}_i \phi E_R + \mathrm{h.c.}\right]$$

↓

EFT theory: SMEFT6

Limitations of Matchmaker

- γ_5 issues: require renormalizable UV model (but warning is issued)
- The matching for SMEFT is done in the unbroken phase, so the translation to the broken phase should follow (and is done by the user)
- The Green basis and the reduction to the physical basis are required from the user.
- No new heavy gauge boson are allowed in the loops

RGE mode: computing anomalous dimensions

A special kind of matching

“UV model”

A theory with only light particles

→

“eft model”

Same theory including redundant operators

Normally the hard region expansion of the “UV” would vanish

$$\int[dk]\,\frac{1}{k^2\,(k+p)^2} = \underbrace{\int[dk]\,\frac{1}{k^4} - \frac{p^2}{k^6} + \frac{(4\,p\cdot k)^2}{k^8}}_{\text{scale-less integrals}} = 0$$

…but we retain the UV poles only (no finite parts)

$$\int[dk]\,\frac{1}{k^2}\,,\quad \int[dk]\,\frac{1}{k^6}\,,\quad \int[dk]\,k^2 \quad = 0$$

but

$$\int[dk]\,\frac{1}{k^4} \approx \frac{1}{\epsilon_{UV}} - \frac{1}{\epsilon_{IR}}$$

…the “matching coefficients” are now renormalization constants and then standard techniques lead to beta’s

```
alpha6 -> a6 - (15*a4*a6*invepsilonbar)/(32*Pi^2),
```

```
\[Beta][alpha6] -> (15*a4*a6)/(16*Pi^2)
```

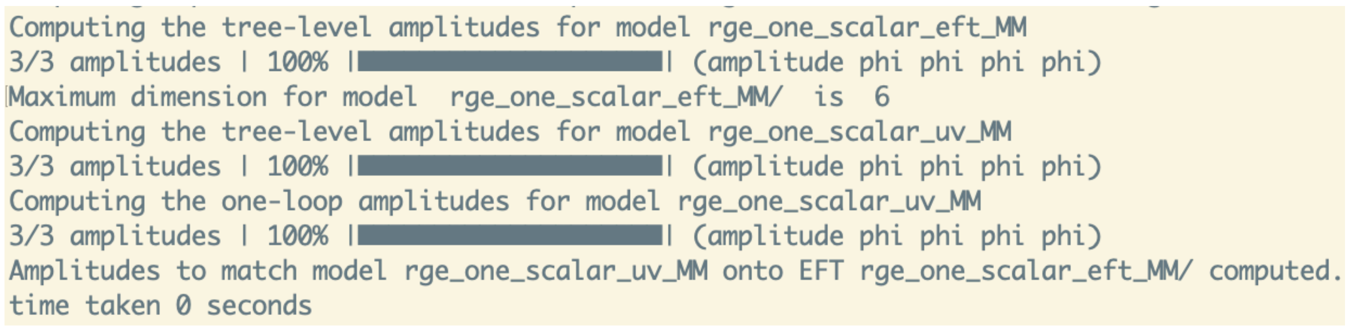
RGE mode: computing anomalous dimensions

Redundant operators

- Treated exactly the same way as before: they are removed during the reduction to the physical basis.
- This is guaranteed to always work: redundant operators renormalize among themselves, they can always be eliminated from anomalous dim matrix

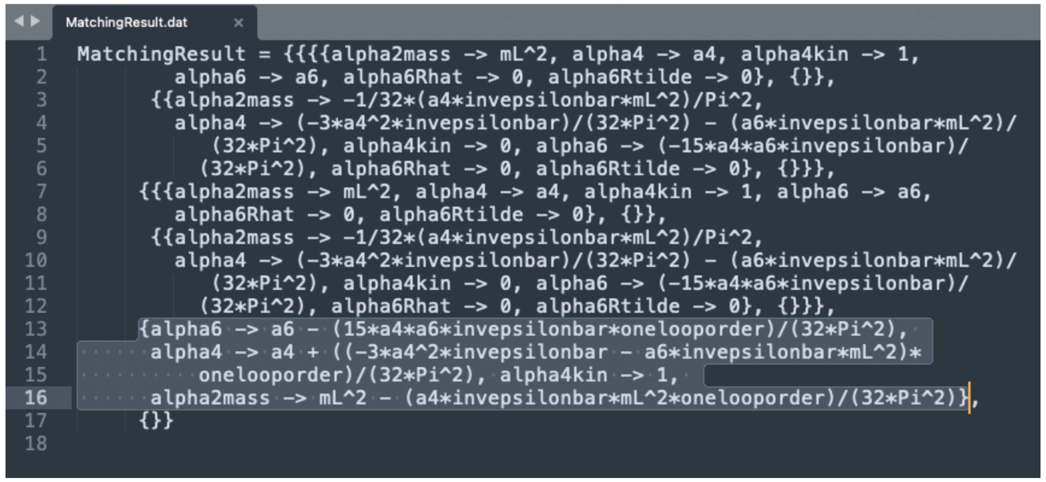
The command to invoke is

```
compute_rge_model_to_eft UV_model_MM eft_model_MM
```



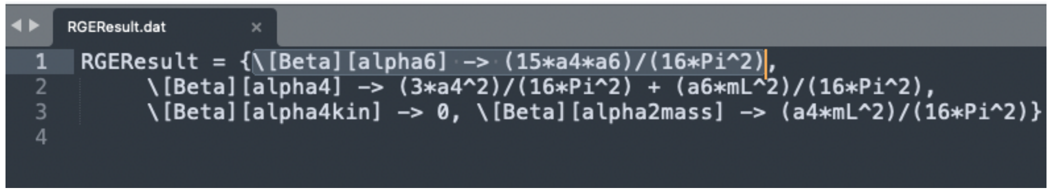
```
Computing the tree-level amplitudes for model rge_one_scalar_eft_MM
3/3 amplitudes | 100% |████████████████████| (amplitude phi phi phi phi)
Maximum dimension for model  rge_one_scalar_eft_MM/  is  6
Computing the tree-level amplitudes for model rge_one_scalar_uv_MM
3/3 amplitudes | 100% |████████████████████| (amplitude phi phi phi phi)
Computing the one-loop amplitudes for model rge_one_scalar_uv_MM
3/3 amplitudes | 100% |████████████████████| (amplitude phi phi phi phi)
Amplitudes to match model rge_one_scalar_uv_MM onto EFT rge_one_scalar_eft_MM/ computed.
time taken 0 seconds
```

Output as before: MatchingResult.dat



```
MatchingResult = {{{{alpha2mass -> mL^2, alpha4 -> a4, alpha4kin -> 1,
        alpha6 -> a6, alpha6Rhat -> 0, alpha6Rtilde -> 0}, {}},
      {{alpha2mass -> -1/32*(a4*invepsilonbar*mL^2)/Pi^2,
        alpha4 -> (-3*a4^2*invepsilonbar)/(32*Pi^2) - (a6*invepsilonbar*mL^2)/
           (32*Pi^2), alpha4kin -> 0, alpha6 -> (-15*a4*a6*invepsilonbar)/
          (32*Pi^2), alpha6Rhat -> 0, alpha6Rtilde -> 0}, {}}},
     {{{alpha2mass -> mL^2, alpha4 -> a4, alpha4kin -> 1, alpha6 -> a6,
        alpha6Rhat -> 0, alpha6Rtilde -> 0}, {}},
      {{alpha2mass -> -1/32*(a4*invepsilonbar*mL^2)/Pi^2,
        alpha4 -> (-3*a4^2*invepsilonbar)/(32*Pi^2) - (a6*invepsilonbar*mL^2)/
           (32*Pi^2), alpha4kin -> 0, alpha6 -> (-15*a4*a6*invepsilonbar)/
          (32*Pi^2), alpha6Rhat -> 0, alpha6Rtilde -> 0}, {}}},
     {alpha6 -> a6 - (15*a4*a6*invepsilonbar*oneloopoder)/(32*Pi^2),
      alpha4 -> a4 + ((-3*a4^2*invepsilonbar - a6*invepsilonbar*mL^2)*
          oneloopoder)/(32*Pi^2), alpha4kin -> 1,
      alpha2mass -> mL^2 - (a4*invepsilonbar*mL^2*oneloopoder)/(32*Pi^2)},
     {}}
```

Special output in RGE mode: RGEresult.dat



```
RGEResult = {\[Beta][alpha6] -> (15*a4*a6)/(16*Pi^2),
     \[Beta][alpha4] -> (3*a4^2)/(16*Pi^2) + (a6*mL^2)/(16*Pi^2),
     \[Beta][alpha4kin] -> 0, \[Beta][alpha2mass] -> (a4*mL^2)/(16*Pi^2)}
```

Tadpoles

If there are tadpoles in the theory (due to trilinear interactions), Matchmakereft will include their contributions to matching and anomalous dimensions, even if there is no corresponding tadpole term declared in the Lagrangian.

Redefinition of the fields to remove tadpole contributions are left (as an exercise) to the user.

See example in paper on how to correctly compute the running

Future developments

a wish list

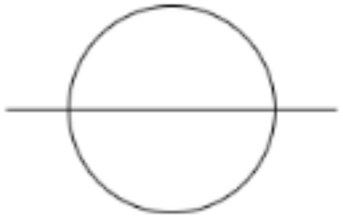
- Support for spin-2 fields
- Flavor for heavy massive particles: a nightmare of indices in propagators, indices that repeat more than twice etc.
- Extension to dim-8 matching at one loop
- Incorporating heavy spin-1 gauge bosons
- Improvements in performance: possibility to run in parallel when the number of diagrams necessary is large, e.g. phi^6 or phi^8. Also possible to speed up by recognising diagrams that are identical up to momentum mappings, e.g. s<->t or s<->u replacements, using mapping recognition routines in form.
- Support at least tHV as a general approach to gamma5 treatment
- Extension to two loops

Future developments

Two loops

- Starting from two-loop RGEs.
- Brute force the BPHZ algorithm: the nested singularities cancel if the one loop counterterms are included
- Check: dedicated FORM program that computes the anomalous dimensions by employing the R* operation, by Ben Ruijl
- (successfully reproduces the two loop QCD anomalous dimensions and more, in dev.)
- Evanescent operators and mixing overload will be foreseeable bottlenecks

At two loops we need a new master integral (sunset) after the hard expansion is performed



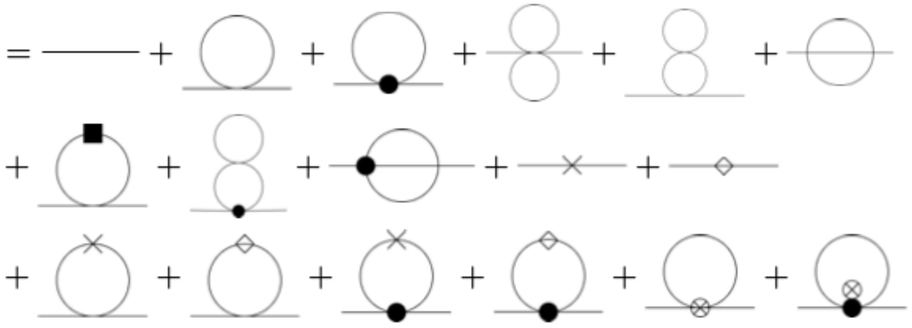
$$I_{\alpha,\beta,\gamma} = \mu^{4\epsilon} \int \frac{d^D k}{(2\pi)^D} \int \frac{d^D l}{(2\pi)^D} \left(\frac{1}{k^2 - \sigma^2}\right)^\alpha \left(\frac{1}{l^2 - \sigma^2}\right)^\beta \left(\frac{1}{(k-l)^2 - \sigma^2}\right)^\gamma$$

The tensor reduction needs a bit more attention than at one loop, but this is all known technology

No obvious bottleneck appears in the expansion and loop integral computation. After that, the real fun begins!

A toy model with scalars only, using Matchmakereft infrastructure: 2-point function

$i\Gamma_R^{(2)} =$

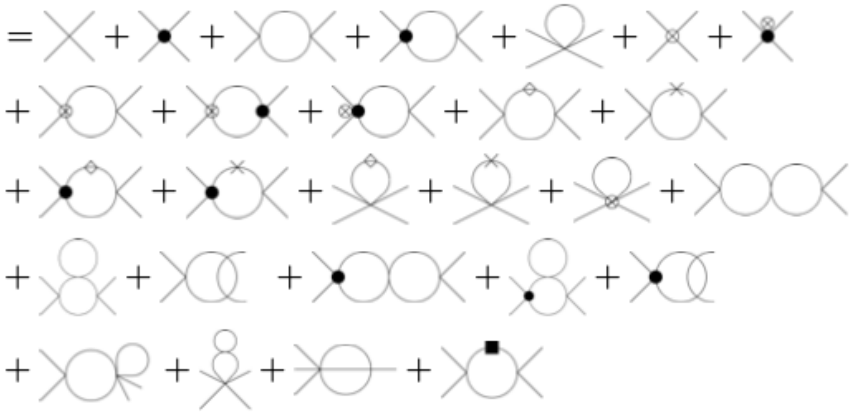


$$Z_\phi = 1 + \frac{\tilde{C}_6}{(4\pi)^2}\frac{m^2}{M^2}\frac{1}{4\epsilon} - \frac{C_4^2}{(4\pi)^4}\frac{1}{24}\frac{1}{\epsilon} + \frac{C_4\tilde{C}_6}{(4\pi)^4}\frac{m^2}{M^2}\left[\frac{1}{4\epsilon^2} - \frac{1}{16\epsilon}\right]$$

$$Z_{m^2} = 1 + \frac{C_4}{(4\pi)^2}\frac{1}{2\epsilon} - \frac{\tilde{C}_6}{(4\pi)^2}\frac{m^2}{M^2}\frac{1}{4\epsilon} + \frac{C_4^2}{(4\pi)^4}\left[\frac{1}{2\epsilon^2} - \frac{1}{4\epsilon}\right]$$
$$+ \frac{C_4\tilde{C}_6}{(4\pi)^4}\frac{m^2}{M^2}\left[-\frac{5}{4\epsilon^2} + \frac{3}{8\epsilon}\right] + \frac{C_6}{(4\pi)^4}\frac{m^2}{M^2}\left[\frac{1}{8\epsilon^2}\right] - \frac{C_4\hat{C}_6}{(4\pi)^2}\frac{m^2}{M^2}\left[\frac{3}{2\epsilon}\right]$$

… and four-point function

$i\Gamma_R^{(4)} =$



$$Z_4 = 1 + \frac{C_4}{(4\pi)^2}\frac{3}{2}\frac{1}{\epsilon} - \frac{\tilde{C}_6}{(4\pi)^2}\frac{m^2}{M^2}\frac{3}{\epsilon} + \frac{C_6}{C_4}\frac{1}{(4\pi)^2}\frac{m^2}{M^2}\frac{1}{2}\frac{1}{\epsilon}$$
$$+ \frac{C_4^2}{(4\pi)^4}\left\{\frac{9}{4}\frac{1}{\epsilon^2} - \frac{3}{2}\frac{1}{\epsilon}\right\} + \frac{C_6}{(4\pi)^4}\frac{m^2}{M^2}\left\{\frac{11}{4}\frac{1}{\epsilon^2} - \frac{1}{\epsilon}\right\}$$
$$+ \frac{C_4\tilde{C}_6}{(4\pi)^4}\frac{m^2}{M^2}\left\{\cdots\frac{1}{\epsilon^2} + \cdots\frac{1}{\epsilon}\right\} + \frac{C_4\hat{C}_6}{(4\pi)^2}\frac{m^2}{M^2}\frac{9}{\epsilon}$$

THANK YOU FOR YOUR ATTENTION AND REMEMBER:

SMEFT-Tools 2022 PAULI CENTER Zürich

We are all more than happy to work with you to extend the features and capabilities of Matchmakereft