

Institute of High Energy Physics

# *Matchete与Matchmakereft的使用*

**报告人：黎栩**

**导师：周顺 研究员**

**高能物理研究所**

**2023.12 珠江理论物理冬季学校**

# Matchete 使用

<https://gitlab.com/matchete/matchete>

[J. F.-Martín, et al, arXiv: 2212.04510]

**下载与安装：** Matchete 是一个 Mathematica 包，可以直接通过运行以下命令安装：

```
In[1]:= Import["https://gitlab.com/matchete/matchete/-/raw/master/install.m"]
```

**加载程序：**

```
In[2]:= << Matchete`
```

**加载标准模型：**（如果我们考虑的是标准模型加上新粒子和新作用项）

```
In[3]:= LSM=LoadModel["SM",ModelParameters->{"μ"->mH,"λ"->λh}];  
In[4]:= LSM//NiceForm
```

→ 显示拉氏量

或者，我们也自己构造标准模型，通过以下命令：

```
定义规范群： DefineGaugeGroup[...]  
定义味指标： DefineFlavorIndex[...]  
定义场：      DefineField[...]  
定义耦合：    DefineCoupling[...]
```

# Matchete 使用

UV模型示例：标准模型加上一个重标量粒子

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} (\partial_\mu \phi)^2 - \frac{1}{2} M_\phi^2 \phi^2 - \frac{\mu}{3!} \phi^3 - \frac{\lambda_\phi}{4!} \phi^4 - \frac{\kappa}{2} \phi^2 |H|^2$$

定义新场：

```
In[5]:= DefineField[phi, Scalar, SelfConjugate->True, Mass->{Heavy, M}]
```

定义耦合：

标量粒子

电荷共轭性质

定义重场

```
In[6]:= DefineCoupling[A, SelfConjugate -> True]
In[7]:= DefineCoupling[kappa, SelfConjugate -> True]
In[8]:= DefineCoupling[mu, SelfConjugate -> True]
In[9]:= DefineCoupling[lamphi, SelfConjugate -> True]
```

定义拉氏量：

规范指标, 味指标等等

```
In[10]:= Lint = FreeLag[phi]+(- 1/2 kappa[] Bar@H[i] H[i] phi[] phi[] - 1/3! mu[] phi[]^3
- 1/4! lambda phi[] phi[]^4) //RelabelIndices;
In[12]:= Lint // NiceForm
```

```
In[13]:= LUV = LSM + Lint;
In[14]:= LUV // NiceForm
In[15]:= CheckLagrangian@LUV
```

定义完整UV拉氏量

检查厄米性, 规范不变性等等

# Matchete 使用

Tree level matching:

微扰展开阶数

EFT展开阶数

```
In[16]:= LEFT0 = Match[LUV, LoopOrder -> 0, EFTOrder -> 6];  
In[17]:= LEFT0 - LSM // HcSimplify // NiceForm
```

收集厄米共轭项

IBP和其他恒等式化简——形成Green basis:

```
In[18]:= LEFT0OffShell = LEFT0 // GreensSimplify;  
In[19]:= LEFT0OffShell - LSM // CollectOperators // HcSimplify // NiceForm
```

EOM化简——形成物理基:

```
In[20]:= LEFT0OnShell = LEFT0 // EOMSimplify;  
In[21]:= LEFT0OnShell - LSM // CollectOperators // HcSimplify // NiceForm
```

# Matchete 使用

One-loop level matching:

```
In[22]:= LEFT = Match[LUV, LoopOrder -> 1, EFTOrder -> 6] /.  $\epsilon^{-1} \rightarrow 0$ ;  
In[23]:= LEFT - LSM // HcSimplify // NiceForm
```

IBP和EOM化简等等:

```
In[24]:= LEFTOffShell = LEFT // GreensSimplify;  
In[25]:= LEFTOffShell - LSM // CollectOperators // HcSimplify // NiceForm  
In[26]:= LEFTOnShell = LEFT // EOMSimplify;  
In[27]:= LEFTOnShell - LSM // CollectOperators // HcSimplify // NiceForm
```

$$\begin{aligned} & -\frac{1}{4} B^{\mu\nu 2} - \frac{1}{4} G^{\mu\nu A 2} - \frac{1}{4} W^{\mu\nu I 2} + D_\mu \bar{H}_i D_\mu H^i + \left( mH^2 + \frac{1}{2} \hbar \kappa M^2 \left( 1 + \text{Log} \left[ \frac{\bar{\mu}^2}{M^2} \right] \right) \right) \bar{H}_i H^i + i (\bar{d}_a^r \cdot \gamma_\mu P_R \cdot D_\mu d^{ap}) \delta^{pr} + \\ & i (\bar{e}^r \cdot \gamma_\mu P_R \cdot D_\mu e^p) \delta^{pr} + i (\bar{l}_i^r \cdot \gamma_\mu P_L \cdot D_\mu l^{ip}) \delta^{pr} + i (\bar{q}_{ai}^r \cdot \gamma_\mu P_L \cdot D_\mu q^{aip}) \delta^{pr} + i (\bar{u}_a^r \cdot \gamma_\mu P_R \cdot D_\mu u^{ap}) \delta^{pr} + \\ & \left( -\frac{1}{2} \lambda h - \frac{1}{12} \hbar \frac{1}{M^2} \kappa^2 \left( mH^2 - 3 M^2 \text{Log} \left[ \frac{\bar{\mu}^2}{M^2} \right] \right) \right) \bar{H}_i \bar{H}_j H^i H^j - \bar{Y} d^{pr} \bar{H}_i (\bar{d}_a^r \cdot P_L \cdot q^{aip}) - \bar{Y} e^{pr} \bar{H}_i (\bar{e}^r \cdot P_L \cdot l^{ip}) - \\ & Y e^{rp} H^i (\bar{l}_i^r \cdot P_R \cdot e^p) - Y d^{rp} H^i (\bar{q}_{ai}^r \cdot P_R \cdot d^{ap}) - Y u^{rp} \bar{H}_i (\bar{q}_{aj}^r \cdot P_R \cdot u^{ap}) \epsilon^{ji} - \bar{Y} u^{pr} H^j (\bar{u}_a^r \cdot P_L \cdot q^{aip}) \bar{\epsilon}_{ij} + \\ & \frac{1}{12} \hbar \frac{1}{M^2} \kappa^2 (-\kappa + \lambda h) \bar{H}_i \bar{H}_j \bar{H}_k H^i H^j H^k - \frac{1}{12} \hbar \frac{1}{M^2} \kappa^2 D_\mu \bar{H}_i \bar{H}_j D_\mu H^i H^j + \frac{1}{24} \hbar \bar{Y} d^{pr} \frac{1}{M^2} \kappa^2 \bar{H}_i \bar{H}_j H^j (\bar{d}_a^r \cdot P_L \cdot q^{aip}) + \\ & \frac{1}{24} \hbar \bar{Y} e^{pr} \frac{1}{M^2} \kappa^2 \bar{H}_i \bar{H}_j H^j (\bar{e}^r \cdot P_L \cdot l^{ip}) + \frac{1}{24} \hbar Y e^{rp} \frac{1}{M^2} \kappa^2 \bar{H}_i H^i H^j (\bar{l}_j^r \cdot P_R \cdot e^p) + \frac{1}{24} \hbar Y d^{rp} \frac{1}{M^2} \kappa^2 \bar{H}_i H^i H^j (\bar{q}_{aj}^r \cdot P_R \cdot d^{ap}) + \\ & \frac{1}{24} \hbar Y u^{rp} \frac{1}{M^2} \kappa^2 \bar{H}_i \bar{H}_j H^j (\bar{q}_{ak}^r \cdot P_R \cdot u^{ap}) \epsilon^{ki} + \frac{1}{24} \hbar \bar{Y} u^{pr} \frac{1}{M^2} \kappa^2 \bar{H}_i H^i H^k (\bar{u}_a^r \cdot P_L \cdot q^{ajp}) \bar{\epsilon}_{jk} \end{aligned}$$

# Matchete 使用

但是上述化简命令没办法自动转化到Warsaw, 有些需要手动处理替换规则, 比如

$$(\bar{H}H)(D_\nu \bar{H} D^\nu H) \rightarrow \frac{1}{2}(\bar{H}H)\partial^2(\bar{H}H) - \frac{1}{2}(D^2 \bar{H}H + \bar{H}D^2 H)(\bar{H}H)$$

```
In[28]:= GetHBox = Evaluate[Bar@H[i_] H[i_] CD[v, Bar@H[j_]] CD[v, H[j_]] /. v->v_ :> Evaluate[1/2 (Op["HBox"] - Bar@H[i] H[i] (CD[{v, v}, Bar@H[j]] H[j] + Bar@H[j] CD[{v, v}, H[j]]))]
```

但是显然又引进了新的冗余算符, 需要再次使用 `GreensSimplify` `EOMSimplify`

在所有经过所有简化之后, 我们能直接通过以下命令读取Warsaw基的算符和系数,

例如,  $(H^\dagger H)^3$  顶点的匹配系数提取为

包含的场

包含的偏导数目

```
In[29]:= SelectOperatorClass[LEFTOnShell, {Bar@H,H,Bar@H,H,Bar@H,H}, 0] // NiceForm
```

Out[29]:=  $C_{H^6} = -\frac{\kappa^3}{192\pi^2 M_\phi^2}$  如果没做简化, 你将得到  $C_{H^6} = -\frac{\kappa^2(\kappa - \lambda_h)}{192\pi^2 M_\phi^2}$

还有更多的算符是需要手动写替换代码来手动变换的, 比如一些包含费米子的算符等等, 并不是完全自动化! [See examples for more details](#)

# Matchmakereft的使用

[A. Carmona, et al, arXiv: 2112.10787]

<https://pypi.org/project/matchmakereft/>

下载与安装: Matchmakereft是一个需要在终端运行的程序

所需条件:

FORM <http://www.nikhef.nl/~form/>  
QGRAF <http://cfif.ist.utl.pt/~paulo/qgraf>  
Python  
FeynRules <https://feynrules.irmp.ucl.ac.be/>  
pip(或者conda)



这两个是画图和计算振幅的程序, 需保证能随时在终端运行

安装:

```
> pip install matchmakereft
```

或者

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

运行:

```
> matchmakereft
```

如果这一步不能运行, 可能需要指定路径

产生模型:

```
> create_model modfile1.fr ... modfilen.fr
```

将1到n的fr文件联合起来产生模型, 叫做 ...\_MM (保证fr文件路径能被识别)

匹配:

```
> match_model_to_eft ..._MM SMEFT_Green_Bpreserving_MM
```

结果将产生一个.dat文件

你产生的模型  
match程序使用-黎栩

系统自带的EFT模型

# Matchmakereft的使用

## fr文件的书写

UV模型:

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} (\partial_\mu \phi)^2 - \frac{1}{2} M_\phi^2 \phi^2 - \frac{\mu}{3!} \phi^3 - \frac{\lambda_\phi}{4!} \phi^4 - \frac{\kappa}{2} \phi^2 |H|^2$$

定义场:

```
M$ClassesDescription = {  
  S[105] == {  
    ClassName      -> sphl,  
    SelfConjugate  -> True,  
    Mass           -> Msphi,  
    FullName       -> "heavy",  
    Width          -> 0  
  }  
};
```

定义耦合:

```
M$Parameters = {  
  kappa == {  
    ParameterType  -> Internal,  
    ComplexParameter -> False  
  }  
};
```

指定是否是重场

拉氏量定义:

```
LNP := Block[{mu, ii, jj},  
  del[sphi, mu]*del[sphi, mu]/2 - Msphi^2 sphi^2/2 - kappa/2  
  Phibar[ii] Phi[ii] sphi^2 - mu/6 sphi^3 - lambdaphi/24  
  sphi^4]  
  
Ltot := LSM + LNP;
```

注意, 最后模型必须定义为Ltot



# Matchmakereft的使用

## 顶点读取(使用 *mathematica*)

```
result = Get["./MatchingResult.dat"]
```

单圈匹配产生的.dat文件

```
res1 = Select[result[[3]], If[#[[2]]==0&&NumberQ[#[[2]]], 1, 0]==0&]
```

例如：读取Higgs二次项的阈值修正，我们可以用

```
CoefficientList[Select[res1, #[[1]] == alphaOmuH2 &][[1, 2]], onelooporder]
```

$$\delta\mathcal{L} \supset -\delta\mu_H^2 H^\dagger H \quad \rightarrow \quad \delta\mu_H^2 = \frac{\kappa M_\phi^2 \left( \log \frac{M_\phi^2}{\mu^2} - 1 \right)}{32\pi^2}$$

耦合的名称,

见：[SMEFT\\_Green\\_Bpreserving.fr](#)

按阶数排序

例如：读取 $(H^\dagger H)^3$ 顶点的匹配系数，我们可以用

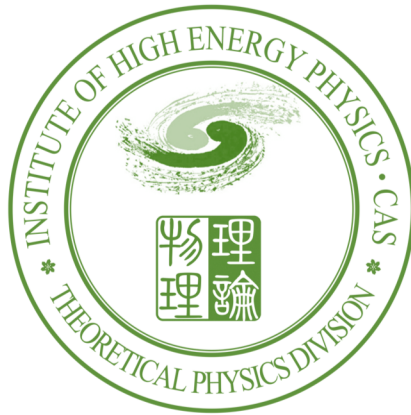
```
CoefficientList[Select[res1, #[[1]] == alphaOH &][[1, 2]], onelooporder]
```

$$\rightarrow C_{H^6} = -\frac{\kappa^3}{192\pi^2 M_\phi^2}$$



由于耦合是直接定义到Warsaw基，所以程序能从振幅出发直接得到Warsaw基的算符系数

1. Matchete和Matchmakereft是两个处理单圈匹配的自动化程序，分别基于泛函方法和费曼图方法
2. Matchete运行时间要长于Matchmakereft，而且不能直接得到Warsaw基的算符系数，有些算符需要手动处理
3. Matchmakereft需要的运行条件要比Matchete更复杂，相反地Matchete是对mathematica用户更友好
4. 两个程序得到的结果应该是一致的，都基于区域展开方法



谢谢! O\_5