

FeynRules tutorial

阿部智広 (KEK)

のじりコライダーズスクール

2015.12.9

前回までの流れ

僕の好きな模型が入っていないんだけど…

pythia

- モンテカルロをしてイベントをつくる
- Detectorには通していないイベントが作られる
- hepmcなる形式のファイルに結果を書き出せる

hepmcファイルを渡す

Delphes

- Detector simulation をする
- hepmcなる形式のイベントファイルを読み込んで simulation できる
- 結果を rootなる形式に書き出せる

rootファイルを渡す

root

- グラフを描く
- C++的な命令文を読み込ませると色々プロットできる

絵ができる

物理をやる！！

どうするの？

Q1. Pythia に自分のやりたい模型が入っていない。どうしよう。

Q2. Calchep や **Madgraph** にも入っていない。どうしよう。

A1. 著者に連絡して入れて自分のやりたい模型を入れてもらう

A2. 自分で模型ファイルをつくって Calchep や Madgraph を動かす。

模型ファイル = Feynman rules、質量、スピン、parameters とかの情報を Calchep とかが理解できるフォーマットで書かれたもの。

- 模型ファイルを自分で書くのは大変。
(わけの分からないファイルを1000行以上書く必要ある)
- **FeynRules** は、模型ファイルをつくってくれる
- (Calchep だけなら **LanHEP** で作れるらしい)

FeynRules

著者

Adam Alloul, Neil D. Christensen, Celine Degrande, Claude Duhr, Benjamin Fuks

ダウンロード先 (マニュアルやチュートリアルもある)

<http://feynrules.irmp.ucl.ac.be>

必要なもの

- テキストエディタ (メモ帳、emacs、…)
- Mathematica
- **エラーにくじけない心 (根気と時間)**
- **マニュアルを読もうとする気持ち**

C とか C++ がわからなくてもFeynRulesは使える！

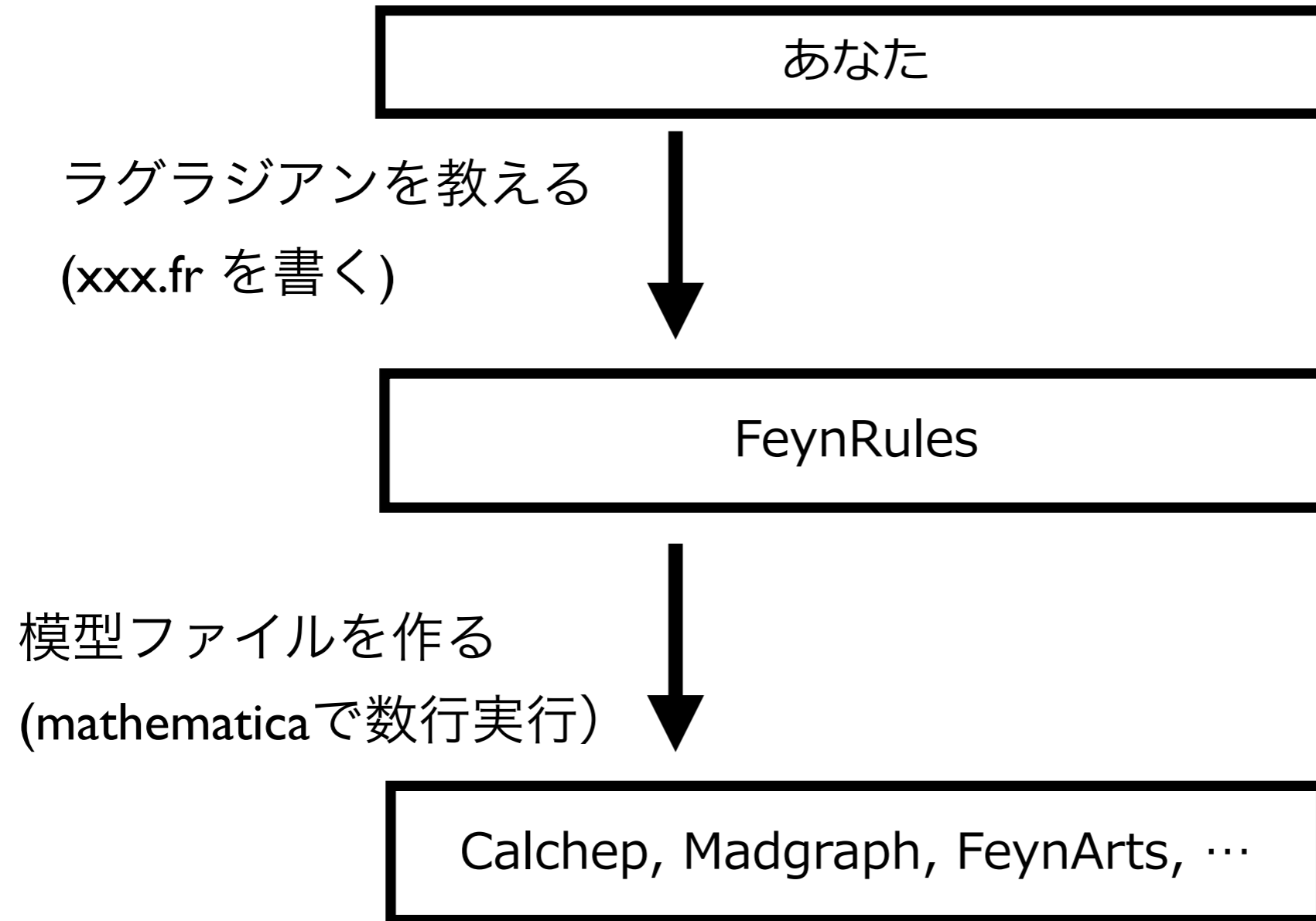
私のプログラミング言語の経験

- Fortranは cteq のためにやったが覚えてない。
- Cは micrOMEGAs のためにやったが覚えてない。
- C と C++ は学部で勉強した記憶があるが、よくわかってない。
 - ★ printf, cout, for, if はわかる
 - ★ ポインタに慣れてない (「->」 , 「*」 , 「&」 , 「xxx.yy()」 ???????)
 - ★ 構造体？クラス？継承？

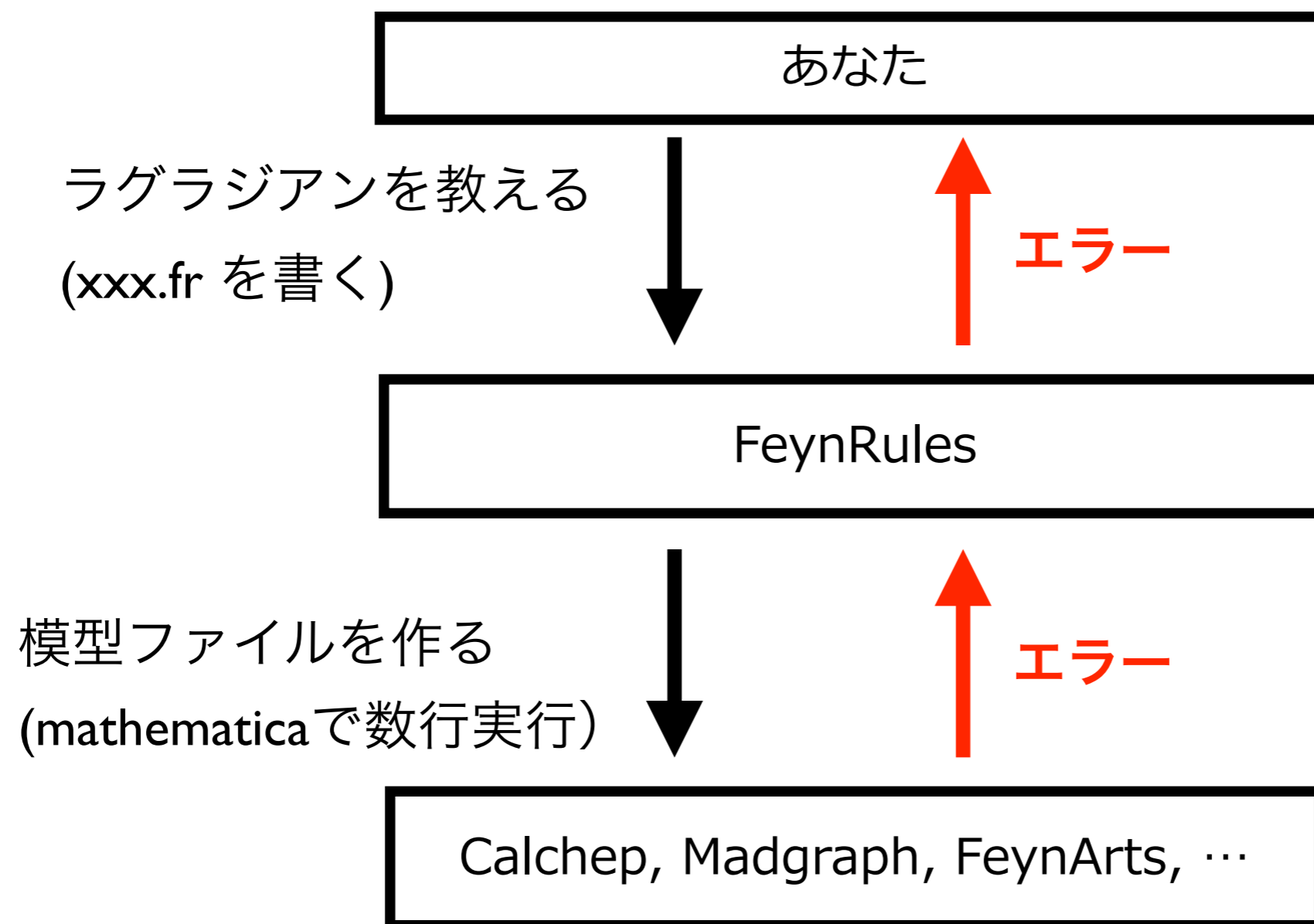
FeynRulesの経験

- モデルは6つ作った (5つは論文で使った [[TA Kitano Sato arXiv:1411.1335](#)])
 - ★ SM + scalar DM (SU(2)の singlet, doublet, triplet)
 - ★ SM + fermion DM (SU(2) の singlet + doublet, doublet + triplet)
 - ★ W' 模型 [[TA and Kitano '13](#), [TA Kitahara Nojiri '15](#)]

流れ



ありがちな流れ



エラー10回くらいでへこたれない

人間が Feynman rule を導出するとき

1. ゲージ対称性を指定
2. 場の表現を指定 (gauge eigenstates を指定)
(例: H は SU(2) の基本表現で hypercharge 1/2 など)
3. ラグランジアンを書き下す (パラメタも指定)
4. 質量を対角化する (mass eigenstates と mixing angle の指定)
(例)
$$\begin{pmatrix} Z \\ \gamma \end{pmatrix} = \begin{pmatrix} \cos \theta_W & -\sin \theta_W \\ \sin \theta_W & \cos \theta_W \end{pmatrix} \begin{pmatrix} W^3 \\ B \end{pmatrix}$$
5. ラグランジアンの場を全部 mass eigenstates で書き直す。
6. vertex から Feynman rules を読み取る。

FeynRules を使う場合

1. ゲージ対称性を指定

2. 場の表現を指定 (**gauge eigenstates** を指定)

(例: H は SU(2) の基本表現で hypercharge 1/2 など)

3. ラグランジアンを書き下す (パラメタも指定)

4. 質量を対角化する (**mass eigenstates** と **mixing angle** の指定)

(例)

$$\begin{pmatrix} Z \\ \gamma \end{pmatrix} = \begin{pmatrix} \cos \theta_W & -\sin \theta_W \\ \sin \theta_W & \cos \theta_W \end{pmatrix} \begin{pmatrix} W^3 \\ B \end{pmatrix}$$

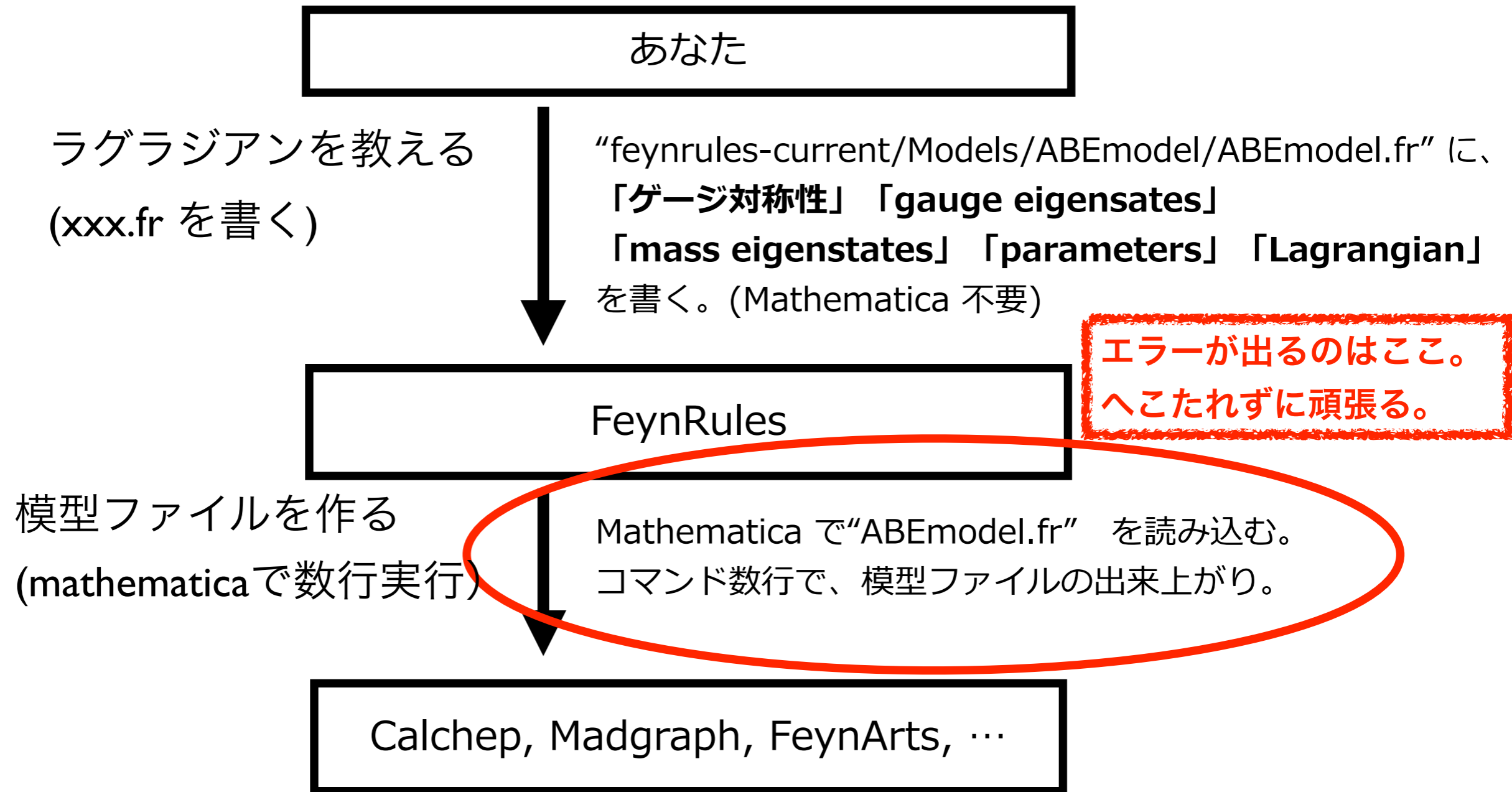
ここまでは人間がやる

5. ラグランジアンを全部 mass eigenstates で書き直す。

6. vertex から Feynman rules を読み取る。

FeynRules がやってくれる

流れ



このトークでやること、やらないこと

- やること

- ★ FeynRules の作業の流れを手を動かして体験する
- ★ 簡単な模型 (SM + scalars) なら作れるようになる

- 時間の都合でやれないこと

- ★ フェルミオン, ゲージ場, ゴースト場の書き方
- ★ 複雑な模型 (SM + fermions, W' 模型, など)

- やらないこと

- ★ SUSY models (誰かが既に作っているから)

イントロおしまい

ここまでで質問ありますか？

もくじ

- **イントロ (終わりました)**
 - ★ FeynRules て何？
 - ★ 使うのに必要な概略
- **SM + scalar singlet DM を作ってみよう**
 - ★ 一連の流れと、文法 (の一部) に慣れることが目標。
 - ★ 手取り足取りやります。
- **SM のファイルを見してみる**
 - ★ 上の例で扱えなかった文法を眺める。
 - ★ 完全な理解は目指さない。無理。
- **SM + VEVをもつ scalar singlet を作ってみよう**
 - ★ mixing angle を導入できるようになることが目標。
- **(自習) 好きな模型でやってみよう。**
 - ★ 実際の研究に使えるものを作るのが目標。
 - ★ やりたいことによっては、ここで難易度が急上昇します。

SM + scalar singlet DM をつくってみよう

模型の定義

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} \partial^\mu S \partial_\mu S - \frac{\mu_S^2}{2} S^2 - \frac{\lambda_S}{24} S^4 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H$$

- S の性質
 - ★ scalar gauge singlet
 - ★ 真空期待値 もたない
 - ★ 暗黒物質(DM)
- H は SM Higgs boson
- 簡単なので練習に最適！
- DM模型としてもポピュラーなので実用的！

SM + scalar singlet DM をつくってみよう

模型の定義

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} \partial^\mu S \partial_\mu S - \frac{\mu_S^2}{2} S^2 - \frac{\lambda_S}{24} S^4 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H$$

何やるんだったっけ？

1. **ゲージ対称性を指定**
2. **場の表現を指定 (gauge eigenstates を指定)**
3. **ラグランジアンを書き下す (パラメタも指定)**
4. **質量を対角化する (mass eigenstates と mixing angle の指定)**

SM + scalar singlet DM をつくってみよう

模型の定義

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} \partial^\mu S \partial_\mu S - \frac{\mu_S^2}{2} S^2 - \frac{\lambda_S}{24} S^4 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H$$

何やるんだったっけ？

1. ゲージ対称性を指定

→ SMと同じ

ここだけやればいい

(SMの部分は、SM.fr を利用する)

2. 場の表現を指定 (gauge eigenstates を指定)

→ S は singlet scalar

3. ラグランジアンを書き下す (パラメタも指定)

→ 上に書いてあるやつ。新しいパラメタは3つ

4. 質量を対角化する (mass eigenstates と mixing angle の指定)

→ S は VEV を持たないので、S がそのまま mass eigenstate

→ S の質量は $m_{\text{DM}}^2 = \mu_S^2 + (\lambda_{hS})^2 v^2 / 2$

SM + scalar singlet DM をつくってみよう

模型の定義

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} \partial^\mu S \partial_\mu S - \frac{\mu_S^2}{2} S^2 - \frac{\lambda_S}{24} S^4 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H$$

この部分は最初から模型ファイルがある
(Models/SM/SM.fr)

ここだけ自分で書けばよい！

SM + scalar singlet DM をつくってみよう

やることは簡単だけど、ファイルを書くのは面倒。

似たような fr ファイルを持ってきて変更するのが楽。バグも減る。

(1) ファイルを入手 (SMScalar.fr)

- Feynrules のページに行く
- Go to model database をクリック
- Simple extensions of the SM をクリック
- 下から9個目くらいの“Standard model + Scalars” をクリック
- SMScalars.fr をクリック
- 下に“異なるフォーマットでダウンロード”とあるのでクリック

(2) 作業ディレクトリ作成。ファイルの移動。ファイル名を“singletDM.fr”に。

```
cd feynrules-current/Models/  
mkdir singletDM  
cd singlet DM  
mv ~/Download/SMScalar.fr ./singletDM.fr
```

singletDM.fr を開くとこんな感じ

```
(***** This is the FeynRules mod-file for the extension of the SM where additional scalars are
(*)
(*)   This model file needs the Standard Model model file SM.fr.
(*)
(*)   This model can be used in both the unitary and the Feynman gauge.
(*)
(*)   The number of scalars is by default set to 4. It can be changed at wish just by changing the va
(*****

Nf = 4;           ← 消す

M$ModelName = "SM_Plus_Scalars";   ← "singletDM" に変える

M$Information = {Authors -> "C. Duhr",           (← 名前や所属も変える)
                  Institutions -> "Universite catholique de Louvain (CP3).",
                  Emails -> "claude.duhr@uclouvain.be",
                  Date -> "14. 06. 2009",
                  Version -> "1.0",
                  URLs -> "http://feynrules.phys.ucl.ac.be/view/Main/StandardModelScalars",
                  References -> "\"The minimal non-minimal Standard Model\"", J.J. van der Bij, Phys.Lett.B

(***** Index definitions *****)
```

ちょっと下に行くと…

いらないので消す

```
(***** Index definitions *****)  
  
(* This is the declaration for the generation index of the new scalars *)  
  
IndexRange[ Index[SGen] ] = Range[Nf]  
  
IndexStyle[SGen, k]  
  
(***** Parameters *****)
```

さらにちょっと下に行くと… パラメタが定義してある

```
(***** Parameters *****)
```

```
M$Parameters = { { と } の対応があります
```

```
(* The parameters of the new sector *)
```

```
\[Lambda]S == {  
  InteractionOrder -> {QED, 2},  
  ParameterName -> lS,  
  Value -> 0.5,  
  BlockName -> INVSCALAR},
```

λ_S (S^4 の係数)

- ¥[Lambda]S : ラグランジアンを書く時に使う。
- 残りは、MadGraphとかで使う。
- } のあと, で区切る。次のパラメタがあるので。

```
\[Omega] == {  
  InteractionOrder -> {QED, 2},  
  ParameterName -> om,  
  Value -> 0.5,  
  BlockName -> INVSCALAR}
```

λ_{hS} ($S^2|H|^2$ の係数)

2点変更 notation

- ¥[Omega] を ¥[Lambda]hS
- om を lhS

```
}
```

mass parameter (μ_S) が定義されて無いじゃん！定義してみよう

```
(***** Parameters *****)
(* The parameters of the new sector *)
M$Parameters = {
```

```
\[Lambda]S == {
  InteractionOrder -> {QED, 2},
  ParameterName -> \lS,
  BlockName -> INVSCALAR},
```

} を }, に変更 (コンマつけました)

```
\[Lambda]hS == {
  InteractionOrder -> {QED, 2},
  ParameterName -> \lhS,
  Value -> 0.5,
  BlockName -> INVSCALAR},
```

1行目： μ_S は他のパラメタで決まると宣言

2行目： $\mu_S = \text{Sqrt}[m_{DM}^2 - (\lambda_{hS})^2 v^2/2]$

3行目：FeynArtsの表示に使う

4行目：説明。未来の自分がわかれば何でも良い。

```
muS == {
  ParameterType -> Internal,
  Value -> Sqrt[2 MDM^2 - vev^2 \[Lambda]hS]/Sqrt[2],
  TeX -> \[Mu]S,
  Description -> "Coefficient of the quadratic piece of the singlet potential"
}
```

MDM (DMの質量) はあとで
粒子を定義するところで一緒に定義する

Higgs VEV
Models/SM/SM.fr で
定義されている

下に行くと、粒子を定義しているよ。

```
(***** Particle Classes *****)
```

```
M$ClassesDescription = { { と } の対応があるよ
```

```
(* The new scalar sector *)
```

```
S[4] == { SDM に変更
```

```
  ClassName -> Sk,
```

```
  ClassMembers :=> Table[Symbol["S" <> ToString[k]], {k, Nf}], 消す
```

```
  SelfConjugate -> True,
```

```
  Indices -> {Index[SGen]}, 消す
```

```
  FlavorIndex -> SGen,
```

```
  Mass -> {MSk, 40},
```

```
  Width -> 0} MDM に変更 400 に変更
```

```
}
```

```
(*****)
```

言われた通りに変更しました

```
M$ClassesDescription = {  
  (* The new scalar sector *)  
  S[4] == {  
    ClassName -> SDM,  
    SelfConjugate -> True,  
    Mass -> {MDM, 400},  
    Width -> 0}
```

}

1~3行目 : FeynArts に必要

4~6行目 : CalcHePやMadGraph

(重要) DMなので、"~" で始まる名前にする

さらに変更してみよう

```
M$ClassesDescription = {  
  (* The new scalar sector *)  
  S[4] == {  
    ClassName -> SDM,  
    SelfConjugate -> True,  
    Mass -> {MDM, 400},  
    Width -> 0,  
    PropagatorLabel -> "~S",  
    PropagatorType -> D,  
    PropagatorArrow -> None,  
    PDG -> 990025,  
    ParticleName -> "~S",  
    FullName -> "~S"
```

}

最後にラグランジアンを定義しよう。下の方に行くと

(* The lagrangian for the new sector *)

```
LScalar := 1/2 del[Sk[k], mu]^2 - 1/2 MSk^2 Sk[k]^2 - \[Lambda]S/(8 Nf) Sk[k1]^2  
Sk[k2]^2 - \[Omega]/(4 Sqrt[Nf]) Sk[k]^2 HC[Phi].Phi;
```

これを変更していきます。

簡単ですが、説明の便宜上、全部で4step踏みます)

step1: ただ書いてみる

`LScalar := 1/2 del[SDM, mu]^2 - 1/2 muS^2 SDM^2 - \[Lambda]S/(24) SDM^4 - \[Lambda]hS/(2) SDM^2 Phibar[ii] Phi[ii];`

- 今付け足したいラグランジアンを “LScalar” と呼びます。
- “:=” (コロンとイコール) で式をつなぎます。 (Mathematica文法)
- 右辺の意味
 - ★ 1 項目 : kinetic term
 - ★ 2 項目 : mass term
 - ★ 3 項目 : S^4
 - ★ 4 項目 : $S^2 |H|^2$
- 4 項目の説明
 - ★ “ii” はSU(2)の足。ダミーの添字。
 - ★ 場の名前に “bar” とつけるとダガーの意味 (もしくはディラック共役の意味)
- 関数の最後はセミコロン (;) でフィニッシュを決めるのが安全 (Mathematica文法)

step2: ダミーの添字を教える

```
LScalar := Block[{ii, mu, feynmangaugerules},  
1/2 del[SDM, mu]^2 - 1/2 mu S^2 SDM^2 - \[Lambda] S/(24) SDM^4 - \[Lambda] h S/(2) SDM^2 Phibar[ii] Phi[ii]  
];
```

- **Block[{ii, mu, feynmangaugerules}, ...];**
 - ★ この文法は Mathematica のもの。
 - ★ 関数内で使う local なパラメータを定義するときのやりかた。
 - ★ feynmangaugerules は後で使う
- “ii” と “mu” はダミーの添字で、他の部分で使わないパラメータなので、こうしておく。
- **Phi[ii]** の右にあったセミコロンは最後に移動したので注意。

step3: SU(2)の場を展開

```
LScalar := Block[{ii,mu, feynmangaugerules},  
ExpandIndices[1/2 del[SDM, mu]^2 - 1/2 muS^2 SDM^2 - \[Lambda]S/(24) SDM^4 - \[Lambda]hS/(2) SDM^2 Phibar[ii] Phi[ii]  
, FlavorExpand->{SU2D,SU2W}]  
];
```

- **ExpandIndices[...**
, FlavorExpand->{SU2D, SU2W}]
 - ★ FeynRules の文法。
 - ★ SU(2)の場をcomponent field で書くように命令する。
($|H|^2$ を $h^2 + NG_{\text{bosons}}^2$ に直す。)
- SU2Dは SU(2)の基本表現(H とか)
- SU2Wは SU(2)のadjoint表現 (ゲージ場とか)
- SU2W は今書かなくてよいが、SU2Dと一緒に書く癖を付けた方が安全。
共変微分使う場合は必須。

step4: Feynman gauge と Unitary gauge を選べるようにする

```
LScalar := Block[{ii,mu, feynmangaugerules},  
feynmangaugerules = If[Not[FeynmanGauge], {G0|GPIGPbar ->0}, {}];  
ExpandIndices[1/2 del[SDM, mu]^2 - 1/2 muS^2 SDM^2 - \[Lambda]S/(24) SDM^4 - \[Lambda]hS/(2) SDM^2 Phibar[ii] Phi[ii]  
, FlavorExpand->{SU2D,SU2W}]/.feynmangaugerules  
];
```

- 模型ファイルを作るときに、ゲージを選べるようにする。
 - ★ Calcchep用模型ファイルは feynman gague で書く（のがよい）。
 - ★ Madgraph用模型ファイルはどっちでもよいらしい。
- **feynmangaugerules** の中身を定義（おまじない。書き写す。）
- `/.feynmangaugerules` で、定義をラグランジアンに適用する。

fr ファイルの編集は以上

やったこと

- パラメタ($\lambda_S, \lambda_{HS}, \mu_S$) の定義
- 場 (S) の定義
- ラグランジアンを書いた

これは簡単な模型なので簡単だった。一般には

- **mass eigenstate \neq gauge eigenstate**
 - ★ 場の定義が 2 倍になる
 - ★ パラメタに mixing angle を定義しないといけない。
- **gauge non-singlet (SU(2) doublet にするとか)**
 - ★ doublet の場と component 場の両方定義しないといけない
 - ★ 足のつぶり方の指定 (ラグランジアンを書くのが手間)
- **ゲージ群の拡張**
 - ★ SM.fr の再利用が出来ないので、全部書く必要がある。

モデルファイルを作ろう

次に、このファイルをもとに、Calchepで読めるファイルを作ります。
Mathematica を起動してください。

目的：書いたファイルにバグが無いか確認する

Mathematica に FeynRules を認識させる

- Feynrules-current を置いているディレクトリを教えます。
- 下の"/Users/abetomo/..." を各自書き換えてください。
- 書き換え方がわからない場合は、ターミナルでfeynrules-current のディレクトリで pwd と打ち、出てきたやつを使いましょう。

`Quit[]`

```
$FeynRulesPath = SetDirectory["/Users/abetomo/work/programs/feynrules-current/"];  
<< FeynRules`  
SetDirectory["/Users/abetomo/work/programs/feynrules-current/Models/singletDM"];
```

- FeynRules` の最後の点 (`) は、
 - ★ Shift+@ (Macの日本語キーボード)
 - ★ 左上の esc の1つ下 (MacのUSキーボード)
 - ★ どこかにあります (その他のキーボード)
- (Mathematica file を "singletDM/" に保存している場合は、最後の行は次でも良いです。)

```
SetDirectory[NotebookDirectory[]];
```


実行すると

```
Quit[]

$FeynRulesPath = SetDirectory["/Users/abetomo/work/programs/feynrules-current/"];
<< FeynRules`
SetDirectory[NotebookDirectory[]];

- FeynRules -

Version: 2.0.3 (23 October 2013).
Authors: A. Alloul, N. Christensen, C. Degrande, C. Duhr, B. Fuks

Please cite:
- arXiv:1310.1921;
- Comput.Phys.Commun.180:1614-1641,2009 (arXiv:0806.4194).

http://feynrules.phys.ucl.ac.be

The FeynRules palette can be opened using the command FRPalette[].

Get::noopen: Cannot open FAToFR.m.
Get::noopen: Cannot open FAToFR.m.
```

これは無視してよいエラー

次にこれを実行。（ラグランジアンを読み込む）

```
LoadModel["../SM/SM.fr", "singletDM.fr"]
```

エラーが無ければこんなだけ出ます。

```
Merging model-files...
```

```
This model implementation was created by
```

```
T. Abe
```

```
Model Version: 1.0
```

```
For more information, type ModelInformation[].
```

- Loading particle classes.
- Loading gauge group classes.
- Loading parameter classes.

```
Model singletDM loaded.
```

次にこれを実行。(UFOファイルの作成)

```
FeynmanGauge = True;  
WriteUFO [LSM + LScalar  
  , FlavorExpand -> {True, SU2W}  
  , Output -> "UFO"]
```

- **FeynmanGauge=True;**

- ★ feynman gauge で模型ファイルをつくります
- ★ **True**の代わりに **False** を使うと unitary gauge になります

- **WriteUFO[...**

- , FlavorExpand->{True, SU2W}**
- , Output->"UFO"]**

- ★ 定義したラグランジアンを書き込む (LScalar以外は SM.fr に定義されている)
- ★ SU(2)の足を持つものはcomponent で展開
- ★ 出来たファイルは"UFO"というディレクトリに書き込む(適当に変えてよい)

できましたか？

エラーが起こったら、`singletDM.fr` ファイルを見直しましょう。

- 必要なコンマ (,) はありますか？
- 不必要なコンマ (,) はありませんか？
- 変なところにセミコロン (;) を書いていませんか？
- 定義していない場を使ってませんか？
- { と } は対応してますか？
- その小文字は大文字じゃないですか？
- など

`singletDM.fr` ファイルを直したら、`mathematica` は最初からやり直し。
面倒ですが、`Quit[]` から打ち直してください。

```
Quit[]
```

```
$FeynRulesPath = SetDirectory["/Users/abetomo/work/programs/feynrules-current/"];  
<< FeynRules`  
SetDirectory["/Users/abetomo/work/programs/feynrules-current/Models/singletDM"];
```

UFO以外のは manual に書いてあるとおり。(1310.1921 の第6章)

- **Calchep**

WriteCHOutput[...]

- **FeynArts**

WriteFeynArtsOutput[...]

- **MadGraph**

WriteUFO[...]

注意：続けざまに模型ファイルを作ろうとするとバグる場合があります

- 例えば WriteFeynArtsOutput[] 実行後 WriteCHOutput[]を実行するとバグる場合あり
- 1つ模型ファイルを作ったら、Quit[] して模型ファイルを読み込むところから、やり直しましょう。面倒だがそれほど手間ではない。

その他

Calchep や Madgraph にどう読み込ませるの？

- 午前中の遠藤さんの話、もしくはマニュアルを参照

あなた「念願の FeynRules をマスターしたぞ！」

- **まだ覚えなないといけないことがあります**
 - ★ mass eigenstate \neq gauge eigenstate のとき
 - ★ フェルミオン場、ゲージ場、ゴースト場の扱い
 - ★ その他こまごまとしたTips
- **case by case で必要となることは異なるので
使いながら少しずつ覚えていきましょう**

ここまでの模型ファイルはこちらから入手できます

<http://bit.ly/singletDM>

じゃあ SM.fr を見てみましょう

ねらい

- スカラー場以外はどんな感じで書かれているか、眺める
- 結構上の方に VEV を定義する欄があることを、眺める
- gauge vs mass eigenstates をどうやるか眺める
((W^3, B) と (photon と Z) の関係を見て理解する)

テキストエディタ (メモ帳、emacs、…) で次を開く

“feynrules-current/Models/SM/SM.fr”

**このファイルは後ですっと使うので、変更しないこと
(変更したら、ダウンロードからやり直し。)**

SM + VEV をもつ scalar singlet をつくってみよう (1)

模型の定義

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2} \partial^\mu S \partial_\mu S - \frac{\mu_S^2}{2} S^2 - \frac{\lambda_S}{24} S^4 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H$$

- S の性質
 - ★ scalar gauge singlet
 - ★ 真空期待値 を持つ

ねらい

- mixing がある場合の練習
- gauge eigenstates \neq mass eigenstates
- Higgs の signal strength が一様にずれるベンチマーク模型

SM + VEV をもつ scalar singlet をつくってみよう (2)

VEV (v と v_S) の周りの揺らぎ (gauge eigenstate の定義)

$$S = v_S + \sigma_1, \quad H = \begin{pmatrix} -i\pi_{W^+} \\ \frac{1}{\sqrt{2}}(v + \sigma_2 + i\pi_Z) \end{pmatrix}$$

- σ_1 と σ_2 は gauge eigenstates \neq mass eigenstate
- π_W, π_Z は would-be NG boson (今の場合は mass eigenstates)

mass eigenstates (h と H) との関係を“定義”する

$$\begin{pmatrix} \sigma_2 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} \cos \theta_h & -\sin \theta_h \\ \sin \theta_h & \cos \theta_h \end{pmatrix} \begin{pmatrix} h \\ h' \end{pmatrix}$$

- θ_h はポテンシャルから質量行列を求めて、そこから求まるやつです。

SM + VEV をもつ scalar singlet をつくってみよう (3)

scalar potential

$$V = -\mu_1^2 H^\dagger H + \lambda (H^\dagger H)^2 + \frac{\mu_S^2}{2} S^2 + \frac{\lambda_S}{24} S^4 + \frac{\lambda_{hS}}{2} S^2 H^\dagger H$$

VEVと μ_1, μ_S の関係

$$\begin{aligned}\mu_1^2 &= \frac{1}{2} (2v^2 \lambda + v_S^2 \lambda_{hS}), \\ \mu_S^2 &= \frac{1}{6} (-v_S^2 \lambda_S - 3v^2 \lambda_{hS})\end{aligned}$$

SM + VEV をもつ scalar singlet をつくってみよう (4)

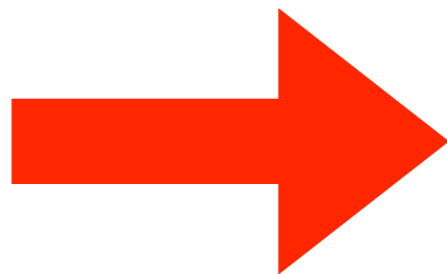
input parameterの選び方

- $(v, v_S, \lambda, \lambda_S, \lambda_{hS})$
- $(v, v_S, m_H, m_{H'}, \cos\theta_h)$ (\leq 僕の好み)

$(\lambda, \lambda_S, \lambda_{hS})$ vs $(m_H, m_{H'}, \cos\theta_h)$

質量行列とmixing angle と質量固有値の関係を使う

$$\begin{pmatrix} 2v^2\lambda & vv_S\lambda_{hS} \\ vv_S\lambda_{hS} & v_S^2\lambda_S/3 \end{pmatrix} = \begin{pmatrix} \cos\theta_h & -\sin\theta_h \\ \sin\theta_h & \cos\theta_h \end{pmatrix} \begin{pmatrix} m_H^2 & 0 \\ 0 & m_{H'}^2 \end{pmatrix} \begin{pmatrix} \cos\theta_h & \sin\theta_h \\ -\sin\theta_h & \cos\theta_h \end{pmatrix}$$



$$\begin{aligned} \lambda &= \frac{1}{2v^2} (\cos^2\theta_h m_H^2 + \sin^2\theta_h m_{H'}^2), \\ \lambda_S &= \frac{3}{v_S^2} (\sin^2\theta_h m_H^2 + \cos^2\theta_h m_{H'}^2), \\ \lambda_{hS} &= \frac{1}{vv_S} (\sin\theta_h \cos\theta_h (m_H^2 - m_{H'}^2)) \end{aligned}$$

SM + VEV をもつ scalar singlet をつくってみよう (5)

実際に作ってみましょう

- 新しいディレクトリを作成(仮に singletScalar/ とする)
- singletDM.fr をコピーしてくる。singletScalar.fr と名前を変える
- SM.fr をコピーしてくる。SMmod.fr と名前を変える

大雑把には、次のことをやります

- VEV を定義
- mass eigenstate を定義
- gauge eigenstate を定義
(例えば、SMmod.fr のS[11] にある H は適当に書き換える)
- パラメータを定義

じゃあ頑張ってください



わからない方は、こちらに模型
ファイルがあるのでご参照下
さい

<http://bit.ly/singletScalar>

リンク一覧

feynrules ダウンロード先 (マニュアルやチュートリアルもある)

<http://feynrules.irmp.ucl.ac.be>

SM + singlet DM

<http://bit.ly/singletDM>

SM + singlet scalar

<http://bit.ly/singletScalar>

Other tips (単なる自分用の覚え書き。 役立つかは保証しない。)

<http://bit.ly/feynrules>