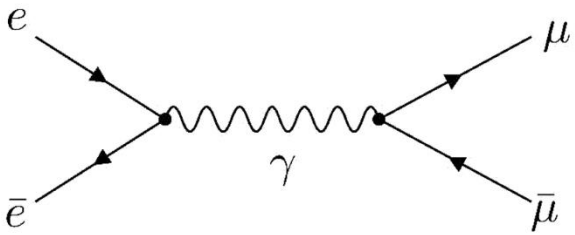
FEYNRULES. BASIC USE

AN INTRODUCTION TO MATHEMATICA TOOLS FOR HEP

MARTÍNEZ MARTÍN, JAVIER



- Wolfram Mathematica for High-Energy Physics
- FeynRules, FeynArts, FeynCalc

24-26 February 2026
UCM-IPARCOS, Madrid

FACULTAD DE CIENCIAS FÍSICAS U. COMPLUTENSE MADRID

IPARCOS
@IPARCOS-UCM
/iparcos

MINISTERIO DE CIENCIA E INNOVACIÓN

UNIÓN EUROPEA
FONDO EUROPEO DE DESARROLLO REGIONAL
"Una manera de hacer Europa"

AGENCIA ESTATAL DE INVESTIGACIÓN

VNIVERSITAS COMPLVTENSIS MATRITENSIS

Organization (UCM-IPARCOS):
Javier Martínez Martín
Juan José Sanz Cillero

FEYNRULES. INTRODUCTION

WHAT IS FEYNRULES? - WHAT IS A MODEL FILE? - FLOWCHART - FIRST STEPS

WHAT IS FEYNRULES?

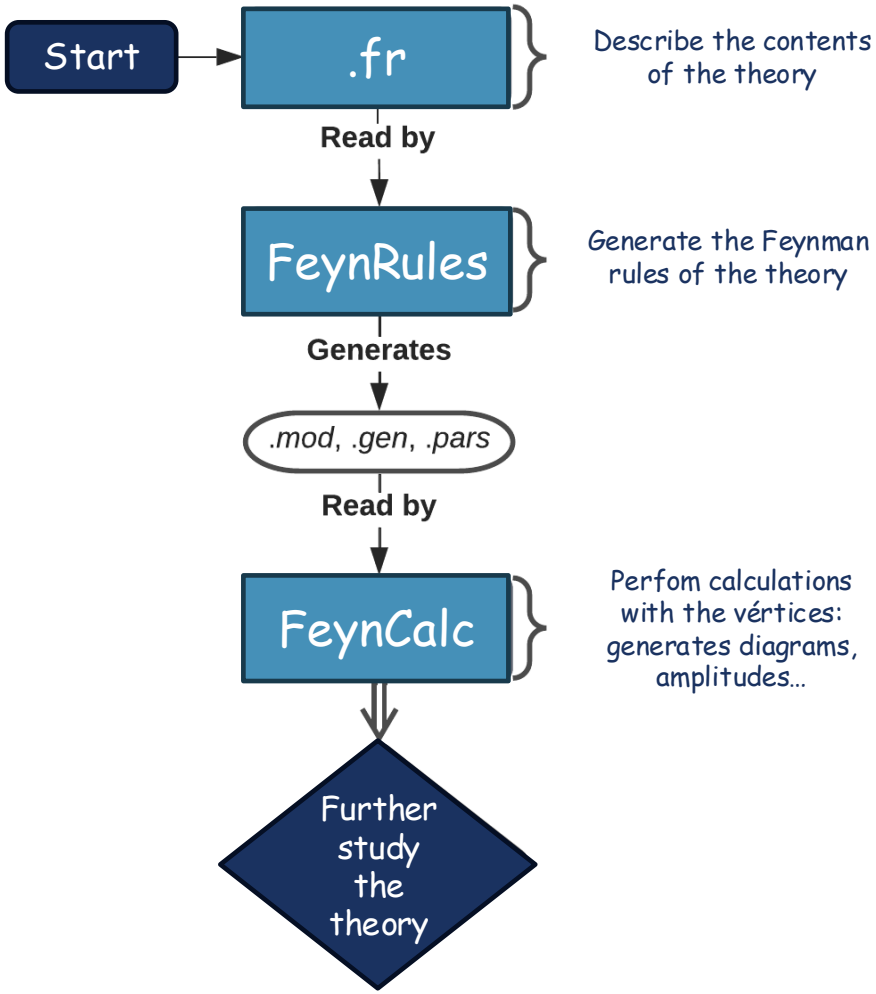
- "FeynRules is a Mathematica package that allows the calculation of Feynman rules in momentum space for any QFT physics model" [1]
- "a Mathematica-based package which addresses the implementation of particle physics models, which are given in the form of a list of fields, parameters and a Lagrangian, into high-energy physics tools. It calculates the underlying Feynman rules and outputs them to a form appropriate for various programs such as CalcHep, FeynArts, MadGraph, Sherpa and Whizard" [2]

[1] HTTPS://FEYNRULES.IRMP.UCL.AC.BE
[2] ARXIV:1310.1921 **[HEP-PH]**

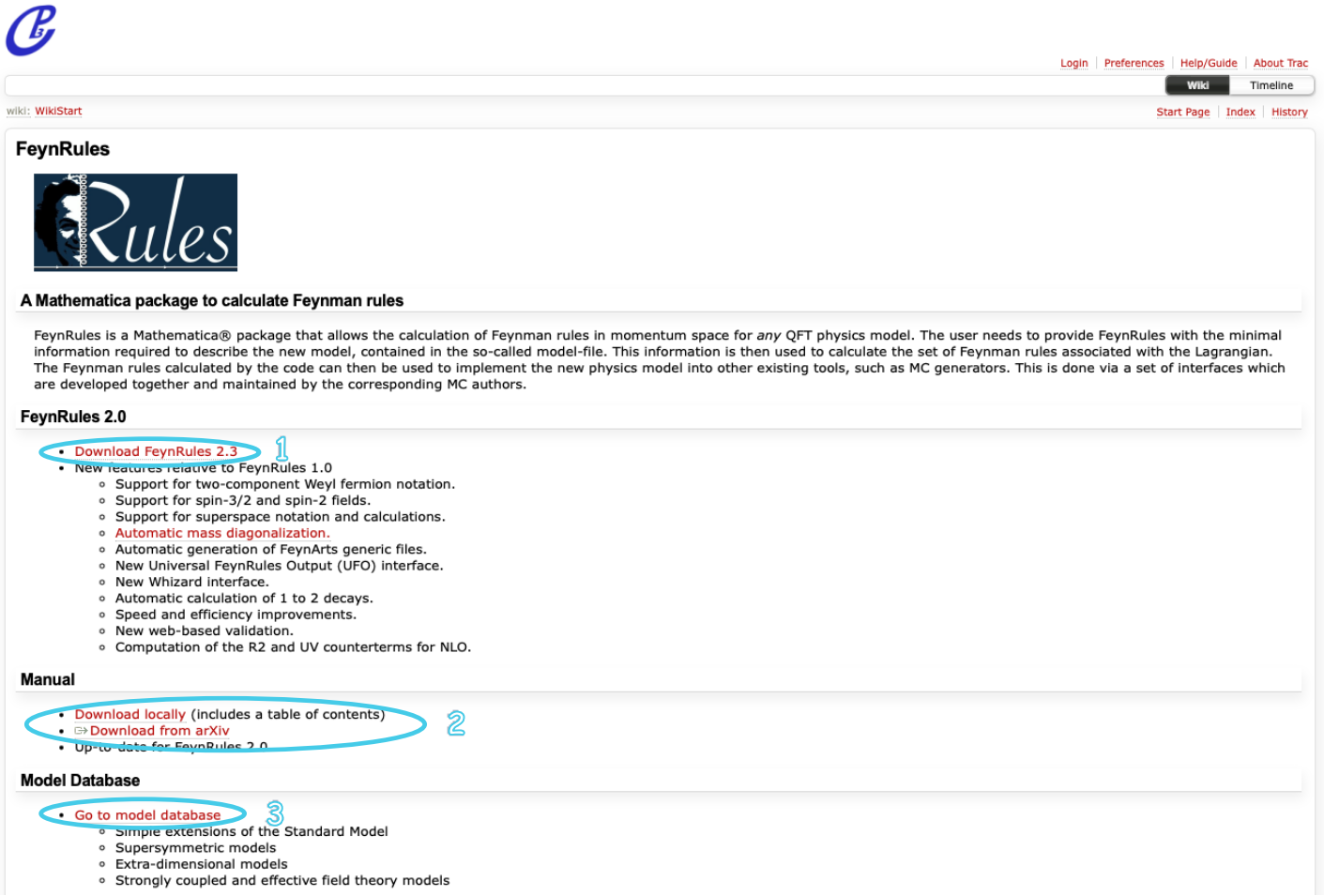
WHAT IS A MODEL FILE?

- A file written in Mathematica syntax where the user specifies the different contents of the model which includes the gauge groups, the particles and the lagrangian.
- While some of them could be directly defined on the Mathematica notebook, it is recommended that most of the model ends up in a different text file, with .fr extension.

FLOWCHART


Start
.fr
Describe the contents of the theory
Read by
FeynRules
Generate the Feynman rules of the theory
Generates
.mod, .gen, .pars
Read by
FeynCalc
Perfom calculations with the vértices: generates diagrams, amplitudes…
Further study the theory

FIRST STEPS



https://feynrules.irmp.ucl.ac.be

Previously:

1. Download the FeynRules package

2. Download the Manual

3. Find pre-built QFT

   Download the SM to follow this session, also found inside the FeynRules installation

MODEL FILE

MODEL INFROMATION - INDICES – PARAMETERS – PARTICLES – GAUGE GROUPS – LAGRANGIAN - EXTRAS

MODEL FILE

- We will be learning some of the commands needed to build a QFT model file as well as its basic syntaxis.
- We will follow the FeynRules manual [2] and the SM.fr [3] model file for the examples.

[2] ARXIV:1310.1921 **[HEP-PH]**
[3] HTTPS://FEYNRULES.IRMP.UCL.AC.BE/ATTACHMENT/WIKI/STANDARDMODEL/SM.FR

MODEL INFORMATION

```
(*****************************************************************************************************************)
(******                        This is the FeynRules mod-file for the Standard model                          ******)
(******                                                                                                       ******)
(******     Authors: N. Christensen, C. Duhr, B. Fuks                                                         ******)
(******                                                                                                       ******)
(****** Choose whether Feynman gauge is desired.                                                             ******)
(****** If set to False, unitary gauge is assumed.                                                             ****)
(****** Feynman gauge is especially useful for CalcHEP/CompHEP where the calculation is 10-100 times faster. ***)
(****** Feynman gauge is not supported in MadGraph and Sherpa.                                                ****)
(*****************************************************************************************************************)

(* ************************** *)
(* *****  Information   ***** *)
(* ************************** *)
M$ModelName = "Standard Model";

M$Information = {
  Authors      -> {"N. Christensen", "C. Duhr", "B. Fuks"},
  Version      -> "1.4.6",
  Date         -> "15. 04. 2014",
  Institutions -> {"Michigan State University", "Universite catholique de Louvain (CP3)", "IPHC Strasbourg / University of Strasbourg"},
  Emails       -> {"neil@pa.msu.edu", "claude.duhr@uclouvain.be", "benjamin.fuks@cnrs.in2p3.fr"},
  URLs         -> "http://feynrules.phys.ucl.ac.be/view/Main/StandardModel"
};
```

Table 1: Model Information

`M$ModelName`	A string, specifying the name of the model. The default value is the name of the model file.
`M$Information`	A replacement list, acting as an electronic signature of the model implementation.
`ModelInformation[]`	Prints the contents of `M$Information`.
Options for `M$Information`	
`Authors`	A list of strings, specifying the authors of the model implementation.
`Institutions`	A list of strings, specifying the institutions of the authors.
`Emails`	A list of strings, specifying the email addresses of the authors. The order of the email addresses is the same as the order in which the names of the authors have been specified.
`Date`	A string specifying the date.
`References`	A list of strings, containing the references that the authors would like to be cited whenever the model implementation is used.
`URLs`	A list of strings, containing Internet addresses where more information about the model can be found.
`Version`	A version number for the model. Each time the model is improved, this version number should be increased and a note written in the model file describing the change.

INDICES

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

IndexRange[Index[SU2W      ]] = Unfold[Range[3]];
IndexRange[Index[SU2D      ]] = Unfold[Range[2]];
IndexRange[Index[Gluon     ]] = NoUnfold[Range[8]];
IndexRange[Index[Colour    ]] = NoUnfold[Range[3]];
IndexRange[Index[Generation]] = Range[3];

IndexStyle[SU2W,       j];
IndexStyle[SU2D,       k];
IndexStyle[Gluon,      a];
IndexStyle[Colour,     m];
IndexStyle[Generation, f];
```

Table 2: Index Information

`Index[name, value]`	Represents an index of type name with a value value.
`IndexRange[Index[name]]`	Declares an index of type name along with its range.
`Range[n]`	Internal MATHEMATICA command, returning the range of integers $\{1, \ldots, n\}$. Allows to declare the range of an index to be from 1 to n.
`IndexStyle[name, label]`	Fixes the `StandardForm` and `TraditionalForm` of an index of type name to be printed as a symbol starting with label.
`NoUnfold`	FEYNARTS command. This is only used by the FEYNARTS interface.
`Unfold`	Indices of this type will always be expanded out in the interfaces. See Section 4.

Predefined types of indices

`Lorentz`	Name for Lorentz indices, ranging from 1 to 4, and printed as μ.
`Spin`	Name for Dirac indices, ranging from 1 to 4, and printed as s.
`Spin1`	Name for left-handed Weyl indices, ranging from 1 to 2, and printed as α.
`Spin2`	Name for right-handed Weyl indices, ranging from 1 to 2, and printed as $\dot{\alpha}$.

PARAMETERS

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

  (* External parameters *)
  aEWM1 == {
    ParameterType    -> External,
    BlockName        -> SMINPUTS,
    OrderBlock       -> 1,
    Value            -> 127.9,
    InteractionOrder -> {QED,-2},
    Description      -> "Inverse of the EW coupling constant at the Z pole"
  },

  (* Internal Parameters *)
  aEW == {
    ParameterType    -> Internal,
    Value            -> 1/aEWM1,
    InteractionOrder -> {QED,2},
    TeX              -> Subscript[\[Alpha], EW],
    Description      -> "Electroweak coupling contant"
  },
```

Table 3: Attributes of the parameter class common for scalar and tensorial parameters.

`ParameterType`	Specifies the nature of a parameter. The allowed choices are `External` or `Internal`. By default, scalar parameters are considered as external and tensorial parameters as internal.
`Value`	Refers to a real number (for external parameters) or to an analytical formula defining the parameter that can be expressed in terms of other parameters (for internal parameters). By default, the `Value` attribute takes the value 1.
`Definitions`	Refers to a list of MATHEMATICA replacement rules that are applied by FEYNRULES before computing the interaction vertices of a model.
`ComplexParameter`	Defines whether a parameter is a real (`False`) or complex (`True`) quantity. By default, scalar parameters are real while tensorial parameters as complex. External parameters must be real.
`BlockName`	This provides information about the name of the Les Houches block containing an external parameter (see Section 6.1.4). By default, the block name is taken as `FRBlock`.
`OrderBlock`	Provides information about the position of an external parameter within a given Les Houches block. By default, this number starts at one as is incremented after the declaration of each parameter. It must be left unspecified for tensorial parameters (see Section 6.1.4).
`TeX`	Teaches FEYNRULES how to write the TEX form of a parameter. By default, it refers to (the string of) the associated MATHEMATICA symbol.
`Description`	Refers to a string containing a description of the physical meaning of the parameter.

Table 4: Additional attributes of the parameter class related to Feynman diagram calculators

`ParameterName`	Specifies what to replace the symbol by before writing out the Feynman diagram calculator model files. By default, it is taken equal to the symbol representing the parameter. See Section 6.1.1 for more details.
`InteractionOrder`	Specifies the order of the parameter according to a specific interaction. It refers to a pair with the interaction name, followed by the order, or a list of such pairs. This option has no default value. See Section 6.1.7 for more details.

PARAMETERS

```
  CKM == {
    ParameterType -> Internal,
    Indices       -> {Index[Generation], Index[Generation]},
    Unitary       -> True,
    Value         -> {CKM[1,1] -> Cos[cabi],  CKM[1,2] -> Sin[cabi], CKM[1,3] -> 0,
                      CKM[2,1] -> -Sin[cabi], CKM[2,2] -> Cos[cabi], CKM[2,3] -> 0,
                      CKM[3,1] -> 0,          CKM[3,2] -> 0,         CKM[3,3] -> 1},
    TeX         -> Superscript[V,CKM],
    Description -> "CKM-Matrix"}
};
```

Table 5: Attributes of the parameter class associated to tensorial parameters.

`Indices`	Mandatory. Provides, as a list, the indices carried by a parameter.
`Unitary`	Defines a matrix as unitary (`True`) or not (`False`). The default value is `False`.
`Hermitian`	Defines a matrix as Hermitian (`True`) or not (`False`). The default value is `False`.
`Orthogonal`	Defines a matrix as orthogonal (`True`) or not (`False`). The default value is `False`.
`AllowSummation`	See the description in the text. By default this option is set to `False`. Moreover, it is only available for parameters with one single index.

PARTICLES

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

(* Gauge bosons: physical vector fields *)
  V[1] == {
    ClassName       -> A,
    SelfConjugate   -> True,
    Mass            -> 0,
    Width           -> 0,
    ParticleName    -> "a",
    PDG             -> 22,
    PropagatorLabel -> "a",
    PropagatorType  -> W,
    PropagatorArrow -> None,
    FullName        -> "Photon"
  },
  V[2] == {
    ClassName       -> Z,
    SelfConjugate   -> True,
    Mass            -> {MZ, 91.1876},
    Width           -> {WZ, 2.4952},
    ParticleName    -> "Z",
    PDG             -> 23,
    PropagatorLabel -> "Z",
    PropagatorType  -> Sine,
    PropagatorArrow -> None,
    FullName        -> "Z"
  },

  F[3] == {
    ClassName        -> ug,
    ClassMembers     -> {u, c, t},
    Indices          -> {Index[Generation], Index[Colour]},
    FlavorIndex      -> Generation,
    SelfConjugate    -> False,
    Mass             -> {Mu, {MU, 2.55*^-3}, {MC,1.27}, {MT,172}},
    Width            -> {0, 0, {WT,1.50833649}},
    QuantumNumbers   -> {Q -> 2/3},
    PropagatorLabel  -> {"ug", "u", "c", "t"},
    PropagatorType   -> Straight,
    PropagatorArrow  -> Forward,
    PDG              -> {2, 4, 6},
    ParticleName     -> {"u",  "c",  "t" },
    AntiParticleName -> {"u~", "c~", "t~"},
    FullName         -> {"u-quark", "c-quark", "t-quark"}
  },
```

- `S`: scalar fields;
- `F`: Dirac and Majorana spinor fields;
- `W`: Weyl fermions (both left- and right-handed);
- `V`: vector fields;
- `R`: four-component Rarita-Schwinger fields (spin-3/2 fields);
- `RW`: two-component Rarita-Schwinger fields (both left- and right-handed spin-3/2 fields);
- `T`: spin-2 fields;
- `U`: ghost fields (only complex ghosts are supported).

Table 6: Particle Class Options

`S, F, W, V, R, RW, T, U`	Particle classes.
`ClassName`	Mandatory. This option gives the symbol by which the class is represented.
`SelfConjugate`	Mandatory. Takes the values `True` or `False`.
`Indices`	The list of indices carried by the field. Note that Lorentz indices (`Lorentz`, `Spin`, `Spin1`, `Spin2`) are implicit and not included in this list.
`FlavorIndex`	The name of the index making the link between the generic class symbol and the class members.
`QuantumNumbers`	A replacement rule list, containing the $U(1)$ quantum numbers carried by the class.
`ClassMembers`	A list with all the members of a class. If the class contains only one member, this is by default the `ClassName`.
`Mass`	A list with the masses for the class members. A mass can be entered either as the symbol which is representing it, or by a two-component list, the first element being the associated symbol and the second one its numerical value. A generic symbol with a default numerical value equal to 1 is generated if omitted.
`Width`	A list with the decay rates for the class members. Similar to `Mass`.
`Ghost`	Option for ghost fields specifying the `ClassName` of the gauge boson the ghost is associated to.
`Goldstone`	For a scalar field, tagging it as the Goldstone boson related to the gauge boson which this option is referring to.
`MajoranaPhase`	The Majorana phase of a Majorana fermion. The default is 0.
`Chirality`	Option for Weyl fermions, specifying their chirality, `Left` or `Right`. The default is `Left`.

PARTICLES

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

(* Gauge bosons: physical vector fields *)
  V[1] == {
    ClassName       -> A,
    SelfConjugate   -> True,
    Mass            -> 0,
    Width           -> 0,
    ParticleName    -> "a",
    PDG             -> 22,
    PropagatorLabel -> "a",
    PropagatorType  -> W,
    PropagatorArrow -> None,
    FullName        -> "Photon"
  },
  V[2] == {
    ClassName       -> Z,
    SelfConjugate   -> True,
    Mass            -> {MZ, 91.1876},
    Width           -> {WZ, 2.4952},
    ParticleName    -> "Z",
    PDG             -> 23,
    PropagatorLabel -> "Z",
    PropagatorType  -> Sine,
    PropagatorArrow -> None,
    FullName        -> "Z"
  },

  F[3] == {
    ClassName        -> ug,
    ClassMembers     -> {u, c, t},
    Indices          -> {Index[Generation], Index[Colour]},
    FlavorIndex      -> Generation,
    SelfConjugate    -> False,
    Mass             -> {Mu, {MU, 2.55*^-3}, {MC,1.27}, {MT,172}},
    Width            -> {0, 0, {WT,1.50833649}},
    QuantumNumbers   -> {Q -> 2/3},
    PropagatorLabel  -> {"ug", "u", "c", "t"},
    PropagatorType   -> Straight,
    PropagatorArrow  -> Forward,
    PDG              -> {2, 4, 6},
    ParticleName     -> {"u",  "c",  "t" },
    AntiParticleName -> {"u~", "c~", "t~"},
    FullName         -> {"u-quark", "c-quark", "t-quark"}
  },
```

Particle Class Options (continued)

`WeylComponents`	Option for the Dirac and Majorana field classes, mapping them to their left- and right-handed Weyl components. For Majorana particles, only the left-handed piece needs to be specified.
`Definitions`	A list of replacement rules that should be applied by FeynRules before calculating the interaction vertices.
`ParticleName`	A list of strings, corresponding to the particle names as they should appear in the output files for the Feynman diagram calculation programs. By default, the value is the same as the one of `ClassMembers`. This name must satisfy the constraints of whatever Feynman diagram calculation package the user wishes to use it with. See Section 6.1.1.
`AntiParticleName`	Similar to `ParticleName`.
`TeXParticleName`	A list of strings with the TeX-form of each class member name. The default is the same as `ParticleName`. See Section 6.1.1.
`TeXAntiParticleName`	Similar to `TeXParticleName`.
`Unphysical`	If `True`, declares that the field does not have to be included in the particle list written for another code by a FeynRules interface but replaced instead by the value of the `Definitions` attribute. The default is `False`.
`PDG`	A list of the PDG codes of the particles. An automatically generated PDG code is assigned if this option is omitted. See Section 6.1.2.
`PropagatorLabel`	A list of strings propagators should be labeled with when drawing Feynman diagrams. The default value is the same as the `ParticleName`. See Section 6.1.8.

Particle Class Options (continued)

`PropagatorArrow`	Whether to put an arrow on the propagator (`True`) or not (`False`). `False` by default. See Section 6.1.8.
`PropagatorType`	This specifies how to draw the propagator line for this field. The default value is inferred from the class. The allowed choices are `ScalarDash` (straight dashed line), `Sine` (sinusoidal line), `Straight` (straight solid line), `GhostDash` (dashed line), and `Curly` (curly *gluonic* line). See Section 6.1.8.
`FullName`	A string, specifying the full name of the particle, or a list containing the names for each class member. By default `FullName` is the same as `ParticleName`.
`MixingPartners`	FeynArts option. See Ref. [6].
`InsertOnly`	FeynArts option. See Ref. [6].
`MatrixTraceFactor`	FeynArts option. See Ref. [6].

GAUGE GROUPS

```
(* ************************* *)
(* *****  Gauge groups  ***** *)
(* ************************* *)
M$GaugeGroups = {
  U1Y  == {
    Abelian          -> True,
    CouplingConstant -> g1,
    GaugeBoson       -> B,
    Charge           -> Y
  },
  SU2L == {
    Abelian           -> False,
    CouplingConstant  -> gw,
    GaugeBoson        -> Wi,
    StructureConstant -> Eps,
    Representations   -> {Ta,SU2D},
    Definitions       -> {Ta[a_,b_,c_]->PauliSigma[a,b,c]/2, FSU2L[i_,j_,k_]:> I Eps[i,j,k]}
  },
  SU3C == {
    Abelian           -> False,
    CouplingConstant  -> gs,
    GaugeBoson        -> G,
    StructureConstant -> f,
    Representations   -> {T,Colour},
    SymmetricTensor   -> dSUN
  }
};
```

Table 10: Gauge group options

`Abelian`	Mandatory, specifying whether the gauge group is abelian (`True`) or not (`False`).
`GaugeBoson`	Refers to the name of the gauge boson associated with the gauge group. This attribute must be specified, except if a superfield is related to the gauge group through the attribute `Superfield`. In addition, the gauge boson must be properly declared in the particle list (see Section 2.4).
`Superfield`	Refers to the name of the vector superfield associated with the gauge group. This attribute must be specified, except if a vector boson is related to the gauge group through the attribute `GaugeBoson`. The superfield must be declared in the superfield list (see Section 2.5).
`CouplingConstant`	Refers to the parameter standing for the coupling constant associated with the gauge group.
`Charge`	Mandatory for abelian groups, referring to the symbol associated with the $U(1)$ charge connected with this gauge group.
`StructureConstant`	Mandatory for non-abelian groups, referring to the symbol associated with the structure constants of the gauge group.
`SymmetricTensor`	Refers to the symbol associated with the totally symmetric tensor of a non-abelian group.

Table 10: Gauge group options (continued)

`Representations`	Refers to a list of two-component lists containing all the representations defined for this gauge group. The first component of these lists consists of the symbol by which the generators of the representation are denoted, while the second component is the name of the index it acts on.
`Definitions`	Contains a list of replacement rules that should be applied by FeynRules before calculating vertices, expressing representation matrices and/or structure constants in terms of the model parameters and Mathematica standard objects.

Table 12: Field strength tensor and covariant derivative

`FS[A, mu, nu]`	Constructs the field strength tensor $F_{\mu\nu}$ connected with the $U(1)$ gauge boson `A`.
`FS[A, mu, nu, a]`	Constructs the field strength tensor $F^a_{\mu\nu}$ connected with the non abelian gauge boson `A`.
`SuperfieldStrengthL[V, sp]`	Calculates the left-handed superfield strength tensor associated with the abelian vector superfield `V`.
`SuperfieldStrengthL[V, sp, a]`	Calculates the left-handed superfield strength tensor associated with the non-abelian vector superfield `V`.
`SuperfieldStrengthR[V, spdot]`	Calculates the right-handed superfield strength tensor associated with the abelian vector superfield `V`.
`SuperfieldStrengthR[V, spdot, a]`	Calculates the right-handed superfield strength tensor associated with the non-abelian vector superfield `V`.
`DC[phi, mu]`	The covariant derivative acting on the field `phi` with a Lorentz index `mu`.

$$F^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g f^a{}_{bc} A^b_\mu A^c_\nu$$

$$D_\mu \phi = \partial_\mu \phi - i g A^a_\mu T_a \phi$$

LAGRANGIAN

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

LGauge := Block[{mu,nu,ii,aa},
  ExpandIndices[-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],
FlavorExpand->SU2W]];
```

$$\mathcal{L}_{\text{Gauge}} = -\frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}W^{i}_{\mu\nu}W^{\mu\nu i} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu a}$$

```
(* Gauge bosons: unphysical vector fields *)
  V[11] == {
    ClassName     -> B,
    Unphysical    -> True,
    SelfConjugate -> True,
    Definitions   -> { B[mu_] -> -sw Z[mu]+cw A[mu]}
  },
  V[12] == {
    ClassName     -> Wi,
    Unphysical    -> True,
    SelfConjugate -> True,
    Indices       -> {Index[SU2W]},
    FlavorIndex   -> SU2W,
    Definitions   -> { Wi[mu_,1] -> (Wbar[mu]+W[mu])/Sqrt[2], Wi[mu_,2] -> (Wbar[mu]-W[mu])/(I*Sqrt[2]), Wi[mu_,3] -> cw
Z[mu] + sw A[mu]}
  },
```

Table 14: Special symbols for Lagrangians

`del[`ϕ, μ`]`	Partial derivative of ϕ with respect to the space-time coordinate x^μ.
`DC[`ϕ, μ`]`	Gauge covariant derivative of ϕ with respect to the space-time coordinate x^μ.
`ME[`μ, ν`]`	Minkowski metric tensor $\eta_{\mu\nu}$.
`IndexDelta[`i,j`]`	Kronecker delta δ_{ij}.
`Eps[a,...,b]`	Totally antisymmetric Levi-Civita tensor with respect to the indices `a,...,b`. In the case of four indices, the conventions on the ϵ-tensors are important when employing interfaces to matrix element generators. They are based on $\epsilon_{0123} = +1$.
`HC[expr ]`	Hermitian conjugate of expr.
`CC[`ψ`]`	The charge conjugate ψ^c of a Dirac field ψ.
`FS`	Field strength tensor.
`Ga[`μ`]`, `Ga[`μ, i, j`]`	Dirac Matrix γ^μ, γ^μ_{ij}.
`ProjP`, `ProjP[`i, j`]`	Chiral projection operator $\frac{1+\gamma^5}{2}$, $\left(\frac{1+\gamma^5}{2}\right)_{ij}$.
`ProjM`, `ProjM[`i, j`]`	Chiral projection operator $\frac{1-\gamma^5}{2}$, $\left(\frac{1-\gamma^5}{2}\right)_{ij}$.
`ProjP[`μ`]`, `ProjP[`μ, i, j`]`	$\gamma^\mu\frac{1+\gamma^5}{2}$, $\left(\gamma^\mu\frac{1+\gamma^5}{2}\right)_{ij}$.
`ProjM[`μ`]`, `ProjM[`μ, i, j`]`	$\gamma^\mu\frac{1-\gamma^5}{2}$, $\left(\gamma^\mu\frac{1-\gamma^5}{2}\right)_{ij}$.
`right[`ψ`]`	The right-handed part of the four-component fermion field ψ, $\frac{1+\gamma^5}{2}\psi$.
`left[`ψ`]`	The left-handed part of the four-component fermion field ψ, $\frac{1-\gamma^5}{2}\psi$.
`si[`μ`]`, `si[`μ, α, $\dot{\alpha}$`]`	Pauli matrix σ^μ, $\sigma^\mu{}_{\alpha\dot{\alpha}}$.
`sibar[`μ`]`, `sibar[`μ, $\dot{\alpha}$, α`]`	Pauli matrix $\bar{\sigma}^\mu$, $\bar{\sigma}^{\mu\dot{\alpha}\alpha}$.

LAGRANGIAN

```
LFermions := Block[{mu},
  ExpandIndices[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]),
  FlavorExpand->{SU2W,SU2D}]/.{CKM[a_,b_] Conjugate[CKM[a_,c_]]->IndexDelta[b,c], CKM[b_,a_] Conjugate[CKM[c_,a_]]-
>IndexDelta[b,c]}];
```

$$\mathcal{L}_{\text{Fermions}} = i \sum_f \bar{f} \gamma^\mu \mathcal{D}_\mu f \quad , \quad f = Q_L, L_L, u_R, d_R, l_R$$

```
(* Fermions: unphysical fields *)
  F[11] == {
    ClassName      -> LL,
    Unphysical     -> True,
    Indices        -> {Index[SU2D], Index[Generation]},
    FlavorIndex    -> SU2D,
    SelfConjugate  -> False,
    QuantumNumbers -> {Y -> -1/2},
    Definitions    -> { LL[sp1_,1,ff_] :> Module[{sp2}, ProjM[sp1,sp2] vl[sp2,ff]], LL[sp1_,2,ff_] :> Module[{sp2},
ProjM[sp1,sp2] l[sp2,ff]] }
  },
  F[12] == {
    ClassName      -> lR,
    Unphysical     -> True,
    Indices        -> {Index[Generation]},
    FlavorIndex    -> Generation,
    SelfConjugate  -> False,
    QuantumNumbers -> {Y -> -1},
    Definitions    -> { lR[sp1_,ff_] :> Module[{sp2}, ProjP[sp1,sp2] l[sp2,ff]] }
  },
  F[13] == {
    ClassName      -> QL,
    Unphysical     -> True,
    Indices        -> {Index[SU2D], Index[Generation], Index[Colour]},
    FlavorIndex    -> SU2D,
    SelfConjugate  -> False,
    QuantumNumbers -> {Y -> 1/6},
    Definitions    -> {
      QL[sp1_,1,ff_,cc_] :> Module[{sp2}, ProjM[sp1,sp2] uq[sp2,ff,cc]],
      QL[sp1_,2,ff_,cc_] :> Module[{sp2,ff2}, CKM[ff,ff2] ProjM[sp1,sp2] dq[sp2,ff2,cc]] }
  },
  F[14] == {
    ClassName      -> uR,
    Unphysical     -> True,
    Indices        -> {Index[Generation], Index[Colour]},
    FlavorIndex    -> Generation,
    SelfConjugate  -> False,
    QuantumNumbers -> {Y -> 2/3},
    Definitions    -> { uR[sp1_,ff_,cc_] :> Module[{sp2}, ProjP[sp1,sp2] uq[sp2,ff,cc]] }
  },
  F[15] == {
    ClassName      -> dR,
    Unphysical     -> True,
    Indices        -> {Index[Generation], Index[Colour]},
    FlavorIndex    -> Generation,
    SelfConjugate  -> False,
    QuantumNumbers -> {Y -> -1/3},
    Definitions    -> { dR[sp1_,ff_,cc_] :> Module[{sp2}, ProjP[sp1,sp2] dq[sp2,ff,cc]] }
  },
```

Table 14: Special symbols for Lagrangians

`del[`ϕ, μ`]`	Partial derivative of ϕ with respect to the space-time coordinate x^μ.
`DC[`ϕ, μ`]`	Gauge covariant derivative of ϕ with respect to the space-time coordinate x^μ.
`ME[`μ, ν`]`	Minkowski metric tensor $\eta_{\mu\nu}$.
`IndexDelta[`i,j`]`	Kronecker delta δ_{ij}.
`Eps[a,...,b]`	Totally antisymmetric Levi-Civita tensor with respect to the indices `a,...,b`. In the case of four indices, the conventions on the ϵ-tensors are important when employing interfaces to matrix element generators. They are based on $\epsilon_{0123} = +1$.
`HC[expr ]`	Hermitian conjugate of expr.
`CC[`ψ`]`	The charge conjugate ψ^c of a Dirac field ψ.
`FS`	Field strength tensor.
`Ga[`μ`]`, `Ga[`μ, i, j`]`	Dirac Matrix γ^μ, γ^μ_{ij}.
`ProjP`, `ProjP[`i, j`]`	Chiral projection operator $\frac{1+\gamma^5}{2}$, $\left(\frac{1+\gamma^5}{2}\right)_{ij}$.
`ProjM`, `ProjM[`i, j`]`	Chiral projection operator $\frac{1-\gamma^5}{2}$, $\left(\frac{1-\gamma^5}{2}\right)_{ij}$.
`ProjP[`μ`]`, `ProjP[`μ, i, j`]`	$\gamma^\mu \frac{1+\gamma^5}{2}$, $\left(\gamma^\mu \frac{1+\gamma^5}{2}\right)_{ij}$.
`ProjM[`μ`]`, `ProjM[`μ, i, j`]`	$\gamma^\mu \frac{1-\gamma^5}{2}$, $\left(\gamma^\mu \frac{1-\gamma^5}{2}\right)_{ij}$.
`right[`ψ`]`	The right-handed part of the four-component fermion field ψ, $\frac{1+\gamma^5}{2}\psi$.
`left[`ψ`]`	The left-handed part of the four-component fermion field ψ, $\frac{1-\gamma^5}{2}\psi$.
`si[`μ`]`, `si[`μ, α, $\dot{\alpha}$`]`	Pauli matrix σ^μ, $\sigma^\mu{}_{\alpha\dot{\alpha}}$.
`sibar[`μ`]`, `sibar[`μ, $\dot{\alpha}$, α`]`	Pauli matrix $\bar{\sigma}^\mu$, $\bar{\sigma}^{\mu\dot{\alpha}\alpha}$.

EXTRAS

- The model file can describe some other features we are not able to cover in these talks: superfields, restrictions and mixing relations.
- One should now be capable to further explore its capabilities using the manual, we encourage you to do so.
- Develope your own model! Contribute to the database!

RUNNING FEYNRULES

LOADING THE MODEL - CHECKING THE LAGRANGIAN - THE FEYNMAN RULES - OUTPUTS

RUNNING FEYNRULES

- We will be learning how to load a pre-built model, check its consistency and obtain the Feynmann rules of the lagrangian.
- We will again follow the FeynRules manual [2], use the SM.fr [3] model file and the SM.nb [4] for Mathematica.
- Copy-paste SM.fr and SM.nb in the same folder.

[2] ARXIV:1310.1921 **[HEP-PH]**
[3] HTTPS://FEYNRULES.IRMP.UCL.AC.BE/ATTACHMENT/WIKI/STANDARDMODEL/SM.FR
[4] HTTPS://FEYNRULES.IRMP.UCL.AC.BE/ATTACHMENT/WIKI/STANDARDMODEL/SM.NB

LOADING THE MODEL

- Open SM.nb notebook with Mathematica
- Steps 1-3 are already discussed

4. Set the working directory to the notebook directory

It is best to set the directory to the notebook's so that every exported file is easily found

```
In[•]:= Quit[];                                                    1
        deten nucle

2 In[1]:= $FeynRulesPath = SetDirectory["/Applications/Mathematica.app/Contents/AddOns/feynrules"];
                            establece directorio

In[2]:= << FeynRules`;                                             3

 - FeynRules -

Version: 2.3.49 (29 September 2021).

Authors: A. Alloul, N. Christensen, C. Degrande, C. Duhr, B. Fuks


Please cite:

    - Comput.Phys.Commun.185:2250-2300,2014 (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[].

In[3]:= SetDirectory[NotebookDirectory[]];                         4
        establece directorio directorio de cuaderno
```

LOADING THE MODEL

5. Load the model

Remind that it must be located in the directory the Mathematica session is set.

The Standard Model

We first load in the Standard Model model-file

```
In[4]:= LoadModel["SM.fr"]
```

5

```
This model implementation was created by
N. Christensen
C. Duhr
B. Fuks
Model Version: 1.4.6
http://feynrules.phys.ucl.ac.be/view/Main/StandardModel
For more information, type ModelInformation[].

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

Model Standard Model loaded.

LoadRestriction["Massless.rst", "DiagonalCKM.rst"]
```

CHECKING THE LAGRANGIAN

- Just writing the name given in the Model file for the lagrangian will show the full expression

Unitary Gauge

The full lagrangian in unitary gauge can be accessed immediatly via

In[5]:= `LSM;`

Gauge sector

The part of the lagrangian representing the gauge sector can be accessed via

In[6]:= `LGauge`

Out[6]=
$$-\frac{1}{4}c_w^2\,\partial_{nu}[A_{mu}]^2-\frac{1}{4}s_w^2\,\partial_{nu}[A_{mu}]^2+\frac{1}{2}c_w^2\,\partial_{nu}[A_{mu}]\,\partial_{mu}[A_{nu}]+\frac{1}{2}s_w^2\,\partial_{nu}[A_{mu}]\,\partial_{mu}[A_{nu}]-\frac{1}{4}c_w^2\,\partial_{mu}[A_{nu}]^2-\frac{1}{4}s_w^2\,\partial_{mu}[A_{nu}]^2-\frac{1}{4}\,\partial_{nu}[G_{mu,aa}]^2+\frac{1}{2}\,\partial_{nu}[G_{mu,aa}]\,\partial_{mu}[G_{nu,aa}]-\frac{1}{4}\,\partial_{mu}[G_{nu,aa}]^2-$$
$$\frac{1}{2}\,\partial_{nu}[W_{mu}]\,\partial_{nu}[W_{mu}^\dagger]+\frac{1}{2}\,\partial_{mu}[W_{nu}]\,\partial_{nu}[W_{mu}^\dagger]+\frac{1}{2}\,\partial_{nu}[W_{mu}]\,\partial_{mu}[W_{nu}^\dagger]-\frac{1}{2}\,\partial_{mu}[W_{nu}]\,\partial_{mu}[W_{nu}^\dagger]-\frac{1}{4}c_w^2\,\partial_{nu}[Z_{mu}]^2-\frac{1}{4}s_w^2\,\partial_{nu}[Z_{mu}]^2+\frac{1}{2}c_w^2\,\partial_{nu}[Z_{mu}]\,\partial_{mu}[Z_{nu}]+\frac{1}{2}s_w^2\,\partial_{nu}[Z_{mu}]\,\partial_{mu}[Z_{nu}]-\frac{1}{4}c_w^2\,\partial_{mu}[Z_{nu}]^2-\frac{1}{4}s_w^2\,\partial_{mu}[Z_{nu}]^2+$$
$$\frac{1}{4}g_s\,\partial_{nu}[G_{mu,aa}]\,f_{aa,bb\$6689,cc\$6689}\,G_{mu,bb\$6689}\,G_{nu,cc\$6689}-\frac{1}{4}g_s\,\partial_{mu}[G_{nu,aa}]\,f_{aa,bb\$6689,cc\$6689}\,G_{mu,bb\$6689}\,G_{nu,cc\$6689}+\frac{1}{4}g_s\,\partial_{nu}[G_{mu,aa}]\,f_{aa,bb\$6690,cc\$6690}\,G_{mu,bb\$6690}\,G_{nu,cc\$6690}-\frac{1}{4}g_s\,\partial_{mu}[G_{nu,aa}]\,f_{aa,bb\$6690,cc\$6690}\,G_{mu,bb\$6690}\,G_{nu,cc\$6690}-$$
$$\frac{1}{4}g_s^2\,f_{aa,bb\$6689,cc\$6689}\,f_{aa,bb\$6690,cc\$6690}\,G_{mu,bb\$6689}\,G_{mu,bb\$6690}\,G_{nu,cc\$6689}\,G_{nu,cc\$6690}+\frac{1}{2}\,i\,e\,A_{nu}\,\partial_{nu}[W_{mu}^\dagger]\,W_{mu}-\frac{1}{2}\,i\,e\,A_{nu}\,\partial_{mu}[W_{nu}^\dagger]\,W_{mu}-\frac{1}{2}\,i\,e\,A_{mu}\,\partial_{nu}[W_{mu}^\dagger]\,W_{nu}+\frac{1}{2}\,i\,e\,A_{mu}\,\partial_{mu}[W_{nu}^\dagger]\,W_{nu}-\frac{1}{2}\,i\,e\,A_{nu}\,\partial_{nu}[W_{mu}]\,W_{mu}^\dagger+$$
$$\frac{1}{2}\,i\,e\,A_{nu}\,\partial_{mu}[W_{nu}]\,W_{mu}^\dagger-\frac{1}{2}\,e^2\,A_{nu}^2\,W_{mu}\,W_{mu}^\dagger+\frac{1}{2}\,e^2\,A_{mu}\,A_{nu}\,W_{nu}\,W_{mu}^\dagger+\frac{1}{2}\,i\,e\,\partial_{nu}[A_{mu}]\,W_{nu}\,W_{mu}^\dagger-\frac{1}{2}\,i\,e\,\partial_{mu}[A_{nu}]\,W_{nu}\,W_{mu}^\dagger+\frac{i\,c_w\,e\,\partial_{nu}[Z_{mu}]\,W_{nu}\,W_{mu}^\dagger}{2\,s_w}-\frac{i\,c_w\,e\,\partial_{mu}[Z_{nu}]\,W_{nu}\,W_{mu}^\dagger}{2\,s_w}+\frac{e^2\,W_{nu}^2\,\left(W_{mu}^\dagger\right)^2}{4\,s_w^2}+\frac{1}{2}\,i\,e\,A_{mu}\,\partial_{nu}[W_{mu}]\,W_{nu}^\dagger-$$
$$\frac{1}{2}\,i\,e\,A_{mu}\,\partial_{mu}[W_{nu}]\,W_{nu}^\dagger+\frac{1}{2}\,e^2\,A_{mu}\,A_{nu}\,W_{mu}\,W_{nu}^\dagger-\frac{1}{2}\,i\,e\,\partial_{nu}[A_{mu}]\,W_{mu}\,W_{nu}^\dagger+\frac{1}{2}\,i\,e\,\partial_{mu}[A_{nu}]\,W_{mu}\,W_{nu}^\dagger-\frac{i\,c_w\,e\,\partial_{nu}[Z_{mu}]\,W_{mu}\,W_{nu}^\dagger}{2\,s_w}+\frac{i\,c_w\,e\,\partial_{mu}[Z_{nu}]\,W_{mu}\,W_{nu}^\dagger}{2\,s_w}-\frac{1}{2}\,e^2\,A_{mu}^2\,W_{nu}\,W_{nu}^\dagger-\frac{e^2\,W_{mu}\,W_{nu}\,W_{mu}^\dagger\,W_{nu}^\dagger}{2\,s_w^2}+\frac{e^2\,W_{mu}^2\,\left(W_{nu}^\dagger\right)^2}{4\,s_w^2}-$$
$$\frac{i\,c_w\,e\,\partial_{nu}\left[W_{mu}^\dagger\right]\,W_{nu}\,Z_{mu}}{2\,s_w}+\frac{i\,c_w\,e\,\partial_{mu}\left[W_{nu}^\dagger\right]\,W_{nu}\,Z_{mu}}{2\,s_w}+\frac{c_w\,e^2\,A_{nu}\,W_{nu}\,W_{mu}^\dagger\,Z_{mu}}{2\,s_w}+\frac{i\,c_w\,e\,\partial_{nu}[W_{mu}]\,W_{mu}^\dagger\,Z_{mu}}{2\,s_w}-\frac{i\,c_w\,e\,\partial_{mu}[W_{nu}]\,W_{mu}^\dagger\,Z_{mu}}{2\,s_w}+\frac{c_w\,e^2\,A_{mu}\,W_{nu}\,W_{mu}^\dagger\,Z_{mu}}{2\,s_w}-\frac{c_w\,e^2\,A_{mu}\,W_{nu}\,W_{nu}^\dagger\,Z_{mu}}{s_w}-\frac{c_w^2\,e^2\,W_{nu}\,W_{nu}^\dagger\,Z_{mu}^2}{2\,s_w^2}+\frac{i\,c_w\,e\,\partial_{nu}\left[W_{mu}^\dagger\right]\,W_{mu}\,Z_{nu}}{2\,s_w}-$$
$$\frac{i\,c_w\,e\,\partial_{mu}\left[W_{nu}^\dagger\right]\,W_{mu}\,Z_{nu}}{2\,s_w}-\frac{i\,c_w\,e\,\partial_{nu}[W_{mu}]\,W_{nu}^\dagger\,Z_{nu}}{2\,s_w}+\frac{i\,c_w\,e\,\partial_{mu}[W_{nu}]\,W_{mu}^\dagger\,Z_{nu}}{2\,s_w}-\frac{c_w\,e^2\,A_{nu}\,W_{mu}\,W_{mu}^\dagger\,Z_{nu}}{s_w}+\frac{c_w\,e^2\,A_{nu}\,W_{nu}\,W_{mu}^\dagger\,Z_{nu}}{2\,s_w}+\frac{c_w\,e^2\,A_{mu}\,W_{mu}\,W_{nu}^\dagger\,Z_{nu}}{2\,s_w}+\frac{c_w^2\,e^2\,W_{nu}\,W_{mu}^\dagger\,Z_{mu}\,Z_{nu}}{2\,s_w^2}+\frac{c_w^2\,e^2\,W_{mu}\,W_{nu}^\dagger\,Z_{mu}\,Z_{nu}}{2\,s_w^2}-\frac{c_w^2\,e^2\,W_{mu}\,W_{mu}^\dagger\,Z_{nu}^2}{2\,s_w^2}$$

Scalar sector

The part of the lagrangian representing the scalar sector can be accessed via

In[7]:= `LHiggs;`

Fermion sector

The part of the lagrangian representing the fermion sector can be accessed via

In[8]:= `LFermions;`

Yukawa sector

The part of the lagrangian representing the Yukawa sector can be accessed via

In[9]:= `LYukawa;`

CHECKING THE LAGRANGIAN

Checking the Lagrangian

```
FeynmanGauge = False;
```
false

Checking hermiticity

The hermiticity of the Lagrangian can be checked via

```
In[10]:= CheckHermiticity[LSM]

Checking for hermiticity by calculating the Feynman rules contained in L-HC[L].
If the lagrangian is hermitian, then the number of vertices should be zero.
Starting Feynman rule calculation.
Expanding the Lagrangian...
Expanding the indices over 10 cores
Collecting the different structures that enter the vertex.
No vertices found.
0 vertices obtained.
The lagrangian is hermitian.

Out[10]= {}
```

The same command can be applied to the flavor - expanded Lagrangian

Table 16: Checking the consistency of a Lagrangian

All the functions below share the same options as `FeynmanRules`, which can be found in Table 19.

`CheckHermiticity[` $\mathcal{L}$, options `]`

Checks if the Lagrangian $\mathcal{L}$ is Hermitian.

`CheckDiagonalKineticTerms[` $\mathcal{L}$, options `]`

Checks if all the kinetic terms in the Lagrangian $\mathcal{L}$ are diagonal.

`CheckDiagonalMassTerms[` $\mathcal{L}$, options `]`

Checks if all the mass terms in the Lagrangian $\mathcal{L}$ are diagonal.

`CheckDiagonalQuadraticTerms[` $\mathcal{L}$, options `]`

Checks if all the quadratic terms in the Lagrangian $\mathcal{L}$ are diagonal.

`CheckKineticTermNormalisation[` $\mathcal{L}$, options `]`

Checks if all the kinetic terms in the Lagrangian $\mathcal{L}$ are correctly diagonalized and normalized.

`CheckMassSpectrum[` $\mathcal{L}$, options `]`

Checks if all the mass terms in the Lagrangian $\mathcal{L}$ are correctly diagonalized and if their value corresponds to the numerical value given in the model file.

THE FEYNMAN RULES

`FeynmanRules[`$\mathcal{L}$, options`]`

Calculates the Feynman rules associated with the Lagrangian $\mathcal{L}$. The list of available options are given in Table 19.

Gauge sector

```
In[25]:= vertsGauge = FeynmanRules[LGauge, FlavorExpand → SU2W]
```

```
Starting Feynman rule calculation.
Expanding the Lagrangian...
Expanding the indices over 10 cores
Collecting the different structures that enter the vertex.
8 possible non-zero vertices have been found -> starting the computation: 8 / 8.
8 vertices obtained.
```

Out[25]=

$$\Big\{\{\{G,1\},\{G,2\},\{G,3\}\},\ g_s\, f_{a_1,a_2,a_3}\, p_1^{\mu_3}\,\eta_{\mu_1,\mu_2} - g_s\, f_{a_1,a_2,a_3}\, p_2^{\mu_3}\,\eta_{\mu_1,\mu_2} - g_s\, f_{a_1,a_2,a_3}\, p_1^{\mu_2}\,\eta_{\mu_1,\mu_3} + g_s\, f_{a_1,a_2,a_3}\, p_3^{\mu_2}\,\eta_{\mu_1,\mu_3} + g_s\, f_{a_1,a_2,a_3}\, p_2^{\mu_1}\,\eta_{\mu_2,\mu_3} - g_s\, f_{a_1,a_2,a_3}\, p_3^{\mu_1}\,\eta_{\mu_2,\mu_3}\Big\},$$

$$\Big\{\{\{G,1\},\{G,2\},\{G,3\},\{G,4\}\},\ i\, g_s^2\, f_{a_1,a_3,\text{Gluon\$1}}\, f_{a_2,a_4,\text{Gluon\$1}}\,\eta_{\mu_1,\mu_4}\,\eta_{\mu_2,\mu_3} + i\, g_s^2\, f_{a_1,a_2,\text{Gluon\$1}}\, f_{a_3,a_4,\text{Gluon\$1}}\,\eta_{\mu_1,\mu_4}\,\eta_{\mu_2,\mu_3} +$$
$$i\, g_s^2\, f_{a_1,a_4,\text{Gluon\$1}}\, f_{a_2,a_3,\text{Gluon\$1}}\,\eta_{\mu_1,\mu_3}\,\eta_{\mu_2,\mu_4} - i\, g_s^2\, f_{a_1,a_2,\text{Gluon\$1}}\, f_{a_3,a_4,\text{Gluon\$1}}\,\eta_{\mu_1,\mu_3}\,\eta_{\mu_2,\mu_4} - i\, g_s^2\, f_{a_1,a_4,\text{Gluon\$1}}\, f_{a_2,a_3,\text{Gluon\$1}}\,\eta_{\mu_1,\mu_2}\,\eta_{\mu_3,\mu_4} - i\, g_s^2\, f_{a_1,a_3,\text{Gluon\$1}}\, f_{a_2,a_4,\text{Gluon\$1}}\,\eta_{\mu_1,\mu_2}\,\eta_{\mu_3,\mu_4}\Big\},$$

$$\Big\{\{\{A,1\},\{W,2\},\{W^\dagger,3\}\},\ -i\, e\, p_1^{\mu_3}\,\eta_{\mu_1,\mu_2} + i\, e\, p_2^{\mu_3}\,\eta_{\mu_1,\mu_2} + i\, e\, p_1^{\mu_2}\,\eta_{\mu_1,\mu_3} - i\, e\, p_3^{\mu_2}\,\eta_{\mu_1,\mu_3} - i\, e\, p_2^{\mu_1}\,\eta_{\mu_2,\mu_3} + i\, e\, p_3^{\mu_1}\,\eta_{\mu_2,\mu_3}\Big\},\ \Big\{\{\{A,1\},\{A,2\},\{W,3\},\{W^\dagger,4\}\},\ i\, e^2\,\eta_{\mu_1,\mu_4}\,\eta_{\mu_2,\mu_3} + i\, e^2\,\eta_{\mu_1,\mu_3}\,\eta_{\mu_2,\mu_4} - 2\, i\, e^2\,\eta_{\mu_1,\mu_2}\,\eta_{\mu_3,\mu_4}\Big\},$$

$$\Big\{\{\{W,1\},\{W^\dagger,2\},\{Z,3\}\},\ -\frac{i\, c_w\, e\, p_1^{\mu_3}\,\eta_{\mu_1,\mu_2}}{s_w} + \frac{i\, c_w\, e\, p_2^{\mu_3}\,\eta_{\mu_1,\mu_2}}{s_w} + \frac{i\, c_w\, e\, p_1^{\mu_2}\,\eta_{\mu_1,\mu_3}}{s_w} - \frac{i\, c_w\, e\, p_3^{\mu_2}\,\eta_{\mu_1,\mu_3}}{s_w} - \frac{i\, c_w\, e\, p_2^{\mu_1}\,\eta_{\mu_2,\mu_3}}{s_w} + \frac{i\, c_w\, e\, p_3^{\mu_1}\,\eta_{\mu_2,\mu_3}}{s_w}\Big\},$$

$$\Big\{\{\{W,1\},\{W,2\},\{W^\dagger,3\},\{W^\dagger,4\}\},\ -\frac{i\, e^2\,\eta_{\mu_1,\mu_4}\,\eta_{\mu_2,\mu_3}}{s_w^2} - \frac{i\, e^2\,\eta_{\mu_1,\mu_3}\,\eta_{\mu_2,\mu_4}}{s_w^2} + \frac{2\, i\, e^2\,\eta_{\mu_1,\mu_2}\,\eta_{\mu_3,\mu_4}}{s_w^2}\Big\},\ \Big\{\{\{A,1\},\{W,2\},\{W^\dagger,3\},\{Z,4\}\},\ -\frac{2\, i\, c_w\, e^2\,\eta_{\mu_1,\mu_4}\,\eta_{\mu_2,\mu_3}}{s_w} + \frac{i\, c_w\, e^2\,\eta_{\mu_1,\mu_3}\,\eta_{\mu_2,\mu_4}}{s_w} + \frac{i\, c_w\, e^2\,\eta_{\mu_1,\mu_2}\,\eta_{\mu_3,\mu_4}}{s_w}\Big\},$$

$$\Big\{\{\{W,1\},\{W^\dagger,2\},\{Z,3\},\{Z,4\}\},\ \frac{i\, c_w^2\, e^2\,\eta_{\mu_1,\mu_4}\,\eta_{\mu_2,\mu_3}}{s_w^2} + \frac{i\, c_w^2\, e^2\,\eta_{\mu_1,\mu_3}\,\eta_{\mu_2,\mu_4}}{s_w^2} - \frac{2\, i\, c_w^2\, e^2\,\eta_{\mu_1,\mu_2}\,\eta_{\mu_3,\mu_4}}{s_w^2}\Big\}\Big\}$$

Table 19: `FeynmanRules` and `SelectVertices` options

`ScreenOutput`	Option of `FeynmanRules`. If turned to `False`, the derived Feynman rules do not appear on the screen. The default is `True`.
`FlavorExpand`	Option of `FeynmanRules`. Expands over the flavor indices specified by the list which this option is referring to. If turned to `True`, all flavor indices are expanded.
`ConservedQuantumNumbers`	Option of `FeynmanRules`. Checks whether the quantum numbers specified by the list which this option is referring to are conserved. If `True` (`False`), the conservation of all (no) quantum numbers is checked. The default is `True`.
`MinParticles`	Only vertices involving at least the specified number of external fields are derived.
`MaxParticles`	Only vertices involving at most the specified number of external fields are derived.
`MinCanonicalDimension`	Option of `FeynmanRules`. Only vertices of at least the specified canonical dimension are derived.
`MaxCanonicalDimension`	Option of `FeynmanRules`. Only vertices of at most the specified canonical dimension are derived.
`SelectParticles`	Calculates only the vertices specified in the list which this option is referring to.
`Contains`	Only the vertices which contain at least the particles contained in the list which this option is referring to are derived.
`Free`	Only the vertices which do not contain any of the particles contained in the list which this option is referring to are derived.

OUTPUTS

- The information of a model can be exported to different formats:
  - CalcHep / CompHep
  - FeynArts / FeynCalc /FormCalc
  - UFO
  - …
- Use advanced options to save just the relevant information for your model

Outputs and interfaces

FeynArts output

The FeynArts output for the SM can be obtained via

```
In[30]:= FeynmanGauge = False;
         (false
```

```
WriteFeynArtsOutput[LGauge, LHiggs, LFermions, LYukawa, FlavorExpand → SU2W, Output → "SM"(*,CouplingRename→False, MaxParticles → 4*)]
 - - - FeynRules interface to FeynArts - - -
      C. Degrande C. Duhr, 2013
      Counterterms: B. Fuks, 2012
Creating output directory: SM
Calculating Feynman rules for L1
Starting Feynman rules calculation for L1.
Expanding the Lagrangian...
Expanding the indices over 10 cores
Collecting the different structures that enter the vertex.
8 possible non-zero vertices have been found -> starting the computation: 3 / 8.
8 vertices obtained.
Calculating Feynman rules for L2
Starting Feynman rules calculation for L2.
Expanding the Lagrangian...
Collecting the different structures that enter the vertex.
6 possible non-zero vertices have been found -> starting the computation: 3 / 6.
6 vertices obtained.
Calculating Feynman rules for L3
Starting Feynman rules calculation for L3.
Expanding the Lagrangian...
Expanding the indices over 10 cores
Collecting the different structures that enter the vertex.
18 possible non-zero vertices have been found -> starting the computation: 3 / 18.
13 vertices obtained.
Calculating Feynman rules for L4
Starting Feynman rules calculation for L4.
Expanding the Lagrangian...
Collecting the different structures that enter the vertex.
3 possible non-zero vertices have been found -> starting the computation: 3 / 3.
3 vertices obtained.
mytimecheck,after LGC
Writing FeynArts model file into directory SM
Writing FeynArts generic file on SM.gen.
```

FeynArts also supports the Feynman gauge

HANDS-ON EXAMPLES

A

HANDS-ON EXAMPLES

https://feynrules.irmp.ucl.ac.be/wiki/ModelDatabaseMainPage

- Let's practice with the following models from the Database:

1. SM
2. Higgs Effective Lagrangian [5]
3. The SMEFT in the Warsaw basis[6]

GOAL. Extracting the vertices:

- (a) $hh \to hh$
- (b) $W^+W^- \to ZZ$
- (c) $e^+e^- \to \gamma$

[5] LSILH for (a) and (b) and no (c).

[6] Use: Flavor = U35, Scheme = alphaScheme. LHiggs for (a), LGauge for (b) and LFermions for (c).

You are now ready to start using FeynArts and FeynCalc