LanHEP - a package for automatic generation of Feynman rules from the Lagrangian. Updated version 3.1

A. Semenov.

Joint Institute of Nuclear research, JINR, 141980 Dubna, Russia

Abstract

We present a new version 3.1 of the LanHEP software package. New features of the program include tools for the models with extra dimensions, implementation of the particle classes for FeynArts output and using templates with LanHEP statements.

Introduction

The LanHEP program [1] is developed for Feynman rules generation from the Lagrangian. It reads the Lagrangian written in a compact form, close to the one used in publications. It means that Lagrangian terms can be written with summation over indices of broken symmetries and using special symbols for complicated expressions, such as covariant derivative and strength tensor for gauge fields. Supersymmetric theories can be described using the superpotential formalism and the 2-component fermion notation. The output is Feynman rules in terms of physical fields and independent parameters in the form of CompHEP [2] or CaclHEP [3] model files, which allows one to start calculations of processes in the new physical model. Alternatively, Feynman rules can be generated in FeynArts [4] format or as LaTeX table. The program can also generate one-loop counterterms in the FeynArts format. This note describes new features of the version 3.1 of the LanHEP package, including tools for the models with extra dimensions, implementation of the particle classes for FeynArts output and using templates with LanHEP statements.

1 Models with extra dimensions

A new feature in LanHEP helps to generate Kaluza-Klein modes for particles in models with additional dimensions. In case of the Minimal Universal Extra Dimension Model [5], the photon field in 5 dimensions can be projected into 4-dimensional space

$$
\begin{aligned}
A_\mu\left(x^\mu, y\right) &= \frac{1}{\sqrt{\pi R}}\left\{A_\mu^{(0)}(x)+\sqrt{2}\sum_{n=1}^{\infty} A_\mu^{(n)}(x)\cos\left(\frac{ny}{R}\right)\right\}, \\
A_5\left(x^\mu, y\right) &= \sqrt{\frac{2}{\pi R}}\sum_{n=1}^{\infty} A_5^{(n)}(x)\sin\left(\frac{ny}{R}\right)
\end{aligned}
$$

LanHEP allows to expand the 4-dimensional field $A_\mu(x^\mu)$ into the sum $A_\mu^{(0)}(x)$ and Kaluza-Klein modes $A_\mu^{(n)}(x)$ as in the right-hand part of the equation above, by the statement

```
transform A -> A + (A1*cos(1) + A2*cos(2))*Sqrt2.
```

Here `sin(`N`)`, `cos(`N`)` correspond to KK-mode number N, and these function will be integrated out using orthogonality relation after constructing the Lagrangian. We have introduced two KK modes in this example, but one can write only one or add more. Similar prescription can be written for scalars and spinors. The `transform` statements allows to expand with KK modes particles in the existing LanHEP model without modifying the statements which describe Lagrangian terms.

One also can define the scalar field corresponding to the fifth component of the photon:

```
let A5 = (s1*sin(1) + s2*sin(2))*Sqrt2.
```

Here `s1`, `s2` are scalar fields which should be declared as particles before. One can write the interactions of the 5th components of vector fields explicitly, using symbols like `A5` defined above and the symbol `deriv5` for ∂_5. Note that `deriv5` differentiate only `sin` and `cos` functions, and multiply it by the mode number, so the scale factor should be written explicitly next to `deriv5`.

LanHEP can generate the interaction of 5th components automatically, by adding the product of the 5th components to each convolution of vector indices. To do this, one should use the `ued_5th` statement to define the 5th components of vectors:

```
ued_5th deriv -> deriv5/R, A -> A5.
```

Here R is the scale parameter. One also can specify the scale parameter in the second argument of `sin` or `cos` function, like `cos(1,invR)`, where invR is defined as 1/R. In this case, one should write `deriv->deriv5` in the `ued_5th` statement.

2 Classes in FeynArts output

FeynArts allows to combine particles with similar properties into *classes*. By default, LanHEP generates the model where each particle has its own class. It is possible to combine several particles into a class by the `class` statement. For example

```
class lpc=[e,m,l].
```

joins the electron, muon and tau-lepton into the class `lpc` (charged lepton). So, the vertices with these three particles will be joined to describe generic lepton interaction with other fields. This feature allows to decrease the number of vertices and speeds up calculations. The particles in the class must have the same spin and color, however it is possible to combine into a class particles with different electric charge, or scalars that are CP-even and CP-odd scalars.

3 Using templates

A model description often includes several statements with the same structure. For example, the declaration of the parameters which are elements of some mixing matrix, evaluated by external function reads

```
parameter Zn11=MixMatr(neu,1,1), Zn12=MixMatr(neu,1,2), ...
```

where '...' stands for all other definitions for this matrix. The declaration of these parameters can be written in a simplier form:

```
_x=1-4, _y=1-4 in parameter Zn_x_y=MixMatr(neu, _x, _y).
```

Here the `parameter` statement will be invoked several times for all possible combinations of values for symbols like _x, making the substitution when _x appears into one of values, and creating names `Zn11, Zn12, ...` from template `Zn_x_y`. These symbols must have one letter. The values can be set as `_x=1-4` or `_x=[1,2,3,4]`. The latter form is useful when substitution values are not sequential numbers, for example values can be particle names. The prefix with keyword `'in'` can be applied to any LanHEP statement.

Another way to execute a statement several times with different names of parameters is to use the keyword `'where'`. This feature was already used in earlier versions in the lagrangian terms. For example

```
lterm   anti(psi)*gamma*(1+g5)/2*(i*deriv - Y*g1*B1)*psi  where
           psi=e, Y= -1;   psi=m, Y= -1;   psi=l, Y= -1;
           psi=u, Y=  2/3; psi=c, Y=  2/3; psi=t, Y=  2/3;
           psi=d, Y= -1/3; psi=s, Y= -1/3; psi=b, Y= -1/3.
```

describes the gauge interaction for quarks and leptons, Y is hypercharge. Now the substitution with the keywors `'where'` can be applied as postfix to any LanHEP statement, and the description of substitutions can be made in a shorter form:

```
lterm ...  where psi=[e,m,l,u,c,t,d,s,b], Y=[-1,-1,-1,2/3,2/3,2/3,-1/3,
-1/3,-1/3].
```

The lists of the values for names of substitutions must be the same length, and this length is the number of times the statement is executed. At each execution the next values from the lists are used for substitution symbols.

When it is necessary to execute the statement with all combinations of substitutions in two (or more) list, one can use nested keywords `'where'`. For example, deriving Yukawa terms from the superpotential may read

```
(lterm -df(superW,Ai,Aj)*Fi*Fj/2 + AddHermConj
where Ai=[h1,h2],Fi=[fh1,fh2] ) where Aj=[h1,h2],Fj=[fh1,fh2].
```

Both `'in'` and `'where'` keywords can be used if it is necessary to combine templates for names like `Zn_i_j` and substitutions for group of symbols. The construction will look:

`_x=name in` *statement* `where name=` *values, ...* .

In this example the keyword `'where'` will substitute `name` with given values, which then will be used by keyword `'in'` to substitute symbol `'_x'` in the *statement*. If the keyword `'in'` is needed to make substitutions before `'where'`, brackets can be used:

`_x=[u,d,c,s,t,b] in (` *statement* `where mass=M_x, ...)` .

In general, any construction with `'in'` or `'where'` can be included in brackets and next `'in'` or `'where'` appended. The outermost keyword makes its substitution first.

Acknowledgments

This work was supported in part by the GDRI-ACPP of CNRS and by the ANR project `ToolsDMColl`, BLAN07-2-194882. This work was also supported by the Russian foundation for Basic Research, grant RFBR-08-02-92499-a.

References

[1] A. Semenov, *Comput.Phys.Commun.* 180:431-454,2009. e-Print: arXiv:0805.0555 [hep-ph].

[2] E. Boos et al., *Nucl.Instrum.Meth.* A534:250-259,2004. e-Print: hep-ph/0403113.

[3] A. Pukhov. e-print hep-ph/0412191.

[4] J. Küblbeck, M. Böhm, and A. Denner, *Comp. Phys. Commun.* **60** (1990) 165;
H. Eck and J. Küblbeck, *Guide to FeynArts 1.0* (Würzburg, 1992);
H. Eck, *Guide to FeynArts 2.0* (Würzburg, 1995);
T. Hahn, *Comp. Phys. Commun.* **140** (2001) 418; [arXiv:hep-ph/0012260].

[5] A. Perez-Lorenzana, *An introduction to extra dimensions*, *J.Phys. Conf. Ser* **18** (2005) 224–269, hep-ph/0503177.