What is new in QGRAF 3.1[1]

P. Nogueira

CFIF, Instituto Superior Técnico,
Universidade Técnica de Lisboa (TULisbon), Lisbon, Portugal.

Abstract

This document is a sort of addendum to the guide for version 3.0, and describes what has changed since that version was released. The manual for version 3.0 remains the main documentation source, to be complemented by this document.

[1] version 2, October 2012

1. The optional statements ‘vsum’ and ‘psum’

The major new feature of version 3.1 consists in having a new way of selecting diagrams that is based on the somewhat recent ability to define a few types of functions in the model file — that is, parameters associated to fields, propagators, and vertices. It is now possible to impose certain numerical constraints that depend on those parameters that are themselves of numerical type (and defined in the model file).

In that regard, the existence of a mechanism to eg constrain the power of a certain coupling constant seems particularly relevant, specially in models with two or more independent gauge coupling constants. In fact, partial radiative corrections based on subsets of diagrams defined by such conditions have been routinely considered in the scientific literature.

Two new optional statements that may be used in the file `qgraf.dat` form the basis of this new ‘filter’ (see also the guide for version 3.0, subsection 4.3). Let us assume that `g_power` is a vertex function that maps every vertex to a number representing the power of a certain coupling constant g in the Feynman rule for that vertex. To constrain the sum of such values for the vertices of a diagram (and thus the power of g in the diagram amplitude) one may write eg

```
true = vsum[ g_power , 4, 4] ;
```

The new operator `vsum` has three arguments: the first is a vertex function and the other two are numerical and are similar to the corresponding arguments of operators `iprop`, `brigde`, etc, with the exception that they can be negative. The function values have to be integer, at least for now, which means that definitions like

```
g_power = '1/2'

g_power = ' 2'
```

are not accepted. In principle it is possible to convert a constraint involving rationals into another one depending on integers only. However, integers appearing in such functions must be smaller than 10^4 in absolute value, assuming your compiler provides at least 32-bits for the standard Fortran `INTEGER` type (16-bit integers should be a faint memory by now).

As it is obvious, this new operator allows the user to consider different types of weights, not just powers of coupling constants. To provide another simple example let us consider a vertex function `one` that takes the value `1` for every vertex. By using `one` as the first argument of a `vsum` statement one is constraining the number of vertices in the diagrams.

There is also an analogous statement for propagator weights. For example,

```
true = psum[ pweight , -1, 1] ;
```

constrains the sum of the values of the propagator function `pweight` taken over the diagram propagators.

2. Minor differences

There are other small (and hopefully harmless) visible differences between versions 3.1 and 3.0. Some of those are discussed below.

The scope of option `floop` has been generalized (the same class of diagrams is eliminated, but the model doesn’t have to be strict QED).

There is a different presentation of the model partitions (screen ouput), which now includes some additional types of propagators. For example, the output will now include something like

```
4P  ---  2+  2-  ---  1N+  1C+  1N-  1C-
```

which is to be interpreted according to the following rules:

- The signs '+' and '-' denote the propagator sign (ie, whether the propagators are defined in terms of commuting or anti-commuting fields).
- The letter 'N' refers to 'neutral' propagators, ie those for which the particle is equal to the anti-particle.
- The letter 'C' refers to 'charged' propagators, ie those for which the particle and the anti-particle differ.

For example, the propagators of Dirac fermions contribute to the coefficient of `C-` (the same goes for some ghost propagators) while the propagators of Majorana fermions contribute to the coefficient of `N-`.

So the previous example tells us that the model file has four propagators (string `4P`), two of which are bosonic (string `2+`) and the other two fermionic (string `2-`). In the case of the bosonic propagators, one is neutral (string `1N+`) and the other is charged (string `1C+`). The fermionic propagators also split in the same manner.

3. Changelog for versions 3.1.x

3.1.1 (April 2008)

A bug of the `psum` operator was fixed, as well as a minor bug related to the screen output (one of the propagator numbers could be wrong).

3.1.2 (September 2010)

A bug of the `vsum` operator was fixed.

3.1.3 (November 2011)

Some nonstandard options of the `OPEN` statement were removed from the source code, following a suggestion made by the `GoSam` collaboration. In that way the source code became automatically compatible with `gfortran` without further ado.

3.1.4 (October 2012)

This release is an attempt to make `QGRAF` accept input files that contain ASCII text prepended by an `utf8` byte order mark, which is included by some applications when saving text. The text characters themselves must still be the ASCII ones defined in the manual for version 3.0.