

# FeynEdit – a tool for drawing Feynman diagrams

T. Hahn<sup>a</sup>, P. Lang<sup>b</sup>

<sup>a</sup>Max-Planck-Institut für Physik  
Föhringer Ring 6, D–80805 Munich, Germany  
<sup>b</sup>Rieslingstr. 62, D–74343 Lauffen

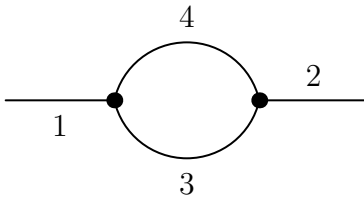
November 8, 2007

## Abstract

We describe the FeynEdit tool for drawing Feynman diagrams. Input and output is done using the  $\text{\LaTeX}$  macros of FeynArts, which also implies that diagrams drawn by FeynArts can be edited with FeynEdit. The  $\text{\LaTeX}$  code can be conveniently transferred using copy-and-paste.

## 1 Introduction

The FeynArts package [1] can paint Feynman diagrams and export them as  $\text{\LaTeX}$  code, such that they can be included directly in publications. For example, the diagram



is represented by the  $\text{\LaTeX}$  code

```
\FAProp(0.,10.)(6.,10.)(0.,){/Straight}{0}
\FALabel(3.,9.18)[t]{$1$}
\FAProp(20.,10.)(14.,10.)(0.,){/Straight}{0}
\FALabel(17.,10.82)[b]{$2$}
\FAProp(6.,10.)(14.,10.)(0.8,){/Straight}{0}
\FALabel(10.,5.98)[t]{$3$}
\FAProp(6.,10.)(14.,10.)(-0.8,){/Straight}{0}
\FALabel(10.,14.02)[b]{$4$}
\FAVert(6.,10.){0}
\FAVert(14.,10.){0}
```

The elements of the diagram are easy to recognize and it is straightforward to make changes e.g. to the label text. It is less straightforward, however, to alter the geometry of the diagram, i.e. to move vertices and propagators.

The new tool FeynEdit fills this gap by allowing the user to copy-and-paste their  $\text{\LaTeX}$  code of the Feynman diagram into the editor, visualize the diagram, modify it using the mouse, and finally copy-and-paste it back into the text.

## 2 Installation

The FeynEdit package can be downloaded from <http://www.feynarts.de>. The program is written in Java and should thus run on almost any platform. The pre-compiled Java bytecode is contained in the file `FeynEdit.jar`. Users with access to a Unix-ish environment may want to run `make` after unpacking the tar file, i.e.

```
gunzip -c FeynEdit-n.m.tar.gz | tar xvf -
cd FeynEdit-n.m
make
```


This turns the jar-file into a regular executable by adding the script signature with the full path of the local system's Java interpreter. Alternately, "`java -jar FeynEdit.jar`" starts the program.


FeynEdit presently does not run with GNU's Java interpreter, `gij`, which is unfortunately installed as the default Java interpreter on some (mostly Debian-derived) Linux distributions. The genuine Sun Java Runtime Environment (JRE) can be obtained from <http://www.java.com>.

On Windows with a JRE installed and on Mac OS X, `FeynEdit.jar` executes directly when double-clicked.

## 3 Usage

The editor is started by typing `FeynEdit` at the command line or by double-clicking on the `FeynEdit.jar` icon on Windows. The start-up screen is shown in Fig. 1.

The window is divided into an upper panel for the diagram display and a lower panel which shows the  $\text{\LaTeX}$  code. To display an existing Feynman diagram, mark its  $\text{\LaTeX}$  code with the mouse and paste it into the lower dialog box. Then press the  button to display the diagram. Otherwise, start with an empty canvas and use the mouse to add elements.

When finished with editing, press the  button to turn the diagrams into  $\text{\LaTeX}$  code, then pick up the latter with the mouse and paste it (back) into your text.

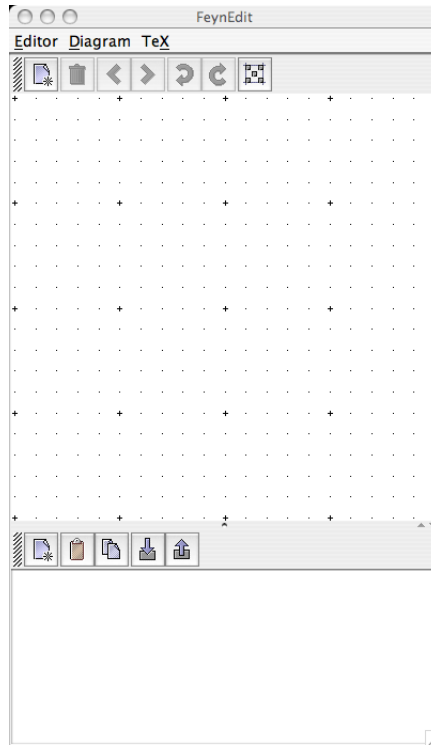


Figure 1: The start-up screen of FeynEdit.

Just as in the Topology Editor of FeynArts, vertices are marked with red, propagators with blue, and labels with green squares. When clicking on a square, the corresponding entity becomes marked and the square is drawn a little larger.

Propagators come in two varieties which have to be distinguished for editing purposes: tadpoles, with coincident initial and final vertices, and ‘ordinary’ propagators.

The diagram can be edited with the mouse. Four editing functions are available:

- Move vertices, propagators, and labels: Click on the corresponding box (red, blue, green) and drag it to the desired position.
- Draw tadpoles: Click on the ‘footpoint’ of the tadpole and drag it to the desired size and orientation.
- Draw ‘ordinary’ propagators: Click on the starting point and drag to the end point.
- Delete objects: Click on the square (red, blue, green) corresponding to the object you want to delete. When deleting a vertex, the propagators adjacent to this vertex are also deleted. When deleting a propagator, the corresponding label is also deleted.

In the default setup, the left mouse button moves objects, the middle mouse button draws tadpoles, and the right mouse button draws propagators. The assignment of the mouse button can be changed in the Mouse Button Assignment menu (Fig. 2, right panel).

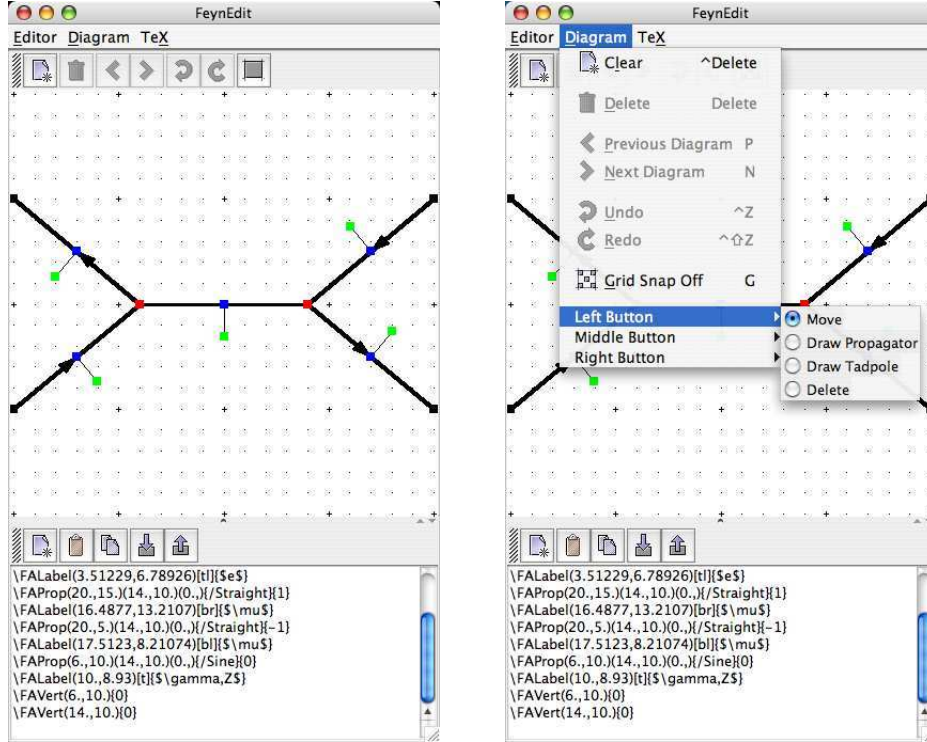


Figure 2: Left panel: The diagram pasted into FeynEdit and displayed. Right panel: The Mouse Button Assignment Menu in FeynEdit.

Following is an overview of all buttons:



Interpret the  $\text{\LaTeX}$  code and display the diagrams.



Turn the edited diagrams into  $\text{\LaTeX}$  code.



Turn on Grid Snap, i.e. restrict a dragged object's location to lie on a grid position ('quantize' the drag movement). This is to aid the aligned placement of items.



Turn off Grid Snap, i.e. allow objects to be dragged to arbitrary positions.



Move forward one diagram. For this button to become active, the  $\text{\LaTeX}$  code must contain more than one diagram, separated by `\FADiagram` directives.



Move back one diagram.



Undo last change.



Redo last undone change.



Clear panel.



Delete the currently marked entity.



Copy the contents of the  $\text{\LaTeX}$  panel into the copy-and-paste buffer.



Paste the current copy-and-paste buffer into the  $\text{\LaTeX}$  panel.

Note that details of the diagram, such as line attributes and label texts, are neither displayed by the editor, nor can they be specified for new tadpoles and propagators. Thus, for example, a gluon line is not drawn as a cycloid. This is largely for performance reasons (think of dragging a gluon line) and may be added in a future version. At any rate, it is not a serious deficit because that information can easily be added in the  $\text{\LaTeX}$  code. For instance, the line

```
\FAProp(0.,10.)(6.,10.)(0.,){/Straight}{0}
```

refers to a straight line. Simply changing “**Straight**” to “**Cycles**” makes it a gluon line. The next section gives all such details on the FeynArts  $\text{\LaTeX}$  style.

## 4 Graphics Primitives in feynarts.sty

The FeynArts style is included in a L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  document with

```
\usepackage{feynarts}
```

It makes three graphics primitives available with which Feynman diagrams can be drawn:

- `\FAProp` draws a propagator,
- `\FAVert` draws a vertex,
- `\FALabel` places a label.

In addition, it provides formatting/geometry directives:

- `\begin...end{feynartspicture}` delineates a sheet of Feynman diagrams,
- `\FADiagram` advances to the next diagram.

Since `feynarts.sty` emits direct PostScript primitives, the interpretation of which is non-standard across PostScript renderers, it is guaranteed to work only with `dvips`.

### 4.1 Geometry

A single Feynman diagram is always drawn on a  $20 \times 20$  canvas. Several such canvasses are combined into a rectangular sheet which can optionally carry a title. See Fig. 3 for illustration. Such a sheet of Feynman diagrams is enclosed in a `feynartspicture` environment in L<sup>A</sup>T<sub>E</sub>X:

```
\begin{feynartspicture}(s_x,s_y)(n_x,n_y)
...
\end{feynartspicture}
```

This sheet has a size of  $s_x \times s_y$  (in units of L<sup>A</sup>T<sub>E</sub>X's `\unitlength`) with room for  $n_x \times n_y$  Feynman diagrams.  $n_y$  need not be an integer and the extra space implied by the fractional part is allocated at the top for the sheet label.

Note that it is not possible to distort the aspect ratio of a Feynman diagram. If the ratio  $n_x/\lfloor n_y \rfloor$  is chosen different from the ratio  $s_x/s_y$ , the sheet will fit the smaller dimension exactly and be centered in the larger dimension.

Inside the `feynartspicture`, the macro

```
\FADiagram{dtitle}
```

advances to the next diagram, which has the title *dtitle*. The size of *dtitle* can be changed by redefining `\FADiagramLabelSize` with one of the usual L<sup>A</sup>T<sub>E</sub>X font-size specifiers, e.g.

```
\renewcommand\FADiagramLabelSize{\scriptsize}
```

The default size is `\small`.

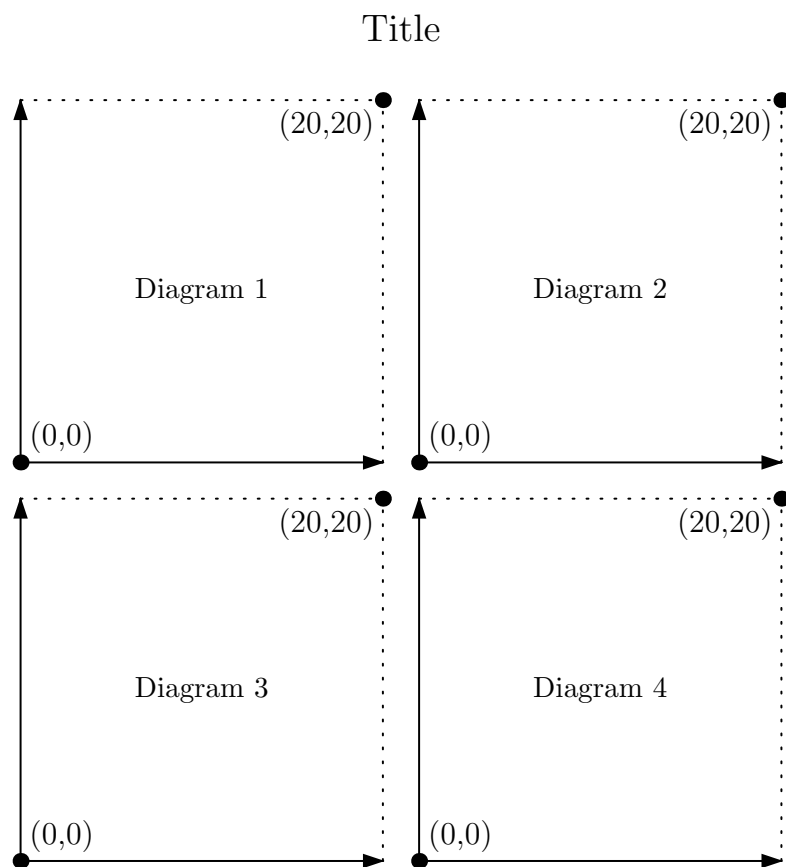
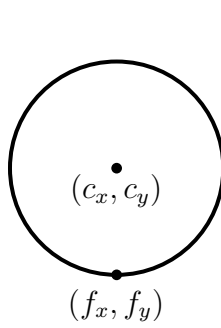
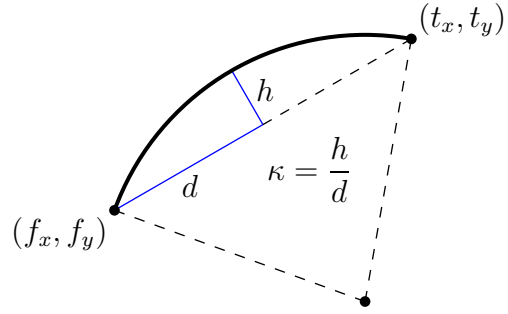


Figure 3: Geometry of a  $2 \times 2$  feynartspicture sheet.



`\FAProp(f_x, f_y)(f_x, f_y)(c_x, c_y){g}{a}`



`\FAProp(f_x, f_y)(t_x, t_y)(\kappa,){g}{a}`

The latter two arguments,  $g$  and  $a$ , respectively determine line and arrow style:

	$g = \text{/Straight}$		$a = 0$
	$g = \text{/ScalarDash}$		$a = 1$
	$g = \text{/GhostDash}$		$a = -1$
	$g = \text{/Sine}$		
	$g = \text{/Cycles}$		

Note the slash (/) in the line-style directive which is necessary because the directive is directly handed to the PostScript interpreter.

Figure 4: The geometrical layout of a propagator and the corresponding arguments of the `\FAProp` macro. Left: tadpole-type propagators (coincident initial and final vertex). Right: ‘ordinary’ propagators (non-coincident initial and final vertex).

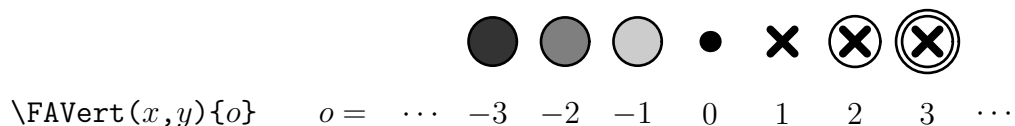
## 4.2 Propagators

All propagators are circular arcs in the FeynArts style. This includes conceptually the straight line as the infinite-radius limit. Propagators furthermore come in two variants: tadpole propagators, where the initial and final vertex coincide, and ‘ordinary’ propagators with distinct initial and final vertex. This distinction is necessary because the information that has to be stored is different for the two cases. The arguments of the `\FAProp` macro and their geometrical meaning are shown in Fig. 4 for both variants.



### 4.3 Vertices

Vertices mark the points where propagators join. Each propagator has a counter-term order associated with it.



The FeynArts notion is that positive  $o$  refer to counter-terms of order  $o$ , negative  $o$  to 1PI insertions of  $|o|$ -loop order, and zero to ordinary (tree-level) vertices.

### 4.4 Labels

Labels are usually associated with propagators, but can in principle be set anywhere. They have a pair of coordinates and an alignment, given in the usual  $\text{\TeX}$  manner, i.e. a code of up to two letters for vertical and horizontal alignment:  $\{\text{t} = \text{top}, (\text{empty}) = \text{center}, \text{b} = \text{bottom}\} \otimes \{\text{l} = \text{left}, (\text{empty}) = \text{center}, \text{r} = \text{right}\}$ , e.g. `[t]` or `[rb]`. The alignment makes it possible to change the label's text, in particular its width, without having to reposition the coordinates.

`\FALabel(x,y)[align]{text}`

## 5 Summary

FeynEdit is a Java program for editing Feynman diagrams. It uses the  $\text{\LaTeX}$  representation of FeynArts for input and output. Diagrams are entered into and retrieved from the editor through cut-and-paste with the mouse. This makes it unnecessary to first save the diagrams one wants to edit in a separate file.

The editor does not show details such as line styles and the actual labels. This is currently done for performance reasons. With the FeynArts  $\text{\LaTeX}$  format, it is not difficult to change these elements later, however.

The package is open source and is licensed under the LGPL. It can be downloaded from <http://www.feynarts.de> and runs on all platforms with a Java interpreter.

## Acknowledgements

We thank M. Schmaus for significant help with the Java programming.

## References

- [1] T. Hahn, *Comp. Phys. Commun.* **140** (2001) 418 [hep-ph/0012260].