

# The feyn font

Norman Gray  
(<https://nxg.me.uk>)

Version 0.4.2, 2022 July 18

This document describes the font *feyn*, which can be used to produce relatively simple Feynman diagrams within equations in a  $\text{\LaTeX}$  document. The package and font aims to be capable of producing all of the diagrams of one- and two-loop QED and QCD, plus  $\phi^4$  theory.

The other Feynman diagram package which exists is Thorsten Ohl's *feynmf/feynmp* package. That works by creating Metafont or MetaPost figures using a preprocessor. It's more general than this package, but is at its best when creating relatively large diagrams, for figures. In contrast, the present system consists of a carefully-designed font with which you can easily write simple diagrams, within equations or within text, in a size matching the surrounding text size.

**Contents:**    §1 Description, p2; §1.1 More complicated diagrams, p5; §2 Examples, p6; §3 Installation, p14; §3.1 Installing by hand, p14; §4 Licence, p15;

## Status

The propagators and vertices which are implemented are those of one- and two-loop QED and QCD, plus  $\phi^4$  theory, plus a few extras which commonly appear in non-figure displays (the practical definition of 'commonly appear' is 'which I wanted' or 'which people asked me for'). I'd be delighted to add others which folk feel would be useful, as long as they're still in the rather simple scope of the font.

# 1 Description

You use the font by including the package `feyn.sty`, as in

```
\usepackage{feyn}
```

If you use the ‘`amsmath`’ package, you should load that before loading the ‘`feyn`’ package.<sup>1</sup>

The characters in the font are shown in table 1. These are shown in `textstyle` size, but all characters are also available in `displaystyle` size. All of the `displaystyle` characters are additionally offset vertically up to the math axis, which has height  $a$ . All the dimensions are shown as multiples of the length of a module which is the length of the short fermion. The  $\circ$  shows the reference point of each character. The proper vertex and the complete vertex are specified in terms of a ‘blob-radius’ which is about half a module. They have a width of 2 radii, and a height and depth of 1 radius.

Each of the characters marked with a † in the table has two arrowed variants, so that `\feyn{fA + gV}` produces  $\longrightarrow + \curvearrowright$ . The ‘A’ variants produce arrows pointing rightwards or upwards, and the ‘V’ variants arrows pointing leftwards or downwards. The characters marked with a ‡ also have an upside-down variant, thus ‘flu’ is an upside-down variant of the fermion loop, and ‘fluV’ an arrowed variant.

Note that the fermion, gluon and ghost half-loop characters are slightly shorter than the height you might expect. These characters are deliberately slightly squashed partly because that makes them look slightly better (in my opinion!) in a one-loop self-energy diagram, but also so that the ‘gl glu’ or ‘fl flu’ loop in a two-loop diagram looks better. The loop formed by the pair of ‘g1 g2’ is the size you might expect. See section 2 below for some illustrations of this.

The unassigned positions in the *feyn* font are filled with an obviously wrong dummy character, so that `$_\feyn{A}$`, for example, produces  $\mathfrak{A}$ .

If you were to print out a font table, you would discover that many of the glyphs in the font can be obtained using single characters, rather than the ligatures mentioned in the table. Do not be tempted to use this as a shortcut, since the assignment of glyphs to character positions may be changed without warning from release to release, as long as the ligature maps remain valid.

The file `feyn.sty` additionally defines a number of macros, described here.

---

<sup>1</sup>Because of a slight peculiarity of the `amsmath` package, the active ‘!’ command described below will not work properly within `amsmath align` or `align*` environments unless the ‘!’ is made active globally. This is done by default if the ‘`amsmath`’ package is loaded first, and not done by default otherwise. If you have to override this behaviour for some very arcane reason, then you can force the global or local declaration of ‘!’ using the `globalbang` and `noglobalbang` options to the `\usepackage{feyn}` command.
































Character	Name	Width	Height	Depth
	$f$ fermion <sup>†</sup>	2	0	0
	$fs$ short fermion	1	0	0
	$fl$ fermion loop <sup>†‡</sup>	0	2–	0
	$flS$ fermion loop (small) <sup>†‡</sup>	0	1	0
	$flo$ fermion loop (for $\phi^4$ theory) <sup>†</sup>	0	2	0
	$fu$ upward fermion <sup>†</sup> (45°)	$\sqrt{2}$	$\sqrt{2}$	0
	$fd$ downward fermion <sup>†</sup>	$\sqrt{2}$	$\sqrt{2}$	0
	$fv$ vertical fermion <sup>†</sup>	0	2	0
	$f0$ spacer	2	0	0
	$fs0$ short spacer	1	0	0
	$g$ gluon/photon <sup>†</sup>	2	0	0
	$gl$ gluon loop <sup>†‡</sup>	0	2–	0
	$glB$ gluon loop (big) <sup>†</sup>	0	2.67	0
	$glS$ gluon loop (small) <sup>†</sup>	0	1.33	0
	$g1, g2$ gluon loop, 1st & 2nd quadrants <sup>†</sup>	0	2	0
	$g4, g3$ gluon loop, 4th & 3rd quadrants <sup>†</sup>	0	0	-2
	$gu$ upward gluon <sup>†</sup> (45°)	$\sqrt{2}$	$\sqrt{2}$	0
	$gd$ downward gluon <sup>†</sup>	$\sqrt{2}$	$\sqrt{2}$	0
	$gv$ vertical gluon <sup>†</sup>	0	2	0
	$m$ massive fermion <sup>†</sup>	2	0	0
	$ms$ short massive fermion	1	0	0
	$h$ ghost <sup>†</sup>	2	0	0
	$hs$ short ghost	1	0	0
	$hu$ upward ghost <sup>†</sup> (45°)	$\sqrt{2}$	$\sqrt{2}$	0
	$hd$ downward ghost <sup>†</sup>	$\sqrt{2}$	$\sqrt{2}$	0
	$hl$ ghost loop <sup>†‡</sup>	0	2–	0
	$x$ counterterm vertex	0	$a$	$-a$
	$p$ proper vertex	-	-	-
	$P$ proper vertex (variant)	-	-	-
	$c$ complete vertex	-	-	-
	$a$ arrow <sup>†</sup>	0	0	0

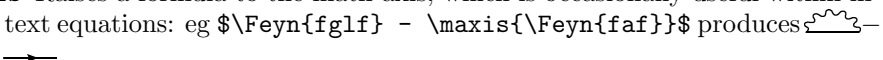
Table 1: The characters in font *feyn*, and their reference points, marked with circles. Characters marked <sup>†</sup> have arrowed variants indicated with ‘A’ and ‘V’ suffixes; and characters marked <sup>‡</sup> have an upside-down variant indicated with a ‘u’ suffix. The ‘u’ must come before the ‘A’ or ‘V’. Character ‘a’ has an ‘arrowed variant’ in the sense that ‘aV’ produces a backward arrow. A length of ‘2–’ indicates a length slightly less than two units, for aesthetic reasons.

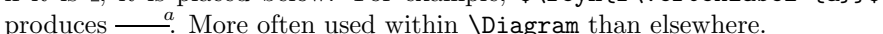
`\feyn` Selects the diagram font. This must be used within math mode. `\Feyn` selects the text-mode diagram font.

`\momentum{ch}{text}` sets the character ‘ch’, which may be a ligature, and puts the given text at the recommended annotation position for that character. `\momentum[pos]{ch}{text}` is the same, except that the optional argument indicates the position relative to the annotation position. It must be one of ‘top’, ‘urt’, ‘lrt’, ‘bot’, ‘llft’ or ‘ulft’. This is rarely used as such, since the `\feyn` and `\Feyn` commands make ‘!’ a temporarily active character, and define it to be `\momentum`. See section 2 for examples and further discussion.

`\belowl{}` A common pattern in diagrams is to have a pair of incoming or outgoing propagators. If the lower one of this pair is enclosed within `\belowl{...}`, then it is set so that it occupies zero horizontal space, and moved leftwards and downwards, to fit under the upper of the pair of propagators. The command `\belowr{...}` is the same, except that the argument is moved rightwards. See examples below.

`\Diagram{}` This is for building more complicated diagrams. It takes one argument, which is like the contents of an `{array}` environment—a series of formulae separated by `&`’s and `\\`. See below for an example. The result is a box on the math-axis.

`\maxis` Raises a formula to the math-axis, which is occasionally useful within in-text equations: eg `$_{\Feyn{fglf}} - \maxis{\Feyn{faf}}$` produces .

`\vertexlabel{p}{text}` Allows you to label a vertex. If the first parameter  $p$  is  $\wedge$ , the *text* is placed above the point at which the command is given, if it is  $\_$ , it is placed below. For example, `$_{\feyn{f}\vertexlabel^{\wedge}{a}}$` produces . More often used within `\Diagram` than elsewhere.

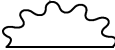

`\feynstrut{h}{d}` For use within an `(eqn)array` environment, or the like. It modifies the control sequence `\strut` to be a strut of height  $h$  and depth  $d$  modules, which can therefore be used to space the array out. Note that it modifies `\strut`, rather than being the strut itself.

`\annotate{x}{y}{text}` Puts the *text*, between dollars, in a zero width box at offset  $(x, y)$  modules from the position of the `\annotate` command (which is generally the first command(s) after entering math-mode). Because the *text* is in math-mode, anything that is not maths should be in an `\mbox`. You will probably use this only rarely, as the `\momentum` command supports most of the annotation you will need.

The *feyn* font is modular, in the sense that all the dimensions are in terms of a module of 10pt, and all sizes given below are in units of modules. As well as 10pt, the font has versions for 11pt, 12pt, 18pt and 24pt text, and the most appropriate one is selected depending on the text size declared (or defaulted) in

the `\documentclass` command. This range of sizes seems to be appropriate for the sizes required, but the author would welcome comments on this. The distribution contains a sample file using the 18pt *feyn* font in a *foiltex* document.

You draw a diagram by going into math-mode (between  $\dots$ ), and selecting the diagrams font by the font-changing command `\feyn{}` (exactly as you might use `\mathrm` in math-mode). With a couple of exceptions, all the characters are obtained by typing a single letter, or a couple of letters which form a ligature, so that the letter *f* in the diagrams font ( $\mathsf{\feyn{f}}$ ) produces a fermion of length 2 (modules), and *fs* produces a short fermion of length 1. As usual in maths mode, space characters are ignored, so you can add whitespace as required to make the expressions more legible. Some of the more heavily used characters are available in both display size and text size, with the text-size version invoked

by `\Feyn{...}`, so that `\feyn{fglf}` gives  and `\Feyn{fglf}` gives . The displaystyle characters are all on the math-axis, the textstyle ones are on the text line.

The display and text sizes of diagrams (that is, as produced by `\feyn` and `\Feyn` respectively) are intended to be used within equations, and they are sized accordingly. When diagrams have significantly complicated annotation, however, this size can look unhelpfully cramped, and it is for this case that there is an ‘extended’ set of diagram characters, which are approximately double the size, and are available, in `\displaystyle` only, with the `\FEYN` and `\DIAGRAM` macros. These are used exactly like the `\feyn` and `\Diagram` macros. Note that the line width, the module used by the (rare) `\annotate` command, and the math axis, do not change from the sizes they have for the `\feyn` command. That is, this font is *not* the appropriate one to use if you simply want a larger font size – in that case, simply increase the L<sup>A</sup>T<sub>E</sub>X font size and the *feyn* font will change in size accordingly. See below for an example of the use of `\FEYN`.

## 1.1 More complicated diagrams

The *feyn* font is designed so that simple diagrams are simple to produce, and are as far as possible writable as a single line of characters. There are a couple of more complicated diagrams which are drawable without too much effort.

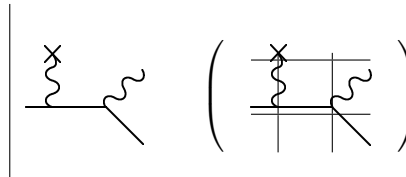
A common pattern in diagrams is to have a pair of incoming or outgoing propagators. If the lower one of this pair is enclosed within `\belowl{...}`, then it is set so that it occupies zero horizontal space, and moved leftwards and downwards, to fit under the upper of the pair of propagators. For example the sequence `\feyn{f gu \belowl{fd}}` would put the downward fermion directly below the upward gauge boson. See below for this and other examples.

The second way of composing more complicated diagrams is to use the `\Diagram` macro. This allows you to lay out a diagram in an array-like environment, each cell of which is set within a `\feyn{...}` macro. On the right is the same diagram drawn with lines indicating the boundaries between the cells.

```

\Diagram{   & x \\
           fs & gv f & gu \\
           &      & fd }

```



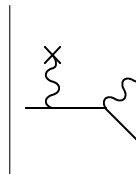
The `\Diagram` and `\DIAGRAM` macros take an optional argument of either ‘centre’ or ‘bottom’: with ‘centre’, the diagram is centred on the math axis, with ‘bottom’ the reference point of the diagram is the reference point of the bottom-most row of the array. The default is ‘centre’.

Before the introduction of the `\belowl` macro in *feyn* v0.4, this was the only way to produce this pair of angled propagators one on top of the other. With `\belowl`, however, an alternative way of producing this diagram is as shown below.

```

\feyn{fs
  \Diagram[bottom]{x\gv}
  f gu\belowl{fd}}

```



Diagrams much more complicated than this are probably better dealt with using Thorsten Ohl’s *feynmf/feynmp* package.

## 2 Examples

A simple propagator:

```

\feyn{\vertexlabel^a
      !{fA}p \vertexlabel^b}
= \displaystyle
  \frac{i\delta^{ab}}
    {\pslash-m_0}

```

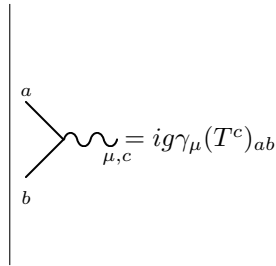
$$\begin{array}{c}
 a \quad p \quad b \\
 \longrightarrow \\
 \end{array} = \frac{i\delta^{ab}}{\not{p} - m_0}$$

The `\Diagram` command lays out its arguments as an array.

```

\Diagram{\vertexlabel^a \\\
fd \\\
& g\vertexlabel_{\mu,c} \\\
\vertexlabel_b fu\\
}
= ig\gamma_{\mu} (T^c)_{ab}

```

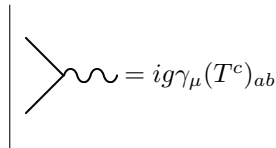


The `\Diagram` is here necessary only for the vertex labels. Without those, it would be as simple as:

```

\feyn{fd \belowl{fu} g
= ig\gamma_{\mu} (T^c)_{ab}}

```

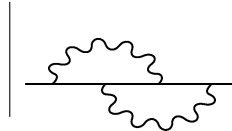


An assortment of two-loop diagrams:

```

\feyn{fs f gl f glu f fs}

```



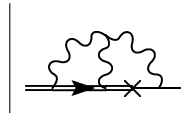
This is a “short-fermion fermion gluon-loop fermion gluon-loop-upside-down fermion short-fermion”.

We can include vertical gluons:

```

\feyn{ms mA g1 gv g2 ms x f}

```

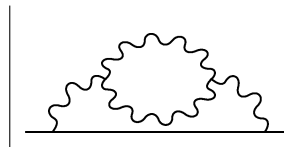


Note, in this example, that the gluon loop is formed by a pair of ‘g1 g2’ characters. This is because the ‘gl’ half-loop character is deliberately slightly squashed, and wouldn’t meet the vertical gluon properly. The half-loop character is squashed (as are the fermion and ghost half-loops) partly because it looks slightly better in a one-loop self-energy diagram, but also so that the ‘gl glu’ two-gluon loop in a two-loop diagram looks better:

```

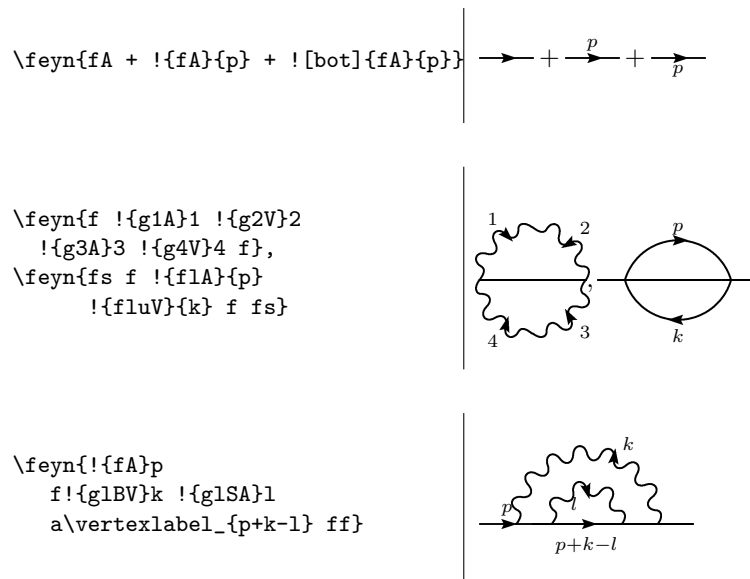
\Diagram{&gl glu\\
fsf g1 f&f g2 ffs \\\}

```

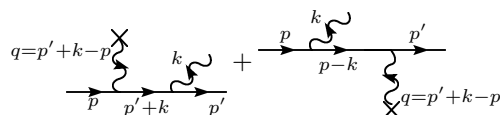
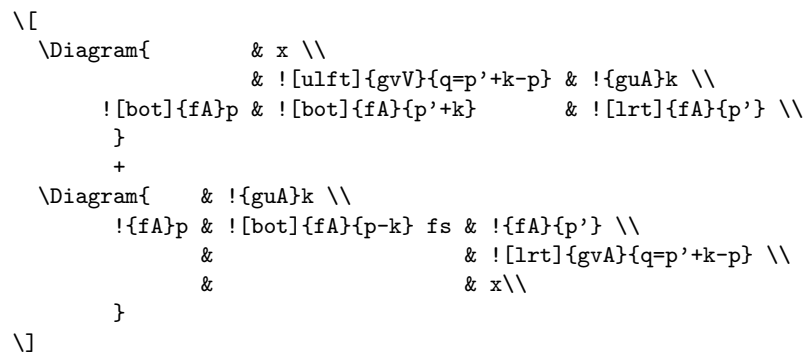


The diagrams for 'hl hlu' or 'flA fluV', for example, are very similar.

The `\momentum` command (more usually called with simply the '!' character) allows you to annotate a propagator with a momentum or a particle label. For example



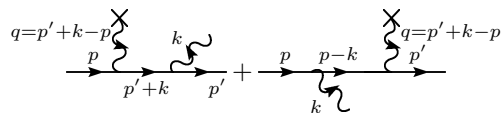
Bremsstrahlung:





And now the same, with `\belowl` and `\belowr`:

```
\[
\def\stack#1#2{\Diagram[bottom]{#1\#2}}
\feyn{
  !{fA}p
  \stack{x}{![ulft]{gvV}{q=p'+k-p}}
  ![bot]{fA}{p'+k}
  \stack{![guA]{k}{![lrt]{fA}{p'}}}
  +
  !{fA}p
  \belowr{![llft]{gdA}{k} !{fA}{p-k}}
  fs
  \stack{x{![gvV]{q=p'+k-p}} !{fA}{p'}}}
\]
```



The `\stack` macro isn't defined by the *feyn* package, but it is useful enough that you might want to define it as one of your own macros.

The expression for the heavy-fermion self-energy in the OPE is:

```

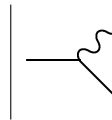
\def\bracket#1{\langle #1 \rangle}
\def\stack#1#2{\Diagram[bottom]{#1\\#2}}
\begin{eqnarray*}
-i\Sigma_{\rm ope} &=&
\left[ \text{\feyn{faf + fsfglfs + \cdots}} \right] 1 \\
&& \text{\feynstrut{1}{1}\strut} \\
&& \quad \text{\quad} \left[ \text{\feyn{ fxfsglfs0xf + \cdots}} \right] \\
&& \quad \text{\quad} \text{\bracket{\overline{\psi} M \psi}} \\
&& \quad \text{\quad} \left[ \text{\feyn{fs \stack x{gv} fs \stack x{gv} fs} + \cdots} \right] \\
&& \quad \text{\quad} \text{\bracket{G_{\mu\nu}^a G_{\mu\nu}^a}} \\
\end{eqnarray*}

```

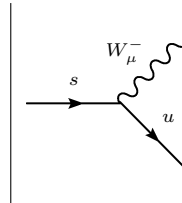
$$\begin{aligned}
-i\Sigma_{\text{ope}} = & \left[ \text{---} + \text{---} + \dots \right] 1 \\
& + \left[ \text{---} \times \times \text{---} + \dots \right] \langle \bar{\psi} M \psi \rangle \\
& + \left[ \text{---} \times \times \text{---} + \dots \right] \langle G_{\mu\nu}^a G_{\mu\nu}^a \rangle
\end{aligned}$$

Now for two examples using `\FEYN`:

`\feyn{f gu \belowl{fd}}`



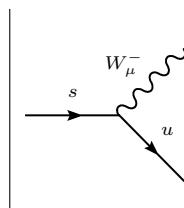
`\FEYN{
!{fA}s !{gu}{W^-_\mu}
\belowl{!{fdA}u}
}`



```

\DIAGRAM{
  !{fA}s &!{gu}{W^-_\mu}\\
  & !{fdA}u
}

```



The Feynman Rules are as follows:

---

```

\def\arraystretch{3} \arraycolsep=0.2cm
\[ \begin{array}{rcl}
\feyn{\vertexlabel^a !{fA}p \vertexlabel^b} % quark prop
&=& \displaystyle
\{i\delta^{ab} \over \pslash- m_0\} \\\

\Diagram{\vertexlabel^a \\\ fd \\\ & g\vertexlabel_{\mu,c} \\\
\vertexlabel_b fu\}\} % 3-point vertex
&=& \displaystyle
ig\gamma_{\mu} (t^c)_{ab} \\\

\feyn{\vertexlabel_{\mu, a} !{gA}k \vertexlabel_{\nu, b}} % gluon prop
&=& \displaystyle
\{-i\over k^2\} \left[g_{\mu\nu} + (a_0 - 1) \{k_{\mu} k_{\nu} \over k^2\}\right]
\delta^{ab} \\\

\Diagram{\vertexlabel^{\alpha,a} \\\
!{gdA}p \\\
& & !{gA}r \vertexlabel_{\gamma,c} \\\
!{guA}q \\\
\vertexlabel_{\beta,b}
} % 3-gluon vertex
&=& \displaystyle
-g f^{abc} (
g_{\beta\gamma} (q-r)_{\alpha} +
g_{\gamma\alpha} (r-p)_{\beta} +
g_{\alpha\beta} (p-q)_{\gamma}
) \\\

\feyn{\vertexlabel^a !hp \vertexlabel^b} % ghost prop
&=& \displaystyle
\{i \over p^2\} \delta_{ab} \\\

\Diagram{\vertexlabel^b \\\ hd \\\ & g\vertexlabel^{\mu,a} \\\
\vertexlabel_c hu\}\} % ghost-gluon vertex
&=& \displaystyle
-g f^{abc} p_{\mu}
\quad\raise 1ex\hbox{\vtop{\hsize=15em
where $\mathbf{p}$ is the momentum of the outgoing positive energy
ghost\par}}
\end{array} \]

```

---

--

\_\_\_\_\_

$$\begin{aligned}
 \text{---} \bigcirc \text{---} &= \text{---} \rightarrow \text{---} + \text{---} \textcircled{\text{X}} \text{---} + \text{---} \textcircled{\text{X}} \textcircled{\text{X}} \text{---} + \dots & (1) \\
 &= \sum_{n=0}^{\infty} \text{---} \rightarrow \textcircled{\text{X}} \rightarrow \text{---} )^n & (2) \\
 &= \frac{\text{---} \rightarrow \text{---}}{1 - \textcircled{\text{X}} \rightarrow \text{---}}. & (3)
 \end{aligned}$$

### 3 Installation

The nominal distribution point for the Feyn package is CTAN, under `fonts/feyn`<sup>2</sup>.

The feyn package is included in the TeXLive distribution<sup>3</sup>, so you may have it on your machine already. If not, or if you need a more up-to-date version, read on.

Since the package is on CTAN, it should be easily installed and updated using the tools which support this in your TeX distribution. For TeXLive, that's 'tlmgr'<sup>4</sup>, and for MikTeX see the maintenance section of the manual<sup>5</sup>.

#### 3.1 Installing by hand

If, for some reason, you need to install the package by hand, then you should be able to find generic instructions for installing L<sup>A</sup>T<sub>E</sub>X files at <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=installthings>.

In brief: If you have not done so already, you need to extract the package file from the distribution file using `latex feyn.ins`. The resulting `feyn.sty` should be installed in the usual location for style files – that is, something like `texmf.local/tex/latex/feyn.sty`. The Metafont source files `*.mf` should be installed in a directory with a location such as `texmf.local/fonts/source/feyn/`. That should be all you need to do as regards the installation of the fonts – if your T<sub>E</sub>X setup is installed correctly, then the usual font-generation commands (such as `mktexpk` for example) should work as normal, and these will usually be invoked by default when L<sup>A</sup>T<sub>E</sub>X or pdfL<sup>A</sup>T<sub>E</sub>X is run.

If you are using a T<sub>E</sub>X system based on TeXLive (which is true of most Unix installations, and most macOS installations), then you can determine the style-file and Metafont search paths using the commands

```
% kpsepath tex
% kpsepath mf
```

respectively. After you have installed them, you will probably need to give the command `mktexlsr` to rebuild the paths database, and you can confirm that the files are findable with the commands `kpsewhich feyn.sty` and `kpsewhich feyn10.mf`.

---

<sup>2</sup><https://ctan.org/pkg/feyn>

<sup>3</sup><http://www.tug.org/texlive/>

<sup>4</sup><https://www.tug.org/texlive/tlmgr.html>

<sup>5</sup><https://docs.miktex.org/faq/maintenance.html>

## 4 Licence

The following is the text of the BSD 2-clause licence.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. in no event shall the copyright holder or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

## Acknowledgements

Thanks to Purnendu Chakraborty, Fabio Correa, Bryan Chen, Hoang Nghia Nguyen and Luciano Pandola for suggestions and testing (and encouragement) at various times.