Guide to FeynHelpers

A collection of interfaces between FeynCalc and other HEP-related tools

Vladyslav Shtabovenko

July 26, 2022

This documentation snapshot was generated on **2022-07-26 20:06:45 UTC** from the commit **f294de**

Contents

1 Useful information

1.1 Citations

If you use FeynHelpers in your research, please cite

- V. Shtabovenko, *FeynHelpers: Connecting FeynCalc to FIRE and Package-X*, Comput. Phys. Commun., 218, 48-65, 2017, arXiv:1611.06793

Apart from that, you should also provide the proper academic attribution to the authors of the tools that you are using via the interfaces provided by FeynHelpers. Please check out the respective websites for the proper citation instructions.

The tools that should be cited are

- FERMAT if you are using **`Fer*`** functions
- FIRE if you are using **`FIRE*`** functions
- Package-X if you are using **`PaX*`** functions
- pySecDec if you are using **`PSD*`** functions
- LoopTools if you are using **`LTools*`** functions
- QGRAF if you are using **`QG*`** functions

1.2 Installation

The installation of FeynHelpers from scratch is a two step process. Apart from installing the actual FeynCalc addon (step 1) it is also necessary to set up the tools to which FeynHelpers provides FeynCalc interfaces (step 2).

Due to the fact that these tools are written in different programming languages and usually must be compiled on user's machine, it is not possible to fully automatize this second step. Instead, we expect the user to install the relevant tools manually.

In the following we provide some relevant instructions but without any warranty that this will work on your machine. *If you are experiencing issues setting up a particular tools please contact the corresponding developer team.*

1.2.1 FeynHelpers

Automatic installation

Run the following instruction in a Kernel or Notebook session of Mathematica to install the stable version

```
Import["https://raw.githubusercontent.com/FeynCalc/feynhelpers/master/install.m"]
InstallFeynHelpers[]
```

If you like the bleeding edge and you are already using the development version of FeynCalc, you can also install the development version of FeynHelpers

```
Import["https://raw.githubusercontent.com/FeynCalc/feynhelpers/master/install.m"]
InstallFeynHelpers[InstallFeynHelpersDevelopmentVersion->True]
```

Manual installation

Create a directory *FeynHelpers* inside

```
FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns"}]
```

and put the source code there.

1.2.2 Fermat

You can download FERMAT binaries for Linux or macOS from the developer's website.

Linux

Copy the directory **`ferl64`** to **`FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns", "FeynHelpers", "ExternalTools", "Fermat"}]`**.

macOS

Copy the directory **`ferm64`** to **`FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns", "FeynHelpers", "ExternalTools", "Fermat"}]`**.

Windows

Currently there is no native Windows version of FERMAT. The Linux version appears to be usable via WSL, but currently there is no support for that in FeynHelpers.

1.2.3 FIRE

You can download the source code of FIRE from the developer's website. The content should be extracted to **`FileNameJoin[{$UserBaseDirectory, "Applications", "FIRE"}]`**.

Linux

The instructions for compiling FIRE from source on Linux are provided here.

macOS

The instructions for compiling FIRE from source on macOS can be found here.

Windows

There is no native Windows port of FIRE. It should be possible to compile FIRE on WSL with an Ubuntu installation, but currently there is no support for that in FeynHelpers.

1.2.4 LoopTools

On the developer's website you can download not only the source code but also precompiled binaries

Linux or macOS

Copy the self-compiled or precompiled MathLink executable **`LoopTools`** to **`FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns", "FeynHelpers", "ExternalTools", "LoopTools"}]`**.

Windows

Rename the self-compiled or precompiled MathLink executable **`LoopTools.exe`** to **`LoopTools`** and copy it to **`FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns", "FeynHelpers", "ExternalTools", "LoopTools"}]`**.

1.2.5 pySecDec

The installation instructions for pySecDec can be found here. FeynHelpers does not require pySecDec to *generate* the corresponding Python scripts. However, in order to actually compute the given loop integrals one has to run those scripts, which is possible only when pySecDec is installed.

Linux or macOS

It should be possible to install pySecDec via pip automatically.

Windows

It could be possible to set up pySecDec on WSL, but currently it is unclear whether this can work.

1.2.6 Package-X

Package-X is not being developed or even maintained anymore. However, the author of the tool, Hiren H. Patel, kindly gave us permission to ship **`OneLoop.m`** (part of Package-X containing library of analytic results for Passarino-Veltman functions) together with FeynHelpers.

Therefore, no separate installation of Package-X is needed.

Please notice that you are using this library on your own risk. The lack of maintenance means that newly discovered bugs are not going to be fixed and there is no guarantee that it will continue to work with newer Mathematica versions.

1.2.7 QGRAF

You can download the source code of QGRAF from the developer's website.

Linux and macOS

The compilation instructions can be found in the section "Compiling" of the manual **`qgraf-3.x.y.pdf`** inside the tarball. For example, on a Linux system equipped with a GNU Fortran compiler something like

```
gfortran qgraf-3.6.3.f08 -o qgraf
```

should do the job. Having compiled the program you need to move the binary file **`qgraf`** to **`FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns", "FeynHelpers", "ExternalTools", "QGRAF", "Binary"}]`**.

Windows

To extract the source code tarball of QGRAF you need a tool that can deal with tar.gz archives, e.g. 7-zip.

To build QGRAF from source you need a FORTRAN compiler. You can use the MinGW compiler via the *MinGW-W64 Online Installer* (**`MinGW-W64-install.exe`**). When the **`Settings`** page appears in the installation wizard, change **`Architecture`** from **`i686`** to **`x86_64`**.

Unfortunately, as of June 2022 the installer is broken and fails with the error message "file was downloaded incorrectly". A possible workaround is described here. When you reach the **`Installation folder`** page in the installation wizard, open

```
C:\Users\YOUR_USER_NAME\AppData\Local\Temp\gentee
```

and drop there the file **x86_64-8.1.0-release-win32-seh-rt_v6-rev0.7z** that you can donwload from SourceForge. This skips the broken process of downloading the file by the installer and should get you through the installation.

Finally, open **MiniGW-W64 project** -> **Open Terminal** via the start menu. Go to the folder where you extracted the source code of QGRAF and compile it with **gfortran.exe -static qgraf-3.x.y.f08 -o qgraf.exe**, where **x** and **y** denote the current version numbers.

Run **qgraf.exe** to make sure that it works properly.

Finally, move **qgraf.exe** to **FileNameJoin[{$UserBaseDirectory, "Applications", "FeynCalc", "AddOns", "FeynHelpers", "ExternalTools", "QGRAF", "Binary"}]**.

1.3 Tensor reduction with Fermat

One of the most useful functions exposed by the Fermat interface is **FerSolve** that is vastly superior to Mathematica's **Solve** when dealing with very large symbolic systems of equations.

A typical situation where one needs to solve such equations is the derivation of tensor decomposition formulas. To this aim FeynCalc's **Tdec** can directly use **FerSolve**, once FeynHelpers is loaded. One just needs to set the option **Solve** to **FerSolve**.

The following example calculates tensor reduction formula for a rank 6 integral with 2 loop momenta and two external momenta. The Fermat part requires only 40 seconds on a modern laptop to solve the corresponding 52×52 symbolic system.

```
Tdec[{{p1, mu1}, {p1, mu2}, {p1, mu3}, {p1, mu4}, {p2, mu5}, {p2, mu6}},
{Q1, Q2}, Solve -> FerSolve, UseTIDL -> False, FCVerbose -> 1]
```

2 Generic functions

2.1 FeynHelpersHowToCite

`FeynHelpersHowToCite[]` lists publications that should be cited when mentioning FeynHelpers in scientific works.

2.1.1 See also

Overview

2.1.2 Examples

```
FeynHelpersHowToCite[]
```

• V. Shtabovenko, "FeynHelpers: Connecting FeynCalc to FIRE and Package-X", Comput. Phys. Commun., 218, 48

Furthermore, remember to cite the authors of the tools that you are calling from FeynHelpers, which are

• FIRE by A. Smirnov, if you are using functions that begin with FIRE.

• Kira by the Kira collaboration, if you are using functions that begin with Kira.

• Package − X by H. Patel, if you are using functions that begin with PaX.

• Fermat by R. Lewis, if you are using functions that beging with Fer.

• QGRAF by P. Nogueira, if you are using functions that begin with QG.

• LoopTools by T. Hahn, if you are using functions that begin with LT.

• pySecDec by the SecDec collaboration, if you are using functions that begin with PSD.

2.2 $FeynHelpersDirectory

'**$FeynHelpersDirectory** specifies the location of FeynHelpers.

2.2.1 See also

Overview.

2.2.2 Examples

```
$FeynHelpersDirectory
```

/home/vs/.Mathematica/Applications/FeynCalc/AddOns/FeynHelpers

2.3 $FeynHelpersLoadInterfaces

$FeynHelpersLoadIntefaces is a debugging switch that can be used to disable the loading of particular interfaces contained in FeynHelpers.

2.3.1 See also

Overview.

2.3.2 Examples

2.4 $FeynHelpersVersion

$FeynHelpersVersion is a string that represents the version of FeynHelpers.

2.4.1 See also

Overview.

2.4.2 Examples

```
$FeynHelpersVersion
```

2.0.0

3 Fermat interface

3.1 FerImportArrayAsSparseMatrix

`FerImportArrayAsSparseMatrix[outFile, var, {dimX, dimY}]` is an auxiliary function that converts the Fermat array called **`var`** from file **`outFile`** to a **`SparseArray`** matrix with dimensions **`dimX`** and **`dimY`**.

3.1.1 See also

Overview.

3.1.2 Examples

3.2 FerMatrixToFermatArray

`FerMatrixToFermatArray[mat, varName]` is an auxiliary function that converts the matrix **`mat`** to a Fermat array named **`varName`**, where the latter must be a string.

The function returns a string that represents the matrix, a list of auxiliary variables (introduced to be compatible with the restrictions of Fermat) and a replacement rule for converting auxiliary variables back into the original variables.

3.2.1 See also

Overview.

3.2.2 Examples

```
FerMatrixToFermatArray[{{a, b}, {c, d}}, "mat"]
```

{Array mat[2,2];\n[mat]:=[[fv1, fv3, fv2, fv4]];, {fv1, fv2, fv3, fv4}, {fv1 → a, fv2 → b, fv3 → c, fv4 → d}}

3.3 FerCommand

`FerCommand[str, args]` is an auxiliary function that returns a Fermat command corresponding to **`str`** (possibly using arguments **`args`**) as a list of strings.

At the moment only a very limited set of Fermat instructions is implemented.

3.3.1 See also

Overview.

3.3.2 Examples

```
FerCommand["Quit"]
```

&q;

```
FerCommand["StopReadingFromTheInputFile"]
```

&x;

```
FerCommand["EnableUglyDisplay"]
```

&(U=1);

```
FerCommand["ReadFromAnInputFile", "myFile.txt"]
```

&(R='myFile.txt');

```
FerCommand["SaveToAnOutputFile", "myFile.txt"]
```

&(S='myFile.txt');

```
FerCommand["SaveSpecifiedVariablesToAnOutputFile", {x, y, z}]
```

!(&o, x, y, z);

```
FerCommand["ReducedRowEchelonForm", "mat"]
```

Redrowech(mat);

```
FerCommand["AdjoinPolynomialVariable", x]
```

&(J=x);

3.4 FerRunScript

FerRunScript[scriptFile] is an auxiliary function that runs the script **scriptFile** containing Fermat instructions. The script file is expected to be a valid Fermat code.

The option **FerPath** should point to an executable Fermat binary

3.4.1 See also

Overview, FerRowReduce, FerSolve.

3.4.2 Examples

3.5 FerRowReduce

FerRowReduce[mat] uses Fermat to obtain the row-reduced echelon form of matrix **mat**. An important difference to Mathematica's **RowReduce** is that Fermat does not assume all symbolic variables to be nonzero by default.

The location of script, input and output files is controlled by the options **FerScriptFile**, **FerInputFile**, **FerOutputFile**. When set to **Automatic** (default), these files will be automatically created via **CreateTemporary[]**. If the option **Delete** is set to **True** (default), the files will be deleted after a successful Fermat run.

3.5.1 See also

Overview, FerRowReduce.

3.5.2 Examples

The syntax of FerSolve is very similar to that of **Solve**

```
res1 = RowReduce[{{3, 1, a}, {2, 1, b}}]
```

$$\begin{pmatrix} 1 & 0 & a-b \\ 0 & 1 & 3b-2a \end{pmatrix}$$

```
res2 = FerRowReduce[{{3, 1, a}, {2, 1, b}}] // Normal

(*FerRunScript: Running Fermat.
FerRunScript: Done running Fermat, timing: 0.5115*)
```

$$\begin{pmatrix} 1 & 0 & a-b \\ 0 & 1 & 3b-2a \end{pmatrix}$$

```
res1 === res2
```

True

This is an example for Mathematica SE, where RowReduce assumes $a \neq 0$

```
RowReduce[{{1, a, 2}, {0, 1, 1}, {-1, 1, 1}}]
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
FerRowReduce[{{1, a, 2}, {0, 1, 1}, {-1, 1, 1}}] // Normal



(*FerRunScript: Running Fermat.
FerRunScript: Done running Fermat, timing: 0.1393*)
```

$$\begin{pmatrix} 1 & 0 & 2-a \\ 0 & 1 & 1 \\ 0 & 0 & 2-a \end{pmatrix}$$

3.6 FerSolve

FerSolve[eqs, vars] solves the system of linear equations **eqs** for the variables **vars** by calculating the row-reduced form of the corresponding augmented matrix using Fermat by R. Lewis.

The location of script, input and output files is controlled by the options **FerScriptFile**, **FerInputFile**, **FerOutputFile**. When set to **Automatic** (default), these files will be automatically created via **CreateTemporary[]**. If the option **Delete** is set to **True** (default), the files will be deleted after a successful Fermat run.

3.6.1 See also

Overview, FerRowReduce.

3.6.2 Examples

The syntax of FerSolve is very similar to that of **Solve**

```
FerSolve[{a x + y == 7, b x - y == 1}, {x, y}]

(*FerRunScript: Running Fermat.
FerRunScript: Done running Fermat, timing: 0.6790
FerSolve: Verifying the result.
FerSolve: Done verifying the result, timing: 0.000630*)
```

$$\left\{x \to \frac{8}{a+b}, y \to -\frac{a-7b}{a+b}\right\}$$

```
Solve[{a x + y == 7, b x - y == 1}, {x, y}]
```

$$\left\{\left\{x \to \frac{8}{a+b}, y \to -\frac{a-7b}{a+b}\right\}\right\}$$

```
FerSolve[{2 x + y c == 2, 4 x == c}, {x, y}]

(*FerRunScript: Running Fermat.
FerRunScript: Done running Fermat, timing: 0.1560
FerSolve: Verifying the result.
FerSolve: Done verifying the result, timing: 0.000496*)
```

$$\left\{x \to \frac{c}{4}, y \to -\frac{c-4}{2c}\right\}$$

```
Solve[{2 x + y c == 2, 4 x == c}, {x, y}]
```

$$\left\{\left\{x \to \frac{c}{4}, y \to -\frac{c-4}{2c}\right\}\right\}$$

However, for larger systems of equations **FerSolve** is superior to **Solve** owing to better algorithms implemented in Fermat

```
eqs = {M0 == (1 - f1 - f2 - f3 - f4 - f5 - f6 - f7 - f8 - f9 - f10 -
              f11 - f12 - f13 - f14 - f15 - f16)*m0,
     M1 == f1*m0 + (1 - f1 - f2 - f3 - f4 - f5 - f6 - f7 - f8 - f9 -
               f10 - f11 - f12 - f13 - f14 - f15 - f16)*m1,
     M2 == f1*m1 +
         f2*m0 + (1 - f1 - f2 - f3 - f4 - f5 - f6 - f7 - f8 - f9 - f10 -
               f11 - f12 - f13 - f14 - f15 - f16)*m2,
     M4 == f1*m3 + f2*m2 + f3*m1 + f4*m0,
     M5 == f2*m3 + f3*m2 + f4*m1 + f5*m0,
     M6 == f3*m3 + f4*m2 + f5*m1 + f6*m0,
     M7 == f4*m3 + f5*m2 + f6*m1 + f7*m0,
     M8 == f5*m3 + f6*m2 + f7*m1 + f8*m0,
     M9 == f6*m3 + f7*m2 + f8*m1 + f9*m0,
     M10 == f7*m3 + f8*m2 + f9*m1 + f10*m0,
     M11 == f8*m3 + f9*m2 + f10*m1 + f11*m0,
     M12 == f9*m3 + f10*m2 + f11*m1 + f12*m0,
     M13 == f10*m3 + f11*m2 + f12*m1 + f13*m0,
     M14 == f11*m3 + f12*m2 + f13*m1 + f14*m0,
     M15 == f12*m3 + f13*m2 + f14*m1 + f15*m0,
     M16 == f13*m3 + f14*m2 + f15*m1 + f16*m0};
```

```
vars = {f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14,
     f15, f16};
```

```
sol1 = FerSolve[eqs, vars];

(*FerRunScript: Running Fermat.
FerRunScript: Done running Fermat, timing: 11.21
FerSolve: Verifying the result.
FerSolve: Done verifying the result, timing: 0.3670*)
```

```
sol1[[1]]
```

$$\text{f1} \to -\frac{\text{M0 m1} - \text{m0 M1}}{\text{m0}^2}$$

3.7 FerInputFile

FerInputFile is an option for multiple functions of the Fermat interface. It specifies the location of the file containing the input for a Fermat calculation. If set to **Automatic** (default), a temporary file will be automatically created and removed after a successful evaluation.

3.7.1 See also

Overview, FerSolve, FerRowReduce, FerRunScript.

3.7.2 Examples

3.8 FerOutputFile

FerOutputFile is an option for various functions of the Fermat interface. It specifies the location of the file containing the output of Fermat calculation. If set to **Automatic** (default), a temporary file will be automatically created and removed after a successful evaluation.

3.8.1 See also

Overview, FerSolve, FerRowReduce, FerRunScript.

3.8.2 Examples

3.9 FerPath

FerPath is an option for **FerRunScript** and other functions of the Fermat interface and multiple other Fer* functions. It specifies the full path to the Fermat binary.

If set to Automatic, Fermat binaries are expected to be located in **FileNameJoin[{$FeynHelpersDirectory, "ExternalTools", "Fermat", "ferl6", "fer64"}]** and **FileNameJoin[{$FeynHelpersDirectory, "ExternalTools", "Fermat", "ferm6", "fer64"}]** for Linux and macOS respectively.

3.9.1 See also

Overview, FerSolve, FerRowReduce, FerRunScript.

3.9.2 Examples

3.10 FerScriptFile

FerScriptFile is an option for various functions of the Fermat interface. It specifies the location of the file containing instructions for Fermat. If set to **Automatic** (default), a temporary file will be automatically created and removed after a successful evaluation.

3.10.1 See also

Overview, FerSolve, FerRowReduce, FerRunScript.

3.10.2 Examples

4 Package-X interface

4.1 PaXEvaluate

PaXEvaluate[expr, q] evaluates scalar 1-loop integrals (up to 4-point functions) in **expr** that depend on the loop momentum **q** in **D** dimensions.

The evaluation is using H. Patel's Package-X.

4.1.1 See also

Overview, PaXEvaluateIR, PaXEvaluateUV, PaXEvaluateUVIRSplit.

4.1.2 Examples

```
FAD[{q, m}]
PaXEvaluate[%, q, PaXImplicitPrefactor -> 1/(2 Pi)^(4 - 2 Epsilon)]
```

$$\frac{1}{q^2 - m^2}$$

$$\frac{im^2}{16\pi^2\varepsilon} - \frac{im^2\left(-\log\left(\frac{\mu^2}{m^2}\right) + \gamma - 1 - \log(4\pi)\right)}{16\pi^2}$$

```
FAD[{l, 0}, {q + l, 0}]
PaXEvaluate[%, l, PaXImplicitPrefactor -> 1/(2 Pi)^(4 - 2 Epsilon)]
```

$$\frac{1}{l^2.(l+q)^2}$$

$$\frac{i}{16\pi^2\varepsilon} + \frac{i\log\left(-\frac{4\pi\mu^2}{q^2}\right)}{16\pi^2} - \frac{i(\gamma - 2)}{16\pi^2}$$

PaVe functions do not require the second argument specifying the loop momentum

```
PaVe[0, {0, Pair[Momentum[p, D], Momentum[p, D]], Pair[Momentum[p, D],
Momentum[p, D]]}, {0, 0, M}]
PaXEvaluate[%]
```

$$C_0\left(0, p^2, p^2, 0, 0, M\right)$$

$$\frac{1}{\varepsilon M - \varepsilon p^2} - \frac{\gamma - \log\left(\frac{\mu^2}{\pi M}\right)}{M - p^2} + \frac{\log\left(\frac{M}{M-p^2}\right)}{M - p^2} + \frac{M \log\left(\frac{M}{M-p^2}\right)}{p^2\left(M - p^2\right)}$$

4.2 PaXEvaluateUV

PaXEvaluateUV[expr, q] is like **PaXEvaluate** but with the difference that it returns only the UV-divergent part of the result.

The evaluation is using H. Patel's Package-X.

4.2.1 See also

Overview, PaXEvaluateIR, PaXEvaluate, PaXEvaluateUVIRSplit.

4.2.2 Examples

```
int = -FAD[{k, m}] + 2*FAD[k, {k - p, m}]*(m^2 + SPD[p, p])
```

$$\frac{2\left(m^2 + p^2\right)}{k^2.\left((k-p)^2 - m^2\right)} - \frac{1}{k^2 - m^2}$$

```
PaXEvaluateUV[%, k, PaXImplicitPrefactor -> 1/(2 Pi)^D, FCE -> True]
```

$$\frac{im^2}{16\pi^2 \varepsilon_{\text{UV}}} + \frac{ip^2}{8\pi^2 \varepsilon_{\text{UV}}}$$

Notice that with **PaVeUVPart** one can get the same result

```
res = PaVeUVPart[ToPaVe[int, k], Prefactor -> 1/(2 Pi)^D]
```

$$-\frac{i2^{1-2D}\pi^{2-2D}\left(2^{D+1}\pi^{D}m^2-(2\pi)^{D}m^2+2^{D+1}\pi^{D}p^2\right)}{D-4}$$

```
Series[FCReplaceD[res, D -> 4 - 2 EpsilonUV], {EpsilonUV, 0, 0}] // Normal
// SelectNotFree2[#, EpsilonUV] & // ExpandAll
```

$$\frac{im^2}{16\pi^2\varepsilon_{\text{UV}}}+\frac{ip^2}{8\pi^2\varepsilon_{\text{UV}}}$$

```
int2 = TID[FVD[2 k - p, mu] FVD[2 k - p, nu] FAD[{k, m}, {k - p, m}] - 2
MTD[mu, nu] FAD[{k, m}], k]
```

$$\begin{aligned}&\frac{-(1-D)p^2p^{\text{mu}}p^{\text{nu}}-Dp^2p^{\text{mu}}p^{\text{nu}}-4m^2p^2g^{\text{mu nu}}+p^4g^{\text{mu nu}}+4m^2p^{\text{mu}}p^{\text{nu}}}{(1-D)p^2\left(k^2-m^2\right).\left((k-p)^2-m^2\right)}\\&+\frac{2\left(-(1-D)p^2g^{\text{mu nu}}-Dp^{\text{mu}}p^{\text{nu}}-p^2g^{\text{mu nu}}+2p^{\text{mu}}p^{\text{nu}}\right)}{(1-D)p^2\left(k^2-m^2\right)}\end{aligned}$$

```
PaXEvaluateUV[int2, k, PaXImplicitPrefactor -> 1/(2 Pi)^D, FCE -> True]
```

$$\frac{ip^{\text{mu}}p^{\text{nu}}}{48\pi^2\varepsilon_{\text{UV}}}-\frac{ip^2g^{\text{mu nu}}}{48\pi^2\varepsilon_{\text{UV}}}$$

4.3 PaXEvaluateIR

`PaXEvaluateIR[expr, q]` is like **`PaXEvaluate`** but with the difference that it returns only the IR-divergent part of the result.

4.3.1 See also

Overview, PaXEvaluate, PaXEvaluateUV, PaXEvaluateUVIRSplit.

4.3.2 Examples

```
PaXEvaluateIR[B0[SPD[p], 0, m^2], PaXSeries -> {{SPD[p], m^2, 1}},
PaXAnalytic -> True]
```

$$\frac{1}{2\varepsilon_{\text{IR}}}-\frac{p^2}{2m^2\varepsilon_{\text{IR}}}$$

4.4 PaXEvaluateUVIRSplit

PaXEvaluateUVIRSplit[expr, q] is like **PaXEvaluate**, but with the difference that it explicitly distinguishes between UV- and IR-divergences.

The evaluation is using H. Patel's Package-X.

4.4.1 See also

Overview, PaXEvaluateIR, PaXEvaluateUV, PaXEvaluate.

4.4.2 Examples

```
PaXEvaluateUVIRSplit[B0[SPD[p], 0, m^2], PaXSeries -> {{SPD[p], m^2, 1}},
PaXAnalytic -> True]
```

$$\frac{m^2-p^2}{2m^2\varepsilon_{\text{IR}}}-\frac{\left(3m^2-p^2\right)\left(-\log\left(\frac{\mu^2}{m^2}\right)+\gamma-2+\log(\pi)\right)}{2m^2}+\frac{1}{\varepsilon_{\text{UV}}}$$

4.5 PaXContinuedDiLog

PaXContinuedDiLog corresponds to **ContinuedDiLog** in Package-X.

4.5.1 See also

Overview.

4.5.2 Examples

```
(*Just to load Package-X*)
  PaXEvaluate[A0[1]];
```

PaXContinuedDiLog uses **X`ContinuedDiLog** for numerical evaluations

```
PaXContinuedDiLog[{3.2, 1.0}, {1.1, 1.0}]
```

$$-1.7089$$

```
X`ContinuedDiLog[{3.2, 1.0}, {1.1, 1.0}]
```

$$-1.7089$$

The same goes for derivatives and series expansions

```
D[PaXContinuedDiLog[{x, xInf}, {y, yInf}], x]
```

$$-\frac{y(-\log(x+i\ \text{xInf}\epsilon)-\log(y+i\ \text{yInf}\epsilon))}{1-xy}$$

```
Series[PaXContinuedDiLog[{x, xInf}, {y, yInf}], {x, 1, 2}]
```

$$\mathcal{L}_2(1+i\ \text{xInf}\epsilon, y+i\ \text{yInf}\epsilon)-\frac{(x-1)y\log(y+i\ \text{yInf}\epsilon)}{y-1}+\frac{(x-1)^2\left(y^2\log(y+i\ \text{yInf}\epsilon)-y^2+y\right)}{2(y-1)^2}+O\left((x-1)^3\right)$$

4.6 PaXDiLog

PaXDiLog corresponds to **DiLog** in Package-X.

4.6.1 See also

Overview.

4.6.2 Examples

```
(*Just to load Package-X*)
  PaXEvaluate[A0[1]];
```

PaXDiLog uses **X`DiLog** for numerical evaluations

```
PaXDiLog[1, 2]
```

$$\frac{\pi^2}{6}$$

```
X`DiLog[1, 2]
```

$$\frac{\pi^2}{6}$$

The same goes for derivatives and series expansions

```
D[PaXDiLog[x, \[Alpha]], x]
```

$$-\frac{\log(1-x+i\alpha\epsilon)}{x}$$

```
Series[PaXDiLog[x, \[Alpha]], {x, 0, 5}]
```

$$x+\frac{x^2}{4}+\frac{x^3}{9}+\frac{x^4}{16}+\frac{x^5}{25}+O\left(x^6\right)$$

4.7 PaXDiscB

PaXDiscB corresponds to **DiscB** in Package-X.

4.7.1 See also

Overview.

4.7.2 Examples

4.8 PaXEpsilonBar

PaXEpsilonBar corresponds to **DimRegEpsilon** in Package-X, i.e. **1/PaXEpsilonBar** means **1/Epsilon - EulerGamma + Log[4Pi]**.

4.8.1 See also

Overview.

4.8.2 Examples

4.9 PaXKallenLambda

PaXKallenLambda corresponds to **Kallen**λ in Package-X.

4.9.1 See also

Overview.

4.9.2 Examples

4.10 PaXKibblePhi

PaXKibblePhi corresponds to **Kibble**ϕ in Package-X.

4.10.1 See also

Overview.

4.10.2 Examples

4.11 PaXLn

PaXLn corresponds to **Ln** in Package-X.

4.11.1 See also

Overview.

4.11.2 Examples

```
(*Just to load Package-X*)
  PaXEvaluate[A0[1]];
```

PaXLn uses **X`Ln** for numerical evaluations

```
PaXLn[-4.5, 1]
```

$$1.50408 + 3.14159i$$

```
X`Ln[-4.5, 1]
```

$$1.50408 + 3.14159i$$

The same goes for derivatives and series expansions

```
D[PaXLn[x, \[Alpha]], x]
```

$$\frac{1}{x}$$

```
Series[PaXLn[x, \[Alpha]], {x, 1, 2}]
```

$$\log(1 + i\alpha\epsilon) + (x-1) - \frac{1}{2}(x-1)^2 + O\left((x-1)^3\right)$$

4.12 PaXpvA

PaXpvA corresponds to the **PVA** function in Package-X.

4.12.1 See also

Overview.

4.12.2 Examples

4.13 PaXpvB

PaXpvB corresponds to the **PVB** function in Package-X.

4.13.1 See also

Overview.

4.13.2 Examples

4.14 PaXpvC

PaXpvC corresponds to the **PVC** function in Package-X

4.14.1 See also

Overview.

4.14.2 Examples

4.15 PaXpvD

PaXpvD corresponds to the **PVD** function in Package-X.

4.15.1 See also

Overview.

4.15.2 Examples

4.16 PaXAnalytic

PaXAnalytic is an option for **PaXEvaluate**. If set to **True**, **LoopRefine** and **LoopRefineSeries** will be called with **Analytic->True**.

4.16.1 See also

Overview, PaXEvaluate.

4.16.2 Examples

4.17 PaXC0Expand

PaXC0Expand is an option for **PaXEvaluate**. If set to **True**, Package-X function **C0Expand** will be applied to the output of Package-X.

4.17.1 See also

Overview, PaXEvaluate.

4.17.2 Examples

```
PaVe[0, 0, 1, {SP[p, p], SP[p, p], m^2}, {m^2, m^2, m^2}]
PaXEvaluate[%]
```

$$\mathrm{C}_{001}\left(\overline{p}^2, \overline{p}^2, m^2, m^2, m^2, m^2\right)$$

PaXEvaluate: The explicit result for the occurring C0 function(s) is expected to be very complicated. Please rerun PaXEvaluate with the option PaXC0Expand->True to show the result nevertheless. Please set $FCAdvice=False if you do not want to see this message in future.

$$\begin{aligned}
&-\frac{9m^2\overline{p}^4\,\mathrm{C}_0\left(m^2, \overline{p}^2, \overline{p}^2, m^2, m^2, m^2\right)}{2\left(m^2-4\overline{p}^2\right)^2}+\frac{\overline{p}^6\,\mathrm{C}_0\left(m^2, \overline{p}^2, \overline{p}^2, m^2, m^2, m^2\right)}{\left(m^2-4\overline{p}^2\right)^2}\\
&-\frac{m^6\,\mathrm{C}_0\left(m^2, \overline{p}^2, \overline{p}^2, m^2, m^2, m^2\right)}{2\left(m^2-4\overline{p}^2\right)^2}+\frac{3m^4\overline{p}^2\,\mathrm{C}_0\left(m^2, \overline{p}^2, \overline{p}^2, m^2, m^2, m^2\right)}{\left(m^2-4\overline{p}^2\right)^2}\\
&-\frac{\sqrt{3}\pi m^2\overline{p}^2}{\left(m^2-4\overline{p}^2\right)^2}-\frac{11m^2}{36\left(m^2-4\overline{p}^2\right)}+\frac{\pi\overline{p}^4}{\sqrt{3}\left(m^2-4\overline{p}^2\right)^2}\\
&+\frac{19\overline{p}^2}{18\left(m^2-4\overline{p}^2\right)}-\frac{7m^2\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\log\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{3\left(m^2-4\overline{p}^2\right)^2}\\
&-\frac{\overline{p}^2\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\log\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{3\left(m^2-4\overline{p}^2\right)^2}+\frac{\sqrt{3}\pi m^4}{4\left(m^2-4\overline{p}^2\right)^2}\\
&+\frac{2m^4\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\log\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{3\overline{p}^2\left(m^2-4\overline{p}^2\right)^2}\\
&-\frac{1}{12\varepsilon}+\frac{1}{12}\left(-\log\left(\frac{\mu^2}{m^2}\right)+\gamma-\log(4\pi)+2\log(2\pi)\right)
\end{aligned}$$

The full result is a **ConditionalExpression**

```
PaVe[0, 0, 1, {SP[p, p], SP[p, p], m^2}, {m^2, m^2, m^2}]
res = PaXEvaluate[%, PaXC0Expand -> True];
```

$$\mathrm{C}_{001}\left(\overline{p}^2, \overline{p}^2, m^2, m^2, m^2, m^2\right)$$

```
res // Short
res // Last
```

$$\boxed{\frac{1}{12}\left(-\log(4\pi)+\gamma-\frac{1}{\varepsilon}\right)-\frac{1}{12}\log\left(\frac{\mu^2}{m^2}\right)+\langle\langle 6\rangle\rangle+\frac{1}{6}\log(2\pi)\text{ if }m^4-\langle\langle 1\rangle\rangle>0}$$

$$m^4-4m^2\overline{p}^2>0$$

Use **`Normal`** to get the actual expression

```
(res // Normal)
```

$$
\begin{aligned}
&\frac{1}{12}\left(-\log(4\pi)+\gamma-\frac{1}{\varepsilon}\right)-\frac{1}{12}\log\left(\frac{\mu^2}{m^2}\right)\\
&+\frac{\log\left(\frac{2m^2-\overline{p}^2+\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}}{2m^2}\right)\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\left(2m^4-7\overline{p}^2m^2-\overline{p}^4\right)}{3\left(m^2-4\overline{p}^2\right)^2\overline{p}^2}\\
&+\frac{\pi\left(3m^4-12\overline{p}^2m^2+4\overline{p}^4\right)}{4\sqrt{3}\left(m^2-4\overline{p}^2\right)^2}-\frac{1}{2\left(m^2-4\overline{p}^2\right)^2}\left(m^6-6\overline{p}^2m^4+9\overline{p}^4m^2\right.\\
&\left.-2\overline{p}^6\right)\left(-\frac{\mathrm{Li}_2\left(\frac{-\left(\left(m^2-2\overline{p}^2\right)m^2\right)-\sqrt{m^4-4m^2\overline{p}^2}m^2}{-\left(\left(m^2-2\overline{p}^2\right)m^2\right)-\sqrt{3}\sqrt{-m^4}\sqrt{m^4-4m^2\overline{p}^2}}i\left(-m^2+2\overline{p}^2+\sqrt{m^4-4m^2\overline{p}^2}\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\right.\\
&+\frac{\mathrm{Li}_2\left(\frac{m^2\sqrt{m^4-4m^2\overline{p}^2}-m^2\left(m^2-2\overline{p}^2\right)}{-\left(\left(m^2-2\overline{p}^2\right)m^2\right)-\sqrt{3}\sqrt{-m^4}\sqrt{m^4-4m^2\overline{p}^2}}i\left(-m^2+2\overline{p}^2+\sqrt{m^4-4m^2\overline{p}^2}\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\\
&-\frac{\mathrm{Li}_2\left(\frac{-\left(\left(m^2-2\overline{p}^2\right)m^2\right)-\sqrt{m^4-4m^2\overline{p}^2}m^2}{\sqrt{3}\sqrt{-m^4}\sqrt{m^4-4m^2\overline{p}^2}-m^2\left(m^2-2\overline{p}^2\right)}+i\left(m^2-2\overline{p}^2+\sqrt{m^4-4m^2\overline{p}^2}\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\\
&+\frac{\mathrm{Li}_2\left(\frac{m^2\sqrt{m^4-4m^2\overline{p}^2}-m^2\left(m^2-2\overline{p}^2\right)}{\sqrt{3}\sqrt{-m^4}\sqrt{m^4-4m^2\overline{p}^2}-m^2\left(m^2-2\overline{p}^2\right)}+i\left(m^2-2\overline{p}^2+\sqrt{m^4-4m^2\overline{p}^2}\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\\
&-\frac{2\,\mathrm{Li}_2\left(\frac{m^2\overline{p}^2-\overline{p}^2\sqrt{m^4-4m^2\overline{p}^2}}{m^2\overline{p}^2-\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\sqrt{m^4-4m^2\overline{p}^2}}+i\left(m^2+\sqrt{m^4-4m^2\overline{p}^2}\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\\
&+\frac{2\,\mathrm{Li}_2\left(\frac{\overline{p}^2m^2+\overline{p}^2\sqrt{m^4-4m^2\overline{p}^2}}{m^2\overline{p}^2-\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\sqrt{m^4-4m^2\overline{p}^2}}+i\left(m^2+\sqrt{m^4-4m^2\overline{p}^2}\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\\
&-\frac{2\,\mathrm{Li}_2\left(\frac{m^2\overline{p}^2-\overline{p}^2\sqrt{m^4-4m^2\overline{p}^2}}{\overline{p}^2m^2+\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\sqrt{m^4-4m^2\overline{p}^2}}i\left(\sqrt{m^4-4m^2\overline{p}^2}-m^2\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\\
&\left.+\frac{2\,\mathrm{Li}_2\left(\frac{\overline{p}^2m^2+\overline{p}^2\sqrt{m^4-4m^2\overline{p}^2}}{\overline{p}^2m^2+\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\sqrt{m^4-4m^2\overline{p}^2}}i\left(\sqrt{m^4-4m^2\overline{p}^2}-m^2\right)\epsilon\right)}{\sqrt{m^2\left(m^2-4\overline{p}^2\right)}}\right)\\
&-\frac{11m^2-38\overline{p}^2}{36\left(m^2-4\overline{p}^2\right)}+\frac{1}{6}\log(2\pi)
\end{aligned}
$$

4.18 PaXD0Expand

PaXD0Expand is an option for **PaXEvaluate**. If set to **True**, Package-X function **D0Expand** will be applied to the output of Package-X.

4.18.1 See also

Overview, PaXEvaluate.

4.18.2 Examples

```
D0[0, 0, 0, 0, s, t, m^2, m^2, m^2, m^2]
PaXEvaluate[%]
```

$$\mathrm{D}_0\left(0,0,0,0,s,t,m^2,m^2,m^2,m^2\right)$$

PaXEvaluate: The explicit result for the occurring D0 function(s) is expected to be very complicated. Please rerun PaXEvaluate with the option PaXD0Expand->True to show the result nevertheless. Please set $FCAdvice=False if you do not want to see this message in future.

$$\mathrm{D}_0\left(0,0,0,0,s,t,m^2,m^2,m^2,m^2\right)$$

The full result is a **ConditionalExpression**

```
D0[0, 0, 0, 0, s, t, m^2, m^2, m^2, m^2]
res = PaXEvaluate[%, PaXC0Expand -> True];
```

$$\mathrm{D}_0\left(0,0,0,0,s,t,m^2,m^2,m^2,m^2\right)$$

PaXEvaluate: The explicit result for the occurring D0 function(s) is expected to be very complicated. Please rerun PaXEvaluate with the option PaXD0Expand->True to show the result nevertheless. Please set $FCAdvice=False if you do not want to see this message in future.

```
res // Short
res // Last
```

$$\mathrm{D}_0\left(0,0,0,0,s,t,m^2,m^2,m^2,m^2\right)$$

$$m^2$$

Use **Normal** to get the actual expression

```
(res // Normal)
```

$$\mathrm{D}_0\left(0,0,0,0,s,t,m^2,m^2,m^2,m^2\right)$$

4.19 PaXDiscExpand

PaXDiscExpand is an option for **PaXEvaluate**. If set to **True**, Package-X function **DiscExpand** will be applied to the output of Package-X thus replacing **DiscB** by its explicit form.

4.19.1 See also

Overview, PaXEvaluate.

4.19.2 Examples

```
PaVe[0, 0, 1, {SP[p, p], 0, m^2}, {m^2, m^2, m^2}]
PaXEvaluate[%]
```

$$\mathrm{C}_{001}\left(0,\overline{p}^2,m^2,m^2,m^2,m^2\right)$$

$$-\frac{35m^2}{36\left(m^2-\overline{p}^2\right)}+\frac{2\overline{p}^2}{9\left(m^2-\overline{p}^2\right)}+\frac{m^2\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\log\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{2\left(m^2-\overline{p}^2\right)^2}$$
$$-\frac{\overline{p}^2\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\log\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{12\left(m^2-\overline{p}^2\right)^2}+\frac{\pi\left(9\sqrt{3}+\pi\right)m^4}{36\left(m^2-\overline{p}^2\right)^2}$$
$$+\frac{m^4\log^2\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{4\left(m^2-\overline{p}^2\right)^2}+\frac{m^4\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}\log\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{3\overline{p}^2\left(m^2-\overline{p}^2\right)^2}$$
$$-\frac{1}{12\varepsilon}+\frac{1}{12}\left(-\log\left(\frac{\mu^2}{m^2}\right)+\gamma-\log(4\pi)+2\log(2\pi)\right)$$

```
PaVe[0, 0, 1, {SP[p, p], 0, m^2}, {m^2, m^2, m^2}]
PaXEvaluate[%, PaXDiscExpand -> False]
```

$$C_{001}\left(0,\overline{p}^2,m^2,m^2,m^2,m^2\right)$$

$$\frac{m^2\overline{p}^2\left(\Lambda(\overline{p}^2,m,m)\right)}{2\left(m^2-\overline{p}^2\right)^2}-\frac{\overline{p}^4\left(\Lambda(\overline{p}^2,m,m)\right)}{12\left(m^2-\overline{p}^2\right)^2}+\frac{m^4\left(\Lambda(\overline{p}^2,m,m)\right)}{3\left(m^2-\overline{p}^2\right)^2}$$
$$-\frac{m^2\overline{p}^2\left(-6\log\left(\frac{\mu^2}{m^2}\right)+6\gamma-43+6\log(\pi)\right)}{36\left(m^2-\overline{p}^2\right)^2}+\frac{\overline{p}^4\left(-3\log\left(\frac{\mu^2}{m^2}\right)+3\gamma-8+3\log(\pi)\right)}{36\left(m^2-\overline{p}^2\right)^2}$$
$$+\frac{m^4\left(-3\log\left(\frac{\mu^2}{m^2}\right)+\pi^2+9\sqrt{3}\pi+3\gamma-35-3\log(4\pi)+6\log(2\pi)\right)}{36\left(m^2-\overline{p}^2\right)^2}$$
$$+\frac{m^4\log^2\left(\frac{\sqrt{\overline{p}^2\left(\overline{p}^2-4m^2\right)}-\overline{p}^2+2m^2}{2m^2}\right)}{4\left(m^2-\overline{p}^2\right)^2}-\frac{1}{12\varepsilon}$$

4.20 PaXExpandInEpsilon

PaXExpandInEpsilon is an option for **PaXEvaluate**. If **ImplicitPrefactor** is not **1** and **SubstituteEpsilon** is set to **True**, then the value of **PaXExpandInEpsilon** determines whether the final result should be again expanded in **Epsilon**.

The expansion is done only up to $\mathcal{O}(\varepsilon^0)$. The default value is **True**.

4.20.1 See also

Overview, PaXEvaluate.

4.20.2 Examples

4.21 PaXImplicitPrefactor

PaXImplicitPrefactor is an option for **PaXEvaluate**. It specifies the prefactor that does not show up explicitly in the input expression, but is understood to appear in front of every 1-loop integral. For technical reasons, **PaXImplicitPrefactor** should not depend on the number of dimensions **D**. Instead you should explicitly specify what **D** is (e.g. **4-2 Epsilon**). The default value is **1**.

If the standard prefactor $1/(2\pi)^D$ is implicit in your calculations, use **ImplicitPrefactor -> 1/(2 Pi)^(4 - 2 Epsilon)** .

4.21.1 See also

Overview, PaXEvaluate.

4.21.2 Examples

4.22 PaXKallenExpand

PaXKallenExpand is an option for **PaXEvaluate**. If set to **True**, Package-X function **KallenExpand** will be applied to the output of Package-X thus replacing **Kallen**λ by its explicit form.

4.22.1 See also

Overview, PaXEvaluate.

4.22.2 Examples

4.23 PaXKibbleExpand

PaXKibbleExpand is an option for **PaXEvaluate**. If set to **True**, Package-X function **KibbleExpand** will be applied to the output of Package-X thus replacing **Kibble**ϕ by its explicit form.

4.23.1 See also

Overview, PaXEvaluate.

4.23.2 Examples

4.24 PaXLoopRefineOptions

PaXLoopRefineOptions is an option for **PaXEvaluate**. It allows you to directly specify the options supplied to **LoopRefine**, the Package-X function which returns analytic expressions for loop integrals.

The options should be given using **X`** context, i.e. **PaXLoopRefineOptions -> {X`ExplicitC0 -> All}**.

4.24.1 See also

Overview, PaXEvaluate.

4.24.2 Examples

4.25 PaXPath

PaXPath is an option for **PaXEvaluate**. It specifies the directory in which Package-X is installed.

4.25.1 See also

Overview, PaXEvaluate.

4.25.2 Examples

4.26 PaXSeries

PaXSeries is an option for **PaXEvaluate**. It allows to expand Passarino-Veltman functions around given variables.

The expansion is done by **LoopRefineSeries** and the syntax is the same as with the ordinary **Series**, e.g. **PaXSeries -> {{m0, 0, 2}}** or **PaXSeries -> {{SPD[p1, p1], pp1, 0}, {SPD[p2, p2], pp2, 0}}**.

4.26.1 See also

Overview, PaXEvaluate.

4.26.2 Examples

4.27 PaXSimplifyEpsilon

PaXSimplifyEpsilon is an option for **PaXEvaluate**. When set to **True**, **PaXEvaluate** will attempt to simplify the final result by applying simplifications to the **Epsilon**-free parts of the expression. The default value is **True**.

4.27.1 See also

Overview, PaXEvaluate.

4.27.2 Examples

4.28 PaXSubstituteEpsilon

PaXSubstituteEpsilon is an option for PaXEvaluate. For brevity, Package-X normally abbreviates **1/Epsilon - EulerGamma + Log[4Pi]** with **1/Epsilon** (see **DimRegEpsilon** in the Documentation of Package-X).

When **SubstituteEpsilon** is set to **True**, **PaXEvaluate** will undo this abbreviation to obtain the full result.

4.28.1 See also

Overview, PaXEvaluate.

4.28.2 Examples

5 C++ FIRE interface

5.1 FIRECreateConfigFile

FIRECreateConfigFile[topo, fireID, path] automatically generates a FIRE .config file for the given topology **topo** with the FIRE-identifier **fireID** and saves it to **path/topoName** as **topoName.config** where **topoName** is the **FCTopology**-identifier. The function returns the full path to the generated .config file.

If the directory specified in **path/topoName** does not exist, it will be created automatically. If it already exists, its content will be automatically overwritten, unless the option **OverwriteTarget** is set to **False**.

If no **fireID** is given, i.e. the function is called as **FIRECreateConfigFile[topo, path]**, then the default value **4242** is used.

It is also possible to invoke the routine as **FIRECreateConfigFile[{topo1, topo2, ...}, {id1, id2, ...}, {path1, path2, ...}]** or **FIRECreateConfigFile[{topo1, topo2, ...}, {path1, path2, ...}]**if one needs to process a list of topologies.

The syntax **FIRECreateConfigFile[{topo1, topo2, ...}, {id1, id2, ...}, path]** or **FIRECreateConfigFile[{topo1, topo2, ...}, path]** is also allowed. This implies that all config files will go into the corresponding subdirectories of path, e.g. **path/topoName1**, **path/topoName2** etc.

The default name of the file containing loop integrals for the reduction is **"LoopIntegrals.m"**. It can be changed via the option **FIREIntegrals**.

To customize the content of the .config file one can use following options:

- **FIREBucket** (corresponds to **#bucket**, default value **29**)
- **FIRECompressor** (corresponds to **#compressor**, default value **"zstd"**)
- **FIREFThreads** (corresponds to **#fthreads**, default value $2 \times N_{CPU}$)
- **FIRELThreads** (corresponds to **#lthreads**, default value **2**)
- **FIREPosPref** (corresponds to **#pospref**, unset by default)
- **FIRESThreads** (corresponds to **#sthreads**, default value N_{CPU})
- **FIREThreads** (corresponds to **#threads**, default value N_{CPU})

5.1.1 See also

Overview, FIREBucket, FIRECompressor, FIREFthreads, FIRELthreads, FIREIntegrals, FIREPosPref, FIRESthreads, FIREThreads.

5.1.2 Examples

```
topo = FCTopology["asyR1prop2Ltopo01310X11111N1", {SFAD[{{I*p1, 0}, {0,
-1}, 1}], SFAD[{{I*p3, 0}, {-mg^2, -1}, 1}], SFAD[{{0, -2*p3 . q}, {0, -1},
1}], SFAD[{{I*(p1 + q), 0}, {-mb^2, -1}, 1}], SFAD[{{0, p1 . p3}, {0, -1},
1}]}, {p1, p3}, {q}, {SPD[q, q] -> mb^2}, {}]
```

$$\text{FCTopology}\left(\text{asyR1prop2Ltopo01310X11111N1},\left\{\frac{1}{(-\text{p1}^2-i\eta)},\frac{1}{(-\text{p3}^2+\text{mg}^2-i\eta)},\right.\right.$$
$$\left.\left.\frac{1}{(-2(\text{p3}\cdot q)-i\eta)},\frac{1}{(-(\text{p1}+q)^2+\text{mb}^2-i\eta)},\frac{1}{(\text{p1}\cdot\ \text{p3}-i\eta)}\right\},\{\text{p1},\text{p3}\},\{q\},\left\{q^2\to\ \text{mb}^2\right\},\{\}\right)$$

```
fileName = FIRECreateConfigFile[topo, FileNameJoin[{$FeynCalcDirectory,
"Database"}]];
fileName // FilePrint

(*#compressor zstd
#threads 8
#fthreads s16
#lthreads 4
#sthreads 8
#variables d, mb, mg
#bucket 29
#start
#folder ./
#problem 4242 asyR1prop2Ltopo01310X11111N1.start
#integrals LoopIntegrals.m
#output asyR1prop2Ltopo01310X11111N1.tables*)
```

```
fileName = FIRECreateConfigFile[topo, FileNameJoin[{$FeynCalcDirectory,
"Database"}], FIREIntegrals -> "LIs.m", FIRELthreads -> 4];
fileName // FilePrint

(*#compressor zstd
#threads 8
#fthreads s16
#lthreads 4
#sthreads 8
#variables d, mb, mg
#bucket 29
#start
#folder ./
#problem 4242 asyR1prop2Ltopo01310X11111N1.start
#integrals LIs.m
#output asyR1prop2Ltopo01310X11111N1.tables*)
```

```
fileName = FIRECreateConfigFile[topo, FileNameJoin[{$FeynCalcDirectory,
"Database"}], FIREIntegrals -> "LIs.m", FIRELthreads -> 4];
fileName // FilePrint

(*#compressor zstd
```

```
#threads 8
#fthreads s16
#lthreads 4
#sthreads 8
#variables d, mb, mg
#bucket 29
#start
#folder ./
#problem 4242 asyR1prop2Ltopo01310X11111N1.start
#integrals LIs.m
#output asyR1prop2Ltopo01310X11111N1.tables*)
```

```
topos = {
   FCTopology["asyR3prop2Ltopo01310X11111N1", {SFAD[{{I*p1, 0}, {0, -1},
1}], SFAD[{{I*p3, 0}, {-mg^2, -1}, 1}], SFAD[{{0, -2*p3 . q}, {0, -1}, 1}],
SFAD[{{0, -2*p1 . q}, {0, -1}, 1}], SFAD[{{I*(p1 - p3), 0}, {0, -1}, 1}]},
{p1, p3}, {q}, {SPD[q, q] -> mb^2},  {}],

   FCTopology["asyR1prop2Ltopo01310X11111N1", {SFAD[{{I*p1, 0}, {0, -1},
1}], SFAD[{{I*p3, 0}, {-mg^2, -1}, 1}], SFAD[{{0, -2*p3 . q}, {0, -1}, 1}],
SFAD[{{I*(p1 + q), 0}, {-mb^2, -1}, 1}], SFAD[{{0, p1 . p3}, {0, -1}, 1}]},
{p1, p3}, {q}, {SPD[q, q] -> mb^2}, {}]
  }
```

$$\left\{\text{FCTopology}\left(\text{asyR3prop2Ltopo01310X11111N1},\left\{\frac{1}{(-\text{p1}^2-i\eta)},\frac{1}{(-\text{p3}^2+\text{mg}^2-i\eta)},\right.\right.\right.$$
$$\left.\frac{1}{(-2(\text{p3}\cdot q)-i\eta)},\frac{1}{(-2(\text{p1}\cdot q)-i\eta)},\frac{1}{(-(\text{p1}-\text{p3})^2-i\eta)}\right\},\{\text{p1},\text{p3}\},\{q\},\{q^2\to\ \text{mb}^2\},$$
$$\left.\{\}\right),\text{FCTopology}\left(\text{asyR1prop2Ltopo01310X11111N1},\left\{\frac{1}{(-\text{p1}^2-i\eta)},\frac{1}{(-\text{p3}^2+\text{mg}^2-i\eta)},\right.\right.$$
$$\left.\left.\left.\frac{1}{(-2(\text{p3}\cdot q)-i\eta)},\frac{1}{(-(\text{p1}+q)^2+\text{mb}^2-i\eta)},\frac{1}{(\text{p1}\cdot\ \text{p3}-i\eta)}\right\},\{\text{p1},\text{p3}\},\{q\},\{q^2\to\ \text{mb}^2\},\{\}\right)\right\}$$

```
fileNames = FIRECreateConfigFile[topos, {1150, 1160},
FileNameJoin[{$FeynCalcDirectory, "Database"}]];
```

```
FilePrint[fileNames[[1]]];

(*#compressor zstd
#threads 8
#fthreads s16
#lthreads 4
#sthreads 8
#variables d, mb, mg
#bucket 29
```

```
#start
#folder ./
#problem 1150 asyR3prop2Ltopo01310X11111N1.start
#integrals LoopIntegrals.m
#output asyR3prop2Ltopo01310X11111N1.tables*)
```

```
FilePrint[fileNames[[2]]];

(*#compressor zstd
#threads 8
#fthreads s16
#lthreads 4
#sthreads 8
#variables d, mb, mg
#bucket 29
#start
#folder ./
#problem 1160 asyR1prop2Ltopo01310X11111N1.start
#integrals LoopIntegrals.m
#output asyR1prop2Ltopo01310X11111N1.tables*)
```

5.2 FIRECreateIntegralFile

FIRECreateIntegralFile[ex, topo, fireID, path] extracts **GLI** symbols from **ex** that belong to the topology **topo**. The resulting list of integrals is saved to **path/topoName/LoopIntegrals.m** and can be referred to in the corresponding FIRE .config file.

If the directory specified in **path/topoName** does not exist, it will be created automatically. If it already exists, its content will be automatically overwritten, unless the option **OverwriteTarget** is set to **False**.

If no **fireID** is given, i.e. the function is called as **FIRECreateIntegralFile[topo, path]**, then the default value **4242** is used.

Notice that **ex** may also contain integrals from different topologies, as long as all those topologies are provided as a list in the **topo** argument.

It is also possible to invoke the routine as **FIRECreateIntegralFile[ex, {topo1, topo2, ...}, {id1, id2, ...}, {path1, path2, ...}]** or **FIRECreateIntegralFile[ex, {topo1, topo2, ...}, {path1, path2, ...}]**if one needs to process a list of topologies.

The syntax **FIRECreateIntegralFile[ex, {topo1, topo2, ...}, {id1, id2, ...}, path]** or **FIRECreateIntegralFile[ex, {topo1, topo2, ...}, path]** is also allowed. This implies that all config files will go into the corresponding subdirectories of path, e.g. **path/topoName1**, **path/topoName2** etc.

The default name of the file containing loop integrals for the reduction is "LoopIntegrals.m". It can be changed via the option **FIREIntegrals**.

5.2.1 See also

Overview, FIRECreateConfigFile, FIREIntegrals.

5.2.2 Examples

```
ints = la^8*GLI["asyR3prop2Ltopo01310X11111N1", {-7, 1, 1, 9, 1}] +
    la^8*GLI["asyR3prop2Ltopo01310X11111N1", {-6, 0, 2, 8, 1}] -
    la^7*GLI["asyR3prop2Ltopo01310X11111N1", {-6, 1, 1, 8, 1}] -
    la^8*mg^2*GLI["asyR3prop2Ltopo01310X11111N1", {-6, 1, 2, 8, 1}] +
    la^8*GLI["asyR3prop2Ltopo01310X11111N1", {-5, -1, 3, 7, 1}] -
    la^7*GLI["asyR3prop2Ltopo01310X11111N1", {-5, 0, 2, 7, 1}] -
    2*la^8*mg^2*GLI["asyR3prop2Ltopo01310X11111N1", {-5, 0, 3, 7, 1}] +
    la^6*GLI["asyR3prop2Ltopo01310X11111N1", {-5, 1, 1, 7, 1}] +
    la^7*mg^2*GLI["asyR3prop2Ltopo01310X11111N1", {-5, 1, 2, 7, 1}] +
    la^8*mg^4*GLI["asyR3prop2Ltopo01310X11111N1", {-5, 1, 3, 7, 1}];
```

```
topo = FCTopology["asyR3prop2Ltopo01310X11111N1", {SFAD[{{I*p1, 0}, {0,
-1}, 1}], SFAD[{{I*p3, 0}, {-mg^2, -1}, 1}], SFAD[{{0, -2*p3 . q}, {0, -1},
1}], SFAD[{{0, -2*p1 . q}, {0, -1}, 1}], SFAD[{{I*(p1 - p3), 0}, {0, -1},
1}]}, {p1, p3}, {q}, {SPD[q, q] -> mb^2}, {}]
```

$$\text{FCTopology}\left(\text{asyR3prop2Ltopo01310X11111N1}, \left\{\frac{1}{(-\text{p1}^2 - i\eta)}, \frac{1}{(-\text{p3}^2 + \text{mg}^2 - i\eta)}, \frac{1}{(-2(\text{p3}\cdot q) - i\eta)}, \frac{1}{(-2(\text{p1}\cdot q) - i\eta)}, \frac{1}{(-(\text{p1} - \text{p3})^2 - i\eta)}\right\}, \{\text{p1}, \text{p3}\}, \{q\}, \{q^2 \to \text{mb}^2\}, \{\}\right)$$

```
fileName = FIRECreateIntegralFile[ints, topo,
FileNameJoin[{$FeynCalcDirectory, "Database"}]];
fileName // FilePrint
```

FIRECreateIntegralFile: Number of loop integrals: 10

```
(*{{4242, {-7, 1, 1, 9, 1}}, {4242, {-6, 0, 2, 8, 1}},
 {4242, {-6, 1, 1, 8, 1}}, {4242, {-6, 1, 2, 8, 1}},
 {4242, {-5, -1, 3, 7, 1}}, {4242, {-5, 0, 2, 7, 1}},
 {4242, {-5, 0, 3, 7, 1}}, {4242, {-5, 1, 1, 7, 1}},
 {4242, {-5, 1, 2, 7, 1}}, {4242, {-5, 1, 3, 7, 1}}}*)
```

```
fileName = FIRECreateIntegralFile[ints, topo, 1500,
FileNameJoin[{$FeynCalcDirectory, "Database"}]];
fileName // FilePrint
```

FIRECreateIntegralFile: Number of loop integrals: 10

```
(*{{1500, {-7, 1, 1, 9, 1}}, {1500, {-6, 0, 2, 8, 1}},
 {1500, {-6, 1, 1, 8, 1}}, {1500, {-6, 1, 2, 8, 1}},
 {1500, {-5, -1, 3, 7, 1}}, {1500, {-5, 0, 2, 7, 1}},
 {1500, {-5, 0, 3, 7, 1}}, {1500, {-5, 1, 1, 7, 1}},
 {1500, {-5, 1, 2, 7, 1}}, {1500, {-5, 1, 3, 7, 1}}}*)
```

```
FIRECreateIntegralFile[ints, topo, 1500, FileNameJoin[{$FeynCalcDirectory,
"Database"}], FCVerbose -> -1];
```

```
ex2 = c1 GLI[prop1l, {1, 1}] + c2 GLI[prop1l, {2, 0}] + c3 GLI[tad2l, {1,
1, 0}] + c4 GLI[tad2l, {1, 1, 1}] l
```

$$\text{c1}G^{\text{prop1l}}(1,1)+\text{c2}G^{\text{prop1l}}(2,0)+\text{c3}G^{\text{tad2l}}(1,1,0)+\text{c4}lG^{\text{tad2l}}(1,1,1)$$

```
topos = {
   FCTopology[prop1l, {FAD[{p1, m1}], FAD[{p1 + q, m2}]}, {p1}, {q}, {},
{}],

   FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
{p1, p2}, {}, {}, {}]
  }
```

$$\left\{\text{FCTopology}\left(\text{prop1l},\left\{\frac{1}{\text{p1}^2-\text{m1}^2},\frac{1}{(\text{p1}+q)^2-\text{m2}^2}\right\},\{\text{p1}\},\{q\},\{\},\{\}\right),\right.$$
$$\left.\text{FCTopology}\left(\text{tad2l},\left\{\frac{1}{\text{p1}^2-\text{m1}^2},\frac{1}{\text{p2}^2-\text{m2}^2},\frac{1}{(\text{p1}-\text{p2})^2-\text{m3}^2}\right\},\{\text{p1},\text{p2}\},\{\},\{\},\{\}\right)\right\}$$

```
fileNames = FIRECreateIntegralFile[ex2, topos, {112, 113},
FileNameJoin[{$FeynCalcDirectory, "Database"}]];
```

FIRECreateIntegralFile: Number of loop integrals: 2

FIRECreateIntegralFile: Number of loop integrals: 2

```
fileNames[[1]] // FilePrint

(*{{112, {1, 1}}, {112, {2, 0}}}*)
```

```
fileNames[[2]] // FilePrint

(*{{113, {1, 1, 0}}, {113, {1, 1, 1}}}*)
```

```
FIRECreateIntegralFile[ex2, topos, FileNameJoin[{$FeynCalcDirectory,
"Database"}], FCVerbose -> -1,
  FIREIntegrals -> "LIs.m"]
```

{/home/vs/.Mathematica/Applications/FeynCalc/Database/prop1l/LIs.m,
/home/vs/.Mathematica/Applications/FeynCalc/Database/tad2l/LIs.m}

5.3 FIRECreateStartFile

FIRECreateStartFile[path] creates a FIRE .start file using the script **CreateStartFile.m** in **path**. To that aim a Mathematica kernel is started in the background via **RunProcess**. The function returns **True** if the evaluation succeeds and **False** otherwise.

Alternatively, one can use **FIRECreateStartFile[path, topo]** where **topo** is an **FCTopology** symbol and the full path is implied to be **path/topoName/CreateStartFile.m**.

If you need to process a list of topologies, following syntaxes are possible **FIRECreateStartFile[{path1, path2, ...}]**, **FIRECreateStartFile[path, {topo1, topo2, ...}]**

The path to the Mathematica Kernel can be specified via **FIREMathematicaKernelPath**. The default value is **Automatic**.

5.3.1 See also

Overview, FIRECreateConfigFile, FIREPrepareStartFile, FIREMathematicaKernelPath

5.3.2 Examples

```
topo = FCTopology["prop3lX1", {SFAD[{p1, m^2}], SFAD[p2], SFAD[{p3, m^2}],
SFAD[Q - p1 - p2 - p3], SFAD[Q - p1 - p2], SFAD[Q - p1], SFAD[Q - p2],
SFAD[p1 + p3], SFAD[p2 + p3]}, {p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{prop3lX1}, \left\{\frac{1}{(\text{p1}^2 - m^2 + i\eta)}, \frac{1}{(\text{p2}^2 + i\eta)}, \frac{1}{(\text{p3}^2 - m^2 + i\eta)}, \frac{1}{((-\text{p1} - \text{p2} - \text{p3} + Q)^2 + i\eta)}, \frac{1}{((-\text{p1} - \text{p2} + Q)^2 + i\eta)}, \frac{1}{((Q - \text{p1})^2 + i\eta)}, \frac{1}{((Q - \text{p2})^2 + i\eta)}, \frac{1}{((\text{p1} + \text{p3})^2 + i\eta)}, \frac{1}{((\text{p2} + \text{p3})^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

```
fileName = FIREPrepareStartFile[topo, FileNameJoin[{$FeynCalcDirectory,
"Database"}]]
```

/home/vs/.Mathematica/Applications/FeynCalc/Database/prop3lX1/CreateStartFile.m

```
FIRECreateStartFile[fileName, FCVerbose -> 3]
```

FIRECreateStartFile: Full path to the Math Kernel binary: /media/Data/Software/Mathematica/13.0/Executables

FIRECreateStartFile: Working directory: /home/vs/.Mathematica/Applications/FeynCalc/Database/prop3lX1/

FIRECreateStartFile: Script file: CreateStartFile.m

True

5.4 FIREImportResults

FIREImportResults[topoName, path] imports the content of a FIRE .tables file and converts the results to replacement rules for **GLI**s with the id **topoName**.

Notice that **topoName** can be also a list of replacement rules that link FIRE ids to **FCTopology** ids. For the sake of convenience one can also use full **FCTopology** objects instead of their ids as in **FIREImportResults[topo, path]** or **FIREImportResults[{topo1, topo2, ...}, path]**.

If **path** represents a full path to a file, then this file is loaded. If it is just a path to a directory, then **path/topoName/topoName** is assumed to be the full path.

5.4.1 See also

Overview, FIRECreateConfigFile, FIRECreateStartFile, FIRERunReduction

5.4.2 Examples

```
ibpTables = FileNameJoin[{$FeynHelpersDirectory, "Documentation",
"Examples", "prop3L1topo010000100.tables"}];
```

```
ibpRules = FIREImportResults["prop3L1topo010000100", ibpTables];
```

```
ibpRules // Length
```

112

```
ibpRules[[1 ;; 2]]
```

$$\Big\{G^{\text{prop3L1topo010000100}}(1,1,-2,1,1,-2,1,1,0) \to \frac{\left(d^2-8d+16\right)\ \text{m1}^8 G^{\text{prop3L1topo010000100}}(0,1,1,1,1,1,1,0,0)}{16d^2-32d+16}$$
$$+\frac{\left(-35d^5+456d^4-2092d^3+4048d^2-3136d+768\right)\ \text{m1}^6 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,1,0,0)}{216d^5-1296d^4+3000d^3-3360d^2+1824d-384}$$
$$+\frac{\left(42d^6-484d^5+2333d^4-5568d^3+6572d^2-3456d+576\right)\ \text{m1}^4 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,0,0,0)}{216d^6-1620d^5+4944d^4-7860d^3+6864d^2-3120d+576},$$
$$G^{\text{prop3L1topo010000100}}(1,1,-1,1,1,-1,1,1,-1) \to \frac{\left(6d-3d^2\right)\ \text{m1}^6 G^{\text{prop3L1topo010000100}}(0,1,1,1,1,1,1,0,0)}{32d^2-64d+32}$$
$$+\frac{\left(-61d^4+380d^3-860d^2+864d-320\right)\ \text{m1}^4 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,1,0,0)}{144d^4-768d^3+1488d^2-1248d+384}$$
$$+\frac{\left(42d^4-253d^3+550d^2-520d+176\right)\ \text{m1}^2 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,0,0,0)}{72d^4-384d^3+744d^2-624d+192}\Big\}$$

```
ibpRulesTest = FIREImportResults[{3110 -> "prop3L1topo010000100"},
ibpTables];
```

```
ibpRules === ibpRulesTest
```

True

```
ibpRulesTest[[3 ;; 4]]
```

$$\Big\{G^{\text{prop3L1topo010000100}}(1,1,-1,1,1,-2,1,1,$$
$$0) \to \frac{(4-d)\ \text{m1}^6 G^{\text{prop3L1topo010000100}}(0,1,1,1,1,1,1,0,0)}{8d-8}$$
$$+\frac{(8-5d)\ \text{m1}^4 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,1,0,0)}{12d-12}$$
$$+\frac{(3d-8)\ \text{m1}^2 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,0,0,0)}{6d-6}, G^{\text{prop3L1topo010000100}}(1,$$
$$1,-2,1,1,-1,1,1,0) \to \frac{(4-d)\ \text{m1}^6 G^{\text{prop3L1topo010000100}}(0,1,1,1,1,1,1,0,0)}{8d-8}$$
$$+\frac{(8-5d)\ \text{m1}^4 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,1,0,0)}{12d-12}$$
$$+\frac{(3d-8)\ \text{m1}^2 G^{\text{prop3L1topo010000100}}(0,0,1,1,1,1,0,0,0)}{6d-6}\Big\}$$

```
topo = FCTopology[prop3L1topo010000100, {SFAD[{{I*p1, 0}, {0, -1}, 1}],
SFAD[{{I*p2, 0}, {-m1^2, -1}, 1}], SFAD[{{I*p3, 0}, {0, -1}, 1}],
SFAD[{{I*(p1 - p2), 0}, {0, -1}, 1}], SFAD[{{I*(p2 - p3), 0}, {0, -1}, 1}],
SFAD[{{I*(p1 + q1), 0}, {0, -1}, 1}], SFAD[{{I*(p2 + q1), 0}, {-m1^2, -1},
1}], SFAD[{{I*(p3 + q1), 0}, {0, -1}, 1}], SFAD[{{0, -p1 . p3}, {0, -1},
1}]}, {p1, p2, p3}, {q1}, {SPD[q1, q1] -> m1^2}, {}];
```

5.5 FIREPrepareStartFile

`FIREPrepareStartFile[topo, path]` can be used to convert an **`FCTopology`** object **`topo`** into a FIRE. start-file.

The functions creates the corresponding Mathematica script **`CreateStartFile.m`** and saves it in **`path/topoName`**. Notice that the script still needs to be evaluated in Mathematica to generate the actual FIRE .start-file. This can be conveniently done using **`FIRECreateStartFile`**.

Using **`FIREPrepareStartFile[{topo1, topo2, ...}, path]`** will save the scripts to **`path/topoName1`**, **`path/topoName2`** etc. The syntax **`FIREPrepareStartFile[{topo1, topo2, ...}, {path1, path2, ...}]`** is also possible.

The default path to the FIRE package is **`FileNameJoin[{$UserBaseDirectory, "Applications", "FIRE6", "FIRE6.m"}]`**. It can be adjusted using the option **`FIREPath`**.

5.5.1 See also

Overview, FIRECreateConfigFile, FIRECreateStartFile, FIRERunReduction

5.5.2 Examples

```
topo = FCTopology["prop3lX1", {SFAD[{p1, m^2}], SFAD[p2], SFAD[{p3, m^2}],
SFAD[Q - p1 - p2 - p3], SFAD[Q - p1 - p2], SFAD[Q - p1], SFAD[Q - p2],
SFAD[p1 + p3], SFAD[p2 + p3]}, {p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{prop3lX1}, \left\{\frac{1}{(\text{p1}^2 - m^2 + i\eta)}, \frac{1}{(\text{p2}^2 + i\eta)}, \frac{1}{(\text{p3}^2 - m^2 + i\eta)}, \frac{1}{((-\text{p1} - \text{p2} - \text{p3} + Q)^2 + i\eta)}, \frac{1}{((-\text{p1} - \text{p2} + Q)^2 + i\eta)}, \frac{1}{((Q - \text{p1})^2 + i\eta)}, \frac{1}{((Q - \text{p2})^2 + i\eta)}, \frac{1}{((\text{p1} + \text{p3})^2 + i\eta)}, \frac{1}{((\text{p2} + \text{p3})^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

```
fileName = FIREPrepareStartFile[topo, FileNameJoin[{$FeynCalcDirectory,
"Database"}]];
fileName // FilePrint

(*(* Generated on Sun 15 May 2022 20:01:25 *)

Get["/home/vs/.Mathematica/Applications/FIRE6/FIRE6.m"];
Internal={p1, p2, p3};
External={Q};
Propagators={-m^2 + p1^2, p2^2, -m^2 + p3^2, p1^2 + 2*p1*p2 + p2^2 +
2*p1*p3 + 2*p2*p3 + p3^2 - 2*p1*Q - 2*p2*Q - 2*p3*Q + Q^2, p1^2 + 2*p1*p2 +
p2^2 - 2*p1*Q - 2*p2*Q + Q^2, p1^2 - 2*p1*Q + Q^2, p2^2 - 2*p2*Q + Q^2,
p1^2 + 2*p1*p3 + p3^2, p2^2 + 2*p2*p3 + p3^2};
Replacements={};
PrepareIBP[];
Prepare[AutoDetectRestrictions -> True];
SaveStart[
"/home/vs/.Mathematica/Applications/FeynCalc/Database/prop3lX1/prop3lX1"];*
)
```

5.6 FIRERunReduction

`FIRERunReduction[path]` runs C++ FIRE on the FIRE .config file specified by path. To that aim the FIRE binary is started in the background via **`RunProcess`**. The function returns **`True`** if the evaluation succeeds and **`False`** otherwise.

If **`path`** represents a full path to a file, then this file is used as the .config file. If it is just a path to a directory, then **`path/topoName/topoName.config`** is assumed to be the full path.

The default path to the FIRE binary is **`FileNameJoin[{$UserBaseDirectory, "Applications", "FIRE6", "bin", "FIRE6"}]`**. It can be modified via the option **`FIREBinaryPath`**.

5.6.1 See also

Overview, FIRECreateConfigFile, FIRECreateStartFile.

5.6.2 Examples

```
FIRERunReduction[FileNameJoin[{$FeynHelpersDirectory, "Documentation",
"Examples", "asyR2prop2Ltopo13311X01201N1"}], FCVerbose -> 3]
```

FIRERunReduction: Full path to the FIRE binary: /home/vs/.Mathematica/Applications/FIRE6/bin/FIRE6

FIRERunReduction: Working directory: /home/vs/.Mathematica/Applications/FeynCalc/AddOns/FeynHelpers

True

5.7 FIREToFCTopology

`FIREToFCTopology[props, lmoms, emoms]` converts the list of FIRE propagators **`props`** that depend on the loop momenta **`lmoms`** and external momenta **`emoms`** into a proper **`FCTopology`** object.

Use the option **`Names`** to specify the **`id`** of the resulting topology.

5.7.1 See also

Overview, FIRECreateConfigFile, FIREPrepareStartFile.

5.7.2 Examples

```
props1 = {p1^2, p2^2, p3^2, (Q - p1 - p2 - p3)^2, (Q - p1 - p2)^2, (Q -
p1)^2, (Q - p2)^2, (p1 + p3)^2, (p2 + p3)^2}
```

$$\{\text{p1}^2, \text{p2}^2, \text{p3}^2, (-\text{p1}-\text{p2}-\text{p3}+Q)^2, (-\text{p1}-\text{p2}+Q)^2, (Q-\text{p1})^2, (Q-\text{p2})^2, (\text{p1}+\text{p3})^2, (\text{p2}+\text{p3})^2\}$$

```
FIREToFCTopology[props1, {p1, p2, p3}, {Q}]
```

$$\text{FCTopology}\left(\text{fctopology}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}-\text{p3}+Q)^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}+Q)^2+i\eta)}, \frac{1}{((Q-\text{p1})^2+i\eta)}, \frac{1}{((Q-\text{p2})^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

By default the function assumes the standard $i\eta$-prescription as in $1/(p^2-m^2+i\eta)$. However, if you are using "reversed" propagators that are often preferred in FIRE and FIESTA, then what you have is $1/(-p^2+m^2-i\eta)$, although the propagator is still Minkowskian. In this case you should use the option **`EtaSign`** and set it to **`-1`**

```
props2 = {-p1^2 + m^2, -p2^2 + m^2, -p3^2 + m^2, -(-Q + p1 + p2 + p3)^2,
-(p1 + p2 - Q)^2, -(p1 - Q)^2, -(p2 - Q)^2, -(p1 + p3)^2, -(p2 + p3)^2}
```

$$\{m^2-\text{p1}^2, m^2-\text{p2}^2, m^2-\text{p3}^2, -(\text{p1}+\text{p2}+\text{p3}-Q)^2, -(\text{p1}+\text{p2}-Q)^2, -(\text{p1}-Q)^2, -(\text{p2}-Q)^2, -(\text{p1}+\text{p3})^2, -(\text{p2}+\text{p3})^2\}$$

```
FIREToFCTopology[props2, {p1, p2, p3}, {Q}, EtaSign -> -1, Names -> myTopo]
```

$$\text{FCTopology}\left(\text{myTopo}, \left\{\frac{1}{(-\text{p1}^2+m^2-i\eta)}, \frac{1}{(-\text{p2}^2+m^2-i\eta)}, \frac{1}{(-\text{p3}^2+m^2-i\eta)}, \frac{1}{(-(\text{p1}+\text{p2}+\text{p3}-Q)^2-i\eta)}, \frac{1}{(-(\text{p1}+\text{p2}-Q)^2-i\eta)}, \frac{1}{(-(\text{p1}-Q)^2-i\eta)}, \frac{1}{(-(\text{p2}-Q)^2-i\eta)}, \frac{1}{(-(\text{p1}+\text{p3})^2-i\eta)}, \frac{1}{(-(\text{p2}+\text{p3})^2-i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

Notice that the polynomials in the FIRE propagators should not be expanded. Otherwise, there is a high chance that the conversion will fail.

```
FIREToFCTopology[ExpandAll[props1], {p1, p2, p3}, {Q}]
```

```
FIREToFCTopology: Error! FIREToFCTopology has encountered a fatal problem and must abort the computation. The
problem reads: Failed to convert all propagator to the FeynCalc notation.
```

$Aborted

5.8 FIREBinaryPath

FIREBinaryPath is an option for **FIRERunReduction**. It specifies the full path to the C++ FIRE binary.

The default value is **FileNameJoin[{$UserBaseDirectory, "Applications", "FIRE6", "bin", "FIRE6"}]**.

5.8.1 See also

Overview, FIRERunReduction.

5.8.2 Examples

5.9 FIREBucket

FIREBucket is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#bucket** parameter to be set in a FIRE .config-file. The default value is **29**.

5.9.1 See also

Overview, FIREThreads, FIRECompressor, FIREFthreads, FIRELthreads, FIREIntegrals, FIREPosPref, FIRESthreads.

5.9.2 Examples

5.10 FIRECompressor

FIRECompressor is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#compressor** parameter to be set in a FIRE .config-file. The default value is **"zstd"**.

5.10.1 See also

Overview, FIREBucket, FIREFthreads, FIRELthreads, FIREIntegrals, FIREPosPref, FIRESthreads, FIREThreads.

5.10.2 Examples

5.11 FIREFthreads

FIREFthreads is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#fthreads** parameter to be set in a FIRE .config-file. The default value is twice times the number of physical cores (**$ProcessorCount**) on your machine with the separate home.bway.net/lewis/ mode being active.

5.11.1 See also

Overview, FIREBucket, FIRECompressor, FIREThreads, FIRELthreads, FIREIntegrals, FIREPosPref, FIRESthreads.

5.11.2 Examples

5.12 FIREIntegrals

FIREIntegrals is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#integrals** parameter to be set in a FIRE .config file. The default value is **"LoopIntegrals.m"**.

5.12.1 See also

Overview, FIREBucket, FIRECompressor, FIREFthreads, FIRELthreads, FIREThreads, FIREPosPref, FIRESthreads.

5.12.2 Examples

5.13 FIRELthreads

FIRELthreads is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#lthreads** parameter to be set in a FIRE .config-file. The default value is **4**.

5.13.1 See also

Overview, FIREBucket, FIRECompressor, FIREThreads, FIREFthreads, FIREIntegrals, FIREPosPref, FIRESthreads.

5.13.2 Examples

5.14 FIREMathematicaKernelPath

FIREMathematicaKernelPath is an option for **FIRECreateStartFile** and other functions of the FIRE interface.

It specifies the full path to the Mathematica Kernel that will be used to run FIRE. The default value is **Automatic**.

5.14.1 See also

Overview, FIRECreateStartFile.

5.14.2 Examples

5.15 FIREPosPref

FIREPosPref is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#pospref** parameter to be set in a FIRE .config-file. The default value is **Default** meaning that this parameters is not set.

5.15.1 See also

Overview, FIREThreads, FIRECompressor, FIREFthreads, FIRELthreads, FIREIntegrals, FIREPosPref, FIRESthreads.

5.15.2 Examples

5.16 FIRESthreads

FIRESthreads is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#lthreads** parameter to be set in a FIRE .config-file. The default value is the number of physical cores (**$ProcessorCount**) on your machine.

5.16.1 See also

Overview, FIREBucket, FIRECompressor, FIREThreads, FIREFthreads, FIREIntegrals, FIREPosPref, FIRELthreads.

5.16.2 Examples

5.17 FIREThreads

FIREThreads is an option for **FIRECreateConfigFile** and other functions of the FIRE interface.

It specifies the **#threads** parameter to be set in a FIRE .config-file. The default value is the number of physical cores (**$ProcessorCount**) on your machine.

5.17.1 See also

Overview, FIREBucket, FIRECompressor, FIREFthreads, FIRELthreads, FIREIntegrals, FIREPosPref, FIRESthreads.

5.17.2 Examples

6 Mathematica FIRE interface

6.1 FIREBurn

FIREBurn[expr, {q1, q2, ...}, {p1, p2, ...}] reduces loop integrals with loop momenta **q1, q2, ...** and external momenta **p1, p2, ...** with integration-by-parts (IBP) relations.

FIREBurn expects that the input does not contain any loop integrals with linearly dependent propagators. Therefore, prior to starting the reduction, use **ApartFF**.

The evaluation is done on a parallel kernel using A.V. Smirnov's and V.A. Smirnov's FIRE.

6.1.1 See also

Overview.

6.1.2 Examples

```
int = SFAD[{p, m^2, 2}, {{0, 2 p . k}, m^2, 3}]
```

$$\frac{1}{(p^2-m^2+i\eta)^2.(2(k\cdot p)-m^2+i\eta)^3}$$

```
FIREBurn[int, {p}, {k}, Timing -> False]
```

$$-\frac{(D-5)(D-3)k^2\left(Dm^2-4k^2-6m^2\right)}{m^4\left(m^2-4k^2\right)^3(2(k\cdot p)-m^2+i\eta).(p^2-m^2+i\eta)}-\frac{(D-2)\left(2D^2k^2-24Dk^2+66k^2+m^2\right)}{2m^4\left(m^2-4k^2\right)^3(p^2-m^2+i\eta)}$$

6.2 FIREAddPropagators

FIREAddPropagators is an option for **FIREBurn**. Normally, for loop integrals that do not have enough propagators to form a complete basis, **FIREBurn** will automatically include missing propagators and put them to unity after the reduction is complete. In some cases it may be desirable to choose the missing propagators manually. This can be done by specifying the propagators via **AddPropagators->{prop1, prop2, ...}**.

6.2.1 See also

Overview, FIREBurn.

6.2.2 Examples

6.3 FIREConfigFiles

FIREConfigFiles is an option for **FIREBurn**. It specifies, where the three files that contain all the FIRE configuration are saved.

The first file contains the lists of loop momenta, external momenta and propagators. Normally it ends with the FIRE command **SaveStart**.

The second file loads the .start file that was generated previously and reduces the given loop integrals.

Finally, the third file contains replacement rules for the introduced abbreviations. The default location of these files is the **Database** folder inside **$FeynCalcDirectory**.

6.3.1 See also

Overview, FIREBurn.

6.3.2 Examples

6.4 FIREPath

FIREPath is an option for FIREBurn. It specifies the full path the FIRE package.

The default value is **FileNameJoin[{$UserBaseDirectory, "Applications", "FIRE6", "FIRE6.m"}]**.

6.4.1 See also

Overview, FIREBurn.

6.4.2 Examples

6.5 FIRERun

FIRERun is an option for **FIREBurn**. When set to **False**, the interface will only create configuration files specified in **FIREConfigFiles** but will not run FIRE on these files.

6.5.1 See also

Overview, FIREBurn.

6.5.2 Examples

6.6 FIRESilentMode

FIRESilentMode is an option for **FIREBurn**. When set to **True**, all the **Print**-output of FIRE running on the parallel kernel will be suppressed. This does not affect messages, i.e. warnings or error messages will be still visible.

6.6.1 See also

Overview, FIREBurn.

6.6.2 Examples

6.7 FIREStartFile

FIREStartFile is an option for **FIREBurn**. It specifies, where the start file for FIRE (the one generated by **SaveStart[]**) will be saved. The default location is the **Database** folder inside **$FeynCalcDirectory**.

6.7.1 See also

Overview, FIREBurn.

6.7.2 Examples

6.8 FIREUsingFermat

FIREUsingFermat is an option for **FIREBurn**. When set to **True**, FIRE will use FERMAT by R. H. Lewis to speed up the reduction. The default value is **False**.

Before you activate FERMAT, please make sure that you have read and understood its license agreement.

6.8.1 See also

Overview, FIREBurn.

6.8.2 Examples

7 LoopTools interface

7.1 LToolsEvaluate

LToolsEvaluate[expr, q] evaluates Passarino-Veltman functions in **expr** numerically using LoopTools by T. Hahn.

In contrast to the default behavior of LoopTools, the function returns not just the finite part but also the singular pieces proportional to 1ε and $1\varepsilon^2$. This behavior is controlled by the option **LToolsFullResult**.

Notice that the normalization of Passarino-Veltman functions differs between FeynCalc and LoopTools, cf. Section 1.2 of the LoopTools manual. In FeynCalc the overall prefactor is just $1/(i\pi^2)$, while LoopTools employs $1/(i\pi^{D/2}r_\Gamma)$ with $D = 4 - 2\varepsilon$ and $r_\Gamma = \Gamma^2(1-\varepsilon)\Gamma(1+\varepsilon)/\Gamma(1-2\varepsilon)$.

When the option **LToolsFullResult** is set to **True**, **LToolsEvaluate** will automatically account for this difference by multiplying the LoopTools output with $1/\pi^\varepsilon r_\Gamma$.

However, for **LToolsFullResult -> False** no such conversion will occur. This is because the proper conversion between different ε-dependent normalizations requires the knowledge of the poles: when terms proportional to ε multiply the poles, they generate finite contributions. In this sense it is not recommended to use **LToolsEvaluate** with **LToolsFullResult** set to **False**, unless you precisely understand what you are doing.

7.1.1 See also

Overview, LToolsExpandInEpsilon, LToolsFullResult, LToolsImplicitPrefactor, LToolsSetMudim, LToolsSetLambda.

7.1.2 Examples

Before using **LToolsEvaluate** we need to evaluate **LToolsLoadLibray[]** to establish a connection with the Mathlink executable. The value of the option **LToolsPath** contains the full path to this file. You might need to adjust it accordingly if LoopTools is installed in a different directory.

```
OptionValue[LToolsLoadLibrary, LToolsPath]
```

/home/vs/.Mathematica/Applications/FeynCalc/AddOns/FeynHelpers/ExternalTools/LoopTools/LoopTools

Notice that **LToolsEvaluate** can also load the library by itself on the first run, in case you forget to do so.

```
LToolsLoadLibrary[]
```

LoopTools library loaded.

```
(* =================================================
   FF 2.0, a package to evaluate one-loop integrals
 written by G. J. van Oldenborgh, NIKHEF-H, Amsterdam
 =================================================
 for the algorithms used see preprint NIKHEF-H 89/17,
 'New Algorithms for One-loop Integrals', by G.J. van
 Oldenborgh and J.A.M. Vermaseren, published in
 Zeitschrift fuer Physik C46(1990)425.
 ================================================*)
```

The value of the scale μ can be set via the option **LToolsSetMudim**. Evaluating the **PaVe**-function $A_0(m^2)$ at $m^2 = 1/10$ with $\mu^2 = 20$ yields

```
LToolsEvaluate[A0[m^2], LToolsSetMudim -> 20, InitialSubstitutions -> {m^2
-> 1/10}]
```

$$0.457637 + \frac{0.1}{\varepsilon}$$

Cross-checking this result with Package-X yields the same value

```
(PaXEvaluate[A0[1/10]] /. ScaleMu^2 -> 20) // N
```

$$0.457637 + \frac{0.1}{\varepsilon}$$

More complicated input is also possible

```
exp = a FAD[{q, m}, q - p] + b FAD[{q, M, 2}]
```

$$\frac{a}{(q^2 - m^2).(q - p)^2} + \frac{b}{(q^2 - M^2)^2}$$

```
res = LToolsEvaluate[exp, q, InitialSubstitutions -> {m -> 0.12352, M ->
5.14321, SPD[p] -> 0.8813}, LToolsImplicitPrefactor -> 1/(2 Pi)^(4 - 2
Epsilon), LToolsSetMudim -> 23^2]
```

$$\frac{(0.+0.00633257i)((1.+0.i)a+(1.+0.i)b)}{\varepsilon}+(0.0661028+0.01955i)((0.+1.i)a+(0.128951+0.436013i)b)$$

Compare to Package-X

```
chk = (PaXEvaluate[exp, q, PaXImplicitPrefactor -> 1/(2 Pi)^(4 - 2
Epsilon)] /. {ScaleMu^2 ->23^2, m -> 0.12352, M -> 5.14321, FCI@SPD[p] ->
0.8813}) // N
```

$$\frac{(0.+0.00633257i)(a+b)}{\varepsilon}-(0.+0.00633257i)(-14.4075a-4.94944b)+(-0.01955-0.0251337i)a$$

```
FCCompareNumbers[res, chk]
```

FCCompareNumbers: Minimal number of significant digits to agree in: 6

FCCompareNumbers: Chop is set to 1.*^-10

FCCompareNumbers: No number is set to 0. by Chop at this stage.

0

7.2 LToolsLoadLibrary

`LToolsLoadLibrary[]` loads the LoopTools library so that it can be used with FeynCalc. This command must be executed once before using any of the **`LTools*`** functions.

7.2.1 See also

Overview, LToolsEvaluate, LToolsUnLoadLibrary.

7.2.2 Examples

7.3 LToolsUnLoadLibrary

`LToolsUnLoadLibrary[]` is the inverse of **`LToolsLoadLibrary[]`**, i.e. it unloads the LoopTools library.

7.3.1 See also

Overview, LToolsEvaluate, LToolsLoadLibrary.

7.3.2 Examples

7.4 LToolsExpandInEpsilon

LToolsExpandInEpsilon is an option for **LToolsEvaluate**. When set to **True** (default), the result returned by LoopTools and multiplied with proper conversion factors will be expanded around $\varepsilon = 0$ to $\mathcal{O}(\varepsilon^0)$.

The ε-dependent conversion factors arise from the differences in the normalization between Passarino-Veltman functions in FeynCalc and LoopTools. In addition to that, the prefactor specified via **LToolsImplicitPrefactor** may also depend on ε.

Setting this option to **False** will leave the prefactors unexpanded, which might sometimes be useful when examining the obtained results.

7.4.1 See also

Overview, LToolsEvaluate, LToolsImplicitPrefactor.

7.4.2 Examples

```
LToolsLoadLibrary[]
```

LoopTools library loaded.

```
(* ================================================
   FF 2.0, a package to evaluate one-loop integrals
 written by G. J. van Oldenborgh, NIKHEF-H, Amsterdam
 ================================================
 for the algorithms used see preprint NIKHEF-H 89/17,
 'New Algorithms for One-loop Integrals', by G.J. van
 Oldenborgh and J.A.M. Vermaseren, published in
 Zeitschrift fuer Physik C46(1990)425.
 ================================================*)
```

The default behavior of **LToolsEvaluate** is to do the ε-expansion automatically

```
LToolsEvaluate[FAD[q, q - p], q, InitialSubstitutions -> {SPD[p] -> 1}]
```

$$\frac{0. + 9.8696i}{\varepsilon} - (31.0063 - 2.74429i)$$

This can be disabled by setting **LToolsExpandInEpsilon** to **False**

```
LToolsEvaluate[FAD[q, q - p], q, InitialSubstitutions -> {SPD[p] -> 1},
LToolsExpandInEpsilon -> False]
```

$$\frac{(0. + 1.i)\pi^{2-\varepsilon}\Gamma(1-\varepsilon)^2\Gamma(\varepsilon+1)}{\varepsilon\Gamma(1-2\varepsilon)} - \frac{(3.14159 - 2.i)\pi^{2-\varepsilon}\Gamma(1-\varepsilon)^2\Gamma(\varepsilon+1)}{\Gamma(1-2\varepsilon)}$$

7.5 LToolsFullResult

LToolsFullResult is an option for **LToolsEvaluate**. When set to **True** (default), **LToolsEvaluate** will return the full result including singularities and accompanying terms. Otherwise, only the finite part (standard output of LoopTools) will be provided.

The full result is assembled from pieces returned by LoopTools for the λ^2-parameter set to -2, -1 and 0 respectively. The correct prefactor that accounts for the normalization differences between Passarino-Veltman function in FeynCalc and LoopTools is added as well.

As long as **LToolsFullResult** is set to **True**, the value of the **LToolsSetLambda** option is ignored.

Disabling **LToolsFullResult** will most likely lead to incorrect normalization of the results (especially if you are only interested in the finite part). The reason for this are missing contributions to the finite part generated from poles being multiplied by terms proportional to ε or ε^2.

7.5.1 See also

Overview, LToolsEvaluate.

7.5.2 Examples

```
LToolsLoadLibrary[];
```

LoopTools library loaded.

```
(* ================================================
   FF 2.0, a package to evaluate one-loop integrals
 written by G. J. van Oldenborgh, NIKHEF-H, Amsterdam
 ================================================
 for the algorithms used see preprint NIKHEF-H 89/17,
 'New Algorithms for One-loop Integrals', by G.J. van
 Oldenborgh and J.A.M. Vermaseren, published in
 Zeitschrift fuer Physik C46(1990)425.
 ================================================*)
```

```
LToolsEvaluate[A0[m^2], InitialSubstitutions -> {m^2 -> 1}]
```

$$\frac{1.}{\varepsilon} - 0.721946$$

Setting **LToolsFullResult** to **False** will make **LToolsEvaluate** return only the finite part since the default value for **LToolsSetLambda** is **0**. However, the normalization does not agree with the FeynCalc convention

```
LToolsEvaluate[A0[m^2], InitialSubstitutions -> {m^2 -> 1},
LToolsFullResult -> False]
```

$$1.$$

Even though **LToolsEvaluate** includes the correct prefactor to convert to the FeynCalc normalization, the finite contribution generated by the $1/\varepsilon$-pole is missing here.

```
finRes = LToolsEvaluate[A0[m^2], InitialSubstitutions -> {m^2 -> 1},
LToolsFullResult -> False, LToolsExpandInEpsilon -> False]
```

$$\frac{1.\pi^{-\varepsilon}\Gamma(1-\varepsilon)^2\Gamma(\varepsilon+1)}{\Gamma(1-2\varepsilon)}$$

By setting **LToolsSetLambda->-1** we can get the coefficient of the pole. Here it is obvious that the function is IR-finite so that we do not need to check for the $1/\varepsilon^2$-pole

```
poleRes = LToolsEvaluate[A0[m^2], InitialSubstitutions -> {m^2 -> 1},
LToolsFullResult -> False, LToolsExpandInEpsilon -> False, LToolsSetLambda
-> -1]
```

$$\frac{1.\pi^{-\varepsilon}\Gamma(1-\varepsilon)^2\Gamma(\varepsilon+1)}{\Gamma(1-2\varepsilon)}$$

Combining both pieces and expanding in ε up to zeroth order we recover the same result as when using the option **LToolsFullResult**

```
Series[1/Epsilon poleRes + finRes, {Epsilon, 0, 0}] // Normal
```

$$\frac{1.}{\varepsilon} - 0.721946$$

7.6 LToolsImplicitPrefactor

LToolsImplicitPrefactor is an option for **LToolsEvaluate**. It specifies a prefactor that does not show up explicitly in the input expression, but is understood to appear in front of every Passarino-Veltman function. The default value is **1**.

You may want to use **LToolsImplicitPrefactor->1/(2Pi)^D** when working with 1-loop amplitudes, if no explicit prefactor has been introduced from the very beginning.

7.6.1 See also

Overview, LToolsEvaluate.

7.6.2 Examples

```
LToolsLoadLibrary[]
```

LoopTools library loaded.

```
(* ================================================
   FF 2.0, a package to evaluate one-loop integrals
 written by G. J. van Oldenborgh, NIKHEF-H, Amsterdam
 ================================================
 for the algorithms used see preprint NIKHEF-H 89/17,
 'New Algorithms for One-loop Integrals', by G.J. van
 Oldenborgh and J.A.M. Vermaseren, published in
 Zeitschrift fuer Physik C46(1990)425.
 ================================================*)
```

Here the prefactor $i\pi^2$ arises from the conversion of $\int d^D q\, 1/(q^2 - m^2)$ to $A_0(m^2)$

```
LToolsEvaluate[FAD[{q, m}], q, InitialSubstitutions -> {m -> 5}]
```

$$\frac{0. + 246.74i}{\varepsilon} - (0. + 972.359i)$$

```
LToolsEvaluate[FAD[{q, m}], q, InitialSubstitutions -> {m -> 5}, Head ->
keep]
```

$$\frac{i\pi^2\ \text{keep}(25.)}{\varepsilon} + i\pi^2(\text{keep}(-55.4719) - \gamma\ \text{keep}(25.) - \text{keep}(25.)\log(\pi))$$

This recovers the textbook prefactor

```
LToolsEvaluate[FAD[{q, m}], q, InitialSubstitutions -> {m -> 5},
LToolsImplicitPrefactor -> 1/(2 Pi)^(4 - 2 Epsilon)]
```

$$\frac{0. + 0.158314i}{\varepsilon} - (0. + 0.0419639i)$$

```
(PaXEvaluate[FAD[{q, m}], q, PaXImplicitPrefactor -> 1/(2 Pi)^(4 - 2
Epsilon)] /. {m -> 5, ScaleMu^2 -> 1}) // N
```

$$\frac{0. + 0.158314i}{\varepsilon} - (0. + 0.0419639i)$$

If the input expression contains both loop and non-loop terms, only the terms containing a **PaVe**-function will be multiplied by the implicit prefactor

```
LToolsEvaluate[extra + FAD[{q, m}], q, InitialSubstitutions -> {m -> 2},
LToolsExpandInEpsilon -> False]
```

$$\frac{(0. + 4.i)\pi^{2-\varepsilon}\Gamma(\varepsilon+1)\Gamma(1-\varepsilon)^2}{\varepsilon\Gamma(1-2\varepsilon)} - \frac{(0. + 1.54518i)\pi^{2-\varepsilon}\Gamma(\varepsilon+1)\Gamma(1-\varepsilon)^2}{\Gamma(1-2\varepsilon)} + \text{extra}$$

7.7 LToolsSetLambda

LToolsSetLambda corresponds to the **SetLambda** function in LoopTools.

See **?LoopTools`SetLambda** for further information regarding this LoopTools symbol.

LToolsSetLambda is also an option for **LToolsEvaluate** that sets the numerical value for the IR regularization parameter λ^2.

Setting λ^2 to **-2** or **-1** will make LoopTools return the coefficients of the $1/\varepsilon$ and $1/\varepsilon$-poles respectively. The value **0** yields the finite part of the integral where IR divergences are regularized dimensionally.

When λ^2 is set to some positive value (say **2.**), **LoopTools** will return the finite part of the integral with IR divergences being regularized using a fictitious mass. The result will naturally depend on the value of λ^2.

It is important to keep in mind that for $\lambda^2 = -1$ LoopTools also returns the UV-pole, although this not so clearly stated in the official manual.

Notice that the option **LToolsSetLambda** is ignored, as long as **LToolsFullResult** is set to **True**.

7.7.1 See also

Overview, LToolsEvaluate

7.7.2 Examples

7.8 LToolsSetMudim

LToolsSetMudim corresponds to the **SetMudim** function in LoopTools.

See **?LoopTools`SetMudim** for further information regarding this LoopTools symbol.

LToolsSetMudim is also an option for **LToolsEvaluate** that sets the numerical value for the scale parameter μ^2. The default value is **1**.

7.8.1 See also

Overview, LToolsEvaluate.

7.8.2 Examples

7.9 LToolsPath

LToolsPath is an option for **LToolsLoadLibrary**. It specifies the full path, to the LoopTools MathLink executable.

The default value is **FileNameJoin[{$FeynHelpersDirectory, "ExternalTools", "LoopTools", "LoopTools"}]**.

7.9.1 See also

Overview, LToolsLoadLibrary.

7.9.2 Examples

7.10 LToolsA0

LToolsA0 corresponds to the **A0** function in LoopTools.

See **?LoopTools`A0** for further information regarding this LoopTools symbol.

7.10.1 See also

Overview.

7.10.2 Examples

7.11 LToolsA00

LToolsA00 corresponds to the **A00** function in LoopTools.

See **?LoopTools`A00** for further information regarding this LoopTools symbol.

7.11.1 See also

Overview.

7.11.2 Examples

7.12 LToolsA0i

LToolsA0i corresponds to the **A0i** function in LoopTools. The only difference is that the id should be entered as a string, e.g. **"a0"** instead of **a0**.

See **?LoopTools`A0i** for further information regarding this LoopTools symbol.

7.12.1 See also

Overview.

7.12.2 Examples

7.13 LToolsB0

LToolsB0 corresponds to the **B0** function in LoopTools.

See **?LoopTools`B0** for further information regarding this LoopTools symbol.

7.13.1 See also

Overview.

7.13.2 Examples

7.14 LToolsB00

LToolsB00 corresponds to the **B00** function in LoopTools.

See **?LoopTools`B00** for further information regarding this LoopTools symbol.

7.14.1 See also

Overview.

7.14.2 Examples

7.15 LToolsB0i

LToolsB0i corresponds to the **B0i** function in LoopTools. The only difference is that the **id** should be entered as a string, e.g. **"b0"** instead of **b0**.

See **?LoopTools`B0i** for further information regarding this LoopTools symbol.

7.15.1 See also

Overview.

7.15.2 Examples

7.16 LToolsB1

LToolsB1 corresponds to the **B1** function in LoopTools.

See **?LoopTools`B1** for further information regarding this LoopTools symbol.

7.16.1 See also

Overview.

7.16.2 Examples

7.17 LToolsB001

LToolsB001 corresponds to the **B001** function in LoopTools.

See **?LoopTools`B001** for further information regarding this LoopTools symbol.

7.17.1 See also

Overview.

7.17.2 Examples

7.18 LToolsB11

LToolsB11 corresponds to the **B11** function in LoopTools.

See **?LoopTools`B11** for further information regarding this LoopTools symbol.

7.18.1 See also

Overview.

7.18.2 Examples

7.19 LToolsB111

LToolsB111 corresponds to the **B111** function in LoopTools.

See **?LoopTools`B111** for further information regarding this LoopTools symbol.

7.19.1 See also

Overview.

7.19.2 Examples

7.20 LToolsC0

LToolsC0 corresponds to the **C0** function in LoopTools.

See **?LoopTools`C0** for further information regarding this LoopTools symbol.

7.20.1 See also

Overview.

7.20.2 Examples

7.21 LToolsC0i

LToolsC0i corresponds to the **C0i** function in LoopTools. The only difference is that the **id** should be entered as a string, e.g. **"c0"** instead of **c0**.

See **?LoopTools`C0i** for further information regarding this LoopTools symbol.

7.21.1 See also

Overview.

7.21.2 Examples

7.22 LToolsClearCache

LToolsClearCache corresponds to the **ClearCache** function in LoopTools.

See **?LoopTools`ClearCache** for further information regarding this LoopTools symbol.

7.22.1 See also

Overview.

7.22.2 Examples

7.23 LToolsD0

LToolsD0 corresponds to the **D0** function in LoopTools.

See **?LoopTools`D0** for further information regarding this LoopTools symbol.

7.23.1 See also

Overview.

7.23.2 Examples

7.24 LToolsD0i

LToolsD0i corresponds to the **D0i** function in LoopTools. The only difference is that the **id** should be entered as a string, e.g. **"d0"** instead of **d0**.

See **?LoopTools`D0i** for further information regarding this LoopTools symbol.

7.24.1 See also

Overview.

7.24.2 Examples

7.25 LToolsDB0

LToolsDB0 corresponds to the **DB0** function in LoopTools.

See **?LoopTools`DB0** for further information regarding this LoopTools symbol.

7.25.1 See also

Overview.

7.25.2 Examples

7.26 LToolsDB00

LToolsDB00 corresponds to the **DB00** function in LoopTools.

See **?LoopTools`DB00** for further information regarding this LoopTools symbol.

7.26.1 See also

Overview.

7.26.2 Examples

7.27 LToolsDB1

LToolsDB1 corresponds to the **DB1** function in LoopTools.

See **?LoopTools`DB1** for further information regarding this LoopTools symbol.

7.27.1 See also

Overview.

7.27.2 Examples

7.28 LToolsDB11

LToolsDB11 corresponds to the **DB11** function in LoopTools.

See **?LoopTools`DB11** for further information regarding this LoopTools symbol.

7.28.1 See also

Overview.

7.28.2 Examples

7.29 LToolsDebugA

LToolsDebugA corresponds to **DebugA** in LoopTools.

See **?LoopTools`DebugA** for further information regarding this LoopTools symbol.

7.29.1 See also

Overview.

7.29.2 Examples

7.30 LToolsDebugAll

LToolsDebugAll corresponds to **DebugAll** in LoopTools.

See **?LoopTools`DebugAll** for further information regarding this LoopTools symbol.

7.30.1 See also

Overview.

7.30.2 Examples

7.31 LToolsDebugB

LToolsDebugB corresponds to **DebugB** in LoopTools.

See **?LoopTools`DebugB** for further information regarding this LoopTools symbol.

7.31.1 See also

Overview.

7.31.2 Examples

7.32 LToolsDebugC

LToolsDebugC corresponds to **DebugC** in LoopTools.

See **?LoopTools`DebugC** for further information regarding this LoopTools symbol.

7.32.1 See also

Overview.

7.32.2 Examples

7.33 LToolsDebugD

LToolsDebugD corresponds to **DebugD** in LoopTools.

See **?LoopTools`DebugD** for further information regarding this LoopTools symbol.

7.33.1 See also

Overview.

7.33.2 Examples

7.34 LToolsDebugE

LToolsDebugE corresponds to **DebugE** in LoopTools.

See **?LoopTools`DebugE** for further information regarding this LoopTools symbol.

7.34.1 See also

Overview.

7.34.2 Examples

7.35 LToolsDR1eps

LToolsDR1eps corresponds to the **DRResult** function in LoopTools.

See **?LoopTools`DR1eps** for further information regarding this LoopTools symbol.

7.35.1 See also

Overview.

7.35.2 Examples

7.36 LToolsDRResult

`LToolsDRResult` corresponds to the **`DRResult`** function in LoopTools.

See **`?LoopTools`DRResult`** for further information regarding this LoopTools symbol.

7.36.1 See also

Overview.

7.36.2 Examples

7.37 LToolsE0

`LToolsE0` corresponds to the **`E0`** function in LoopTools.

See **`?LoopTools`E0`** for further information regarding this LoopTools symbol.

7.37.1 See also

Overview.

7.37.2 Examples

7.38 LToolsE0i

LToolsE0i corresponds to the E0i function in LoopTools. The only difference is that the **`id`** should be entered as a string, e.g. **`"e0"`** instead of **`e0`**.

See **`?LoopTools`**E0i' for further information regarding this LoopTools symbol.

7.38.1 See also

Overview.

7.38.2 Examples

7.39 LToolsGetCmpBits

LToolsGetCmpBits corresponds to the **SetCmpBits** function in LoopTools.

See **?LoopTools`GetCmpBits** for further information regarding this LoopTools symbol.

7.39.1 See also

Overview.

7.39.2 Examples

7.40 LToolsGetDebugKey

LToolsGetDebugKey corresponds to the **GetDebugKey** function in LoopTools.

See **?LoopTools`GetDebugKey** for further information regarding this LoopTools symbol.

7.40.1 See also

Overview.

7.40.2 Examples

7.41 LToolsGetDelta

LToolsGetDelta corresponds to the **GetDelta** function in LoopTools.

See **?LoopTools`GetDelta** for further information regarding this LoopTools symbol.

7.41.1 See also

Overview.

7.41.2 Examples

7.42 LToolsGetDiffEps

LToolsGetDiffEps corresponds to the **GetDiffEps** function in LoopTools.

See **?LoopTools`GetDiffEps** for further information regarding this LoopTools symbol.

7.42.1 See also

Overview.

7.42.2 Examples

7.43 LToolsGetErrDigits

LToolsGetErrDigits corresponds to the **GetErrDigits** function in LoopTools.

See **?LoopTools`GetErrDigits** for further information regarding this LoopTools symbol.

7.43.1 See also

Overview.

7.43.2 Examples

7.44 LToolsGetLambda

LToolsGetLambda corresponds to the **GetLambda** function in LoopTools.

See **?LoopTools`GetLambda** for further information regarding this LoopTools symbol.

7.44.1 See also

Overview.

7.44.2 Examples

7.45 LToolsGetMaxDev

LToolsGetMaxDev corresponds to the **GetMaxDev** function in LoopTools.

See **?LoopTools`GetMaxDev** for further information regarding this LoopTools symbol.

7.45.1 See also

Overview.

7.45.2 Examples

7.46 LToolsGetMinMass

LToolsGetMinMass corresponds to the **GetMinMass** function in LoopTools.

See **?LoopTools`GetMinMass** for further information regarding this LoopTools symbol.

7.46.1 See also

Overview.

7.46.2 Examples

7.47 LToolsGetMudim

LToolsGetMudim corresponds to the **GetMudim** function in LoopTools.

See **?LoopTools`GetMudim** for further information regarding this LoopTools symbol.

7.47.1 See also

Overview.

7.47.2 Examples

7.48 LToolsGetUVDiv

LToolsGetUVDiv corresponds to the **GetUVDiv** function in **LoopTools**.

See **?LoopTools`GetUVDiv** for further information regarding this LoopTools symbol.

7.48.1 See also

Overview.

7.48.2 Examples

7.49 LToolsGetVersionKey

LToolsGetVersionKey corresponds to the **SetVersionKey** function in LoopTools.

See **?LoopTools`GetVersionKey** for further information regarding this LoopTools symbol.

7.49.1 See also

Overview.

7.49.2 Examples

7.50 LToolsGetWarnDigits

`LToolsGetWarnDigits corresponds` to the **`GetWarnDigits`** function in LoopTools.

See **`?LoopTools`GetWarnDigits`** for further information regarding this LoopTools symbol.

7.50.1 See also

Overview.

7.50.2 Examples

7.51 LToolsGetZeroEps

`LToolsGetZeroEps` corresponds to the **`GetZeroEps`** function in LoopTools.

See **`?LoopTools`GetZeroEps`** for further information regarding this LoopTools symbol.

7.51.1 See also

Overview.

7.51.2 Examples

7.52 LToolsKeyA0

`LToolsKeyA0` corresponds to **`KeyA0`** in LoopTools.

See **`?LoopTools`KeyA0`** for further information regarding this LoopTools symbol.

7.52.1 See also

Overview.

7.52.2 Examples

7.53 LToolsKeyAll

LToolsKeyAll corresponds to **KeyAll** in LoopTools.

See **?LoopTools`KeyAll** for further information regarding this LoopTools symbol.

7.53.1 See also

Overview.

7.53.2 Examples

7.54 LToolsKeyBget

LToolsKeyBget corresponds to **KeyBget** in LoopTools.

See **?LoopTools`KeyBget** for further information regarding this LoopTools symbol.

7.54.1 See also

Overview.

7.54.2 Examples

7.55 LToolsKeyC0

LToolsKeyC0 corresponds to **KeyC0** in LoopTools.

See **?LoopTools`KeyC0** for further information regarding this LoopTools symbol.

7.55.1 See also

Overview.

7.55.2 Examples

7.56 LToolsKeyCEget

LToolsKeyCEget corresponds to **KeyCEget** in LoopTools.

See **?LoopTools`KeyCEget** for further information regarding this LoopTools symbol.

7.56.1 See also

Overview.

7.56.2 Examples

7.57 LToolsKeyD0

LToolsKeyD0 corresponds to **KeyD0** in LoopTools.

See **?LoopTools`KeyD0** for further information regarding this LoopTools symbol.

7.57.1 See also

Overview.

7.57.2 Examples

7.58 LToolsKeyE0

LToolsKeyE0 corresponds to **KeyE0** in LoopTools.

See **?LoopTools`KeyE0** for further information regarding this LoopTools symbol.

7.58.1 See also

Overview.

7.58.2 Examples

7.59 LToolsKeyEget

LToolsKeyEget corresponds to **KeyEget** in LoopTools.

See **?LoopTools`KeyEget** for further information regarding this LoopTools symbol.

7.59.1 See also

Overview.

7.59.2 Examples

7.60 LToolsLi2

LToolsLi2 corresponds to the **Li2** function in LoopTools.

See **?LoopTools`Li2** for further information regarding this LoopTools symbol.

7.60.1 See also

Overview.

7.60.2 Examples

7.61 LToolsLi2omx

LToolsLi2omx corresponds to the **Li2omx** function in LoopTools.

See **?LoopTools`Li2omx** for further information regarding this LoopTools symbol.

7.61.1 See also

Overview.

7.61.2 Examples

7.62 LToolsMarkCache

LToolsMarkCache corresponds to the **MarkCache** function in LoopTools.

See **?LoopTools`MarkCache** for further information regarding this LoopTools symbol.

7.62.1 See also

Overview.

7.62.2 Examples

7.63 LToolsPaVe

LToolsPaVe corresponds to the **PaVe** function in LoopTools.

See **?LoopTools`PaVe** for further information regarding this LoopTools symbol.

7.63.1 See also

Overview.

7.63.2 Examples

7.64 LToolsRestoreCache

LToolsRestoreCache corresponds to the **RestoreCache** function in LoopTools.

See **?LoopTools`RestoreCache** for further information regarding this LoopTools symbol.

7.64.1 See also

Overview.

7.64.2 Examples

7.65 LToolsSetCmpBits

LToolsSetCmpBits corresponds to the **SetCmpBits** function in LoopTools.

See **?LoopTools`SetCmpBits** for further information regarding this LoopTools symbol.

7.65.1 See also

Overview.

7.65.2 Examples

7.66 LToolsSetDebugKey

LToolsSetDebugKey corresponds to the **SetDebugKey** function in LoopTools.

See **?LoopTools`SetDebugKey** for further information regarding this LoopTools symbol.

Use **LToolsSetDebugKey[-1]** to obtain the most complete debugging output. This can be useful when investigating issues with the evaluation of certain kinematic limits in LoopTools.

7.66.1 See also

Overview, LToolsGetDebugKey.

7.66.2 Examples

```
LToolsLoadLibrary[]
```

LoopTools library loaded.

```
(* ================================================
   FF 2.0, a package to evaluate one-loop integrals
 written by G. J. van Oldenborgh, NIKHEF-H, Amsterdam
 ================================================
 for the algorithms used see preprint NIKHEF-H 89/17,
 'New Algorithms for One-loop Integrals', by G.J. van
 Oldenborgh and J.A.M. Vermaseren, published in
 Zeitschrift fuer Physik C46(1990)425.
 ================================================*)
```

```
LToolsEvaluate[C0[0, 0, 0, 0, 1, 0], q]
```

LTools: Warning! LoopTools failed to evaluate the following PaVe function: PaVe[0, {0, 0, 0}, {0, 0, 1}]

$$\text{FeynCalcLoopTools`Private`ltFailed}\left(\text{C}_0(0,0,0,0,0,1)\right) + \frac{1.}{\varepsilon}$$

```
LToolsSetDebugKey[-1]
```

$$-1$$

```
LToolsEvaluate[C0[0, 0, 0, 0, 1, 0], q]

(* Bcoeff           4
   p      =   0.0000000000000000
   m1     =   0.0000000000000000
   m2     =   0.0000000000000000
 bb0        =              (-1.7219455507509331,0.0000000000000000)
 bb1        =              (0.86097277537546657,0.0000000000000000)
 bb11       =             (-0.57398185025031101,0.0000000000000000)
 bb111      =              (0.43048638768773329,0.0000000000000000)
 dbb0       =   (9.9999999999999978E+122,9.9999999999999978E+122)
 dbb0:1     =   (9.9999999999999978E+122,9.9999999999999978E+122)
 dbb1       =   (9.9999999999999978E+122,9.9999999999999978E+122)
 dbb1:1     =   (9.9999999999999978E+122,9.9999999999999978E+122)
 dbb00      =              (0.14349546256257775,0.0000000000000000)
 dbb001     =         (-7.1747731281288762E-002,0.0000000000000000)
```

```
 ==================================================
 Bcoeff           5
   p     =   0.0000000000000000
   m1    =   0.0000000000000000
   m2    =   1.0000000000000000
 bb0       =            (-0.72194555075093314,-0.0000000000000000)
 bb0:1     =               (1.0000000000000000,0.0000000000000000)
 bb1       =              (0.61097277537546657,0.0000000000000000)
 bb1:1     =             (-0.50000000000000000,0.0000000000000000)
 bb00      =       (-5.54863876877332851E-002,0.0000000000000000)
 bb00:1    =              (0.25000000000000000,0.0000000000000000)
 bb11      =           (-0.46287073913919996,-0.0000000000000000)
 bb11:1    =             (0.33333333333333331,0.0000000000000000)
 bb001     =        (6.47687029029332950E-002,0.0000000000000000)
 bb001:1   =            (-0.16666666666666666,0.0000000000000000)
 bb111     =             (0.36798638768773329,0.0000000000000000)
 bb111:1   =            (-0.25000000000000000,0.0000000000000000)
 dbb0      =       (0.50000000000000000,-1.0000000000000001E-050)
 dbb1      =       (-0.16666666666666669,5.0000000000000004E-051)
 dbb00     =  (7.40510181181333327E-002,-8.33333333333333290E-052)
 dbb00:1   =       (-8.33333333333333287E-002,0.0000000000000000)
 dbb11     =  (8.33333333333333148E-002,-3.33333333333333316E-051)
 dbb001    =  (-4.74421757257333238E-002,4.16666666666666645E-052)
 dbb001:1  =        (4.16666666666666644E-002,0.0000000000000000)
 ==================================================
 Ccoeff           6
   p1    =   0.0000000000000000
   p2    =   0.0000000000000000
   p1p2  =   0.0000000000000000
   m1    =   0.0000000000000000
   m2    =   0.0000000000000000
   m3    =   1.0000000000000000
collinear C0, perm = 312
C0collDR, perm = 123
 p1 =   0.0000000000000000
 p2 =   0.0000000000000000
 p3 =   0.0000000000000000
 m1 =   0.0000000000000000
 m2 =   0.0000000000000000
 m3 =   1.0000000000000000
 C0collDR: qltri3
 C0collDR:0 =                                              (NaN,NaN)
 C0collDR:1 =              (1.0000000000000000,0.0000000000000000)
 C0collDR:2 =              (0.0000000000000000,0.0000000000000000)
 cc0       =                                              (NaN,NaN)
 cc0:1     =             (1.0000000000000000,0.0000000000000000)
 cc1       =                                              (NaN,NaN)
 cc1:1     =                                              (NaN,NaN)
 cc1:2     =                                              (NaN,NaN)
 cc2       =                                              (NaN,NaN)
```

```
cc2:1     =                                                    (NaN,NaN)
cc2:2     =                                                    (NaN,NaN)
cc00      =                                                    (NaN,NaN)
cc00:1    =                                                    (NaN,NaN)
cc00:2    =                                                    (NaN,NaN)
cc11      =                                                    (NaN,NaN)
cc11:1    =                                                    (NaN,NaN)
cc11:2    =                                                    (NaN,NaN)
cc12      =                                                    (NaN,NaN)
cc12:1    =                                                    (NaN,NaN)
cc12:2    =                                                    (NaN,NaN)
cc22      =                                                    (NaN,NaN)
cc22:1    =                                                    (NaN,NaN)
cc22:2    =                                                    (NaN,NaN)
cc001     =                                                    (NaN,NaN)
cc001:1   =                                                    (NaN,NaN)
cc001:2   =                                                    (NaN,NaN)
cc002     =                                                    (NaN,NaN)
cc002:1   =                                                    (NaN,NaN)
cc002:2   =                                                    (NaN,NaN)
cc111     =                                                    (NaN,NaN)
cc111:1   =                                                    (NaN,NaN)
cc111:2   =                                                    (NaN,NaN)
cc112     =                                                    (NaN,NaN)
cc112:1   =                                                    (NaN,NaN)
cc112:2   =                                                    (NaN,NaN)
cc122     =                                                    (NaN,NaN)
cc122:1   =                                                    (NaN,NaN)
cc122:2   =                                                    (NaN,NaN)
cc222     =                                                    (NaN,NaN)
cc222:1   =                                                    (NaN,NaN)
cc222:2   =                                                    (NaN,NaN)
cc0000    =                                                    (NaN,NaN)
cc0000:1  =                                                    (NaN,NaN)
cc0000:2  =                                                    (NaN,NaN)
cc0011    =                                                    (NaN,NaN)
cc0011:1  =                                                    (NaN,NaN)
cc0011:2  =                                                    (NaN,NaN)
cc0012    =                                                    (NaN,NaN)
cc0012:1  =                                                    (NaN,NaN)
cc0012:2  =                                                    (NaN,NaN)
cc0022    =                                                    (NaN,NaN)
cc0022:1  =                                                    (NaN,NaN)
cc0022:2  =                                                    (NaN,NaN)
cc1111    =                                                    (NaN,NaN)
cc1111:1  =                                                    (NaN,NaN)
cc1111:2  =                                                    (NaN,NaN)
cc1112    =                                                    (NaN,NaN)
cc1112:1  =                                                    (NaN,NaN)
cc1112:2  =                                                    (NaN,NaN)
```

```
cc1122    =                                                (NaN,NaN)
cc1122:1  =                                                (NaN,NaN)
cc1122:2  =                                                (NaN,NaN)
cc1222    =                                                (NaN,NaN)
cc1222:1  =                                                (NaN,NaN)
cc1222:2  =                                                (NaN,NaN)
cc2222    =                                                (NaN,NaN)
cc2222:1  =                                                (NaN,NaN)
cc2222:2  =                                                (NaN,NaN)
=================================================*)
```

LTools: Warning! LoopTools failed to evaluate the following PaVe function: PaVe[0, {0, 0, 0}, {0, 0, 1}]

$$\text{FeynCalc`LoopTools`Private`ltFailed}\left(\text{C}_0(0,0,0,0,0,1)\right)+\frac{1.}{\varepsilon}$$

7.67 LToolsSetDebugRange

LToolsSetDebugRange corresponds to the **SetVersionKey** function in LoopTools.

See **?LoopTools`SetDebugRange** for further information regarding this LoopTools symbol.

7.67.1 See also

Overview.

7.67.2 Examples

7.68 LToolsSetDelta

LToolsSetDelta corresponds to the **SetDelta** function in LoopTools.

See **?LoopTools`SetDelta** for further information regarding this LoopTools symbol.

7.68.1 See also

Overview.

7.68.2 Examples

7.69 LToolsSetDiffEps

LToolsSetDiffEps corresponds to the **SetDiffEps** function in LoopTools.

See **?LoopTools`SetDiffEps** for further information regarding this LoopTools symbol.

7.69.1 See also

Overview.

7.69.2 Examples

7.70 LToolsSetErrDigits

LToolsSetErrDigits corresponds to the **SetErrDigits** function in LoopTools.

See **?LoopTools`SetErrDigits** for further information regarding this LoopTools symbol.

7.70.1 See also

Overview.

7.70.2 Examples

7.71 LToolsSetMaxDev

LToolsSetMaxDev corresponds to the **SetMaxDev** function in LoopTools.

See **?LoopTools`SetMaxDev** for further information regarding this LoopTools symbol.

7.71.1 See also

Overview.

7.71.2 Examples

7.72 LToolsSetMinMass

LToolsSetMinMass corresponds to the **SetMinMass** function in LoopTools.

See **?LoopTools`SetMinMass** for further information regarding this LoopTools symbol.

7.72.1 See also

Overview.

7.72.2 Examples

7.73 LToolsSetUVDiv

LToolsSetUVDiv corresponds to the **SetUVDiv** function in LoopTools.

See **?LoopTools`SetUVDiv** for further information regarding this LoopTools symbol.

7.73.1 See also

Overview.

7.73.2 Examples

7.74 LToolsSetVersionKey

LToolsSetVersionKey corresponds to the **SetVersionKey** function in LoopTools.

See **?LoopTools`SetVersionKey** for further information regarding this LoopTools symbol.

7.74.1 See also

Overview.

7.74.2 Examples

7.75 LToolsSetWarnDigits

LToolsSetWarnDigits corresponds to the **SetWarnDigits** function in LoopTools.

See **?LoopTools`SetWarnDigits** for further information regarding this LoopTools symbol.

7.75.1 See also

Overview.

7.75.2 Examples

7.76 $LTools

$LTools denotes the LinkObject of the LoopTools MathLink executable.

7.76.1 See also

Overview.

7.76.2 Examples

8 pySecDec interface

8.1 PSDCreatePythonScripts

PSDCreatePythonScripts[int, topo, path] creates a set of Python scripts needed for the evaluation of the integral **int** (in the **GLI** representation) belonging to the topology **topo**. The files are saved to the directory **path/topoNameXindices**. The function returns a list of two strings that point to the generation and integration scripts for pySecDec.

One can also use the **FeynAmpDenominator**-representation as in **PSDCreatePythonScripts[fadInt, lmoms, path]**, where **lmoms** is the list of the loop momenta on which **fadInt** depends. In this case the generation and integration scripts will directly go into **path**.

Another way to invoke the function would be **PSDCreatePythonScripts[{int1, int2, ...}, {topo1, topo2, ...}, path]** in which case the files will be saved to **path/topoName1Xindices1**, **path/topoName2Xindices2** etc. The syntax **PSDCreatePythonScripts[{int1, int2, ...}, {topo1, topo2, ...}, {path1, path2, ...}]** is also possible.

Unless you are computing a single scale integral with the scale variable set to unity, you must specify all external parameters (e.g. masses and scalar products of external momenta) and their numerical values via the corresponding options. For real-valued parameters use the option **PSDRealParameterRules** as **PSDRealParameterRules->{param1->val1, param2->val2, ...}**. For complex-valued parameters use **PSDComplexParameterRules**with the same syntax. The precise numerical values do not matter at the generation stage, one only has to distinguish between real- and complex-valued parameters. As far as the integration stage is concerned, you can easily change the numerical values when running the corresponding Python script. The values supplied via **PSDRealParameterRules** and **PSDComplexParameterRules** will be the default, though.

Notice that the variables passed to pySecDec must be atomic i.e. you can use **qq**, **m**, **m2**, **M** etc. but not something like **Pair[Momentum[q], Momentum[q]]**, **mass[2]**, or**sp["p.q"]**. This means that you need to replace scalar products of external momenta that appear in your integrals with some simple symbols. If this has not been done on the level of replacement rules attached to your **FCTopology** objects (5th argument), you can still use the option **FinalSubstitutions**.

Another important option that you most likely would like to specify is **PSDRequestedOrder** which specifies the order in ε up to which the integral should be evaluated.

The names of generation and integration scripts can be changed via the options **PSDGenerateFileName** and **PSDIntegrateFileName** with the default values being **generate_int.py** and **integrate_int.py** respectively.

The method used for the sector decomposition is controlled by the option **PSDDecompositionMethod**, where **"geometric"** is the default value.

The integrator used for the numerical evaluation of the integral is set by the option **PSDIntegrator**, where **"Qmc"** is the default value. Accordingly, if you want to increase the number of Qmc iterations, you should use the option **PSDMinn**.

If you know in advance that the integral you are computing does not have cuts (i.e. the result is purely real with no imaginary part), then it is highly recommended to disable the contour deformation. This will give you a huge performance boost. The option controlling this pySecDec parameter is called **PSDContourDeformation** and is set to **True** by default.

The prefactor of integrals evaluated by pySecDec is given by $\frac{1}{i\pi^{D/2}}$ per loop, which is the standard choice for multiloop calculations. However, factors of γ_E and $\log(4\pi)$ are not eliminated by default. The FeynHelpers interface takes care of that by adding an extra $e^{\gamma_E \frac{4-D}{2}}$ per loop. This is controlled by the value of the **PSDAdditionalPrefactor** option. When set to **Default**, the overall prefactor is given by $\frac{e^{\gamma_E \frac{4-D}{2}}}{i\pi^{D/2}}$ per loop. Setting this option to a different value, say **x**, will give you $\frac{x}{(i\pi^{D/2})^L}$ as the overall prefactor with L being the number of loops. Notice that in this case **x** must be a string using the pySecDec syntax.

For realistic integrals the generation stage can take a considerable amount of time, especially when done on a laptop. For this reason **PSDCreatePythonScripts** implements some safety measures to prevent the user from accidentally overwriting or corrupting the existing files. First of all, if the files **generate_int.py** and **integrate_int.py** already exist, the function will not overwrite them by default. To change this behavior you need to set the value of the option **OverwriteTarget** to **True**. In addition to that, pySecDec by itself will abort the generation stage if the output directory for the C++ code (specified by the option **PSDOutputDirectory**) already exists. However, you can tweak the corresponding Python script such, that the output directory will be always overwritten without further warnings. To this aim you need to set the option **PSDOverwritePackageDirectory** to **True**.

8.1.1 See also

Overview, PSDIntegrate, PSDLoopIntegralFromPropagators, PSDLoopPackage, PSDLoopRegions.

8.1.2 Examples

```
topo1 = FCTopology[prop1L, {FAD[{p1, m1}], FAD[{p1 + q, m2}]}, {p1}, {q},
{}, {}]
int1 = GLI[prop1L, {1, 1}]
```

$$\text{FCTopology}\left(\text{prop1L}, \left\{\frac{1}{\text{p1}^2 - \text{m1}^2}, \frac{1}{(\text{p1}+q)^2 - \text{m2}^2}\right\}, \{\text{p1}\}, \{q\}, \{\}, \{\}\right)$$

$$G^{\text{prop1L}}(1,1)$$

```
fileNames = PSDCreatePythonScripts[int1, topo1,
FileNameJoin[{$FeynCalcDirectory, "Database"}], PSDRealParameterRules ->
{qq -> 1. , m1 -> 2. , m2 -> 3.}, FinalSubstitutions -> {FCI@SPD[q] -> qq},
OverwriteTarget -> True];
```

```
fileNames[[1]] // FilePrint[#, 1 ;; 14] &

(*#!/usr/bin/env python3
from pySecDec import sum_package, loop_package, loop_regions,
LoopIntegralFromPropagators
import pySecDec as psd

li = LoopIntegralFromPropagators(
['-m1**2 + p1**2', '-m2**2 + (p1 + q)**2'],
loop_momenta = ['p1'],
powerlist = [1, 1],
dimensionality = '4 - 2*eps',
Feynman_parameters = 'x',
replacement_rules = [('q**2','qq')],
regulators = ['eps']
)
*)
```

```
fileNames[[2]] // FilePrint[#, 1 ;; 14] &

(*#!/usr/bin/env python3
from pySecDec.integral_interface import IntegralLibrary,
series_to_mathematica, series_to_maple, series_to_sympy
import sympy as sp

li = IntegralLibrary('loopint/loopint_pylink.so')

li.use_Qmc(
)

num_params_real = [1., 2., 3.]
num_params_complex = []

integral_without_prefactor, prefactor, integral_with_prefactor = li(verbose
= True,
real_parameters = num_params_real,*)
```

```
PSDCreatePythonScripts[SFAD[{p, m^2}], {p},
FileNameJoin[{$FeynCalcDirectory, "Database", "tal1LInt"}],
PSDRealParameterRules -> {m -> 1.}, OverwriteTarget -> True];
```

8.2 PSDIntegrate

`PSDIntegrate[]` is an auxiliary function that creates input for pySecDec's numerical integration routines. The output is returned in form of a string.

PSDIntegrate is used by **PSDCreatePythonScripts** when assembling the integration script.

8.2.1 See also

Overview, PSDCreatePythonScripts, PSDLoopIntegralFromPropagators, PSDLoopPackage, PSDLoopRegions.

8.2.2 Examples

```
PSDIntegrate[PSDRealParameterValues -> {11., 42.}]
```

{.use_Qmc(\n),
(real_parameters = num_params_real,\ncomplex_parameters = num_params_complex\n),
[11., 42.], []}

```
PSDIntegrate[PSDRealParameterValues -> {2., 4.}, PSDIntegrator -> "Vegas",
PSDMinEval -> 10^5]
```

{.use_Vegas(\nminxeval = 100000),
(minxeval = 100000,\nreal_parameters = num_params_real,\ncomplex_parameters = num_params_complex\n),
[2., 4.], []}

8.3 PSDLoopIntegralFromPropagators

PSDLoopIntegralFromPropagators[int, topo] is an auxiliary function that converts the given loop integral (in the **GLI** representation) belonging to the topology **topo** into input for pySecDec's **LoopIntegralFromPropagators** routine. The output is given as a list of two elements, containing a string and the prefactor of the integral. **PSDLoopIntegralFromPropagators**

PSDLoopIntegralFromPropagators is used by **PSDCreatePythonScripts** when assembling the generation script.

8.3.1 See also

Overview, PSDCreatePythonScripts, PSDIntegrate, PSDLoopPackage, PSDLoopRegions.

8.3.2 Examples

```
topo = FCTopology["prop3lX1", {SFAD[{p1, m^2}], SFAD[p2], SFAD[{p3, m^2}],
SFAD[Q - p1 - p2 - p3], SFAD[Q - p1 - p2], SFAD[Q - p1], SFAD[Q - p2],
SFAD[p1 + p3], SFAD[p2 + p3]}, {p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{prop3lX1}, \left\{\frac{1}{(\text{p1}^2 - m^2 + i\eta)}, \frac{1}{(\text{p2}^2 + i\eta)}, \frac{1}{(\text{p3}^2 - m^2 + i\eta)}, \frac{1}{((-\text{p1} - \text{p2} - \text{p3} + Q)^2 + i\eta)}, \frac{1}{((-\text{p1} - \text{p2} + Q)^2 + i\eta)}, \frac{1}{((Q - \text{p1})^2 + i\eta)}, \frac{1}{((Q - \text{p2})^2 + i\eta)}, \frac{1}{((\text{p1} + \text{p3})^2 + i\eta)}, \frac{1}{((\text{p2} + \text{p3})^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

```
PSDLoopIntegralFromPropagators[GLI["prop3lX1", {1, 1, 1, 1, 1, 1, 0, 0,
0}], topo]
```

{LoopIntegralFromPropagators(\n['p2**2', '(-p1 + Q)**2', '-m**2 + p3**2', '-m**2 + p1**2', '(-p1 - p2 + Q)**2', '(-p1 - p
1}

```
PSDLoopIntegralFromPropagators[GLI["prop3lX1", {1, 1, 1, 1, 1, 0, 0, 0,
0}], topo, FinalSubstitutions -> {FCI@SPD[Q] -> QQ, m^2 -> mm}]
```

{LoopIntegralFromPropagators(\n['p2**2', '-m**2 + p3**2', '-m**2 + p1**2', '(-p1 - p2 + Q)**2', '(-p1 - p2 - p3 + Q)**2
1}

8.4 PSDLoopPackage

`PSDLoopPackage[name, loopIntegral, order]` is an auxiliary function that creates input for pySecDec's **`loop_package`** routine. The result is returned as a string.

`PSDLoopPackage` is used by **`PSDCreatePythonScripts`** when assembling the generation script.

8.4.1 See also

Overview, PSDCreatePythonScripts, PSDIntegrate, PSDLoopIntegralFromPropagators, PSDLoopRegions.

8.4.2 Examples

```
PSDLoopPackage["loopint", "li", 2]
```

loop_package(\nname = 'loopint',\nloop_integral = li,\nrequested_orders = [2],\ndecomposition_method = 'geom

```
PSDLoopPackage["loopint", "li", 0, PSDDecompositionMethod -> "iterative",
  PSDAdditionalPrefactor -> "2", PSDContourDeformation -> False]
```

loop_package(\nname = 'loopint',\nloop_integral = li,\nrequested_orders = [0],\ncontour_deformation = False,\na

8.5 PSDLoopRegions

PSDLoopRegions[name, loopIntegral, order, smallnessParameter] is an auxiliary function that creates input for pySecDec's **loop_regions** routine. The results is returned as a string.

PSDLoopPackage is used by **PSDCreatePythonScripts** when assembling the generation script for an asymptotic expansion.

8.5.1 See also

Overview, PSDCreatePythonScripts, PSDIntegrate, PSDLoopIntegralFromPropagators, PSDLoopPackage.

8.5.2 Examples

```
PSDLoopRegions["loopint", "li", 2, z]
```

loop_regions(\nname = 'loopint',\nloop_integral = li,\nsmallness_parameter = 'z',\nexpansion_by_regions_order =

8.6 PSDSumPackage

PSDSumPackage[name, packageGenerators, order] is an auxiliary function that creates input for pySecDec's **sum_package** routine. The result is returned as a string.

8.6.1 See also

Overview, PSDCreatePythonScripts, PSDIntegrate, PSDLoopIntegralFromPropagators, PSDLoopPackage.

8.6.2 Examples

8.7 PSDAdditionalPrefactor

PSDAdditionalPrefactor is an option for **PSDLoopPackage** and other functions of the pySecDec interface. It specifies an implicit prefactor multiplying the loop integral and will be passed to pySecDec's **loop_package** argument **additional_prefactor**. The value should be a string representing a valid pySecDec expression.

8.7.1 See also

Overview, PSDLoopPackage.

8.7.2 Examples

8.8 PSDAddMonomialRegulatorPower

`PSDAddMonomialRegulatorPower` is an option for **`PSDLoopRegions`** and other functions of the pySecDec interface. It specifies the name of the regulator used to introduce monomial factors regulating integrals arising from the expansion by regions.

8.8.1 See also

Overview, PSDLoopRegions.

8.8.2 Examples

8.9 PSDCoefficients

`PSDCoefficients` is an option for **`PSDSumPackage`** and other functions of the pySecDec interface. It specifies coefficients of the integrals in the sum and will be passed to pySecDec's coefficients argument. The default value is **`None`**.

8.9.1 See also

Overview, PSDSumPackage.

8.9.2 Examples

8.10 PSDComplexParameterRules

`PSDComplexParameterRules` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. It is a list of replacement rules containing numerical values for the complex parameters of the integral.

8.10.1 See also

Overview, PSDCreatePythonScripts.

8.10.2 Examples

8.11 PSDComplexParameters

`PSDComplexParameters` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It is a list of symbols (or strings) that will be passed to pySecDec's **`loop_package`** argument **`real_parameters`**.

8.11.1 See also

Overview, PSDLoopPackage.

8.11.2 Examples

8.12 PSDComplexParameterValues

`PSDComplexParameterValues` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It is a list of real numbers that will be passed to pySecDec's **`IntegralLibrary`** function via the argument **`complex_parameters`**.

8.12.1 See also

Overview, PSDIntegrate.

8.12.2 Examples

8.13 PSDContourDeformation

`PSDContourDeformation` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It is a boolean switch that will be passed to pySecDec's **`loop_package`** argument **`contour_deformation`**. The default value is **`True`**. However, if you know in advance that your integral has no imaginary part, setting this option to **`False`** will greatly improve the peformance.

8.13.1 See also

Overview, PSDLoopPackage.

8.13.2 Examples

8.14 PSDCPUThreads

`PSDCPUThreads` is an option for PSDIntegrate and other functions of the pySecDec interface. It specifies the value of the cputhreads parameter to be passed to pySecDec's **`integral_interface`** function. Notice that this option applies only to the Qmc integrator.

8.14.1 See also

Overview, PSDIntegrate.

8.14.2 Examples

8.15 PSDDecompositionMethod

`PSDDecompositionMethod` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It specifies pySecDec's strategy for decomposing the polynomials will be passed to the **`loop_package`** argument **`decomposition_method`**. The default value is **`"geometric"`**.

8.15.1 See also

Overview, PSDLoopPackage.

8.15.2 Examples

8.16 PSDDecreaseToPercentage

`PSDDecreaseToPercentage` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`decrease_to_percentage`** parameter to be passed to pySecDec's numerical integration library.

8.16.1 See also

Overview, PSDIntegrate.

8.16.2 Examples

8.17 PSDDeformationParametersDecreaseFactor

`PSDDeformationParametersDecreaseFactor` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`deformation_parameters_decrease_factor`** parameter to be passed to pySecDec's **`IntegralLibrary`** function.

8.17.1 See also

Overview, PSDIntegrate.

8.17.2 Examples

8.18 PSDDeformationParametersMaximum

`PSDDeformationParametersMaximum` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`deformation_parameters_maximum`** parameter to be passed to pySecDec's **`IntegralLibrary`** function.

8.18.1 See also

Overview, PSDIntegrate.

8.18.2 Examples

8.19 PSDDeformationParametersMinimum

`PSDDeformationParametersMinimum` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`deformation_parameters_minimum`** parameter to be passed to pySecDec's **`IntegralLibrary`** function.

8.19.1 See also

Overview, PSDIntegrate.

8.19.2 Examples

8.20 PSDEnforceComplex

`PSDEnforceComplex` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It specifies whether or not the generated integrand functions should have a complex return type even though they might be purely real. The option value will be passed to the **`loop_package`** argument **`enforce_complex`**.

8.20.1 See also

Overview, PSDLoopPackage.

8.20.2 Examples

8.21 PSDEpsAbs

PSDEpsAbs is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the **epsabs** parameter to be passed to pySecDec's pySecDec's **IntegralLibrary** function.

8.21.1 See also

Overview, PSDIntegrate.

8.21.2 Examples

8.22 PSDEpsRel

PSDEpsRel is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the epsrel parameter to be passed to pySecDec's **IntegralLibrary** function.

8.22.1 See also

Overview, PSDIntegrate.

8.22.2 Examples

8.23 PSDErrorMode

PSDErrorMode is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the errormode parameter to be passed to pySecDec's numerical integration library.

8.23.1 See also

Overview, PSDIntegrate.

8.23.2 Examples

8.24 PSDErrorModeQmc

PSDErrorModeQmc is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the **errormode** parameter to be passed to pySecDec's **integral_interface** function.

Notice that this option applies only to the Qmc integrator.

8.24.1 See also

Overview, PSDIntegrate.

8.24.2 Examples

8.25 PSDEvaluateMinn

PSDEvaluateMinn is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the **evaluateminn** parameter to be passed to pySecDec's **integral_interface** function.

Notice that this option applies only to the Qmc integrator.

8.25.1 See also

Overview, PSDIntegrate.

8.25.2 Examples

8.26 PSDExpansionByRegionsOrder

PSDExpansionByRegionsOrder is an option for **PSDCreatePythonScripts** and other functions of the pySecDec interface. It specifies up to which order the expression should be expanded in a small parameter using the method of regions. The default value is **0**.

The small parameter must be specified via the option **PSDExpansionByRegionsParameter**.

8.26.1 See also

Overview, PSDCreatePythonScripts.

8.26.2 Examples

8.27 PSDExpansionByRegionsParameter

PSDExpansionByRegionsParameter is an option for **PSDCreatePythonScripts** and other functions of the pySecDec interface. It specifies the small parameter in which the given loop integral will be expanded using the method of regions. The default value is **None**, meaning that no expansion takes place.

The order up to which the expansion should be carried out must be specified via the option **PSDExpansionByRegionsOrder**.

8.27.1 See also

Overview, PSDCreatePythonScripts.

8.27.2 Examples

8.28 PSDFitFunction

`PSDFitFunction` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`fitfunction`** parameter to be passed to pySecDec's integral_interface function.

Notice that this option applies only to the Qmc integrator.

8.28.1 See also

Overview, PSDIntegrate.

8.28.2 Examples

8.29 PSDFlags

`PSDFlags` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`flags`** parameter to be passed to pySecDec's **`integral_interface`** function.

Notice that this option applies only to the Vegas, Suave, Cuhre and Divonne integrators.

8.29.1 See also

Overview, PSDIntegrate.

8.29.2 Examples

8.30 PSDFormExecutable

`PSDFormExecutable` is an option for **`PSDSumPackage`** and other functions of the pySecDec interface. It specifies the path to the FORM executable and will be passed to pySecDec's **`form_executable`** argument. The default value is **`None`**.

8.30.1 See also

Overview, PSDSumPackage.

8.30.2 Examples

8.31 PSDFormMemoryUse

`PSDFormMemoryUse` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It specifies the target FORM memory usage and will be passed to pySecDec's **`loop_package`** argument **`form_memory_use`**. The default value is **`None`**.

8.31.1 See also

Overview, PSDLoopPackage.

8.31.2 Examples

8.32 PSDFormOptimizationLevel

`PSDFormOptimizationLevel` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It specifies the optimization level to be used in FORM and will be passed to pySecDec's **`loop_package`** argument form_optimization_level. The default value is **`2.`**

8.32.1 See also

Overview, PSDLoopPackage.

8.32.2 Examples

8.33 PSDFormThreads

`PSDFormThreads` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It specifies the number of threads (T)FORM will use and will be passed to pySecDec's **`form_threads`** argument. The default value is **`2.`**

8.33.1 See also

Overview, PSDLoopPackage.

8.33.2 Examples

8.34 PSDFormWorkSpace

`PSDFormWorkSpace` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It specifies the size of the FORM WorkSpace and will be passed to pySecDec's **`loop_package`** argument **`form_work_space`**. The default value is **`"50M"`**.

8.34.1 See also

Overview, PSDLoopPackage.

8.34.2 Examples

8.35 PSDGenerateFileName

`PSDGenerateFileName` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. It specifies the name of the Python script that generates the C++ package needed for the integral evaluation. The default value **`"generate_int.py"`**.

8.35.1 See also

Overview, PSDCreatePythonScripts.

8.35.2 Examples

8.36 PSDGeneratingVectors

`PSDGeneratingVectors` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`generatingvectors`** parameter to be passed to pySecDec's **`integral_interface`** function.

Notice that this option applies only to the Qmc integrator.

8.36.1 See also

Overview, PSDIntegrate.

8.36.2 Examples

8.37 PSDIntegrateFileName

`PSDIntegrateFileName` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. It specifies the name of the Python script that performs the numerical integration using the previously generated C++ package. The default value is **`"integrate_int.py"`**.

8.37.1 See also

Overview, PSDCreatePythonScripts.

8.37.2 Examples

8.38 PSDIntegrator

`PSDIntegrator` is an option of PSDIntegrate and other functions of the pySecDec interface. It specifies the integrator to be used when performing the numerical evaluation of the integral. The default value is **`"Qmc"`**.

8.38.1 See also

Overview, PSDIntegrate.

8.38.2 Examples

8.39 PSDLoopIntegralName

`PSDLoopIntegralName` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. It specifies the name assigned to the output of pySecDec's **`LoopIntegralFromPropagators`** function. The default value is **`"li"`**.

8.39.1 See also

Overview, PSDCreatePythonScripts.

8.39.2 Examples

8.40 PSDMaxEpsAbs

`PSDMaxEpsAbs` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`max_epsabs`** parameter to be passed to pySecDec's numerical integration library.

8.40.1 See also

Overview, PSDIntegrate.

8.40.2 Examples

8.41 PSDMaxEpsRel

`PSDMaxEpsRel` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`max_epsrel`** parameter to be passed to pySecDec's numerical integration library.

8.41.1 See also

Overview, PSDIntegrate.

8.41.2 Examples

8.42 PSDMaxEval

`PSDMaxEval` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`maxeval`** parameter to be passed to pySecDec's numerical integration library.

8.42.1 See also

Overview, PSDIntegrate.

8.42.2 Examples

8.43 PSDMaxIncreaseFac

`PSDMaxIncreaseFac` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`maxincreasefac`** parameter to be passed to pySecDec's numerical integration library.

8.43.1 See also

Overview, PSDIntegrate.

8.43.2 Examples

8.44 PSDMinDecreaseFactor

`PSDMinDecreaseFactor` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`min_decrease_factor`** parameter to be passed to pySecDec's numerical integration library.

8.44.1 See also

Overview, PSDIntegrate.

8.44.2 Examples

8.45 PSDMinEpsAbs

`PSDMinEpsAbs` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`min_epsabs`** parameter to be passed to pySecDec's numerical integration library.

8.45.1 See also

Overview, PSDIntegrate.

8.45.2 Examples

8.46 PSDMinEpsRel

`PSDMinEpsRel` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`min_epsrel`** parameter to be passed to pySecDec's numerical integration library.

8.46.1 See also

Overview, PSDIntegrate.

8.46.2 Examples

8.47 PSDMinEval

`PSDMinEval` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`mineval`** parameter to be passed to pySecDec's numerical integration library.

8.47.1 See also

Overview, PSDIntegrate.

8.47.2 Examples

8.48 PSDMinm

`PSDMinm` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`minm`** parameter to be passed to pySecDec's **`integral_interface`** function.

Notice that this option applies only to the Qmc integrator.

8.48.1 See also

Overview, PSDIntegrate.

8.48.2 Examples

8.49 PSDMinn

PSDMinn is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the **minn** parameter to be passed to pySecDec's **integral_interface** function. Notice that this option applies only to the Qmc integrator.

8.49.1 See also

Overview, PSDIntegrate.

8.49.2 Examples

8.50 PSDNormalizExecutable

PSDNormalizExecutable is an option for **PSDLoopPackage** and other functions of the pySecDec interface. It specifies the command to run **normaliz** and will be passed to the **loop_package** argument **normaliz_executable**.

8.50.1 See also

Overview, PSDLoopPackage.

8.50.2 Examples

8.51 PSDNumberOfPresamples

PSDNumberOfPresamples is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the **number_of_presamples** parameter to be passed to pySecDec's **IntegralLibrary** function.

8.51.1 See also

Overview, PSDIntegrate.

8.51.2 Examples

8.52 PSDNumberOfThreads

`PSDNumberOfThreads` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`number_of_threads`** parameter to be passed to pySecDec's numerical integration library.

8.52.1 See also

Overview, PSDIntegrate.

8.52.2 Examples

8.53 PSDOutputDirectory

`PSDOutputDirectory` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. It specifies the C++ namespace and the output directory. The default value is **`"loopint"`**.

8.53.1 See also

Overview, PSDCreatePythonScripts.

8.53.2 Examples

8.54 PSDOverwritePackageDirectory

`PSDOverwritePackageDirectory` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. If set to **`True`**, the pySecDec script responsible for the generation of the integral will overwrite an existing C++ package directory.

8.54.1 See also

Overview, PSDCreatePythonScripts.

8.54.2 Examples

8.55 PSDPyLinkQMCTransforms

`PSDPyLinkQMCTransforms` is an option for **`PSDSumPackage`** and other functions of the pySecDec interface. It specifies the required QMC transformations and will be passed to pySecDec's **`pylink_qmc_transforms`** argument.

8.55.1 See also

Overview, PSDSumPackage.

8.55.2 Examples

8.56 PSDRealParameterRules

`PSDRealParameterRules` is an option for **`PSDCreatePythonScripts`** and other functions of the pySecDec interface. It is a list of replacement rules containing numerical values for the real parameters of the integral.

8.56.1 See also

Overview, PSDCreatePythonScripts.

8.56.2 Examples

8.57 PSDRealParameters

`PSDRealParameters` is an option for **`PSDLoopPackage`** and other functions of the pySecDec interface. It is a list of symbols (or strings) that will be passed to pySecDec's **`loop_package`** argument **`real_parameters`**.

8.57.1 See also

Overview, PSDLoopPackage.

8.57.2 Examples

8.58 PSDRealParameterValues

`PSDRealParameterValues` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It is a list of real numbers that will be passed to pySecDec's **`IntegralLibrary`** function via the argument **`real_parameters`**.

8.58.1 See also

Overview, PSDIntegrate.

8.58.2 Examples

8.59 PSDRegulators

PSDRegulators is an option for **PSDLoopIntegralFromPropagators** and other functions of the pySecDec interface. It specifies a list of symbols to be used for the regulators.

8.59.1 See also

Overview, PSDLoopIntegralFromPropagators.

8.59.2 Examples

8.60 PSDRequestedOrder

PSDRequestedOrder is an option for **PSDCreatePythonScripts** and other functions of the pySecDec interface. It specifies the needed order in the ε-expansion. The default value is **0**.

8.60.1 See also

Overview, PSDCreatePythonScripts.

8.60.2 Examples

8.61 PSDResetCudaAfter

PSDResetCudaAfter is an option for **PSDIntegrate** and other functions of the pySecDec interface. It specifies the value of the **reset_cuda_after** parameter to be passed to pySecDec's numerical integration library.

8.61.1 See also

Overview, PSDIntegrate.

8.61.2 Examples

8.62 PSDSplit

PSDSplit is an option for **PSDLoopPackage** and other functions of the pySecDec interface. It specifies whether or not to split the integration domain in order to map singularities from **1** to **0**.

8.62.1 See also

Overview, PSDLoopPackage.

8.62.2 Examples

8.63 PSDTransform

`PSDTransform` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the transform parameter to be passed to pySecDec's **`integral_interface`** function.

Notice that this option applies only to the Qmc integrator.

8.63.1 See also

Overview, PSDIntegrate.

8.63.2 Examples

8.64 PSDVerbose

`PSDVerbose` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`verbose`** parameter to be passed to pySecDec's numerical integration library.

8.64.1 See also

Overview, PSDIntegrate.

8.64.2 Examples

8.65 PSDVerbosity

`PSDVerbosity` is an option for **`PSDIntegrate`** and other functions of the pySecDec interface. It specifies the value of the **`verbosity`** parameter to be passed to pySecDec's **`integral_interface`** function.

Notice that this option applies only to the Qmc integrator.

8.65.1 See also

Overview, PSDIntegrate.

8.65.2 Examples

9 QGRAF interface

9.1 QGConvertToFC

QGConvertToFC[{amp1, amp2, ...}] converts a list of QGRAF amplitudes generated using the styling file **feyncalc.sty** into amplitudes suitable for further evaluation using FeynCalc.

9.1.1 See also

Overview, QGCreateAmp.

9.1.2 Examples

9.2 QGCreateAmp

QGCreateAmp[nLoops, {"InParticle1[p1]", "InParticle1[p2]", ...} -> {"OutParticle1[k1]", "OutParticle1[k2]", ...}] calls **QGRAF** to generate Feynman amplitudes and (optionally) the corresponding diagrams, using the specified model and style files.

The function returns a list with the paths to two files, where the first file contains the amplitudes and the second file the diagrams (graphical representations of the amplitudes).

9.2.1 See also

Overview, QGConvertToFC, QGLoadInsertions.

9.2.2 Examples

9.3 QGLoadInsertions

QGLoadInsertions["insertions.m"] loads insertion rules from **insertions.m** for amplitudes generated with QGRAF.

Specifying only the file name means that **QGLoadInsertions** will search for the file first in **$QGInsertionsDirectory** and then in the current directory. Specifying the full path will force the function to load the file from there directly.

Evaluating **QGLoadInsertions[]** loads some common insertions from **QGCommonInsertions.m** that are shipped with this interface.

9.3.1 See also

Overview, QGConvertToFC, QGCreateAmp.

9.3.2 Examples

9.4 QGPrepareDiagramsTeX

`QGPrepareDiagramsTeX[input_, output_]` processes the LaTeX representation of Feynman diagrams generated by QGRAF using a supported style file. The input file must contain valid LaTeX code. Following styles (to be set via the option **`Style`**) are supported:

- **`"TikZ-Feynman"`** - uses tikz-feynmann packages to visualize Feynman diagrams created with tikz-feynman.sty

The beginning and the end of each .tex file will be pasted from the files specified by the options **`QGTeXProlog`** and **`QGTeXEpilog`** respectively. The resulting .tex file will be saved to **`output`**. If the option **`Split`** is set to **`True`**, each diagram will be saved to a separate .tex file in the directory output.

9.4.1 See also

Overview, QGConvertToFC, QGCreateAmp.

9.4.2 Examples

9.5 QGPolarization

`QGPolarization[psi[index, momentum], mass]` is a placeholder for the external state of the field **`psi`** (e.g. a polarization vector, a spinor or simply unity for a spinless particle). It is introduced in the QGARF style file "feyncalc.sty"

9.5.1 See also

Overview.

9.5.2 Examples

9.6 QGPropagator

`QGPropagator[psi[index1, momentum], psibar[index2, momentum], mass]` is a placeholder for the propagator of the field **`psi`**. It is introduced in the QGARF style file **`"feyncalc.sty"`**.

9.6.1 See also

Overview.

9.6.2 Examples

9.7 QGTruncatedPolarization

QGTruncatedPolarization[psi[index, momentum], mass] is a placeholder for the truncated external state of the field **psi**. It is introduced in the QGARF style file **"feyncalc.sty"**.

9.7.1 See also

Overview.

9.7.2 Examples

9.8 QGVertex

QGVertex[psi1[index1, momentum1], psi2[index2, momentum2], ...] is a placeholder for the interaction vertex of the fields **psi1, psi2, ...**. It is introduced in the QGARF style file **"feyncalc.sty"**.

9.8.1 See also

Overview.

9.8.2 Examples

9.9 $QGInsertionsDirectory

$QGInsertionsDirectory is the string that represents the full path to the directory that contains Feynman rules for vertices, propagators and external states to be inserted into amplitudes generated with QGRAF.

9.9.1 See also

Overview.

9.9.2 Examples

9.10 $QGLogOutputAmplitudes

`$QGLogOutputAmplitudes` contains the full standard output of the QGRAF binary after generating the amplitudes.

9.10.1 See also

Overview.

9.10.2 Examples

9.11 $QGLogOutputDiagrams

`$QGLogOutputDiagrams` contains the full standard output of the QGRAF binary after generating the diagrams.

9.11.1 See also

Overview.

9.11.2 Examples

9.12 $QGModelsDirectory

`$QGModelsDirectory` is the string that represents the full path to the default models directory.

9.12.1 See also

Overview.

9.12.2 Examples

9.13 $QGStylesDirectory

`$QGStylesDirectory` is the string that represents the full path to the default styles directory.

9.13.1 See also

Overview.

9.13.2 Examples

9.14 $QGTeXDirectory

$QGTeXDirectory is the string that represents the full path to the directory that contains LaTeX templates files useful for visualizing Feynman diagrams.

9.14.1 See also

Overview.

9.14.2 Examples

9.15 QGAmplitudeStyle

QGAmplitudeStyle is an option for **QGCreateAmp**, which specifies the QGRAF style file for generating the amplitudes. If you provide only the file name, they style will be loaded from the standard directory containing style files (**$QGStylesDirectory**).

If you specify the full path, the style file will be loaded from there. The default value is a custom style file for FeynCalc **"feyncalc.sty"**.

9.15.1 See also

Overview, QGCreateAmp.

9.15.2 Examples

9.16 QGBinaryFile

QGBinaryFile is an option for **QGCreateAmp**, which specifies the full path to the QGRAF binary. When set to **Automatic**, the default binary is**"qgraf"** on Linux and macOS or **"qgraf.exe"** on Windows, which resides in **FileNameJoin[{$FeynHelpersDirectory, "ExternalTools", "QGRAF", "Binary"}]**.

If you provide a different location, you must ensure that the containing directory is user-writable, since **QGCreateAmp** will need to save an automatically generated **"qgraf.dat"** in that directory and delete it afterwards.

9.16.1 See also

Overview, QGCreateAmp.

9.16.2 Examples

9.17 QGCleanUpOutputDirectory

QGCleanUpOutputDirectory is an option for **QGCreateAmp** which determines whether all temporary files created in the directory that contains the QGRAF binary should be deleted after a successful QGRAF run. The default value is **True**.

9.17.1 See also

Overview, QGCreateAmp.

9.17.2 Examples

9.18 QGDiagramStyle

QGDiagramStyle is an option for **QGCreateAmp**, which specifies the QGRAF style file for generating the diagrams. If you provide only the file name, they style will be loaded from the standard directory containing model and style files (**$QGStylesDirectory**).

If you specify the full path, the style file will be loaded from there. The default value is a custom style file for FeynMP **"latex.sty"**. If the option value is set to an empty string, no diagram file will be generated.

9.18.1 See also

Overview, QGCreateAmp.

9.18.2 Examples

9.19 QGInsertionRule

QGInsertionRule["names"] is a set of replacement rules for inserting explicit vertices, propagators and polarization vectors into the amplitudes generated by QGRAF. These rules are loaded via QGLoadInsertions. Running **QGInsertionRule[]** returns a list of the already loaded sets of rules.

9.19.1 See also

Overview.

9.19.2 Examples

9.20 QGLoopMomentum

QGLoopMomentum is an option for **QGCreateAmp**, which specifies the names of the loop momenta. The default value is **LoopMom**, which means that the loop momenta will be named **LoopMom1**, **LoopMom2**, etc.

9.20.1 See also

Overview, QGCreateAmp.

9.20.2 Examples

9.21 QGModel

QGModel is an option for **QGCreateAmp**, which specifies the QGRAF model file.

If you provide only the file name, they model will be loaded from the standard directory containing model files (**$QGModelsDirectory**). If you specify the full path, the model file will be loaded from there. The default value is a model for one flavour QCD, **"QCDOneFlavor"**.

9.21.1 See also

Overview, QGCreateAmp.

9.21.2 Examples

9.22 QGOptionalStatements

QGOptionalStatements is an option for **QGCreateAmp** which specifies optional statements to be passed to QGRAF. It is a list of strings, where each string is a valid QGRAF optional statement.

9.22.1 See also

Overview, QGCreateAmp.

9.22.2 Examples

9.23 QGOptions

QGOptions is an option for **QGCreateAmp**, which specifies the options to be passed to QGRAF. It is a list of strings, where each string is a valid QGRAF option.

9.23.1 See also

Overview, QGCreateAmp.

9.23.2 Examples

9.24 QGOutputAmplitudes

QGOutputAmplitudes is an option for **QGCreateAmp**. It specifies the file into which the QGRAF output containing the generated amplitudes will be saved. The default name is **"amplitudes.m"** in the current working directory.

9.24.1 See also

Overview, QGCreateAmp.

9.24.2 Examples

9.25 QGOutputDiagrams

QGOutputDiagrams is an option for **QGCreateAmp**. It specifies, the file to which the QGRAF output containing the generated diagrams will be saved. The default name is **"diagrams.tex"** in the current working directory.

9.25.1 See also

Overview, QGCreateAmp.

9.25.2 Examples

9.26 QGOutputDirectory

QGOutputDirectory is an option for **QGCreateAmp**. It specifies the directory in which the QGRAF output containing the generated amplitudes and diagrams will be saved. The default location is the current working directory (**Directory[]**)

9.26.1 See also

Overview, QGCreateAmp.

9.26.2 Examples

9.27 QGOverwriteExistingAmplitudes

QGOverwriteExistingAmplitudes is an option for **QGCreateAmp**, which determines the behavior of the function when the file for the generated amplitudes already exists. The default value is **True**, which means that the file will be silently overwritten. Setting it to **False** will prevent the overwriting by aborting the evaluation.

9.27.1 See also

Overview, QGCreateAmp.

9.27.2 Examples

9.28 QGOverwriteExistingDiagrams

QGOverwriteExistingDiagrams is an option for **QGCreateAmp**, which determines the behavior of the function when the file for the generated diagrams already exists. The default value is **True**, which means that the file will be silently overwritten. Setting it to **False** will prevent the overwriting by aborting the evaluation.

9.28.1 See also

Overview, QGCreateAmp.

9.28.2 Examples

9.29 QGSaveInputFile

QGSaveInputFile is an option for **QGCreateAmp**, which specifies where to save the QGRAF input file **"qgraf.dat"**. This file is automatically created from the input parameters of **QGCreateAmp** but it must be located in the same directory as the QGRAF binary when QGRAF is invoked. The default value is False, which means that **"qgraf.dat"** will be deleted after the successful QGRAF run. When set to **True**, **"qgraf.dat"** will be copied to the current directory.

Specifying an explicit path will make **QGCreateAmp** put **"qgraf.dat"** there. Notice that only the file for generating the amplitudes is saved. The file for generating the diagrams (if exists) is identical except for the difference in the style line.

9.29.1 See also

Overview, QGCreateAmp.

9.29.2 Examples

9.30 QGShowOutput

QGShowOutput is an option for **QGCreateAmp**. When set to **True**, the output of the current QGRAF run will be shown via **Print**. When set to **False** the output is suppressed.

9.30.1 See also

Overview, QGCreateAmp.

9.30.2 Examples

9.31 QGTeXEpilog

GTeXProlog is an option for **QGPrepareDiagramsTeX**, which specifies the file containing LaTeX code to be added to the end of each .tex file containing Feynman diagrams.

If you provide only the file name, they file will be loaded from the standard directory containing .tex files (**$QGTeXDirectory**). If you specify the full path, the file will be loaded from there.

9.31.1 See also

Overview, QGPrepareDiagramsTeX.

9.31.2 Examples

9.32 QGTeXProlog

GTeXProlog is an option for **QGPrepareDiagramsTeX**, which specifies the file containing LaTeX code to be added to the beginning of each .tex file containing Feynman diagrams.

If you provide only the file name, they file will be loaded from the standard directory containing .tex files (**$QGTeXDirectory**). If you specify the full path, the file will be loaded from there.

9.32.1 See also

Overview, QGPrepareDiagramsTeX.

9.32.2 Examples

Index