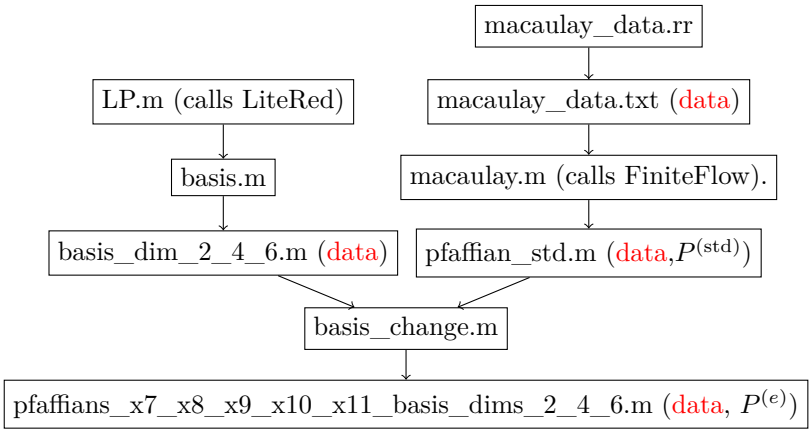
1 Program codes and data to obtain the Pfaffian matrices $P_i^{(e)}$ of one-loop massless pentagon with one massive leg



These program files are in `mma_gkz/1L_0m_1os_pentagon` of `http://www.math.kobe-u.ac.jp/OpenXM/Math/amp-MM/pentagon-dist.zip`. `deq_check.m` is a program to check the agreement of results by LiteRed and our method.

2 Notes

System configuration.

1. These codes are tested on Debian 11.1 with Mathematica 12.3.1, Risa/Asir, FiniteFlow, and LiteRed.
2. Risa/Asir download: `http://www.openxm.org`
3. FiniteFlow download: `https://github.com/peraro/finiteflow`
4. LiteRed download: `https://www.inp.nsk.su/~lee/programs/LiteRed/`
5. The folder `~/.Mathematica/Applications` contains `LiteRed, LiteRed.m, RNL`.
6. The file `~/.Mathematica/Kernel/init.m` should be set properly to run FiniteFlow. A sample `init.m` is in the zip file `pentagon-dist.zip`
7. The command `asir` must be in the command search path and some environmental variables for `asir` should be set properly.
   Please use `/usr/local/OpenXM/rc/dot.bashrc` to set them.
8. `gkz_utils/gkz.m` of `pentagon-dist.zip` contains an interface (for Debian) to Risa/Asir and the package `mt_gkz.m`, `tmp-MM.rr` and `linsolv`. The interface requires that `~/.asirrc` (asir initial run commands) is put

under the home folder. It is in the `pentagon-dist.zip` with the name `dot.asirrc`.

`1L_0m_1os_pentagon/`
Typewriter font is used for variables in codes and math-italic font is used for variables in the paper in the sequel. The page number of the paper is of the version 1 of the arxiv preprint.

1. `macaulay.m`: it constructs Pfaffian matrices $P_i^{(\mathrm{std})}$ by the Macaulay matrix with `FFSparseSolve` function of FiniteFlow. $P_7^{(\mathrm{std})}$ is stored in `p[1]`, ..., $P_{11}^{(\mathrm{std})}$ is stored in `p[5]`.

2. `basis_change.m`: `p7, p8, p9, p10, p11` are Pfaffian matrices $P_i^{(e)}$, $i = 7, \ldots, 11$ for the basis $\{e_i\}$[1]. `g` and `e` give a Gauge transformation from $P_i^{(\mathrm{std})}$ to $P_i^{(e)}$. The diagonal element ${\Lambda_{ii}}^2$ is proportional to `e[[i,i]]` in the code. Precisely speaking, `e[[i,j]]` is equal to Λ_{ii}/Λ'_i times prefactors[3]

3. `basis.m`: The "physical basis" `ints` is obtained by the IBPs section of `LP.m`, which is commented out in the code. `mis` is the output `MIs[pent]`. Each element of `ints` is of the form `j[d0,nu]`, which is a generalized Feynman integral. The function `diffop[]` is defined in `gkz_utils/mma_asir_interface.m`. It constructs a differential operator corresponding to a given differential form expressed by `j[d0,nu]` by calling `mt_gkz.rr` package. `dbaseByD`, which is equal to ${e'_i}^{\mathcal{D}}$ [4] is an expression of `dbase` by differential operators.

`gkz_util/`

1. `math.m`: The function `LPfac[]` gives $c(d_0^{(i)}, \nu^{(i)})$ [5] and the function `kinfac[]` gives $(-s_{12})^{d_0^{(i)}/2-\epsilon-|\nu^{(i)}|-5\epsilon\delta}$ [6]

[1] page 26, (5.36)
[2] page 26, (5.37)
[3] We call $(-s_{12})^{\epsilon}, \ldots$ in (5.36) of page 26 prefactors. The variable `s12` in the program codes stands for $-s_{12}$ in the paper.
[4] page 22, (5.4) and page 27, (5.39)
[5] page 26, (5.37)
[6] page 26, (5.37)