Alpaca v1.01

A Monte Carlo for Axion–Like Particle Production

Users guide

Lucian Harland-Lang
(`lucian.harland-lang@physics.ox.ac.uk`)

Updated: May 20, 2019

Contents

1 Overview

`Alpaca` is a Fortran based Monte Carlo event generator for photon–initiated axion–like particle (ALP) production and decay in beam dump experiments, due to elastic proton–ion collisions (see **ref alpaca** for details). User–defined histograms may be output, as well as unweighted events in HEPEVT and Les Houches formats. The code is completely stand–alone.

2 Installation

A compressed tar file containing all of the relevant code can be downloaded at `http://projects.hepforge.org/alpaca`. To extract it, simply execute `tar -xzvf alpacav1.0.tar.gz` and the `alpacav1.0` directory will be created. This consists of:

- `bin`: the executable `alpaca` and the input card `input.DAT`.
- `doc`: the source for this document.
- `obj`: the object files produced by the compiler.
- `src`: the Fortran source files in various subdirectories.

To compile simply run

```
> make
```

which will create the `alpaca` executable in the `bin` directory as well as the object files in the `obj` directory.

3 Input parameters and runtime options

After compilation, the `alpaca` executable can then be run in the `bin` directory, using the `input.DAT` file to adjust the input parameters by

```
> ./alpaca < input.DAT
```

The adjustable parameters in the input file are described below in order:

- `ebeam`: beam kinetic energy, E_b, in GeV.

- `btype`: beam type, either proton (`prot`) or electron (`elec`).

- `aa`: target mass number, A.

- `az`: target atomic number, Z.

- `lsh`: shielding length, in m.

- `dvol`: decay volume, in m.

- `rmin`: target inner radius, in m.

- `rmax`: target outer radius, in m.

- `dmin`: minimum photon separation, in m.

- `emin`: minimum individual photon energy from ALP decay, in GeV.

- `mres`: ALP mass, in GeV.

- `gax`: ALP–photon–photon coupling, in GeV^{-1}.

- `outtag`: label for file `output[outtag].dat`, created in `bin/outputs` directory, and event record `evrec[outtag].dat`, created in `bin/evrecs` directory. Contains input parameter and cross section information, as well as histograms (see Section 4).

- `ncall`, `itms`: number of iterations and calls in `VEGAS` preconditioning run. Soft survival effects are turned off to decrease the runtime. It is recommended to set the number of calls in units of 1000, and to increase the number when applying cuts.

- `ncall1`, `inccall`, `itend`: number of calls in `VEGAS` main run, and increase per iteration, as well as maximum number of iterations before run will automatically terminate. The run is automatically adjusted so that the number of calls passing cuts is approximately equal to `ncall1`.

- `prec`: the required % accuracy by the user; the main run will terminate once this level of accuracy has been reached.

- `iseed`: random number seed, must be a positive integer.

- `genunw`, `nev`, `erec`: set `genunw` to `.true.` to generate unweighted events, and specify required number of events with `nev`, and event record format (`lhe`, `hepevt` and `hepmc` for Les Houches, HEPEVT and HepMC, respectively) with `erec`.

- `readwt`, `wtmax`: set `readwt` to `.true.` to read in maximum weight `wtmax` input by the user. This should be given by the output 'maximum weight' in `output[outtag].dat` of a run with `genunw=.true.`, with all other physics inputs left unchanged. Setting `readwt` to `.true.` bypasses the maximum weight evaluation stage in the run. This allows for a shorter runtime, and is useful when setting multiple runs with `genunw=.true.` (i.e. each with a different `iseed`, in order to produce multiple event records in parallel).

- `gencuts`: flag to determine if cuts will be placed on photons due to ALP decay.

4 Output, histograms and cuts

The input parameters for a run, the output cross section, and the generated histograms are stored in the `output[outtag].dat` file in the `bin/ouputs` directory. Further user–defined cuts may be placed in `src/user/cuts.f`.

5 Version Information

Here we provide details of the changes made in the releases from version 1.01 onwards:

- Version 1.01: Bug leading to incorrect cross section at low beam energies fixed. ALP vertex position correctly included in event record. Exit statement added for when cross section evaluates to zero. Various minor changes, most notably some subroutines renamed.