ReneSANCe

Release 1.3.0

SANC team

Dec 14, 2022

CONTENTS:

ReneSANCe is a Monte Carlo event generator for simulation of processes at different colliders. Based on the SANC (Support for Analytic and Numeric Calculations for experiments at colliders) modules, the new generator takes into account complete one-loop and some higher-order electroweak radiative corrections with finite particle masses and polarizations. The generator effectively operates in the full phase-space. It is constructed in such a way that new processes can be easily added.

The main difference between the programs ReneSANCe and MCSANC is that ReneSANCe allows you to receive and save unweighted events (in ROOT and LHEF formats), which can be used to build any differential distributions with arbitrary cuts. The MCSANC integrator, in turn, allows you to build a fixed set of distributions that are implemented in it. But the results obtained by the integrator often have a smaller statistical error than those obtained by the ReneSANCe generator.

Please use the following reference for citations:

Comput.Phys.Commun. 256 (2020) 107445
DOI: 10.1016/j.cpc.2020.107445
e-Print: 2001.10755 [hep-ph]
e-Print: 2207.04332 [hep-ph]

CHAPTER

ONE

CONTACTS

Any questions or comments should be sended to renesance-team@cern.ch.

1.1 Downloads

1.2 Getting Started

1.2.1 Installing ReneSANCe

After downloading ReneSANCe, you need to install it on your computer.

The ROOT framework must be available at the stage of compilation. There are several opportunities for its installation, e.g., from distributive repository, from the official site or activate it from CernVM-FS repositories located under /cvmfs. For example, in the case of CERN Scientific Linux 6 (SLC6) with gcc 4.9.* compiler in 64-bit mode

```
source /cvmfs/sft.cern.ch/lcg/views/ROOT-latest/x86_64-slc6-gcc49-opt/setup.sh
```

ReneSANCe_pp binary is for hadron-hadron collision mode only. To compile with support for the hadron collision mode, the LHAPDF library must be installed on the system.

Other dependencies are bundled into archive with ReneSANCe.

The ReneSANCe event generator uses the CMake build system. We highly recommend to build ReneSANCe in the separate directory from the source. To install ReneSANCe run

```
cmake <path_to_source> -DCMAKE_INSTALL_PREFIX=<inst_prefix> <other options>
cmake --build .
cmake --build . --target install
```

Using CMake variable CMAKE_INSTALL_PREFIX you can set installation directory. By default -DCMAKE_INSTALL_PREFIX=<inst_prefix> is set to <path_to_source> directory. To list all available CMake options with the corresponding descriptions, one can run

```
cmake -LH <path_to_source>
```

or use GUI

```
ccmake .
```

For example, hadron mode compilation can be disabled using the cmake option:

```
cmake <path_to_source> -Dhadron_mode=OFF
```

1.2.2 Setup

ReneSANCe needs an access to its schema files at run time to work properly. It searches them relatively to the installation path. If the generator has been relocated and cannot find schema files, run

```
source renesance-init.sh
```

to make it available in the current console or export environment variable RENESANCE_ROOT. For bash shell:

```
export RENESANCE_ROOT=<path_to_root_directory_of_ReneSANCe_installation>
```

Program can be run as

```
ReneSANCe <option1> <option2> ...
```

The ReneSANCe event generator has a multilayer configuration system: default parameters are defined in the program, parameters are provided by a user through configuration files (higher priority) and command line parameters (the highest priority).

Many parameters would be initialized to a default value, which the user can find in the schema files installed in system.

However, some parameters like *pid*, *ecm* do not have default value, so the user must provide it manually using a configuration file or command line interface.

The list of command line options:

-h, --help
: Print help message and exit.

-f <FILE>**, --file** <FILE>
: Set path to index.conf file.

-o,--out DIR
: Set output directory

-s,--seed INT
: Set seed

-p,--pid INT in [101 - 104]
: Set process

-e,--ecm FLOAT
: Set energy of collider

--pol1 FLOAT in [-1 - 1]
: Set first beam polarization

--pol2 FLOAT in [-1 - 1]
: Set second beam polarization

-D,--define TEXT ...

Set other settings as list of key:value

By default the program searches for a configuration file that has the name `index.conf` in the *./input/* directory. However, it can be provided by the `ReneSANCe` using the command line parameter *-f*. Argument of *-f* can be directory containing index.conf or full path including arbitrary name of configuration file.

The input file language was built on JSON syntax, so configuration in JSON format is valid. Our input format is less restrictive. The line comments are allowed with # as a separator. Also, opening and closing brackets { } can be omitted. The parameters are initialized using the key:value or key=value syntax. The type of each key is defined in the schema files.

1.2.3 Output

By default `ReneSANCe` produces only console output with cross-sections. To save this output please use console redirection to file, for example

```
ReneSANCe -f <folder with input files> | tee file_name.txt
```

Also in the run directory there are FOAM*.root files with grid created for each branch. They can be used for further calculations with the same input parameters when you want to produce more events on the same grid. The option explore<branch name>=false switches on the search of existing FOAM*.root files with the existing grid.

Using option `printROOT=true` you can turn on saving of events in `ROOT` format. These events are saved to the Results directory by default, but you can change the pass using option `outputDir=/your/desired/path`. Born events are saved into the file named events_LO.root and NLO events are saved into the file named events_NLO.root. Also the .dat files are generated with technical information which is used by `plot1d` and `plot2d` utilities. You can produce `LHE` events files using the option `printLHE=true`. Born events are saved into the file named events_LO.lhe and NLO events are saved into the file named events_NLO.lhe.

1.2.4 Plots

There is a simple utility to produce plots from ROOT output files. It is compiled and installed by default. Change directory to the folder with ROOT output files and run plot1d to create 1d histograms or plot2d to create 2d histograms. Generally this utility is sample and can be simply modified for your needs.

1.3 Configuration

Configuration files support JSON-like syntax and contain general steering parameters for a run, `FOAM` parameters, kinematic cuts and parameters of the Standard Model. The configuration can be split into several files and included to `index.conf` file by `.include` *<relative or absolute path to file>* macro.

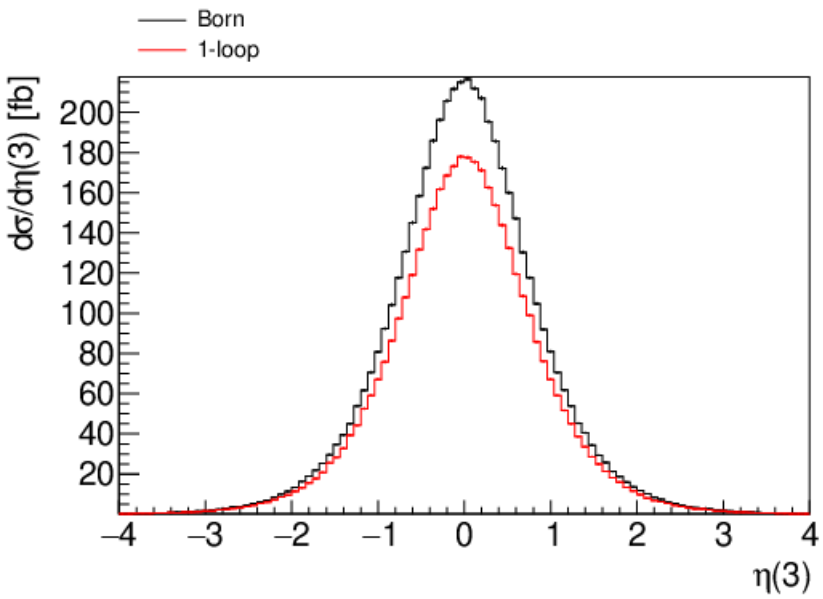

Fig. 1: Example of histogram generated by plot1d.

1.3.1 Cross section structure

The cross section of the electron-positron process at the one-loop level can be divided into four parts:

$\hat{\sigma}^{\text{1-loop}} = \hat{\sigma}^{\text{Born}} + \hat{\sigma}^{\text{virt}}(\lambda) + \hat{\sigma}^{\text{soft}}(\lambda, \omega) + \hat{\sigma}^{\text{hard}}(\omega)$,

and the cross section of the partonic process at the one-loop level can be divided into five parts:

$\hat{\sigma}^{\text{1-loop}} = \hat{\sigma}^{\text{Born}} + \hat{\sigma}^{\text{virt}}(\lambda) + \hat{\sigma}^{\text{soft}}(\lambda, \omega) + \hat{\sigma}^{\text{hard}}(\omega) + \hat{\sigma}^{\text{brdq}}(\omega)$,

where $\hat{\sigma}^{\text{Born}}$ is the Born level cross section, $\hat{\sigma}^{\text{virt}}$ is the contribution of virtual (loop) corrections, $\hat{\sigma}^{\text{soft}}$ corresponds to the soft photon emission, $\hat{\sigma}^{\text{hard}}$ is the hard photon emission part (with energy $E_\gamma > \omega$), and $\hat{\sigma}^{\text{brdq}}$ corresponds to quark mass subtraction. Auxiliary parameters λ (fictitious "photon mass" which regularizes infrared divergences) and ω (photon energy which separates the regions of the phase space associated with the soft and hard emissions) cancel out after summation. Also there is an option in `ReneSANCe` to include in calculation the case with two photons in initial state for NC DY $\hat{\sigma}^{\text{Ibaa}}$ (photon fusion) and the case with one photon and one quark or one gluon and one quark in initial state for NC and CC DY $\hat{\sigma}^{\text{Ibaq}}$ (inverse bremsstrahlung contribution). In the last case there are two parts of quark mass subtraction $\hat{\sigma}^{\text{Ibs1}}$ (NC and CC DY) and $\hat{\sigma}^{\text{Ibs2}}$ (NC DY only).

The calculation of the total cross-section in `ReneSANCe` generator is divided into separate branches according to kinematics and all the branches have their own settings. All branches are named as corresponding part of cross section, while Virt branch means Born+Soft+Virt contributions.

1.3.2 General parameters

`outputDir` [*string*]

set output directory in which results are stored

1.3.3 Process parameters

`pid` [*integer*]

defines a process to calculate. Processes with `pid`>200 can be used only with `ReneSANCe_pp` binary.

- `101`: $e^+e^- \to e^-e^+$,
- `102`: $e^+e^- \to ZH$
- `103`: $e^+e^- \to \mu^-\mu^+$,
- `104`: $e^+e^- \to \tau^-\tau^+$,
- `201`: $hh \to e^-e^+X$,
- `202`: $hh \to \mu^-\mu^+X$
- `203`: $hh \to \tau^-\tau^+X$,
- `211`: $hh \to e^-\bar{\nu}_eX$,
- `212`: $hh \to \mu^-\bar{\nu}_\mu X$,
- `213`: $hh \to \tau^-\bar{\nu}_\tau X$,
- `221`: $hh \to \nu_e e^+X$,
- `222`: $hh \to \nu_\mu \mu^+X$,
- `223`: $hh \to \nu_\tau \tau^+X$.

`ecm` [*double*]

sets collider energy in the center-of-momentum frame.

`alr` [*integer*]

switches generation mode:

- `0`: σ,
- `1`: $\sigma_{RL} - \sigma_{LR}$,
- `2`: $\sigma_{RL} + \sigma_{LR}$,
- `3`: $\sigma_{0L} - \sigma_{0R}$,
- `4`: $\sigma_{0L} + \sigma_{0R}$.

`lamep` [*double*]

polarization degree of the positron.

`lamem` [*double*]

polarization degree of the electron.

`costhcut` [*double*]

cut on $|\cos\theta|$ for both final state particle.

`e5max` [*double*]

cut on maximum energy of hard photon.

`ome` [*double*]

is the parameter separating contributions from soft and hard photon Bremsstrahlung ($\omega = ome\sqrt{s}/2$).

1.3.4 Flags controlling components of the NLO EW computations

`iqed` [*integer*]

- `0`: disables QED corrections,
- `1`: with full QED corrections,
- `2`: only initial state QED radiation (ISR),
- `3`: initial-final QED radiation interference term (IFI),
- `4`: only final state QED radiation (FSR),
- `5`: sum of initial and final state radiation contributions [IFI+FSR],
- `6`: sum of initial state and initial-final QED interference terms [ISR+IFI],
- `7`: sum of initial and final state QED radiation contributions [ISR+FSR].

`iew` [*integer*]

- `0`: disables weak corrections,
- `1`: enables weak corrections.

`iborn` [*integer*]

- `0`: enables NLO corrections,
- `1`: forces LO calculation.

`ifgg` [*integer*]

- `1`: default vacuum polarization $(1 + \mathcal{F}_{\gamma\gamma}(\mathrm{NLO}))$,
- `2`: ressummed vacuum polarization $(1/[1 - \mathcal{F}_{\gamma\gamma}(\mathrm{NLO})])$.

`irun` [*integer*]

- `0`: fixed gauge boson width,
- `1`: running gauge boson width.

`gfscheme` [*integer*]

flag selects electroweak scheme in which the calculation is performed:

- `0`: $\alpha(0)$-scheme,
- `1`: G_μ-scheme,
- `2`: $\alpha(M_Z)$-scheme.

`iqedll` [*integer*]

- `0`: disables LL ISR QED corrections,
- `1`: $O(alphaL)$ gamma corrections. Attention! Double counting is possible! Only for crosscheck.
- `2`: $O(alpha^2L^2)$ gamma corrections,
- `3`: $O(alpha^2L^2)$ el pair corrections,
- `4`: $O(alpha^2L^2)$ mu pair corrections,
- `5`: $O(alpha^3L^3)$ gamma corrections,
- `6`: $O(alpha^3L^3)$ el pair corrections,
- `7`: $O(alpha^3L^3)$ mu pair corrections,
- `8`: $O(alpha^4L^4)$ gamma corrections,
- `9`: $sum^2_{n=2}O(alpha^nL^n)$ corrections,
- `10`: $sum^3_{n=2}O(alpha^nL^n)$ corrections,
- `11`: $sum^4_{n=2}O(alpha^nL^n)$ corrections – best choice.

`deltall` [*double*]

small auxiliary parameter for LL corrections calculation. The result does not depend on it.

`scalell` [*double*]

scale at which LL corrections are calculated.

1.3.5 Special process parameters for hadron mode

`ibeams` [*integer*]

- `1`: corresponds to pp colliding beams,
- `2`: corresponds to $p\bar{p}$ colliding beams.

`pdfname` [*string*]

sets the name of the PDF set connected via LHAPDF.

`pdfmember` [*string*]

sets a member of the PDF set.

`iph` [*integer*]

flag controls which photon induced contributions are calculated.

- `0`: disables photon induced contributions,
- `1`: only $\gamma\gamma$ contribution is enabled,
- `2`: both $\gamma\gamma$ and γq contributions are enabled,
- `3`: only γq contribution is enabled.

1.3.6 Standard Model parameters

`alpha`, `gf`, `alphas`, `conhc` [*double*]

a list of constants and coefficients:
α_{EM}, G_μ, α_S, conversion constant from GeV^{-2} to pb.

`mw`, `mz`, `mh` [*double*]

W, Z, Higgs boson masses.

`wz`, `ww`, `wh`, `wtp` [*double*]

W, Z, Higgs and the top quark widths.

`men`, `mel`, `mmn`, `mmo`, `mtn`, `mta` [*double*]

ν_e, e, ν_μ, μ, ν_τ, τ lepton masses.

`mdn`, `mup`, `mst`, `mch`, `mbt`, `mtp` [*double*]

d, u, s, c, b, t quark masses.

1.3.7 Flags controlling input/output

`explore{Born,Virt,Hard,Brdq,Ibaa,Ibaq,Ibs1,Ibs2}` [*boolean*]

true/false: generates new grid for corresponding branch or use existing grid.

`printROOT` [*boolean*]

true/false: saves events in ROOT format.

`printLHE` [*boolean*]

true/false: saves events in LHE format.

1.4 Changelog

1.4.1 1.3.0 (2022-11-12)

- Implement hadron-hadron collision mode

1.4.2 1.2.1 (2021-12-01)

- Fix an error in the numbering of particles for the leading log contribution

1.4.3 1.2.0 (2021-05-24)

- Add support of higher order corrections to ISR QED in leading log approximation
- Add possibility to set output directory
- Add cut on maximal value of photon energy
- Use single SA module for all 4f processes

1.4.4 1.1.1 (2021-04-16)

- Fix bug in G_μ-scheme realization
- Add new option for gfscheme parameter
- Add new options for alr parameter

1.4.5 1.1.0 (2021-03-05)

- **s-channel processes added:**
  - $e^+e^- \to \mu^-\mu^+$
  - $e^+e^- \to \tau^-\tau^+$

1.4.6 1.0.0 (2020-02-29)

- **Two processes implemented:**
  - Bhabha scattering,
  - Higgs-strahlung process.
- Full 1-loop EW radiative corrections.
- Longitudinal polarization of incoming beams.

INDEX

Symbols

C