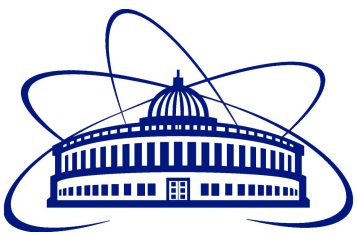
Usage of SANC for polarized event generation

V. Yermolchyk
on behalf of SANC team

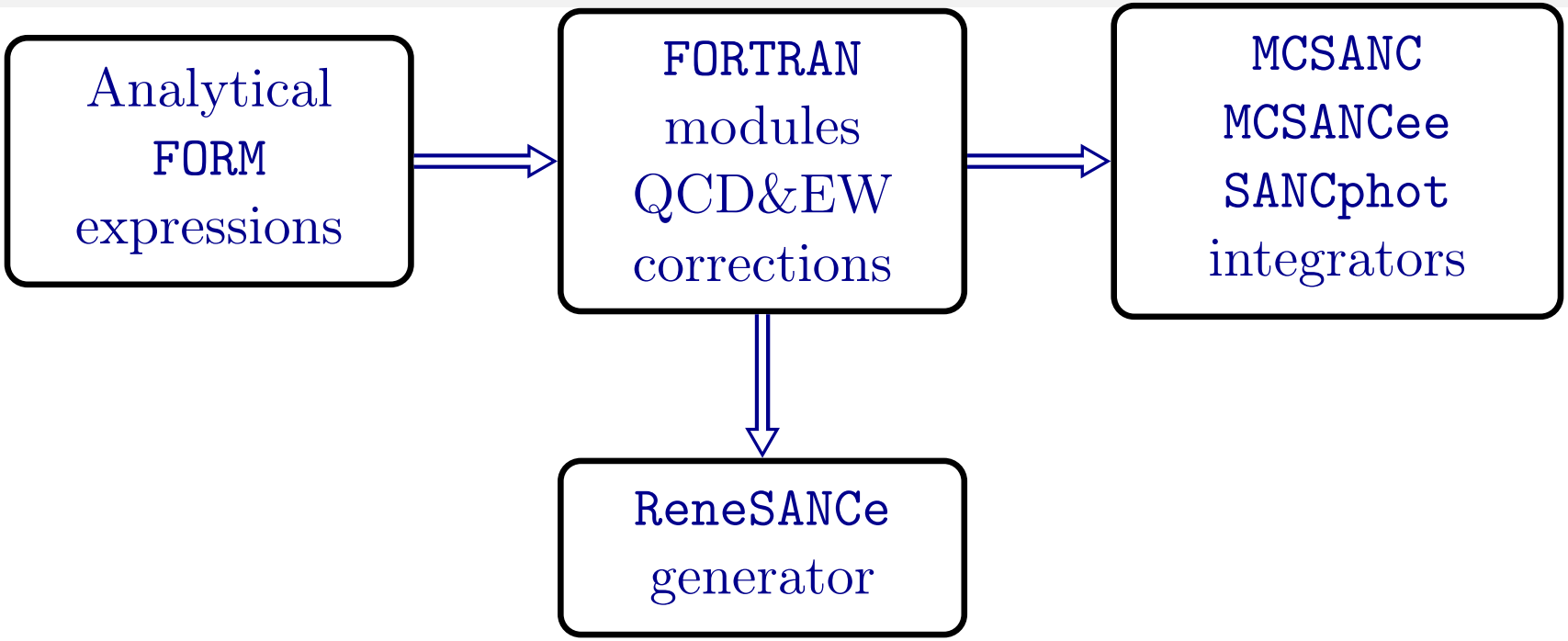
JINR

SPD Physics and MC Meeting N46

22 January 2025



The SANC framework and products family



Publications:

SANC – CPC 174 481-517
MCSANC – CPC 184 2343-2350; JETP Letters 103, 131-136
SANCphot – CPC 294 108929
ReneSANCe – CPC 256 107445; CPC 285 108646

SANC products are available at http://sanc.jinr.ru/download.php
ReneSANCe is also available at http://renesance.hepforge.org

SANC advantages:

- full one-loop electroweak corrections
- higher order corrections
- massive case
- accounting for polarization effects
- full phase space operation
- results of ReneSANCe event generator are thoroughly cross checked
- output of ReneSANCe: events in LHEF, HepMC, HEPEVT or ROOT formats

Installation of the ReneSANCe MC event generator

Stable version 1.4.0 will be available within month. The preliminary version 1.3.8_pol6 is posted on the website `sanc.jinr.ru/download.php`.

Downloads

`ReneSANCe` event generator and documentation is available on `https://renesance.hepforge.org`.

Search Ctrl + K

Contents:

- Downloads
- Getting Started
- Configuration
- Changelog

ReneSANCe: Renewed SANC Monte Carlo event generator

`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.

Dependencies

To compile `ReneSANCe` you need:

- CMake
- Fortran & c++ compilers (with C++11 support)
- ROOT Framework with root-foam module installed
- LHAPDF
- HEPMC3 (optional)

Compilation

```
$ ls -1 ReneSANCe-1.3.8pol6
Analysis/
bin/
build/
cmake/
CMakeLists.txt
external/
Generator/
README.md
share/
work/
```

```
cd build
cmake ..
make -j4 install
```

First run

```
$ ls -1 ReneSANCe-1.3.8pol6
Analysis/
bin/
build/
cmake/
CMakeLists.txt
external/
Generator/
README.md
share/
work/
```

```
cd work
source ../bin/renesance-init.sh
ReneSANCe_pp -p 401 -e 500
```

Configuration options

```
-h,--help                       Print this help message and exit
-f,--file FILE                  Set path to configuretion 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
```

Configuration options, input files

```
$ ls -1 ReneSANCe-1.3.8pol6/share/renesance/examples/201
foam.conf
index.conf
proc.conf
res201.txt
sm.conf
```

proc.conf

```
# Cuts:
eta3cut: true
eta3min: -10.
eta3max: 10.
eta4cut: true
eta4min: -10.
eta4max: 10.
pt3cut: true
pt3min: 1.
pt3max: 50.
pt4cut: true
pt4min: 49.
pt4max: 50.
m34cut: true
m34min: 1.
m34max: 500.
```

Configuration options

List of all parameters is available in schema files:

```
$ ls -1 ReneSANCe-1.3.8pol6/share/renesance/schema/
foam.schema
model.schema
renesance.schema
```

Configuration options, schema files

```
properties: {
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Process id:
pid : {type: integer, minimum: 101, maximum: 403}
    # 101 - e^+e^-  --> e^-e^+
    # 102 - e^+e^-  --> ZH
    # 103 - e^+e^-  --> mu^-mu^+
    # 104 - e^+e^-  --> tau^-tau^+
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# ALR:
alr  : {type: integer, minimum: 0, maximum: 4, default: 0}
    # 0 - sigma, 1 - sigma_RL-sigma_LR, 2 - sigma_RL+sigma_LR,
    # 3 - sigma_0L-sigma_0R, 4 - sigma_0L+sigma_0R
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Longitudinal polarization of initial particles:
lamep :    {type: number, minimum: -1., maximum: 1., default: 0.}
lamem :    {type: number, minimum: -1., maximum: 1., default: 0.}
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# EW scheme:
gfscheme : {type: integer, minimum: 0, maximum: 2, default: 0}
```

Configuration hierarchy

From lower priority to higher:

- Default values.
- Input files.
- Command line.

```
ReneSANCe_pp -f input -p 401 -e 500 -D iscale=1
```

List of processes

ReneSANCe

- 101: $e^+e^- \to e^+e^-$.
- 102: $e^+e^- \to HZ$.
- 103: $e^+e^- \to \mu^+\mu^-$.
- 104: $e^+e^- \to \tau^+\tau^-$.

List of processes

ReneSANCe_pp

- 201: $hh \to e^+e^-$.
- 202: $hh \to \mu^+\mu^-$.
- 203: $hh \to \tau^+\tau^-$.
- 211: $hh \to e^-\bar{\nu}_e$.
- 212: $hh \to \mu^-\bar{\nu}_\mu$.
- 213: $hh \to \tau^-\bar{\nu}_\tau$.
- 221: $hh \to \nu_e e^+$.
- 222: $hh \to \nu_\mu \mu^+$.
- 223: $hh \to \nu_\tau \tau^+$.
- 401: $hh \to g\gamma$.
- 402: $hh \to q\gamma$.
- 403: $hh \to \bar{q}\gamma$.

Cuts

```
costhcut : {type: number, minimum: 0., maximum: 1., default: 1.} # cut on
thetacut : {type: number, minimum: 0., maximum: 90., default: 0.} # cut on
e5max    : {type: number, minimum: 0., maximum: inf, default: inf}
eta3cut: {type: boolean, default: false}
eta3min: {type: number, minimum: -1.e+2, maximum: 1.e+2, default: -2.5}
eta3max: {type: number, minimum: -1.e+2, maximum: 1.e+2, default: 2.5}
eta4cut: {type: boolean, default: false}
eta4min: {type: number, minimum: -1.e+2, maximum: 1.e+2, default: -2.5}
eta4max: {type: number, minimum: -1.e+2, maximum: 1.e+2, default: 2.5}
pt3cut: {type: boolean, default: false}
pt3min: {type: number, minimum: 0., maximum: 1.e+6, default: 25.}
pt3max: {type: number, minimum: 0., maximum: 1.e+6, default: 1.e+6}
pt4cut: {type: boolean, default: false}
pt4min: {type: number, minimum: 0., maximum: 1.e+6, default: 25.}
pt4max: {type: number, minimum: 0., maximum: 1.e+6, default: 1.e+6}
m34cut: {type: boolean, default: false}
m34min: {type: number, minimum: 0., maximum: 1.e+6, default: 50.}
m34max: {type: number, minimum: 0., maximum: 1.e+6, default: 1.e+6}
mt34cut: {type: boolean, default: false}
mt34min: {type: number, minimum: 0., maximum: 1.e+6, default: 0.}
mt34max: {type: number, minimum: 0., maximum: 1.e+6, default: 1.e+6}
y34cut: {type: boolean, default: false}
y34min: {type: number, minimum: -1.e2, maximum: 1.e2, default: -2.4}
y34max: {type: number, minimum: -1.e2, maximum: 1.e2, default: 2.4}
xmin: {type: number, minimum: 0., maximum: 1., default: 1.e-10}
xmax: {type: number, minimum: 0., maximum: 1., default: 1.}
```

Polarization for hadron mode

- iapol=1: $\sigma_{NN} = 0.25 * (\sigma_{++} + \sigma_{+-} + \sigma_{-+} + \sigma_{--})$
- iapol=2: $\sigma_{LL} = 0.25 * (\sigma_{++} - \sigma_{+-} - \sigma_{-+} + \sigma_{--})$
- iapol=3: $\sigma_{LN} = 0.25 * (\sigma_{++} + \sigma_{+-} - \sigma_{-+} - \sigma_{--})$
- iapol=4: σ_{++}
- iapol=5: σ_{+-}
- iapol=6: σ_{-+}
- iapol=7: σ_{--}

Factorization scale

- iscale=1: $Q^2 = s = x_1 * x_2 * Ecm$
- iscale=2: $Q^2 = M_Z^2$
- iscale=3: $Q^2 = s' = M^2(p_3, p_4)$
- iscale=4: $Q^2 = p_T^2$

Output: LHEF, HepMC, HEPEVT, custom ROOT TTree

```
F       IntMC =     0.0038618347 +-    1.4124251e-07  = Result of the MC Integral F
F    mCerelat =    3.6573941e-05    =     Relative error of the MC integral       F
F   <w>/WtMax =       0.98922358    =                MC efficiency, acceptance rate F
F   Sigma/<w> =       0.01646254    =                MC efficiency, variance/ave_wt F
F       WtMax =            0.995    =                WtMax(esp= 0.0005)             F
F       Sigma =      0.016203707    =                variance of MC weight          F
F <OveW>/<W> =                0    =                Contrib. of events wt>MaxWtRej F
F                                                                                   F
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
===================================================================
        MCresult (Born) = 0.0038618347 +- 1.4124251e-07 RelErr = 3.6573941e-05
```

Results/events_LO.lhe

```
<LesHouchesEvents version="1.0">
<!--
File generated with ReneSANCe 1.3.8
on 22 Jan 2025 at 09:46:05
-->
<init>
  2212  2212                  10                  10     3     3 21050 21050    -3     1
    0.00386183468285     1.4124251215e-07                    1            10001
</init>
<event>
     4    10001     1        2.99800033634     0.00729735256928              0.1176
    -2    -1     0     0     0   901                    0                    0       -3.76313116691        3.76313116
     2    -1     0     0   901     0                    0                   -0        2.49178715595        2.49178715
    23     1     1     2     0     0       -1.18588666181        2.75348485416     0.00125452920809        2.99800059
     1     1     1     2     0     0        1.18588666181       -2.75348485416       -1.27259854017        3.25691772
</event>
<event>
```

Conclusion

- The implementation of processes of interest for SPD has begun in the Monte Carlo event generator `ReneSANCe`.
- More processes will be available in the next version 1.4.0.
- Feel free to ask any questions to the developers.

Thank you for paying attention!