Summary of changes to the Guinea-PIG++ program

Strahinja Lukić, INN Vinča, Belgrade, Serbia,

December 2011

Purpose of changes

Adapt and improve the Bhabha-treatment part of the program.

Version used as the starting point

Guinea-PIG++, v1.2.0

Changes in the procedure of rotating and boosting the Bhabha events in the frame of the colliding e^-e^+ pair

In the present version, the velocity of the CM frame with respect to the lab frame, as well as the collision energy $\sqrt{s}'$ and collision axis in the CM frame are calculated from the initial momenta of the colliding e^-e^+ pair. The final momenta of the Bhabha electron, positron and photons, imported from an external calculation with the nominal $\sqrt{s}$ along the z-axis, are scaled by the ratio $\sqrt{s}'/\sqrt{s}$, rotated to match the collision axis and then boosted back to the lab frame.

Format of the input and output files for storing Bhabha samples and results

The following changes were introduced for better tracing of the individual Bhabha events throughout the analysis:

- The Bhabha-event indices are loaded from the `bhabha.ini` file, and stored together with the physical data on the Bhabha particles and photons through all steps of the treatment
- When loading the Bhabha events from the `bhabha.ini` and the `bhabha_photon.ini` files, the event indices of the bhabha-pairs and photons, as well as the number of photons for each event are checked for consistency.

The following was introduced as additional information for the analysis:

- $\sqrt{s}'$ is stored in the zeroth step output file (`bhabha_prod.dat`)

Bhabha input files:

The **`bhabha.ini`** file now contains the following information for each Bhabha event:

```
event index |
px1 | py1 | pz1 | E1 |
px2 | py2 | pz2 | E2 |
number of photons
```

This information can be arranged in one row or in several rows like in the example above.

The **bhabha_photon.ini** file now contains the following information for each photon produced in the Bhabha events:

`event index | $p_{x\gamma}$ | $p_{y\gamma}$ | $p_{z\gamma}$ | $E_\gamma$ |`

Bhabha output files:

The **bhabha_prod.dat** file now contains the following information about each of the used Bhabha events in the zeroth stage of treatment (before scaling, rotation or boost):

`event index | $\sqrt{s'}$ |`

E_{mot1} | E_1 | v_{x1} | v_{y1} | v_{z1} |

E_{mot2} | E_2 | v_{x2} | v_{y2} | v_{z2} |

`number of photons`

These information are all arranged in one row

The **bhphoton_prod.dat** file now contains the following information about each of the photons produced in the Bhabha events in the zeroth stage of treatment (before scaling, rotation or boost):

`photon number | event index | $E_\gamma$ | $p_{x\gamma}$ | $p_{y\gamma}$ | $p_{z\gamma}$ |`

The **pairs0.dat** and **pairs.dat** files now contain some additional information (the event index) **in the case of the Bhabha particles**. The pairs produced in other processes are still stored in the **same way as before**. If one day Bhabhas should be treated in the same run with pairs produced in other processes, a process tag should probably be added for each particle (or pair). In the present version, the number of columns depends on whether the particle comes from a Bhabha event or from other type pf pair-producing events.

The **pairs0.dat** file now contains the following information about each Bhabha event after scaling, rotation and boosting back to the lab frame (physically, immediately after the Bhabha scattering event):

`event index | $E_1$ | $v_{x1}$ | $v_{y1}$ | $v_{z1}$ | $x_1$ | $y_1$ | $z_1$ |`

`event index | $-E_2$ | $v_{x2}$ | $v_{y2}$ | $v_{z2}$ | $x_2$ | $y_2$ | $z_2$ |`

The coordinates *x, y* and *z* are expressed in nm. Velocity components are normalized (expressed in the units of *c*).

The **pairs.dat** file contains the same information as the `pairs0.dat` file, but after tracking (when the particles are "far" from the IP, and have been deflected by the EM field of the bunch they left):

`event index | $E_1$ | $v_{x1}$ | $v_{y1}$ | $v_{z1}$ | $x_1$ | $y_1$ | $z_1$ |`

`event index | $-E_2$ | $v_{x2}$ | $v_{y2}$ | $v_{z2}$ | $x_2$ | $y_2$ | $z_2$ |`

The rows in the `pairs.dat` file are shuffled. They can be sorted using the following shell instruction:

```
sort -k 1n -k2gr -o pairs_sorted.dat pairs.dat
```

The second column is used as the secondary key (`-k2gr`) to order the electron before the positron in each event by the sign before the energy value.

The coordinates *x, y* and *z* are expressed in nm. Velocity components are normalized (expressed in the units of *c*).

The **`bhphotons.dat`** file now contains the following information about each of the photons produced in the Bhabha events after scaling, rotation and boosting back to the lab frame:

`photon number | event index | E`$_\gamma$ `| p`$_{x\gamma}$ `| p`$_{y\gamma}$ `| p`$_{z\gamma}$ `|`

Specific changes in the code

Unused functions

The following functions are not used in the Bhabha treatment anymore and their declarations have been outcommented:

`BHABHA::lorent_bhabha()`

`BHABHA::lorent_bhabha_back()`

`BHABHA::frame_change_part_of_bhabha()`

`BHABHA::lorent_bhabha_transformation()`

`BHABHA_PHOTON_SAMPLES::set_numero_bhabha()`

All Lorentz boosts are now performed by the function `BHABHA::fourboost()`, and the calculation of the frame is performed directly in the `BHABHA::boost_bhabha()` function.

The Bhabha event numbers are now loaded directly with the Bhabha samples from the .ini files, and it is not desirable to be able to change this number later in the program.

New and overloaded functions

`BHABHA::fourboost(float &en, float &px, float &py, float &pz, double beta_x, double beta_y, double beta_z)`

Boosts the four-vector passed in the first four parameters to the frame that moves with the velocity beta (the last three parameters) with respect to the starting frame.

To enable consistent indexing of bhabha pairs, the following overloaded function was introduced:

`PAIR_BEAM::new_pair(const unsigned int evtIndex, <remaining arguments as before>)`

The index passed in the first argument is stored in the existing data member `PAIR_PARTICLE_::label_` of the new `PAIR_PARTICLE_` object.

To store the event indices in all output files concerning the Bhabhas, the following overloaded functions were introduced:

`IFILE_IN_OUT::save_object_on_persistent_file(const int, const ABSTRACT_IO_CLASS* const)`

`FILE_IN_OUT::save_object_on_persistent_file(const int, const ABSTRACT_IO_CLASS* const)`

`FILE_IN_OUT_ASCII::save_object_on_persistent_file(const int, const ABSTRACT_IO_CLASS* const)`

To use the above overloaded functions to save the Bhabha samples, the following new functions were also introduced:

`PAIR_BEAM::save_bhabhas_on_file()`, analogous to `PAIR_BEAM::save_pairs_on_file()`, but invoking the above overloaded functions

`PAIR_BEAM::save_bhabhas0_on_file()`, analogous to `PAIR_BEAM::save_pairs0_on_file()`, but invoking the above overloaded functions

The function `GUINEA::save_results_on_files()` now invokes these new functions if `switches.get_do_bhabhas()` returns true, and the old "`save_pairs...`" functions otherwise.