GUINEAPIG Manual
(SLAC NLC webpage version)

This manual is a revision and update by K.Thompson of the original manual which appeared in Appendix A of Daniel Schulte's Ph.D. thesis. It is intended to go with the version of Guineapig on the SLAC NLC Beam Delivery webpage (http://www-sldnt.slac.stanford.edu/nlc/beamdeliveryhome.htm), and will be kept updated to reflect modifications to this program.

A call to GUINEAPIG is done with three command line arguments. The first is the name of the accelerator, the second the name of a computational parameter set and the third is the output file name. The parameters for the accelerator and the computational parameters are searched for in a data file "acc.dat".

I FORMAT OF MAIN INPUT FILE

This file, acc.dat, contains the definitions of the accelerators and the computational parameter sets.

An entry for an accelerator is defined as

$ACCELERATOR:: name {varl=valuel;var2=value2*value3; ... }

where the definitions in the braces can extend over several lines. The variables can be of different types. The INT and DOUBLE contain single integer and double precision values. The INT_2 and DOUBLE_2 type variables contain two integer or double precision elements each. The elements can be accessed by appending .1 or .2 onto the name of the variable, as for example sigma_x.1=1.0;. Both elements can set simultaneously to the same value via sigma_x=l.0;, but sigma_x can not be used in the definition of a value as in test=sigma_x;. One has then to use test=sigma_x.1;. The last types of numerical values are INT_MIRROR and DOUBLE_MIRROR. They can be used as the INT_2 and DOUBLE_2 except that offset_x=l.0; will be equal to offset_x.1=1.0; offset_x.2=-1.0;. Using a wrong name for a variable will cause the program to write an error message and to stop. The values attributed to a variable can be simple expressions containing brackets, multiplication, division, addition and subtraction. In addition to constants, the names of existing variables may also be used.

particles (DOUBLE_2) The number of particles per bunch in units of $[10^{10}]$.

energy (DOUBLE_2) The energy of the particles in GeV.

espread (DOUBLE_2) The RMS value of the relative energy spread of the beam particles. The shape of the distribution is set via which_espread. NOTE: You must also turn on the switch do_espread to have the beam energy spread included in the simulation!

which_espread (INT_2) The energy distribution of the beam particles. If which_espread is 0 no energy spread, if it is 1 a flat distribution, for 2 two peaks, and for 3 a Gaussian distribution is assumed. Default is 1.

charge_sign (DOUBLE) The relative sign of charge of the two beams_is -1 for an e+e- collider and 1 for ee. If it is set to zero no beam_beam force is assumed. Default is -1.

sigma_x (DOUBLE_2) The horizontal beamsize in nm.

sigma_y (DOUBLE_2) The vertical beamsize in nm.

sigma_z (DOUBLE_2) The longitudinal beamsize in μm, the RMS value should be used in case of distributions different from the Gaussian.

f_rep (DOUBLE) The repetition frequency of the collider, that is the number of bunch trains per second. This value is currently not used since all results are per bunch crossing.

n_b (INT) The number of bunches per bunch train. This value is currently not used.

dist_z (INT_2) The longitudinal charge distribution, for dist_z=0 a normal, for dist_z=l a homogeneous distribution is used (remember to set cut_z appropriately when dist_z=1 is used, i.e. cut_z=$\sqrt{3}$. Default is 0.

trav_focus (INT) For a value not equal to zero the beams are assumed to have a travelling focus.

emitt_x (DOUBLE_2) The normalised horizontal emittance in 10^{-6}mrad.

emitt_y (DOUBLE_2) The normalised vertical emittance in 10^{-6}mrad.

beta_x (DOUBLE_2) The horizontal beta function in mm.

beta_y (DOUBLE_2) The vertical beta function in mm.

offset_x (DOUBLE_MIRROR) The horizontal offset in nm. Default is 0.0.

offset_y (DOUBLE_MIRROR) The vertical offset in nm. Default is 0.0.

waist_x (DOUBLE_MIRROR) The shift in μm of the horizontal waist with respect to the plane of collision. Default is 0.0.

waist_y (DOUBLE_MIRROR) The shift of the vertical waist. Default is 0.0.

angle_x (DOUBLE_2) The horizontal angle in rad, default is 0.0. Keep in mind that the system of beam 2 is lefthanded. To simulate an angle θ_0 between the

two beams (not compensated with the crab crossing scheme) one simply has to set angle_x=$0.5\theta_0$;. It is important to note that the user is responsible for choosing a grid that is sufficiently large to contain the beams.

angle_y (DOUBLE_2) The vertical angle in rad. Default is 0.0.

angle_phi (DOUBLE_2) The transverse tilt angle of the bunch. Default is 0.0.

It is only necessary to define two variables out of the combination of emittance, beta function and beam size. If all three are defined a check for agreement is performed.

Equivalently one has for the computational parameter set

$PARAMETERS:: name {var1=value1;var2=value2; ...}

In this declaration all the parameters from the definition of the accelerator can be addressed and in addition the following:

n_x (INT) The number of cells in horizontal direction.

n_y (INT) The number of cells in vertical direction.

n_z (INT) The number of cells in longitudinal direction.

n_t (INT) The number of timesteps used to move a slice to the next slice of the other bunch.

n_m (INT_2) The number of macroparticles used (per beam).

cut_x (DOUBLE) Size of the grid in horizontal direction in nm. If less than zero, 3·max(sigma_x.1,sigma_x.2) will be used.

cut_y (DOUBLE) The same as cut_x but for the vertical direction

cut_z (DOUBLE) Size of the grid longitudinally in μm.

integration_method (INT) Selects direct (=1), fast Fourier (=2), or iterative method (=3). The default, which is 2, is advised (it is necessary to use powers of two for n_x and n_y in this case)

force_symmetric (INT) If this value is not equal 0 the beams are assumed to be up-down and left-right symmetric. Default is 0.

rndm_save (INT) If not equal to 0, the status of the random number generators is stored in the file rndm.save. Default is 1.

rndm_load (INT) If not equal to 0, the file rndm.save with the status of the random number generators is loaded. If the file does not exist the generators are initialised with the standard values. Default is 1.

do_lumi (INT) If not zero a file lumi.dat containing the energies for e+e- scattering is produced. This file may be used as an input file for the event generator GHOST. Default is 0.

num_lumi (INT) The maximal number of scatters stored in lumi.dat. The actual number will be at least half of this unless the value of lumi_p is too low. The default value is 10000.

lumi_p (DOUBLE) The initial scaling factor for the storage of particles into the lumi.dat file. The value will be adjusted to stay within the allowed maximum number of entries. If lumi_p is too small the number of entries in the file will be less than than half the allowed number. If the value is very high a small overhead is caused at the start of the program till it is adjusted to a sensible value. Default is 1.e-23.

do_gglumi (INT) If not zero a file gglumi.dat containing the energies for $\gamma\gamma$ scattering is produced. Default is 0.

num_gglumi (INT) The maximal number of scatters stored in gglumi.dat. The actual number will be at least half of this unless the value of gglumi_p is too low. The default value is 10000.

gglumi_p (DOUBLE) The initial scaling factor for the storage of particles into the gglumi.dat file. The value will be adjusted to stay within the allowed maximum number of entries. If gglumi_p is too small the number of entries in the file will be less than than half the allowed number. If the value is very high a small overhead is caused at the start of the program till it is adjusted to a sensible value. Default is 1.e-23.

store_beam (INT) If not equal to 0, the spent beams will be stored in the files beaml.dat and beam2.dat respectively. The files are in ASCII. The first of the three values per line is the energy of the particle in GeV, the second and the third are the angles with respect to the beam axis for the horizontal and vertical direction. Default is 0.

ecm_min (DOUBLE) Energy cut-off for output variable lumi_ee_high. Default is 0.

storebeam_ptmin (DOUBLE) Lower limit on transverse momentum for macroparticles to be written to beam1.dat and beam2.dat [GeV]. Default is 0.

storebeam_ptmax (DOUBLE) Upper limit on transverse momentum for macroparticles to be written to beam1.dat and beam2.dat [GeV]. Default is 1.e20.

storebeam_angmin (DOUBLE) Lower limit on angle for macroparticles to be written to beam1.dat and beam2.dat [radians] Default is 0.

storebeam_angmax (DOUBLE) Upper limit on angle for macroparticles to be written to beam1.dat and beam2.dat [radians] Default is 1.e20.

electron_ratio (DOUBLE) Part of the electrons to be used for luminosity spectrum and background calculations. It has to be within $[0 : 1]$. If it set to a value smaller than 10^{-6} these calculations are completely suppressed except for the total luminosity. The choice of this parameter effects the precision (and time consumption)

of the calculations but not the results (except for statistical reasons). Default is 1.

do_photons (INT) If not zero store the photons (needed for background calculations). Default is 0.

photon_ratio (DOUBLE) Part of the photons to be used for background calculations. The same considerations apply as for electron_ratio except that for setting photon_ratio to a value below 10^{-6} (the same as setting do_photons=0;) no photon-electron or photon-photon luminositv will be calculated. Default is 1.

do_hadrons (INT) If not zero calculate hadrons. For do_hadrons=1, Chen-Peskin cross section ($\sigma_{H,3}$ in Chapter 5 of Schulte thesis) is used for photon-photon cross-section; for do_hadrons=2, Pythia cross section ($\sigma_{H,2}$ in Chapter 5 of Schulte thesis) is used. Default is 0.

store_hadrons (INT) If not zero the energies of the initial photons of the hadronic interactions during one bunch crossing are stored in the file hadron.dat. Default is 0.

hadron_ratio (DOUBLE) The weight factor with which the cross section of the hadronic interaction is scaled for the storage in the file. Default is 1.e5.

do_jets (INT) If not zero calculate minijets. Default is 0.

store_jets (INT) If not zero the final states of the minijets are stored in the file minijet.dat. Default is 0.

jet_ptmin (DOUBLE) Minimal transverse momentum of the jets in GeV/c. Default is 2.0.

jet_ratio (DOUBLE) Factor with which the probability of jet production is enhanced (needed for producing a file of these jets). Default is 1.e5.

jet_log (INT) If not zero uses different way to select the energy of the parton in a resolved photon. The default which gives better results is 1.

do_pairs (INT) If not zero calculate pairs. Default is 0.

pair_ratio (DOUBLE) Probability of a produced pair to be stored (used to reduce the total number of particles to be tracked). This value has to be in [0:1]. (Note this also affects scattered compton particles stored in pairs.dat, when do_compt is turned on.) Default is 1.0.

pair_q2 (INT) Selects the Q^2 scale for the pair production, if 0 then $Q^2 = m^2$, if 1 then $Q^2 = m^2 + p_t^2$, if 2 then $Q^2 = \hat{s}/4$. Default is 1.

track_pairs (INT) If not zero the pairs are tracked through the fields of the beams. In order to get reasonable results it is necessary to chose an appropriate value of grids. Default is 0.

grids (INT) The number of grids used for the tracking of the pairs. Should be set

to 7 if they are tracked, zero otherwise. Default is 0.

pair_ecut (DOUBLE) Minimal energy in GeV the particles from pair creation need to have to be stored. Default is electron rest mass.

pair_step (DOUBLE) Scaling factor for the step size of the pairs; if the value is increased the step size is decreased. Default is 1.0.

pairs_ptmin (DOUBLE) Lower limit on transverse momentum for pair particles to be written to pairs.dat [GeV]. Default is 0.

pairs_ptmax (DOUBLE) Upper limit on transverse momentum for pair particles to be written to pairs.dat [GeV]. Default is 1.e20.

pairs_angmin (DOUBLE) Lower limit on angle for pair particles to be written to pairs.dat [radians]. Default is 0.

pairs_angmax (DOUBLE) Upper limit on angle for pair particles to be written to pairs.dat [radians]. Default is 1.e20.

beam_size (INT) If not zero the beam size effect is switched on. For beam_size=1 the impact parameter is calculated as $b = \hbar/q_\perp$, for beam_size=2 this is used as the width of a Gaussian distribution from which the impact parameter is selected. The default and recommended choice is 1.

do_eloss (INT) If not zero the beam particles lose energy due to beamstrahlung. The default is 1.

do_espread (INT) If not zero the beam particles energy spread is taken into account. The energy of the particles is chosen for each collision. Default is 0.

do_isr (INT) If not zero the initial state radiation is taken into account. Default is 0.

do_compt (INT) If not zero the program produces soft particles via the bremsstrahlung process. At present these are stored in the pairs.dat file. Default is 0.

compt_emax (DOUBLE) Maximum energy of compton-scattered electrons to store. Default is 10000 GeV.

compt_x_min (DOUBLE) Minimum momentum transfer for compton-scattered electrons. Default is 0.01.

load_beam (INT) If not zero the initial beams are loaded from the file beam.ini. Default is 0.

load_photons (INT) If not zero the initial photon beams are loaded from the file photon.ini. This is useful for the e-γ and $\gamma\gamma$ options. Default is 0.

store_photons (INT) If not zero the photons of the spent beams are stored in the file photon.dat. Default is 0.

do_prod (INT) If not zero soft particles with a fixed energy prod_e are produced during the beam-beam interaction. The cross section is assumed to be constantly prod_scal. Default is 0.

prod_e (DOUBLE) Energy (in GeV) of the particles produced with the switch do_prod. Default is 0.0.

prod_scal (DOUBLE) The cross section for the production of particles with the switch do_prod. Default is 1.e-29.

do_cross (INT) If not zero the file cross.ini is loaded and the contained cross section is folded with the luminosity spectrum. Default is 0.

II ADDITIONAL OPTIONAL INPUT FILES

A electron.ini

If the switch load_beam is set the initial distribution of electrons and positrons is load from the file electron. ini. On each line this file contains seven parameters describing one particle. The first value gives the particles energy in GeV — if it is negative its absolute value is used and the particle is attributed to beam 2 instead of beam 1. The second value gives the degree of longitudinal polarisation of the particle ranging from -1 to 1. The third and fourth value contain the angles in x and y the particle has in radians. The last three give the position of the particle in the bunch. The first two give the x and y in nm when the particle crosses the plane z = 0. The last value gives the longitudinal position in μm within the bunch — negative values indicate that the particles are in the head of the bunch.

The number of particles for bunch j in the file may not exceed the number n_m.j, but it can be less. The weight of the macro particles is calculated using particles.j/n_m.j. The particles within each bunch have to be sorted according to their z-position.

B photon.ini

This file is comparable to electron.ini. It is used when the switch load_photons is used. Each line of the file contains the seven parameters of one photon. They have the same meaning as in the file electron.ini — the second value is the degree of circular polarisation. The number of photons is not restricted nor need they be ordered.

C rndm.save

This file can be read with the switch rndm_load. If it does not exist the standard initialisation of the random number generators is used. The file can be created using the switch rndm_save. Since it is binary it cannot be transported from one machine to another.

D cross.ini

This file contains in the first line the number of energies at which the cross section is defined n and the number of cross sections per energy n_val. The next two values on this line define how the axis are treated; at the moment they should be simply zero. The following (n) lines each contain the energy at which the cross section is evaluated and the apropriate n_val cross sections. If all cross sections are evaluated simultaneously or if only the first is defined by a flag inside the program: MCROSS. This part of the program is likely to change and comments and suggestions are welcome (as for the rest).

III THE MAIN OUTPUT FILE

The output file contains the settings of the variables in almost the same form as the input file. In addition it of course contains the results. The view program gpv allows to write the results onto the screen (very bad layout) and to produce data files from the histograms stored in the file. They are prepared to be used with GNUPLOT with set data style lines. The variables in the output file containing the results are

lumi_fine: The luminosity in m^{-2} per bunch crossing as calculated from the charge densities.

lumi_ee: The luminosity in m^{-2} per bunch crossing as calculated from the colliding beam particles. This value is zero if electron_ratio=0.0; is used.

lumi_ee_high: The luminosity in m^{-2} per bunch crossing above energy ecm_min.

lumi_eg: The $e\gamma$ luminosity in m^{-2} per bunch crossing. The value is calculated only if electron_ratio and photon_ratio are not equal zero.

lumi_ge: The γe luminosity in m^{-2} per bunch crossing.

lumi_gg: The $\gamma\gamma$ luminosity in m^{-2} per bunch crossing.

lumi_gg_high: The $\gamma\gamma$ luminosity with a centre of mass energy of more than gg_cut in m^{-2} per bunch crossing.

upsmax: Maximal value of Υ that occurred during the interaction.

miss: The relative amount of interacting particles that were outside the grid during one timestep. Should be small compared to 1.

ephot: The average photon energy of the photons of beam 1 or 2.

bpm_vx: The average angle in x in μrad of the particles of each beam after the interaction.

bpm_sig_vx: The RMS angular spread in x of the spent beams.

bpm_vy: The same as bpm_vx only for the y direction.

bpm_sig_vy: The same as bpm_sig_vx only for the y direction.

hadron_cut: The minimal photon-photon centre of mass energies for hadronic events.

hadron_ee: The number of hadronic events per bunch crossing due to the virtual photons in ee collisions.

hadron_eg: The number of hadronic events per bunch crossing due to $e\gamma$ collisions.

hadron_ge: The number of hadronic events per bunch crossing due to γe collisions.

hadron_gg: The number of hadronic events per bunch crossing due to $\gamma\gamma$ collisions.

jets0: The number of minijet events per bunch crossing due to the direct process.

jetsl: The number of minijet events per bunch crossing due to the once resolved process.

jets2: The number of minijet events per bunch crossing due to the twice resolved process.

n_pairs: Total number of pair particles produced per bunch crossing.

e_pairs: Total energy of these particles.

n_BW: The number of particles due to the Breit-Wheeler process.

e_BW: The total energy of the particles due to the Breit-Wheeler process.

n_BH: The number of particles due to the Bethe-Heitler process.

e_BH: The total energy of the particles due to the Bethe-Heitler process.

n_LL: The number of particles due to the Landau-Lifshitz process.

e_LL: The total energy of the particles due to the Landau-Lifshitz process.

high_BW: The number of particles due to the Breit-Wheeler process with a transverse momentum of more than 20 MeV and an angle with respect to the beam axis of more than 150 mrad.

high_BH: The same as high_BW for the Bethe-Heitler process.

high_LL: The same as high_BW for the Landau-Lifshitz process.

high_tot: The total number of these particles.

high_e: Their total energy.

de: The average energy loss of the beam particles in GeV.

cross: The results of the cross section calculation.

cross_var: The errors of the cross section calculation.

IV THE OUTPUT FILES

There are several additional output files which can usually be produced with the switches. So the particles from pair pair creation will be stored in pairs . dat .

A pairs.dat

This file is an ASCII file. Each lines contains one particle. The first value in a line is the particle's energy in GeV. If it is negative the particle is a positron otherwise an electron. The next three values are the velocities in x, y, and z directions divided by the speed of light. Following that are the x, y, z positions of the particle.

B hadron.dat

This file contains the energies of colliding photons that produced a hadronic event. Since the number of these events is small per bunch crossing, the number will be enhanced by the factor hadron_ratio. The file is an ASCII file. Each line contains the energy of the photon from beam 1 and the one from beam 2 in GeV.

C minijet.dat

This file contains the four vectors of the partons before fragmentation on each line. The first two values are the energies of the photon remnant in GeV. They are negative for non-existent remnants. The next two values contain the longitudinal momenta of the two hard jets in GeV. A positive value points into direction of beam 1. The next value is the tranverse momentum of the final state partons in

GeV. In this model it is the same for both jets. The last value is an integer giving the process responsible for the jets.

D lumi.dat

Each line of this file contains the energies of two colliding particles in GeV.

E beam.dat

Actually two files beaml.dat and beam2.dat are produced containing the particles of the first or second beam, respectively. Each line contains the energy of the particle and the angles in x and y in radians. Following that are the z-position within the bunch and then the (x, y)-position at the intersection plane (The particles position is extrapolated back to this plane after the interaction, the real position was of course different.)

F photon.dat

This file contains the photons after the interaction; each line gives the photon energy (if negative, the photon is from the second beam) and angles in x and y.

G rndm.save

This file is automatically created if the switch rndm.save is set. It is a binary file and cannot be transported from one machine to another. With the switch rndm_load it can be reread in the next run.