User Manual for the Halo and Tail generator HTGEN

Ijaz Ahmed, Helmut Burkhardt, Miriam Fitterer

December 2008

Abstract

This is the user manual for the halo and tail generator code HTGEN. It contains a short description, installation instructions and examples of how to use the code. The code and up to date information are available from the web from via `http://hbu.home.cern.ch/hbu/HTGEN.html`

1 Introduction

HTGEN allows studying of potential sources of halo and tail generation in linear colliders with development of analytical models of halo, estimates of halo population, development of computer models for halo/tail generation, simulation studies of halo/tail generation and benchmarking candidate Processes.
The following particle processes are included in HTGEN:

1. **beam gas elastic scattering (Mott scattering):**
   In the elastic process of Mott scattering, the incident beam particle is deflected by the Coulomb potential of the particles in the residual gas. The differential cross section is:
   $$\left(\frac{d\sigma}{d\Omega}\right)_{\text{Mott}} = \left(\frac{Zr_e}{2\gamma\beta^2}\right)^2 \frac{1-\beta^2\sin^2\frac{\theta}{2}}{\sin^4\frac{\theta}{2}} \tag{1}$$
   where Z is the charge of the nucleus, r_e th classical electron radius and γ the Lorentz facotr E/mc^2 of the electron. The total cross section is obtained by integration over the solid angle. Relevant for halo production are scattering angles which exceed the beam divergence, or roughly
   $$\theta_{\min} = \sqrt{\epsilon/\beta_y} = \sqrt{\epsilon_N/\gamma\beta_y} \tag{2}$$
   where $\epsilon_N = \gamma\epsilon$ is the normalized vertical emittance and β_y the local vertical beta function. Using the approximation $\beta \approx 1$ for highly relativistic electrons, we get as a simple estimate for the integrated crossection:
   $$\sigma_{\text{Mott}} \approx \begin{cases} \frac{4\pi Z^2 r_e^2}{\gamma^2 {\theta_{\min}}^2} & \text{for } \theta_{\min} < 10^{-6} \text{ rad} \\ \frac{2\pi Z^2 r_e^2}{\gamma^2(1-\cos\theta_{\min})} & \text{for } \theta_{\min} > 10^{-6} \text{ rad} \end{cases} \tag{3}$$
   Because of numerical instabilities it is better take the taylor series of *cos* for small angles.

2. **inelastic scattering (Bremsstrahlung):**
   The energy spectrum for Bremsstrahlung is:
   $$\frac{d\sigma}{dk} = \frac{A}{N_A X_0}\frac{1}{k}\left(\frac{4}{3} - \frac{4}{3}k + k^2\right) \tag{4}$$
   where k is the photon energy in units of the beam energy, N_A the Avogadro constant and X_0 the radiation length.
   A good approximation for the radiation length is:
   $$X_0 = \frac{A}{4\alpha r_e^2 N_A Z(Z+1)\ln(287/\sqrt{Z})} \quad [\text{ g cm}^{-2}] \tag{5}$$

Integration of (4) over k from $k = k_{min}$ to $k = 1$ yields the crossection:

$$\begin{aligned}\sigma_{Brem} &= \frac{A}{N_A X_0}\left(-\frac{4}{3}\ln k_{min} - \frac{5}{6} + \frac{4}{3}k_{min} - \frac{k_{min}^2}{2}\right) \\ &\overset{(5)}{=} 4\alpha r_e^2 Z\,(Z+1)\ln\left(287/\sqrt{Z}\right)\left(-\frac{4}{3}\ln y_{min} - \frac{5}{6} + \frac{4}{3}y_{min} - \frac{y_{min}^2}{2}\right) \quad (6)\end{aligned}$$

2 Installation

HTGEN is hosted on a CERN Savannah server. You can browse the latest source code here. There is no more need for external libraries for HTGEN (the CLHEP dependence was removed).
Optionally, there are HTGEN standalone tests which use ROOT to make histograms.
For PLACET+HTGEN you will need tcl and gsl libraries. These are all open software libraries which can easily be found on the web and installed on popular linux / unix platforms including macosx.
HTGEN provides both interface libraries for tracking, for example with PLACET as well as standalone tests with analytic estimates of the flux of halo particles with sample input for the ILC and CLIC.
The interface with PLACET is on the PLACET server (mainly in background.cc, background.h).

2.1 Instructions HTGEN

if your shell is csh or tcsh:

```
setenv CVS_RSH ssh
```

if it is bash:

```
export CVS_RSH ssh
```

Getting the code using anonymous checkout should always work with

```
cvs -d :pserver:anonymous@isscvs.cern.ch:/local/reps/htgen checkout
htgen
```

The size of HTGEN is only about 2 MBytes so that you can still do this rather quickly using a modem at home.
If you are a registered CERN user you can also checkout with your password via

```
cvs -d :ext:isscvs.cern.ch:/local/reps/htgen checkout htgen
```

You can also download the tarball file from the cern savannah server (outside CERN).
The very first time or after system changes, you should go then to the HTGEN directory and run through the setup.

```
cd htgen
sh setup.sh
```

and supply the information needed to find the extra libraries in your environment. Type return for unknown or unavailable locations. This script will generate two files env.sh and env.csh.

```
source env.csh
```

to set the environment variables, then compile

```
make libhtgen
make libhtplacet  (if PLACET is defined)
make libhtmerlin  (if MERLIN is defined)
```

if your shell is bash:

```
export CVS_RSH=ssh
cvs -d :pserver:anonymous@isscvs.cern.ch:/local/reps/htgen checkout
htgen
```

or at CERN

```
cvs -d :ext:isscvs.cern.ch:/local/reps/htgen checkout htgen
```

or downloading the tar.
Run the setup script.

```
cd htgen
sh setup.sh
```

and give the information request or hit return

```
source env.sh
```

to set the environment variables, then compile

```
make libhtgen
make libhtplacet   (if PLACET is defined)
make libhtmerlin   (if MERLIN is defined)
```

2.2 Instructions PLACET+HTGEN

You can use HTGEN in PLACET for detailed tracking of core + halo.
For this, you need the HTGEN libraries (libhtgen and libhtplacet produced as described above) and the source code for PLACET, available from the savannah server ; PLACET also requires gsl, the Gnu Scientific Library.
Get the source using anonymous cvs

```
cvs -d :pserver:anonymous@isscvs.cern.ch:/local/reps/placet \
checkout placet-development
```

As PLACET weighs now about 10 MByte, this should still be reasonably fast. As CERN user you can also use

```
cvs -d :ext:isscvs.cern.ch:/local/reps/placet checkout placet-development
```

Now you can create the placet-htgen library:

```
make libhtplacet
```

than go into the PLACET directory

```
cd placet-development
```

and run the configuration script

```
./configure --prefix=$PLACET_DIR --enable-htgen --with-htgendir=$HTGEN_DIR
make
```

This will create the PLACET executable with the background tracking modules included.

3 Examples

3.1 HTGEN

See the CLIC and ILC files in the config directory which comes with the HTGEN installation for HTGEN standalone runs see the test directory which comes with the HTGEN installation. There are few particular examples. Try to check it.
Make always sure you have the correct environment variables set (first time using setup, then just

```
source env.csh
```

or

```
source env.sh
```

depending on your shell, see the installation instructions)

```
cd $HTGEN_DIR/test ; make TestProcessLinac
```

This will produce the executable TestProcessLinac in the `$HTGEN_DIR/test/bin` directory. If your working directory is `/tmp/$LOGNAME/`, you can run there the test program using

```
cd /tmp/$LOGNAME/ ;
$HTGEN_DIR/test/bin/TestProcessLinac $HTGEN_DIR/config/ILC-LIN.par
```

where `$HTGEN_DIR/config/ILC-LIN.par` is the input filename with the parameters for this run.

```
cd $HTGEN_DIR/test ; make TestProcess
```

This will produce the executable TestProcess in the `$HTGEN_DIR/test/bin directory.` If your working directory is `/tmp/$LOGNAME/`, you can run there the test program using

```
cd /tmp/$LOGNAME/ ;
$HTGEN_DIR/test/bin/TestProcess $HTGEN_DIR/config/ILC-LIN.par mott
```

The program prints cross sections for the mott process and produces a file `mott.root` file with histograms. Using `brem` rather than `mott` as second argument will do the equivalent for the Bremsstrahlungs process.
To display the histograms you can use root like

```
root
TFile f1("mott.root");
electron_theta.Draw();
```

similarly:

```
cd $HTGEN_DIR/test ; make TestMultipleScatt
```

3.2 PLACET+HTGEN

A good starting point for PLACET+HTGEN is to checkout or update to the latest distribution versions of HTGEN and placet-development. Make sure you installed PLACET using the `--enable-htgen` option and you run using the module `placet-htgen`. See the directory `$PLACET_DIR/examples/htgen` which comes with the PLACET distribution. Go into that directory and type

```
placet-htgen htgen.bds.tcl
```

or go into your favorite work directory, define this in the tcl file using set `script_dir` and give the full path to this like

```
placet-htgen "fullpath"/placet-development/examples/htgen/htgen.bds.tcl
```

As a result of this command, three types of files will be produced for each beamline elements. One is the file contains beam-generated halos (`halo_nn.txt`), second is the beam particles itself (`beam_nn.txt`) and third, in case of particles lost during passing through a particular element (`lost_nn.txt`). In the first line of the output file you find some information about the Element of interest.
Here is the first line of a `halo_nn.txt`-file:

```
# DCAVITY1  9.045  52  0  1   1
```

From left to right you find: name, longitudinal position [m], number of halo particle created, type of aperture, horizontal aperture [m], vertical aperture [m].
And the first line of a `beam_nn.txt`-file:

```
# BPM1  17.32 10200 at 102 BPM1 0.15 none 1 1 14 7 1.333e-06 300 0.000329 0
```

From left to right: name, longitudinal position [m], number of slices (of the beam), element number, name, length [m], aperture type, horizontal aperture [m], vertical aperture [m], mass number A of material, atomic number Z of material, pressure [Pa], temperature [K], minimal deflection angle [rad] (needed for simulation of mott scattering), radiation length [g/cm^2].
And of a `loss_nn.txt`-file:

```
# Qd2  2.875  1  3  0.0115   0.0115
```

From left to right: name, longitudinal position [m], number of lost particles, type of aperture, horizontal aperture [m], vertical aperture [m] Afterwards there is one row for each particle and one column for each parameter of the particle:

```
# DCAVITY1  9.045  52  0  1   1
    0.15     -162.467   2124.88   4.95482e-317  -1790.7   519.668   9.045   0
    0.1447   1072.76    240.038   9.0365e-313   742.364   944.278   9.045   1
```

From left to right: energy [GeV],horizontal,vertical and longitudinal offset with respect to the design particle [μm],horizontal and vertical angle [μrad],longitudinal position [m] and the index of the particle.
The `loss_nn.txt` lists only the particles which were lost, for example:

```
# Qd2  2.875  1  3  0.0115   0.0115
0.138944  -527.169  13682.3 9.0365e-313  -2065.66  -884.375  2.875  21
```

One can have a look into any file and make a plot quickly using GNUplot or Octave.
For example

```
gnuplot> plot "beam_300.txt" using 1:3 with points linecolor rgb
"red",
"halo_300.txt" using 1:3 with points linecolor rgb "blue"
```

will give you a 2D plot between energy on x-axis and vertical offset on y-axis of the halo and of the beam at the 300'th element of the accelerator.

```
octave> load 'filename.txt'
octave> hist (filename(:,1),200); 1D energy distribution.
```

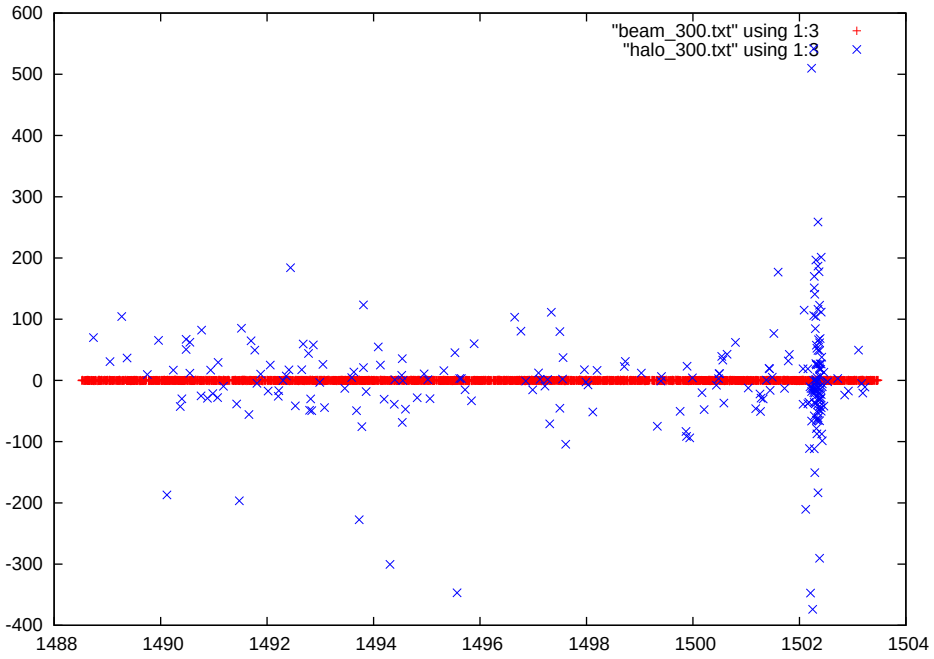

Figure 1: 2D plot of the beam with gnuplot

4 Frequently Asked Questions

1. Placet fails with an invalid command name "`Vacuum`". This will happen if you try to run HTGEN in a PLACET module without HTGEN. Make sure to use PLACET with HTGEN, typically called `placet-htgen`, configured with `--enable-htgen`

2. Starting PLACET fails with something like

   ```
   error while loading shared libraries:  libgsl.so.0:  cannot open
   shared
   ```

   Probably your current environment is not what it was when you created `placet-htgen` and libraries changed or are not within your current path. See where the library is located and make sure it is within your current path, as typically defined using the environment variable `LD_LIBRARY_PATH`

3. How can I get some more output to see what is going on in detail?
   There exist 4 different debugging levels.

```
HT_DEBUG=1   // warnings (by default)
HT_DEBUG=2   // for user information
HT_DEBUG=4   // general
HT_DEBUG=8   // related to process
HT_DEBUG=16   // all
```

   You can set the debug level for example at 16 using the environment variable `HTGEN_DEBUG`. In tcsh, you can set for example

```
setenv HTGEN_DEBUG 16
```

   before starting `placet-htgen` and

```
unsetenv HTGEN_DEBUG 16
```

   later to turn the extra info off.
   For bash, the equivalent commands are:

```
export HTGEN_DEBUG=16
```

   before starting and

```
unexport HTGEN_DEBUG=16
```

   or if this doesn't work

```
unset HTGEN_DEBUG=16
```

4. How can I get a list of the commands used for the input parameters and their arguments?
   start the module without any input, for example

```
placet-htgen
```

   you should see now something like
   ****
   ** PLACET Version No 0.94

   ****
   which shows that the program started followed by the prompt %.
   Just type Help after the prompt:
   `% Help`
   This will give you a list of all input parameters for PLACET and HTGEN as well.
   To get the details of one command, for example for Vacuum, type

```
% Vacuum -help
 -gaslist:      List of the gas mixture
 -temperature:  Temperature in K
                Default value:  293
 -thetamin:     Mott scatt.  cutoff (microradian)
                Default value:  1e-08
 -iel:          Element number (default 0)
                Default value:  0
 -nel:          Number of elements
                Default value:  1
% Material -help
 -X0:   Radiation length of material
        Default value:  -1
 -iel:  Element number (default 0
        Default value:  1e-08
 -nel:  Number of elements
        Default value:  1
% TrackBackground -help
 -beam:          Name of the beam to be used for tracking
 -dir:           Directory for the results defaults to NULL (no output)
 -start:         Starting element number for background tracking
                 Default value:  0
 -fulltracking:  tracking mode:  0 first/last elems recorded - 1 all elems
                 Default value:  0
 -charge:        Number of particles / bunch
                 Default value:  0
 -linac:         set 1 if linac tracking, defaults=0
                 Default value:  0
```

At the end type "exits" to stop the program.

5. Is there an installation ready to use?
   Yes, this exists on afs in /afs/cern.ch/eng/sl/lintrack, ready to be used on lxplus. Just do once per cshell or tcshell session.

   ```
   source /afs/cern.ch/eng/sl/lintrack/HTGEN/env.csh
   ```

   or in case of bash

   ```
   source /afs/cern.ch/eng/sl/lintrack/HTGEN/env.sh
   ```

   which will set the environment with `$HTDIR` and `$PLACET_DIR` and PATH variables to connect to the relevant libraries.

Now you should be able to run the executable for the lxplus x86_64 platform by typing

```
/afs/cern.ch/eng/sl/lintrack/PLACET/x86_64/bin/placet-htgen
```

To run this with the example input file type

```
$PLACET_DIR/../x86_64/bin/placet-htgen $PLACET_DIR/examples/htgen/htgen.bds.tcl
```

Acknowledgement

This work is supported by the Commission of the European Communities under the 6^{th} Framework Programme "Structuring the European Research Area", contract number RIDS-011899.