# GEANT4 : Tests on sensitivity of avalanche probabilities on code parameters

GEANT4 version used : 10.2 Patch-03

***Abstract :***

**We observed a strong dependance of the results of "the probability of avalanche" simulations depending on how simulation parameters are tweaked. There parameters are mostly simulation settings (production cuts, numerical stepper configuration), and not differences on the real physical system.**

***Our goal is to have a simulation set-up that descibes the most accuratly the physical (real) case, and not dependent on non-physical parameters.***

We start with one electron with a given energy $E$ and an uniform field $F$ in air at STP. Then we run >10,000 simulations and and check what is the probability of this system to turn to a Relativistic Runaway Electron Avalanche (RREA). RREA state is reached if we get at least 20 electrons with nore than 1 MeV (after a given time, limited to 10 micro-seconds).

*The simulation set-up is described extensively in the GitLab wiki page.*

### The physical parameters are :

- The initial energy of the electron
- The electric field
- The composition of the medium (air at STP)
- The initial direction of the electron

### Some (non-physical) tweakable parameters are :

- The electro-magnetic physics list can be set to different option.
- The trajectory of the particles within the electric field has to be solved using numerical integrator (Runge-Kutta type solvers).
- This last can be set-up with several options : accuracy, minimum and maximum steps, (among them, we will test MinimumEpsilonStep and MaximumEpsilonStep)
- The ranges cuts, that will be converted to an energy theshold of particle production

***Note :*** *in an uniform field, an analytical solution exists for the electrons/positrons motion equation, but it is not implemented in GEANT4. We tried to make a custom one (G4ExactUniEFStepper), but it is not working properly yet. However, the analytical solution will require several calculation line and the computation of hyperbolic sine functions, so it is not sure it will run faster (in terms of computation time) than the Runge-Kutta methods.*

## Tests' results

For the folowing results, we assume $E = 1\ \mathrm{MeV}$ and $F = 4.10^5\ \frac{V}{m}$.

By default :

- Physics list is set to 01 (fastest).
- Range cuts set to *7 mm* for electron and positrons, and *16 m* for photons; so that energy thresolds are about 10 keV for the three particles.

- The stepper is set to " G4ImplicitEuler " with ...
- ... MinimumEpsilonStep and MaximumEpsilonStep both set equal to 10^-6
- ... the MinStep set to 1 micro-meter
- ... the SetDeltaOneStep set to 1 micro-meter

**NaN values = simulation not ran yet**

**Test 1 :** We keep all the parameters at default, and we change the physics list between 01, LIV, PEN, 04 and LBE.

The probability (%) of having RREA are :

```
P_av.list.01  = 37.5 ;
P_av.list.04  = 59.1 ;
P_av.list.LIV = 58.7 ;
P_av.list.PEN = 58.7 ;
P_av.list.LBE = 90.4 ; % Warning : setting up LBE also changes the production cuts
                       % and enery thresholds
```

**Test 2.1 :** We keep the same physics list (**O1**) and threshold (production cuts), and we change the maximum acceptable step of Runge-Kutta stepper.

The probability (%) of having RREA are :

```
P_av.MaxStep.m1    = 59.2 ;
P_av.MaxStep.cm10  = 88.4 ;
P_av.MaxStep.cm1   = 92.2 ;
P_av.MaxStep.mm1   = 90.9 ;
P_av.MaxStep.um100 = 79.4 ;
P_av.MaxStep.um10  = 51.2 ;
P_av.MaxStep.um2   = NaN ;
```

*RQ : gets really slow (needs a lot of conputation time) below 1 mm*

**Test 2.2 :** We keep the same physics list (**O4**) and threshold (production cuts), and we change the maximum acceptable step of the Runge-Kutta stepper.

The probability (%) of having RREA are :

```
P_av.MaxStep.m1    = 63.5 ;
P_av.MaxStep.cm10  = 87.7 ;
P_av.MaxStep.cm1   = 92.0 ;
P_av.MaxStep.mm1   = 91.25 ;
P_av.MaxStep.um100 = 79.1 ;
P_av.MaxStep.um10  = 51.2 ;
P_av.MaxStep.um2   = NaN ;
```

*RQ : gets really slow (needs a lot of computation time) below 1 mm*

**Test 3.1 :** We keep the same physics list (**O1**) and do not specify the maximum acceptable step. We change the range cuts of photons, electrons and positrons (that changes the production energy thresholds). By default, range cuts set to *7 mm* for electron and positrons, and *16 m* for photons : we devide it by 2, 4, 8, 16, 32, 64, 128, 512, 1024 and 16384.

The probability (%) of having RREA are :

```
P_av.RangeCut.Phot.01.d1     = 37.5 ;
P_av.RangeCut.Phot.01.d2     = 72.6 ;
P_av.RangeCut.Phot.01.d4     = 84.9 ;
P_av.RangeCut.Phot.01.d8     = 84.8 ;
P_av.RangeCut.Phot.01.d16    = 85.0 ;
P_av.RangeCut.Phot.01.d32    = 85.2 ;
P_av.RangeCut.Phot.01.d64    = 85.4 ;
P_av.RangeCut.Phot.01.d128   = 84.95 ;
P_av.RangeCut.Phot.01.d512   = 85.2 ;
P_av.RangeCut.Phot.01.d1024  = 84.7 ;
P_av.RangeCut.Phot.01.d16384 = 85.2 ;
```

*RQ : all are running quite fast*

**Test 3.2 :** We keep the same physics list (**O4**) and do not specify the maximum acceptable step. We change the range cuts of photons, electrons and positrons (that changes the production energy thresholds). By default, range cuts set to *7 mm* for electron and positrons, and *16 m* for photons : we then devide it by 2, 4, 8, 16, 32, 64, 128, 512 and 16384.

The probability (%) of having RREA are :

```
P_av.RangeCut.Phot.04.d1     = 59.1 ;
P_av.RangeCut.Phot.04.d2     = 74.35;
P_av.RangeCut.Phot.04.d4     = 81.3 ;
P_av.RangeCut.Phot.04.d8     = 81.2 ;
P_av.RangeCut.Phot.04.d16    = 81.5 ;
P_av.RangeCut.Phot.04.d32    = 81.4 ;
P_av.RangeCut.Phot.04.d64    = 81.4 ;
P_av.RangeCut.Phot.04.d128   = 81.5 ;
P_av.RangeCut.Phot.04.d512   = 80.8 ;
P_av.RangeCut.Phot.04.d1024  = 81.6 ;
P_av.RangeCut.Phot.04.d16384 = 81.2 ;
```

*RQ : all are running quite fast*

**<u>Test 45:</u>** We keep the same physics list (**O1**). The range cuts are set to default (*7 mm* for electron and positrons, and *16 m* for photons). We change the MinimumEpsilonStep and MaximumEpsilonStep parameters of the stepper to 10^-6, 10^-7, 10^-8, 10^-9, 10^-10, 10^-12

The probability (%) of having RREA are :

```
P_av.RangeCut.Phot.p_6      = 37.5 ; % fast
P_av.RangeCut.Phot.p_7      = 38.0 ; % fast
P_av.RangeCut.Phot.p_8      = 37.6 ; % fast
P_av.RangeCut.Phot.p_9      = 37.6 ;% quite fast
P_av.RangeCut.Phot.p_10     = 37.9 ; % very slow
P_av.RangeCut.Phot.p_12     = NaN ; % very slow,
                                    % and the "equal steps warning" message
                                    % comes very very often
```