Simulation with a fixed interaction point

Sep. 2016

1 Updated use of "FreeC" parameter: What we want to do.

First, suppose a proton primary (we may call it 1ry=primary) of which energy, starting point (=P_c), direction etc are given in a normal way (i.e, P_c may be fixed by the user or randomly sampled using user's input parameters). One special thing here is that the user gives "F" to *FreeC* in *epicsfile*; For this, one may give "F" or "f" for *FreeC* field of *epicsfile*.

This specifies that the 1ry should make the first interaction (hadronic collision) at P_c. In the versions so far ($\leq$ v9.17), no particle tracking is tried before P_c even if there is a lot of materials.

Although the old type simulation may have some application area, one may want to include "possilbe effects" caused by the incident particle before P_c (but excluding the hadronic collision supposed to occur at P_c first time). For such a purpose, one may start the simulaion from P_0 placed at a world boundary. P_0 can be obatined so that $\vec{P}_c - \vec{P}_0$ coincides with the given 1ry direction.

When the 1ry reaches P_c, the hadronic collision is forced to take place. After that, the simulation proceeds in the normaly way. The new scheme helps to do such type of simulations. We call 1ry between P_0 to P_c virtual 1ry. (v1ry)

In the above example, we assumed the interaction to happen at P_c is a hadronic collision. From P_0 to P_c, we must suppress hadronic collisions but allow "possible effects" caused by other interactions. If we consider other primaries such as photon, muon, etc too, the "possilbe effects" to be taken into account are only energy loss by dE/dx and knock-on electron production.

We include multiple scattering during the tracking, but it must be small (i.e, 1ry energy must be high enough) so that the 1ry can reach P_c almost exactly. Also any magnetic field must not present (or very weak as compared with 1ry energy) in the virtual tracking region. These are for charged particle incidents.

For nuetral particles such as photons and neutrons, old and new "FreeC f" are practically the same. (Trace information from P_o to P_c could be obtained by the new scheme).

The forced interaction is dependent on 1ry type and the standard is shown in Table 1. The table also shows other possible choice.

1.1 How to use

1. To use the v1ry function, *config* must have *world*.

Table 1: Interaction type the user can force

1ry	Default int. type	Other possible type
hadron	coll	decay
gamma	pair	photop, comp , mpair
electron	brem	anih
muon	nuci	brem, pair

2. To use the v1ry funciton in the default mode, the user may simply give "F" to *FreeC*.

3. To use the v1ry function without virtural 1ry (i.e, same as old version), the user may give "F" to a parameter "FollowV1ry" like:

   FollowV1ry F /

   in "epicsfile". It's default is T.

   It is also possible to make the function On/Off by

   call epSetFollowV1ry(.true.)

   or

   call epSetFollowV1ry(.false.)

4. To specify non-default interaction, the user must

   call epSetIntType(ptype, itype)

   where "ptype" is 1 character input to specify the 1ry type; one of "h", "g", "e", "m" (for hadron, gamma, electron, muon).
   " itype" is also a character input to specify the interaction type: one of "coll", "pair", "brem", "comp", "anih", "photop", "nuci", "mpair" .

   The user can call this subroutine more than once, say, for, hadron, muon and photon. The most probable use of this call would be:

   call epSetIntType('g', 'photop')

   The combination of ptype and itype must be consistent. For example, If ptype is " h " and itype is "decay", actual 1ry must not be p, heavy ion (no check will be done at present).

   The place to put this call would be in subroutine uiaev in ephook.f

   Even if the call is put in the user program, it's not effective until "FreeC" is made to be "f".

Caution: An invalid combination of ptype and itype may be easily created; (say, ptype="h" and itype="decay" for 1ry proton), in such a case, program execution may result in error or some unknown behavior.

For ptype="e", and itype ="brem", virtual 1ry treatment will not be appropriate since brems takes place so often and suppressing it until the specified P_c is unrealistic, unless the grammage in the virtual 1ry state is $\ll 1$ r.l.

If "FollowV1ry" is "t", the specified interaction point is generally not exactly realized (due to scattering; for 10 GeV proton case, $\sim$0.1mm deviation). If the user specifies, for example,

```
InputP "u+z"
xrange  (2, 3)
yrange  (-1, 1)
zrange  (0.5, 1.0)
```

P_c is sampled uniformly in the region specified by the given parameters. In some case, the sampled point P_c may fall in "sp" or "hollow" (essentially vacuum), then the error is informed to the user and program execution stops. The same error may happen if the P_c is deviated by scattering and the actual point falls in "sp" or "hollow".

2 Tracking scheme

The user needs not read this section: memo only for development time.

Even in the v1ry state, particle tracking is performed normally as much as possible. To know the current tracking is in the v1ry state, a varialbe **V1ry** is prepared. Its default is 0 but reset to 1 when "FreeC" is "f". When V1ry is 1, a new variable "**Dist2colp**" is computed which keeps the distance to the current particle position to P_c.

During V1ry is 1, every interaction except for knock-on is prohibited. If the computed new particle position is beyond P_c, the distance the particle moves is shortened and the "itype" interaction is forced to happen there. V1ry is reset to 0.

Some more details:

We sample a 1ry of which starting point is P_c in the normal way. If *FreeC* in the *epicsfile* is "f" P_0 is computed by using the direction cosine of the 1ry and P_c with information of the world. Then, the 1ry is reset to start from P_0. The place for this is just before calling

epgen in subroutine *s1set* of sepics.f.

The default V1ry value 0 is made to be 1 here. and Dist2colp is computed.

During virtual 1ry tracking, simulation is performed as if normal 1ry were treated. Distances for various physical processes to take place are sampled and the shortest distance one is selected. If the path is too long, truncation is made and the 1ry is moved that distance. The energy loss and multiple scattering calculations are incorporated during the tracking.

The shortest one may be employed for case 1,2,3 below (x: current 1ry pos. B: border of the current component. * is the place the sampled process to occur.) (Note: if 1'), path is truncated at B and the process is discarded)

```
1)  x----*--B---Pc        1')  x-------B--*--Pc
2)  x----*--Pc----B
3)  x-------Pc--*-B
```

The current *Dist2colp* is from x to P_c. Whenever the particle is moved, *Dist2colp* is updated. For case 1), LengthToB (from x to B) is $<$ *Dist2colp*.

For case 1) and 2), the 1ry is moved to * and if the process sampled is *knock-on*, the process is simulated.

If case 3) happens, independently of the sampled process, we reset that the sampled interaction is *ityp*. The distance is adjusted so that the particle moves to P_c . V1ry = 0 is set.

During the tracking with $V1ry = 1$, e- may be produced by the knock-on prcoess; then this e- is tracked while the 1ry is stored in the stack area. If $V1ry$ is still 1, $Dist2colp$ may be affected, so we change $V1ry = -1$ until the incident track appears again (V1ry=1 is restored)

If "FollowV1ry" is "f", "Dist2colp" is simply made to 0 and resetting the 1ry is bypassed.