

Using the “El_hin” mode in multiple Coulomb scattering treatment

KK

Mar. 29, 2017

1 Introduction: Excuse

You may go directly to Sec.3, if sentences around here are already familiar.

The Elsepa-based Hinge method (El_hin method; For details, see a separate document; Treatment of the Multiple Coulomb Scattering (MCS)) to treat multiple Coulomb scattering (MCS) has been implemented more than two years ago but no public announcement has been made. This is due to that

- The standard old method following Moliere’s MCS theory and El_hin gave essentially the same results for a few of tested detector configurations. Agreement by the two completely different schemes suggested correctness of the both methods.
- CERN beam tests were well explained by the Moliere.
- The El_hin is more rigorous than Moliere but it needs much longer computation time.

However, it was found that in the case of the CALET detector configuration, Moliere and El_hin show some differences and a CERN beam test with the CALET model indicates El_hin is better. Also recent CALET cosmic ray observation results seem to favor El_hin (El_hin gives smaller lateral spread).

It’s found that slow El_hin could be improved much by using a sampling table with a different parameter value employed in the current El_hin. For general purposes, several such tables must be prepared and one of them must be selected.

So far selection is made by linking a selected file to a fixed file name (by the unix “ln -s” command). This is awkward and even dangerous in some case. Therefore, we modify implementation so that specification of a table is possible by input parameters. This needs some modification of both of Cosmos and Epics programs and the user must get Cosmos8.01¹ and Epics9.21 from <http://cosmos.n.kanagawa-u.ac.jp>

If you use only Cosmos and have v8.0 already, no need to get v8.01: The El_hin method would give the same result as Moliere, so it is not implemented in Cosmos (El_con has been usable since v7.65).

2 How to use

2.1 For impatient users

- Install Cosmos and Epics

The installation procedure of Cosmos and Epics is the same as older versions²

¹Two big size data (one for Elsepa and one for qgsjetII-04 hadronic interaction model) are included in this Cosmos. They are not included in most earlier versions and must have been down-loaded separately. If you have already the data (Cosmos/Data/Elsepa and Cosmos/Import/Hidden/QGS/qgsjetII-04/qgsdat-II-04), you may save the disk size by, say, deleting the old ones. We use Cosmos/ and Epics/ to mean their top directory

²If not clear, see the general manual on the web page. Also <http://cosmos.n.kanagawa-u.ac.jp/cosmosHome/Cosmos8.00.pdf> may help you

- When your Epics application is made, before its execution, put parameter “MCSmodel” in the “epicsfile” (say, near “Moliere” parameter) from the 2nd column like:

```
MCSmodel 'El_hin'    / default is 'Mol' (then Moliere is referred)
...
Moliere      2      /
....
ALatCor     0      / (0 or 1)
```

That’s all.

Note: This is the fastest option for the El_hin mode, and probably safe for many applications. (In the case of CALET applications, this is confirmed).

However, depending on a detector configuration and purpose, MCSmodel=’Mol’ could be bit faster and give the same result, or slower speed El_hin option might be required for more reliable results,

2.2 More Details

The speed of the MCSmodel=’El_hin’ mode depends on the parameter “MCSparam” which specifies a directory (folder) where (normally) a number of files with media name (such as Pb, PWO, SciFi etc) is contained. Each file contains tables used for sampling a kind of Coulomb scattering angle in the given medium.

An example of MCSparam is “param_1.9-0.05” which is default and contains all media names in Epics/Data/BaseM. Six param folders containing all media are prepared:

```
param_0.1-0.05  param_0.3-0.05  param_0.6-0.05  param_1.0-0.05
param_1.5-0.05  param_1.9-0.05
```

These are contained in Epics/Data/MCS/. The numbers 0.1-0.05 etc means values of two parameters (p and q) used to make a sampling table. The latter q is normally fixed to 0.05 and almost does not affect the speed. The value of p must be in $0 \sim 2$. If $p \rightarrow 0$, MCS treatment approaches to the detailed mode and if $p \rightarrow 2$, it approaches to the condensed mode³

Therefore, MCSparam=’param_0.1-0.05’ is more detailed and rigorous than default, and needs much longer computation time. The user may test using this one and compare the result with MCSmodel=’Mol’ (with incident electron energy $\sim 1\text{GeV}$; if detector is thick, use various starting point of cascade or with a larger energy). Then, if MCSmodel=’Mol’ gives the same result, it should be faster and the user may safely use it for farther mass production. If some non-negligible difference between El_hin and Moliere is found, don’t use Moliere, and try larger p value MCSparam for faster computation.

There are other MCSparam’s prepared with smaller number of media:

```
Acrylic Al2024 CFRP PWO SciFi hollow Al Al7075 G5 SCIN W sp
```

Their param folder names are

```
param_1.0-0.5  param_2.0-0.05  param_0.2-0.05  param_0.7-0.05  param_1.2-0.05
param_0.4-0.05  param_0.8-0.05  param_1.75-0.05  param_0.5-0.05
param_0.9-0.05  param_1.9-0.5
```

They are contained in Epics/Data/MCS2. Since the default location is Epics/Data/MCS, the user must change the the folder by specifying

```
MCSdir='$EPICSTOP/Data/MCS2'.
```

³El.con uses the condensed mode but does not use the hinge method. Moliere is the same.

MCSmodel, MCSparam, MCSdir must be given in “epicsfile”⁴.

Meaning of p : Roughly speaking, p specifies a critical angle θ_c by $p \sim 1 - \cos \theta_c$; scattering angle $> \theta_c$ is regarded as large, and such scattering could lead to fluctuation so that sampling of Coulomb scattering by individual atom is tried. If scattering angle is small, MCS is tried. That’s the mixed mode.

Table 1: MCS control

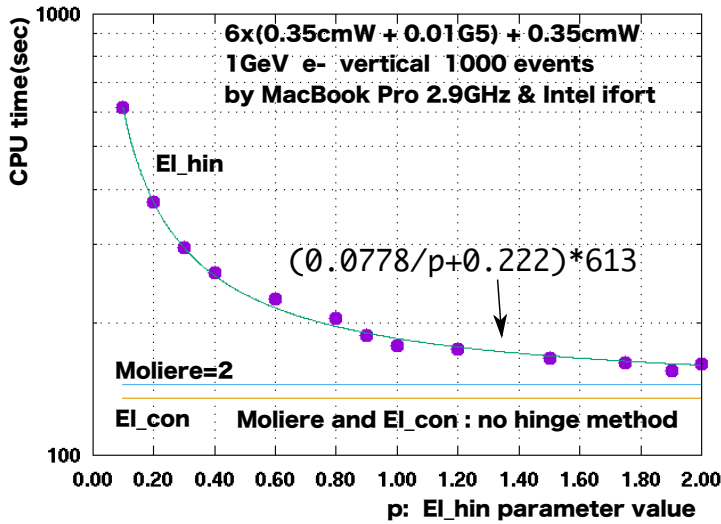
parameter name ¹	default	possible values
MCSmodel	'Mol'	'El_hin' 'El_con' 'Mol' ²
MCSparam	'param_1.9-0.05 '	'param_0.1-0.05' etc
MCSdir	' ' ³	'\$EPICSTOP/Data/MCS2' etc

¹ in Cosmos, must be given in \$PARAM of a “param” file. Epics can also accept this style, but better to put them in “epicsfile”. If both given, those in “epicsfile” have priority.

² Like IntModel, specification like MCSmodel=' "El_hin" 0.1 "Mol" ' is possible; this case means that model changes at 0.1 GeV. Energy must be < 1.

³ If blank, \$EPICSTOP/Data/MCS is employed in Epics and \$COSMOSTOP/Data/MCS in Cosmos.

The figure below shows how the El_hin computation time depends on p . Relative value to $p = 0.1$ case is $\sim 0.0778/p + 0.222$. This is an example of a simple detector case; dependence will change by the configuration.



⁴So the user can put param folder in other arbitrary folder and specify it by MCSdir. The user can make own param folder or add new media to the existing folder; the folder name need not follow param-p-q rule. It can be arbitrary. See Epics/Util/Elemag/MixedMCS and read Readme.

For the user with experience of using El_hin mode in the previous versions: the param folder was placed in Epics/Data/Hidden/ (There were only two: MCSHCSmfp0.05 and MCSHCSmfpLarge. The former is equivalent to parm_0.1-0.05 and a particular param folder in Hidden/ must be specified by Epics/Data/MCS by the ln command: e.g,

ln -s Hidden/MCSHCSmfp0.05 MCS

Now “Hidden” was renamed to “MCS” and old “MCS” was removed. MCSmodel had to be placed in \$param in “param” file. MCSparam and MCSdir were non existent. They are now to be put in “epicsfile”. However, they can also be put in \$param. In that case, if the same parameter appears in “epicsfile”, the latter is employed. For **Cosmos application**, \$param must be used. MCSdir in default is blank, and Cosmos/Data/MCS will be employed in that case (though El_hin cannot be specified).

3 Newly Added updates

Since the “Moliere” and “El_hin” almost give the same results for a simple detector configuration and Moliere is normally faster, in some case, one may want to use “Moliere” for a simple detector configuration part and “El_hin” for complex part. Or in some case, “Moliere” might give a better result than “El_hin” in some detector part in spite of that the latter should give basically better results than the former (by some unknown reason).

For such purposes, several parameters are prepared in V9.22. Suppose one wants to use “El_hin” in a detector region $Z_1 < Z < Z_2$ (world coord. z value) and “Mol” in other part, the use may put (from the 2nd column)

```
MCSmodel 'El_hin' /
MCSzRange (Z1,Z2) /
```

in the epicsfile. However, if $Z_1 = Z_2 = 0$, **all** Z are accepted. The notation

```
MCSmodel 'Mol' 'El_hin' /
```

is possible as usual (i.e, 2nd one is employed). Note: if “Mol” is accepted, “Moliere” parameter is further referred. If one wants to invert the relation, i.e., “Mol” in $Z_1 < Z < Z_2$ and “El_hin” in other part. Then, one may give

```
MCSmodel 'El_hin' /
MCSzRange (Z1,Z2) /
MCSrevert t /
```

If two ranges are wanted, simply give like

```
MCSmodel 'El_hin' /
MCSzRange (Z1,Z2) /
MCSzRange (Z3,Z4) /
```

The regions should not overlap. Although necessity is questionable, one may use up to 5 range specifications.

Regions may be specified by using component numbers. To specify the region covered by the components n_1 to n_2 ,

```
MCSmodel 'El_hin' /
MCSnumRange (n1,n2) /
```

MCSnumRange can be put up to 5, too. MCSrevert will work same as the previous example. MCSzRange and MCSnumRange may be given simultaneously. In that case, **and** of the accepted regions is taken (default). If **or** is wanted, put

```
MCSandor 'or' /
```

(That is, MCSandor 'and' / is default).

To see If your specification is working fine, put

```
MCSdebug t /
```

and generate ~ 10 events with ~ 10 GeV e- incidents. Redirect the **stderr** to a file. After completion of the job, extract lines

```
mol: x y z
```

```
and
```

```
hin: x y z
```

where “x y z” means numerical values of a scattering point (not exact) in the world coord. From these, you can identify which scattering (Mol or El_hin) has taken place. **Never forget to reset MCSdebug**

An example of MCSdebug use is shown below, from which Fig.1 is obtained. (This is applied to the CALET config).

```
MCSmodel 'El_hin' /  
MCSzRange (3, 22) /  
MCSrevert t /  
MCSdebug t /
```

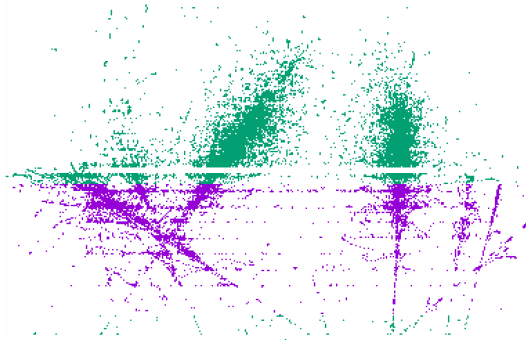


Figure 1: Magenta region is specified by MCSzRange, and “MCSrevert t” is put; El_hin regions are seen bottom and top part—green dots.