Population Control Variance Reduction in MCNP

One of the oldest, but simplest and often effective, variance reduction techniques in MCNP is to control the population of MCNP particles passing through a shield that you are interested in penetrating.

Up to this point in the class, I have always had you either use 0 or 1 for the cell importances (i.e., the imp:n and imp:p entries), indicating whether particles are supposed to exist in the region or not. That is, we have used zero importances in the infinite region that surrounds our calculational volume and one in the regions where we wanted the particles to pass through.

In reality, the rules of the cell importance is more complicated than that, because MCNP uses special rules when particles pass between cells of different importances. In simplest terms, MCNP applies the following rules:

1. When a particle passes to a MORE IMPORTANT region, the particle is divided into multiple pieces, the number of pieces being proportional to the ratio of the user-supplied importances. That is, if it goes from a region of importance 15 to one of 60, it will split into 4 pieces.

   In effect, it is a statistical trick to focus MCNP's effort on regions that the user says have more of an impact on the solution. I often think of this as being like putting a series of amplifiers along the paths that we want to encourage particles down.

2. When a particle passes to a LESS IMPORTANT region, MCNP attempts to "weed" the particle out. It does this by making the particle play a statistical game (appropriately, but gruesomely (and possibly politically-incorrectly), called "Russian roulette"). Basically it gives the particle only a fractional probability of not being killed, the probability being the ratio of the particle importances. Again, if the particle goes from a region of importance 60 to one of 15, the particle has only a 25% chance of not being killed.

I will show you how we can use it using a simple example.

Assume you have a problem where 1 MeV photons need to penetrate a foot of lead. You set up and run a problem with this input deck to get the leakage:

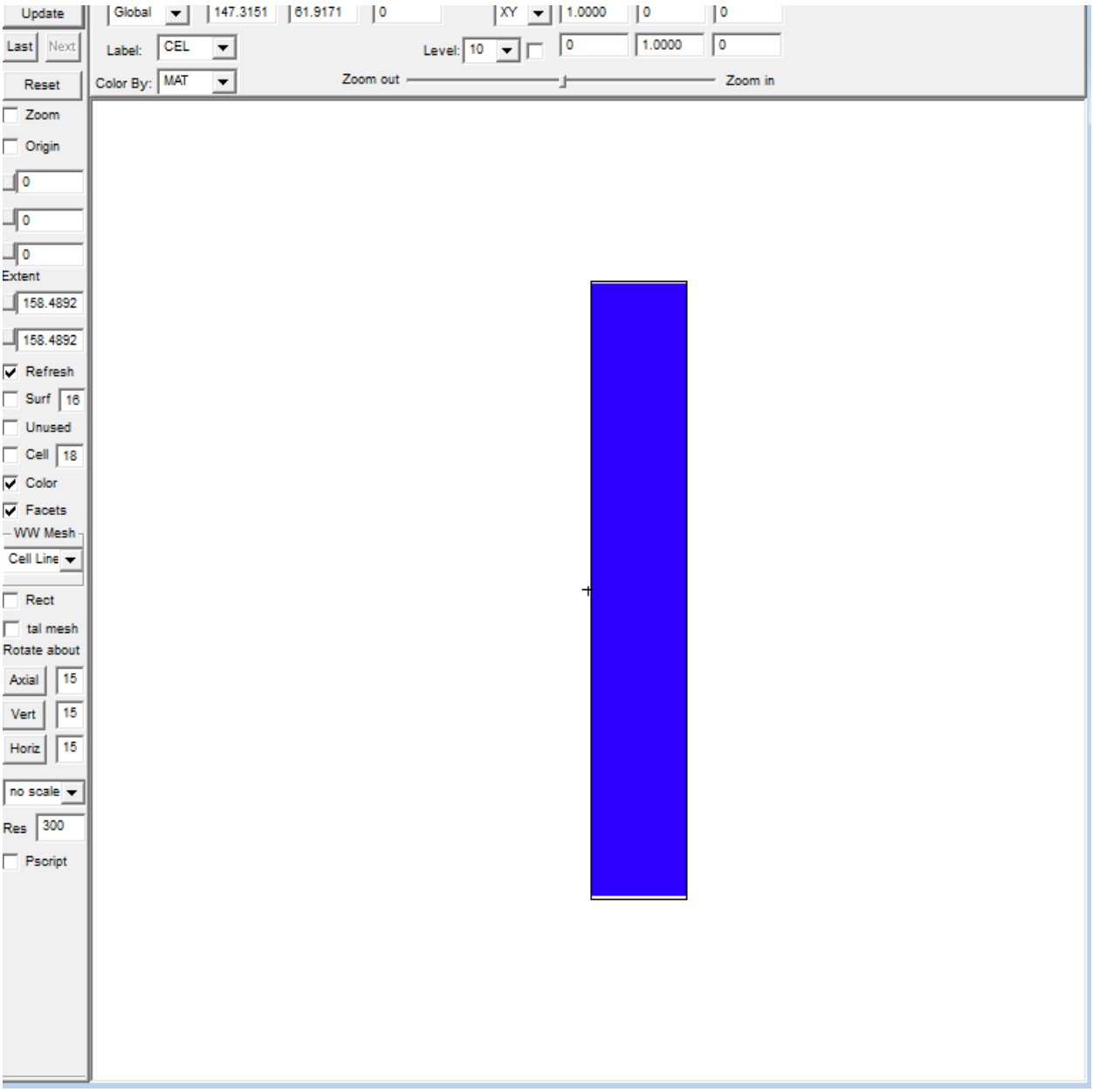
```
Population test 0
c Cells
1 1 -11.35 -1 imp:p 1
2 0        -999 1 imp:p 1
999 0 999 imp:p 0

c Surfaces
1 rpp 1 31.48 -100 100 -100 100
999 sph 1000

c Data
mode p
m1 82000 1
sdef par 2 erg 11 pos 0 0 0
```

```
f1:p 1.1
ctme .25
```

with VisEd view:



Your results come out:

```
tally        1        nps =      576000
           tally type 1    number of particles crossing a surface.
           particle(s): photons

 surface  1.1
                 0.00000E+00 0.0
```

Which tells us that none of the 576 thousand particles got through.

Looking earlier in the output deck (just before the tally results, actually), we see these lines:

	cell	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy
1	1	285915	5476641	8656057	1.5028E+01	3.7619E+00	3.7619E+00
2	2	697458	689508	0	0.0000E+00	7.9364E+00	7.9364E+00
	total	983373	6166149	8656057	1.5028E+01		

What this tells us is that of the 576000 photons born, 285915 of them entered cell 1, which is consistent with just a tad less than half of them hitting the shield (i.e., the other half being emitted from the source going to the left). Okay.

These 285915 resulted in a "population" of a little over 8.6 million. Since MCNP counts each segment of a particle random walk in the "population", this means that each photon must have scattered about 30 times on average. Okay.

Population control technique

To try to see what is going on, we divide up the lead into 3 cm segments, with the last one a bit bigger. The deck now looks like this:

```
Population test 1
c Cells
c 1 1 -11.35 -1 imp:p 1
2 1 -11.35 -2 imp:p 1
3 1 -11.35 -3 imp:p 1
4 1 -11.35 -4 imp:p 1
5 1 -11.35 -5 imp:p 1
6 1 -11.35 -6 imp:p 1
7 1 -11.35 -7 imp:p 1
8 1 -11.35 -8 imp:p 1
9 1 -11.35 -9 imp:p 1
10 1 -11.35 -10 imp:p 1
11 1 -11.35 -11 imp:p 1
99 0   -999 1 imp:p 1
999 0 999 imp:p 0

c Surfaces
1 rpp  1 31.48 -100 100 -100 100
2 rpp  1  4 -100 100 -100 100
3 rpp  4  7 -100 100 -100 100
4 rpp  7 10 -100 100 -100 100
5 rpp 10 13 -100 100 -100 100
6 rpp 13 16 -100 100 -100 100
7 rpp 16 19 -100 100 -100 100
8 rpp 19 22 -100 100 -100 100
9 rpp 22 25 -100 100 -100 100
10 rpp 25 28 -100 100 -100 100
11 rpp 28 31.48 -100 100 -100 100
999 sph 1000

c Data
mode p
m1 82000 1
sdef par 2 erg 11 pos 0 0 0
f1:p 1.1
ctme .25
```

Notice what I did (which I like to do so I can easily reverse the process):

1. I commented out cell 1, but did not reuse the number.
2. I kept surface 1 (so I could use it in the cell 99 description), but divided it up into surfaces 2 through 11.
3. I made cells 2 through 9 (which cover exactly the same region as old cell 1) to just be inside the same-numbered surfaces.

I ran this version and got this in the output deck:

```
photon   activity in each cell
print table 126
```

cell		tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy
1	2	293172	4994555	7650843	1.3283E+01	3.9992E+00	3.9992E+00
2	3	94319	497592	823219	1.4292E+00	2.4756E+00	2.4756E+00
3	4	19766	86221	145044	2.5181E-01	2.1291E+00	2.1291E+00
4	5	4385	17256	29158	5.0622E-02	1.9251E+00	1.9251E+00
5	6	1049	3762	6450	1.1198E-02	1.7103E+00	1.7103E+00
6	7	226	753	1283	2.2274E-03	1.5779E+00	1.5779E+00
7	8	60	188	343	5.9549E-04	1.4456E+00	1.4456E+00
8	9	12	36	62	1.0764E-04	1.3997E+00	1.3997E+00
9	10	1	4	6	1.0417E-05	1.4707E+00	1.4707E+00
10	11	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00
11	99	697517	689567	0	0.0000E+00	7.9353E+00	7.9353E+00
total		1110507	6289934	8656408	1.5028E+01		

```
1tally        1        nps =      576000
           tally type 1    number of particles crossing a surface.
           particle(s): photons

 surface  1.1
                0.00000E+00 0.0000
```

This indicates that none still got through, but looking at the "population" column, we get a picture of how far the calculated particles penetrated; i.e., we get a number for each our new layers, so we see that some of them ALMOST made it. More importantly, we can see how the number of MCNP particles drops as we go through the shield. After a drop of a factor of 10 between cells 2 and 3, they seem to settle into a drop of about a factor of 5 for each layer.

This is where population control comes in. We take our deck and we <u>increase</u> the importance by (about) these factors, with the importances getting bigger and bigger because each has to be that factor larger than the previous one (you could use an EXCEL spreadsheet to get it more exactly, of course). Eyeballing it, the cell cards become:

```
Population test 2
c Cells
c 1 1 -11.35 -1 imp:p 1
2 1 -11.35 -2 imp:p 1
3 1 -11.35 -3 imp:p 10
4 1 -11.35 -4 imp:p 50
5 1 -11.35 -5 imp:p 250
6 1 -11.35 -6 imp:p 1250
7 1 -11.35 -7 imp:p 6000
8 1 -11.35 -8 imp:p 30000
9 1 -11.35 -9 imp:p 120000
10 1 -11.35 -10 imp:p 600000
11 1 -11.35 -11 imp:p 3000000
99 0   -999 1 imp:p 1
```

```
999 0 999 imp:p 0
```

Those importances get pretty big.

But when we run this one, we get:

```
photon   activity in each cell

                      tracks     population   collisions   collisions     number        flux
             cell    entering                              * weight     weighted     weighted
                                                        (per history)    energy       energy

        1        2       30498       520644       797004    1.3324E+01   3.9772E+00   3.9772E+00
        2        3       97096       505393       836110    1.3978E+00   2.4572E+00   2.4572E+00
        3        4      101201       440436       741730    2.4800E-01   2.1119E+00   2.1119E+00
        4        5      111971       436725       743683    4.9730E-02   1.9203E+00   1.9203E+00
        5        6      129156       470516       805353    1.0771E-02   1.8133E+00   1.8133E+00
        6        7      146944       514999       888125    2.4745E-03   1.7433E+00   1.7433E+00
        7        8      175724       598857      1033000    5.7563E-04   1.7000E+00   1.7000E+00
        8        9      172395       574283       992808    1.3831E-04   1.6651E+00   1.6651E+00
        9       10      214018       699013      1211633    3.3759E-05   1.6343E+00   1.6343E+00
       10       11      264385       899836      1593355    8.8789E-06   1.6148E+00   1.6148E+00
       11       99      125670       124856            0    0.0000E+00   7.9174E+00   7.9174E+00

           total       1569058      5785558      9642801    1.5033E+01

1tally        1        nps =       59818
           tally type 1    number of particles crossing a surface.
           particle(s): photons

 surface  1.1
                2.96048E-07 0.0384
```

Looks like I “overmultiplied” (which is like amplifying noise) from about cell 7 on, but going from no answer at all to 4% accuracy is a pretty good tradeoff.