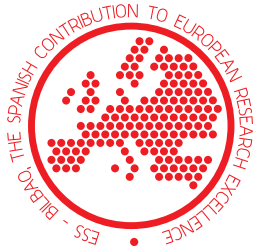


Using Detailed supermirror physics in MCNP6.2

ESS-Bilbao

M.Magán, O. González

February 24, 2020

Introduction

Introduction

Recently, we have performed some developments in the MCNP6 code in order to better simulate the behavour of guides through guides. Said work started in the SINE2020 with Dr. Bergmann porting the MCNPX patch from F.X. Gallmeier to MCNP6, and including the event biasing in reflection, in order to enhance the figure of merits.

Verification and validation

In October 2018, the port to MCNP6 was completed, and we verified the results in a number of tests, in the context of the SINE2020 project. In particular, we benched it againts previous MCNPX implementation. The results fall well between the margin of variance between both codes.

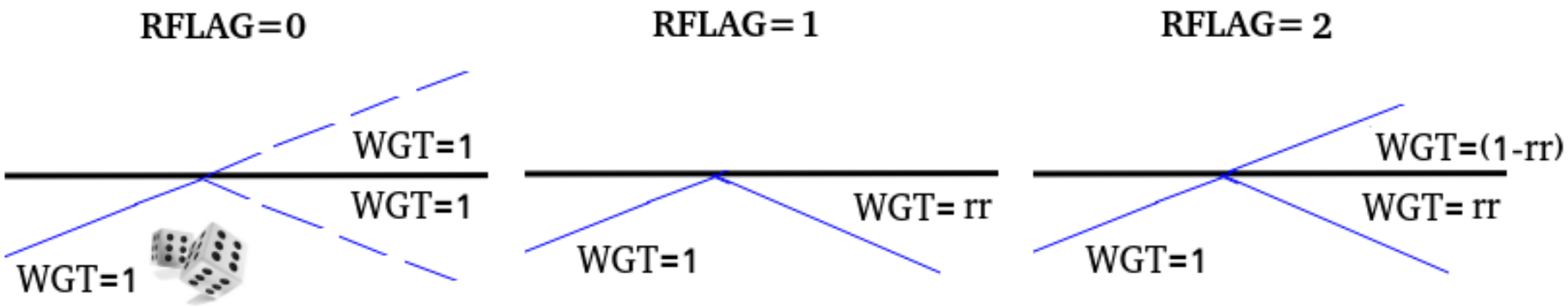
Tally	Distance m	MCNPX Result	MCNP6 Result	% Diff
12	0.5	2.94E-1	2.87E-1	2.59%
22	1.5	1.13E-1	1.10E-1	2.52%
32	2	6.09E-2	6.08E-2	0.13%
42	3	4.74E-2	4.66E-2	1.85%
52	7.5	1.88E-2	1.81E-2	3.76%
62	9.5	1.43E-2	1.36E-2	4.77%
72	25	6.21E-3	5.96E-3	4.70%
82	45	4.50E-3	4.30E-3	5.21%
92	66	3.82E-3	3.64E-3	5.70%

Code porting

Enhancements to the code

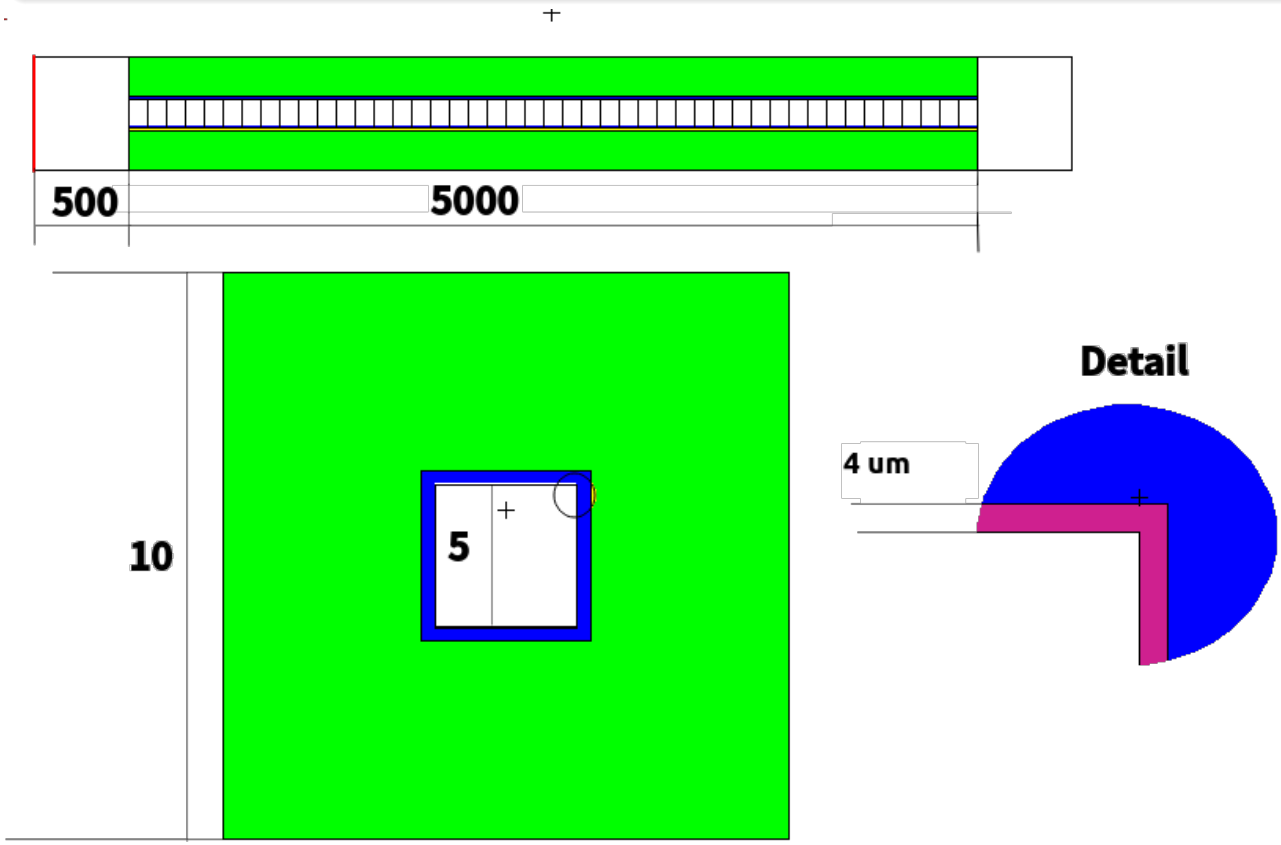
Reflection mode: In-guide only and splitting

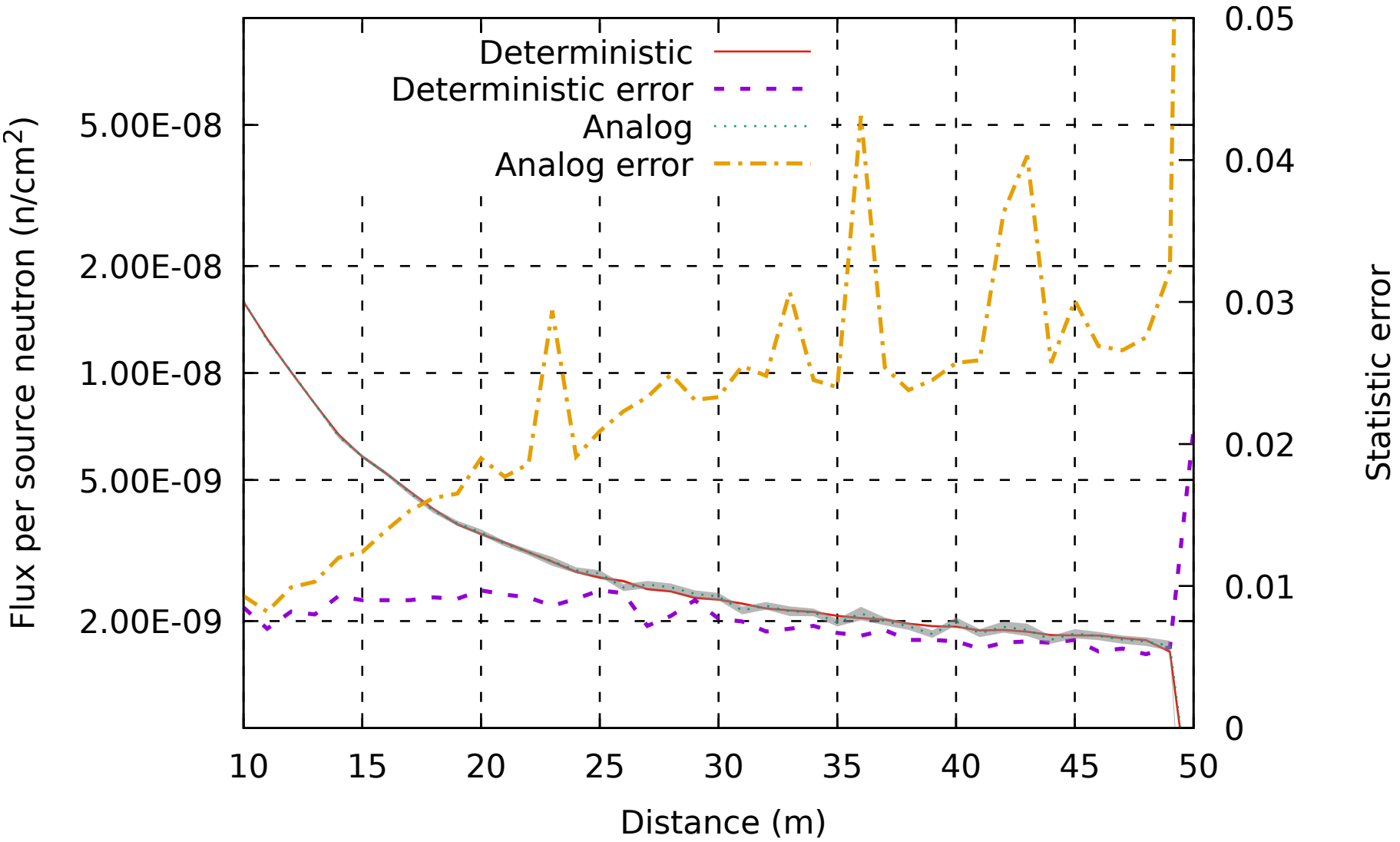
Once the port was done, we focused on implementing improvements that allow for more powerful simulations. The first one is enabling the possibility of three different reflection modes: The first one is the one previously implemented, the second one can be seen similar to a McStas simulation (discarding the lost neutrons), and the third one splits the neutrons in transmitted and reflected parts. Notice that only the first mode implies a RANG() call, so the other two are, effectively, deterministic transport.



Benefits of the splitting at mirrors

Splitting the particle into reflected and transmitted part smoothes the distribution of the particles. Reflected particles progressively lose some weight down the guide, while transmitted particles, specially in scenarios where the reflectivity is close to 1, are much better sampled. This enables better calculation of neutron flux, and generated gamma.




Deterministic
Deterministic error
Analog
Analog error
Flux per source neutron (n/cm^2)
Statistic error
Distance (m)
5.00E-08
2.00E-08
1.00E-08
5.00E-09
2.00E-09
0.05
0.04
0.03
0.02
0.01
0
10
15
20
25
30
35
40
45
50

Enhancements to the code (II)

DXT spheres

Detectors and Direct Transport Spheres (DXT) are incompatible with any kind of reflecting surface, as the MCNP sternly warns. The reason is that it is not possible to calculate the contribution from a collision or source (primary or secondary) through a reflection, but if the reflected particle enters the sphere, it will still get killed.

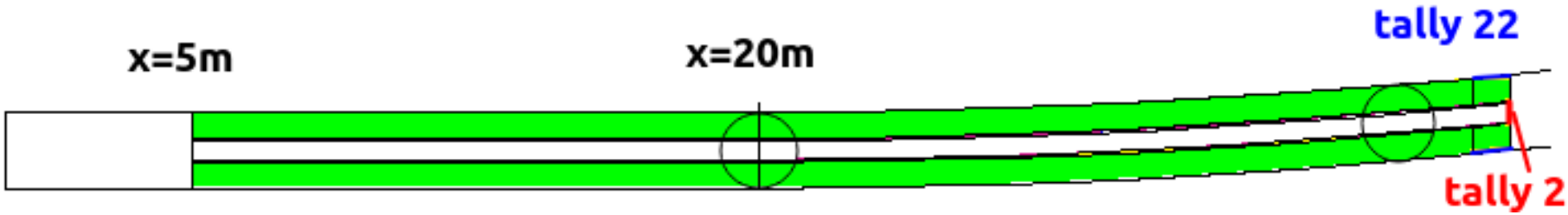
Solution to this problem

Our solution to the problem is to make the DXT "fair" by not killing the particles whose contribution it could not calculate. This is done by setting up a flag that makes the particle invisible to DXT, raising it in a reflection event, and lowering it in a collision. The code for making the sphere invisible to it is actually already present in the function dist_dxtran_sphere().

DXT sphere testbench

Description

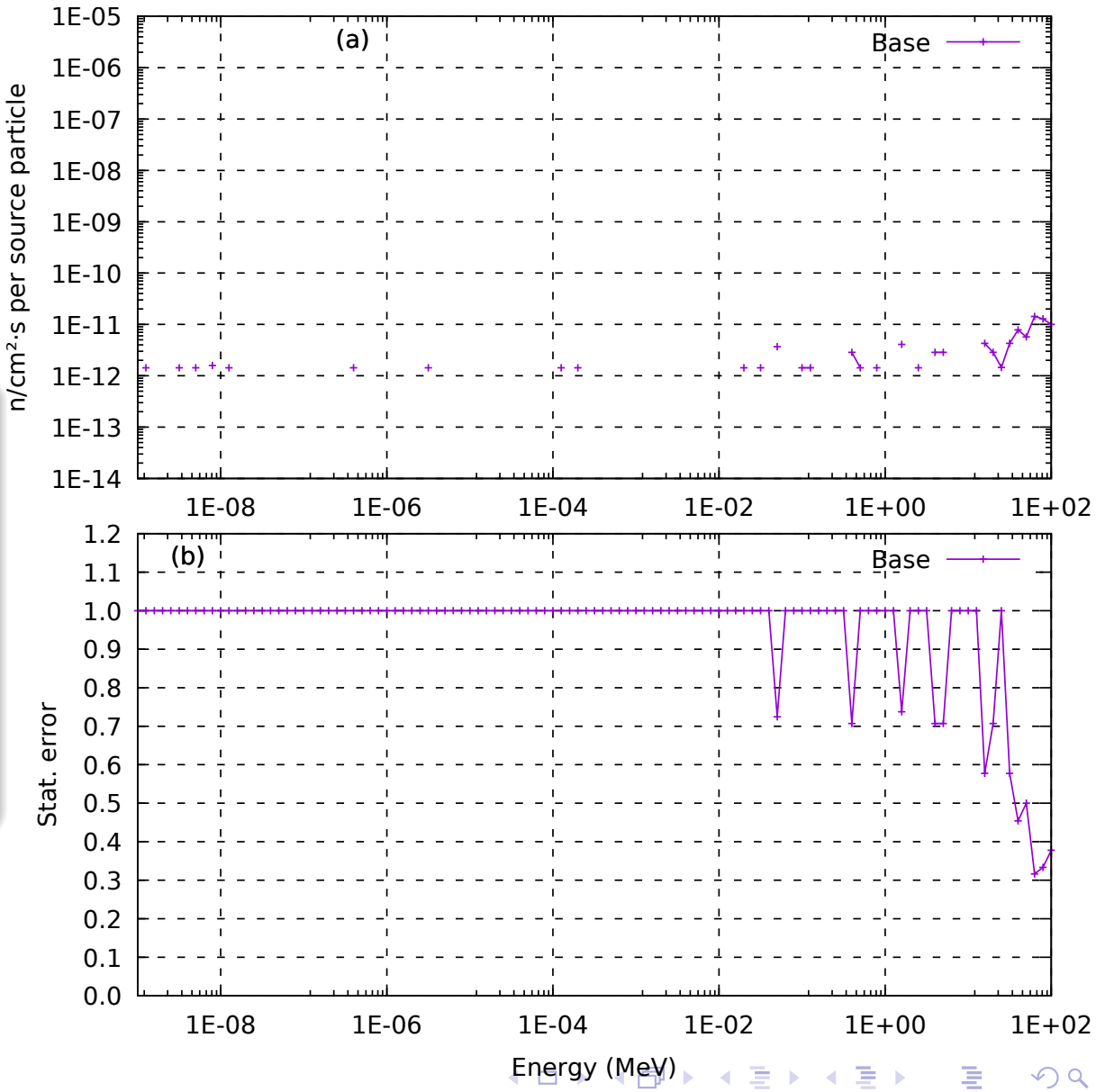
We tested the new DXT code in different configurations, including double and nested DXT to check for cross-talking. This allows us to see the effects of the DXT and the mirror card, in stock and patched configurations. The source is a 0.1 meV to 100 MeV with homogeneous lethargy distribution, in a 4 degrees cone. a 15m straight guide is followed by a 2km radius curve for another 20m. Tally 2 checks the flux at the end of the guide, and tally 22 checks the neutron leak in the final meter.



DXT results(End of guide)

Effects of mirror card and DXT

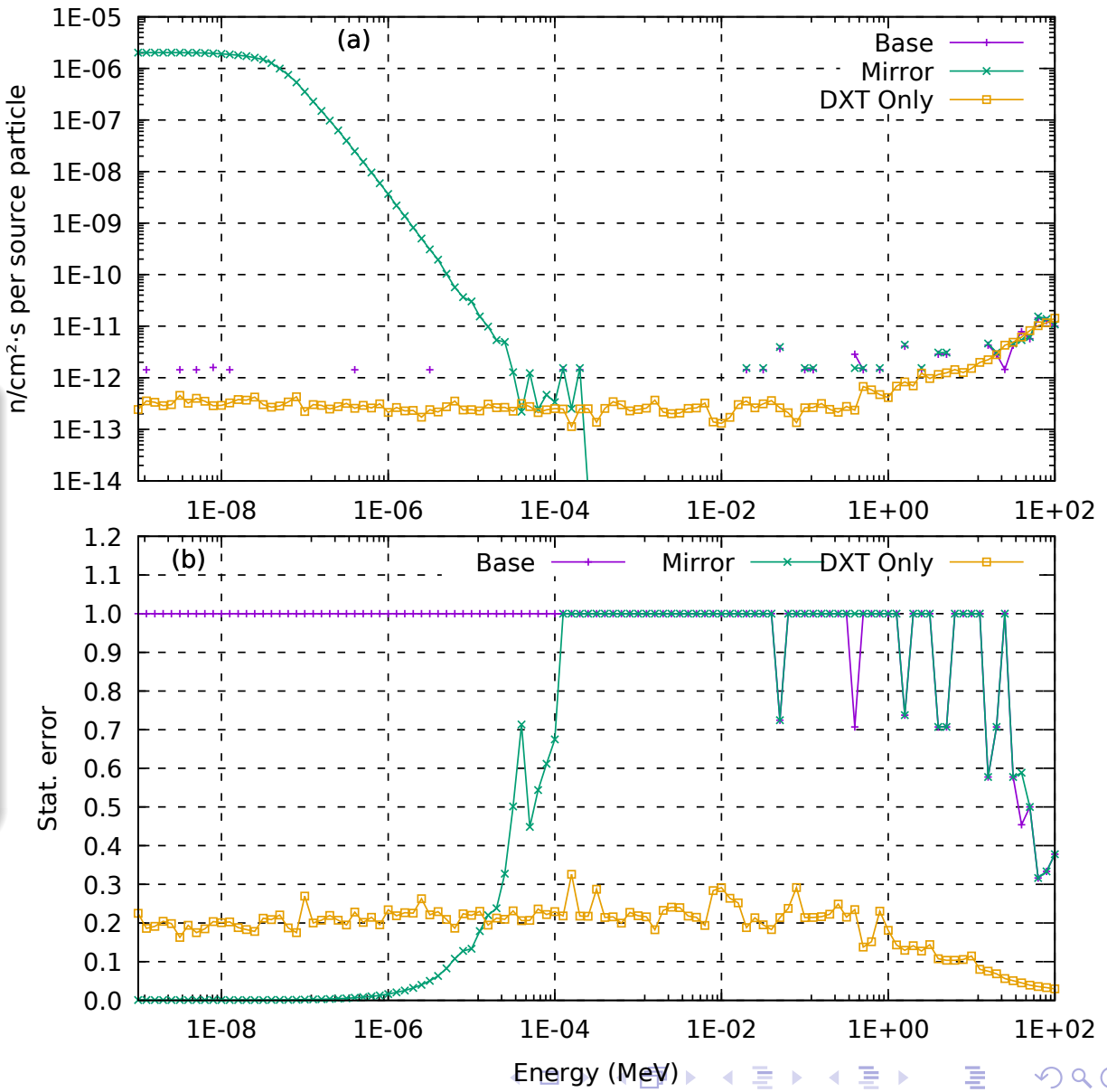
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(End of guide)

Effects of mirror card and DXT

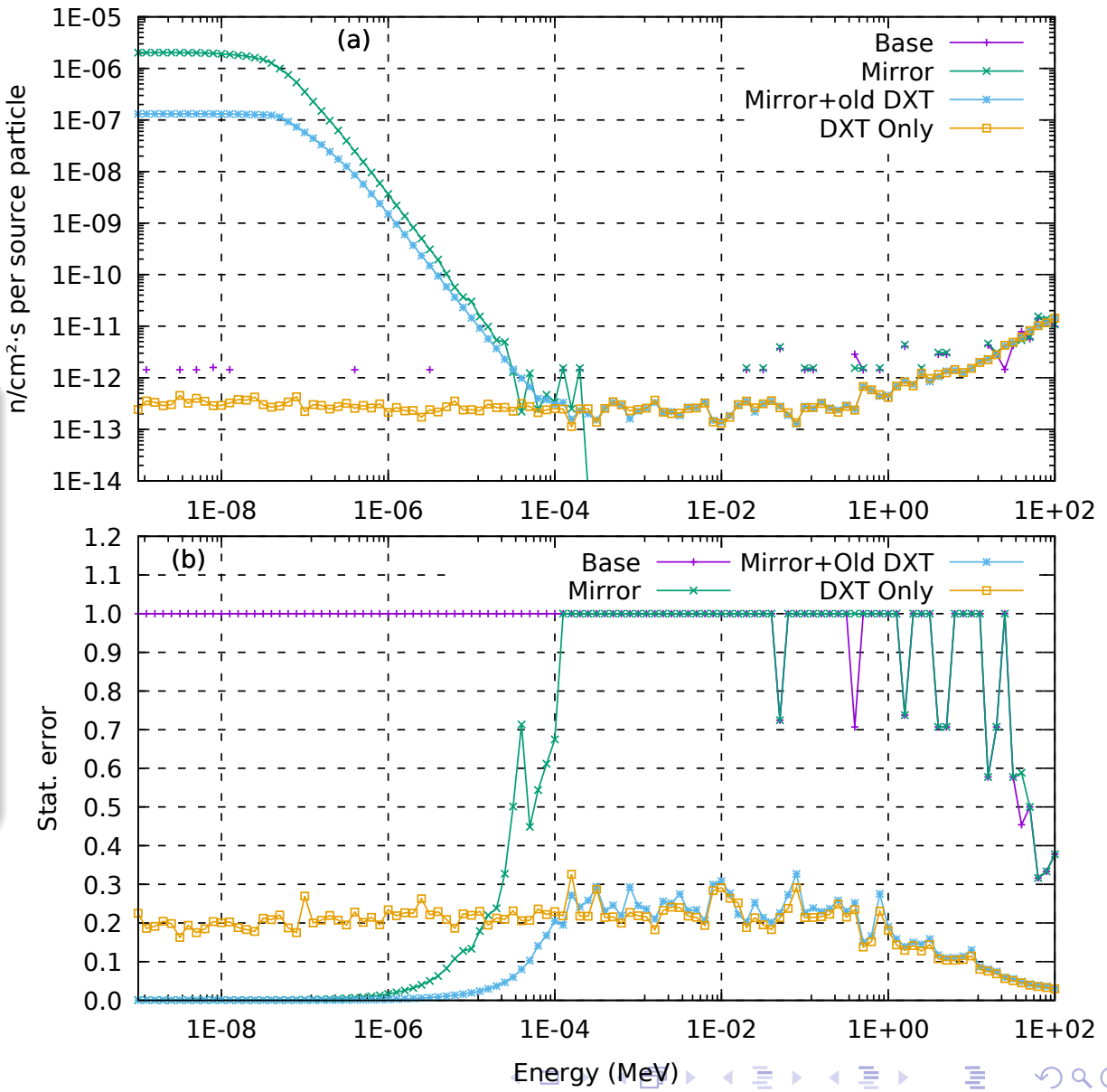
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(End of guide)

Effects of mirror card and DXT

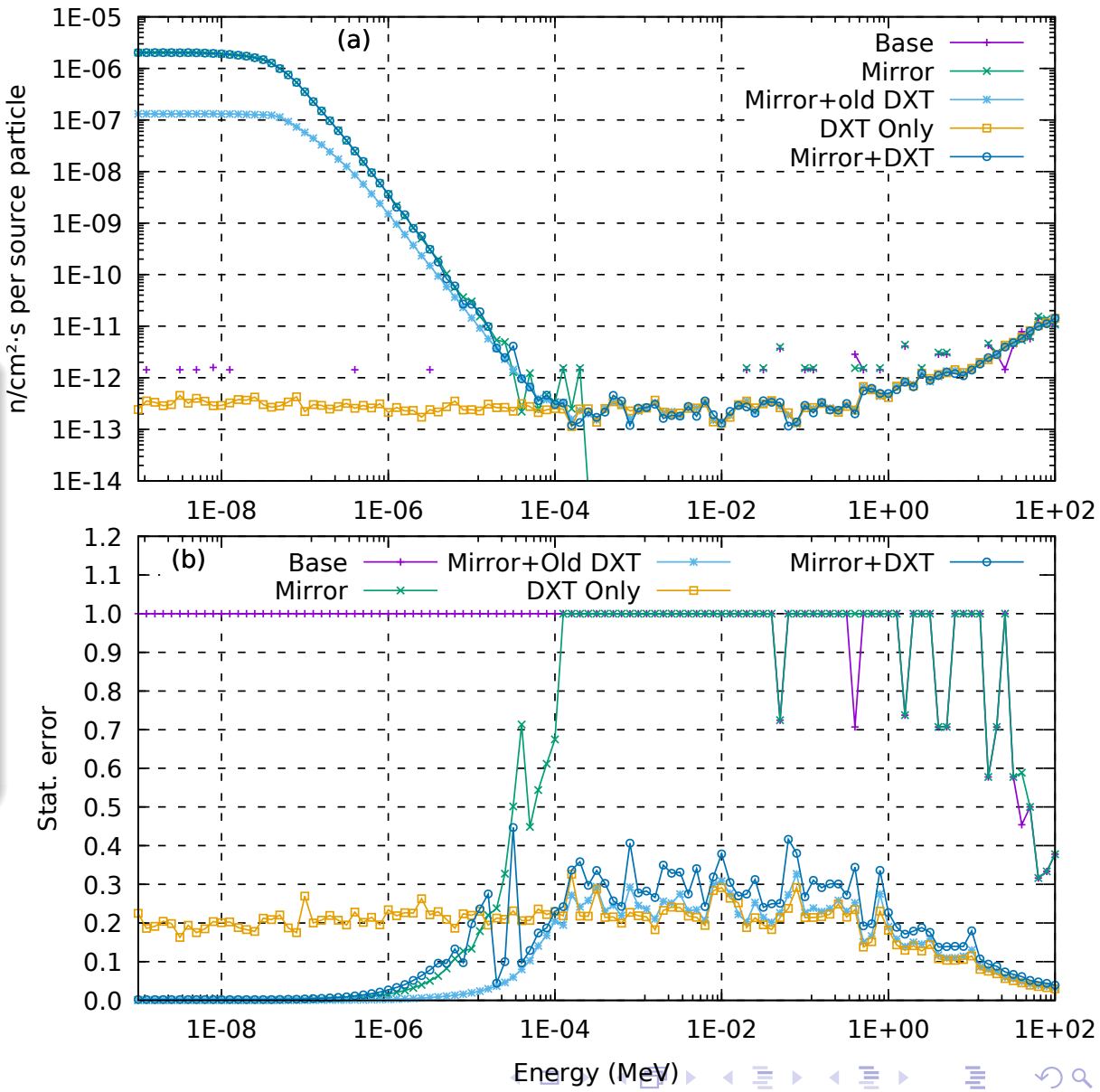
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(End of guide)

Effects of mirror card and DXT

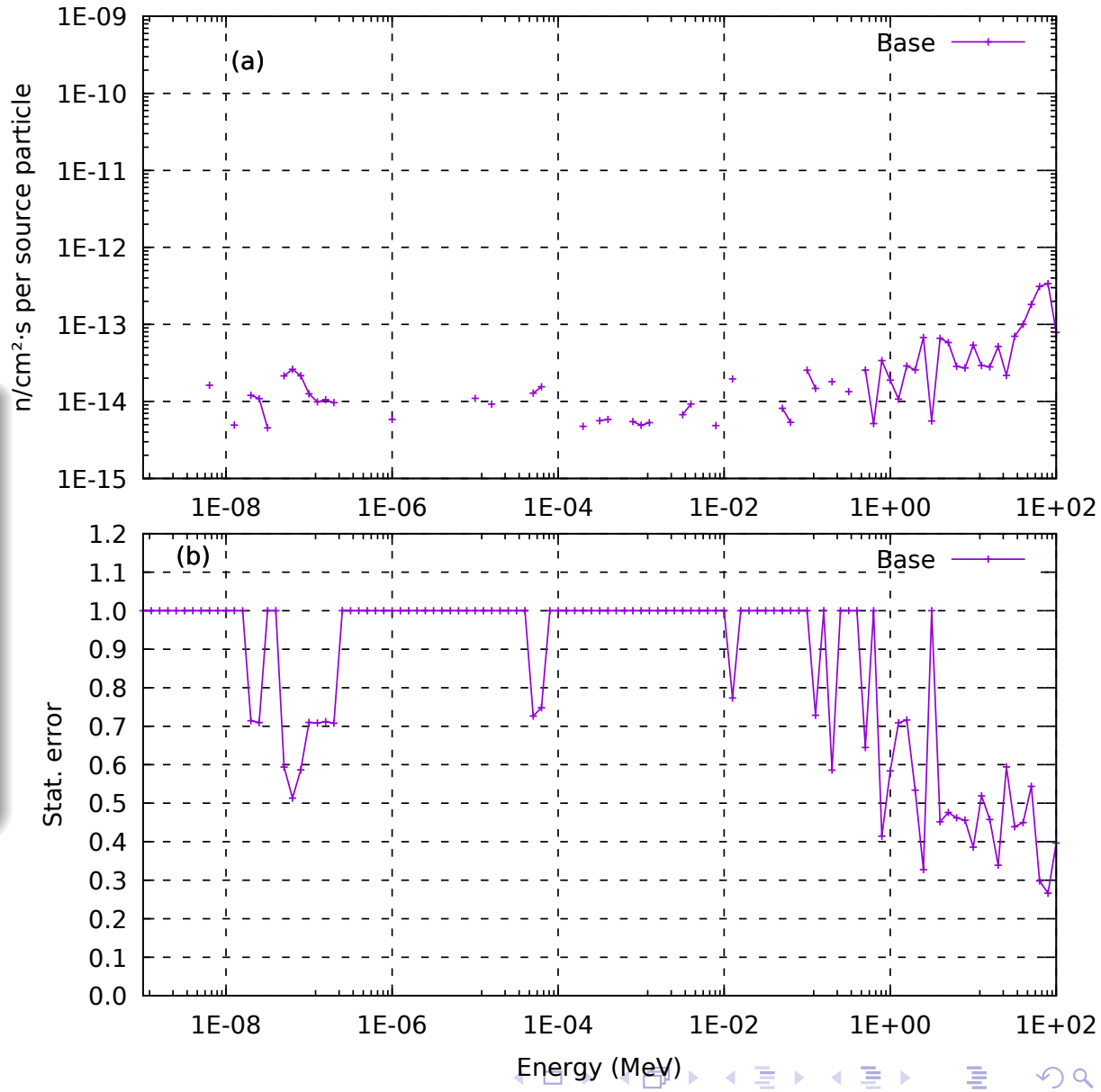
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(Neutron leak)

Effects of mirror card and DXT

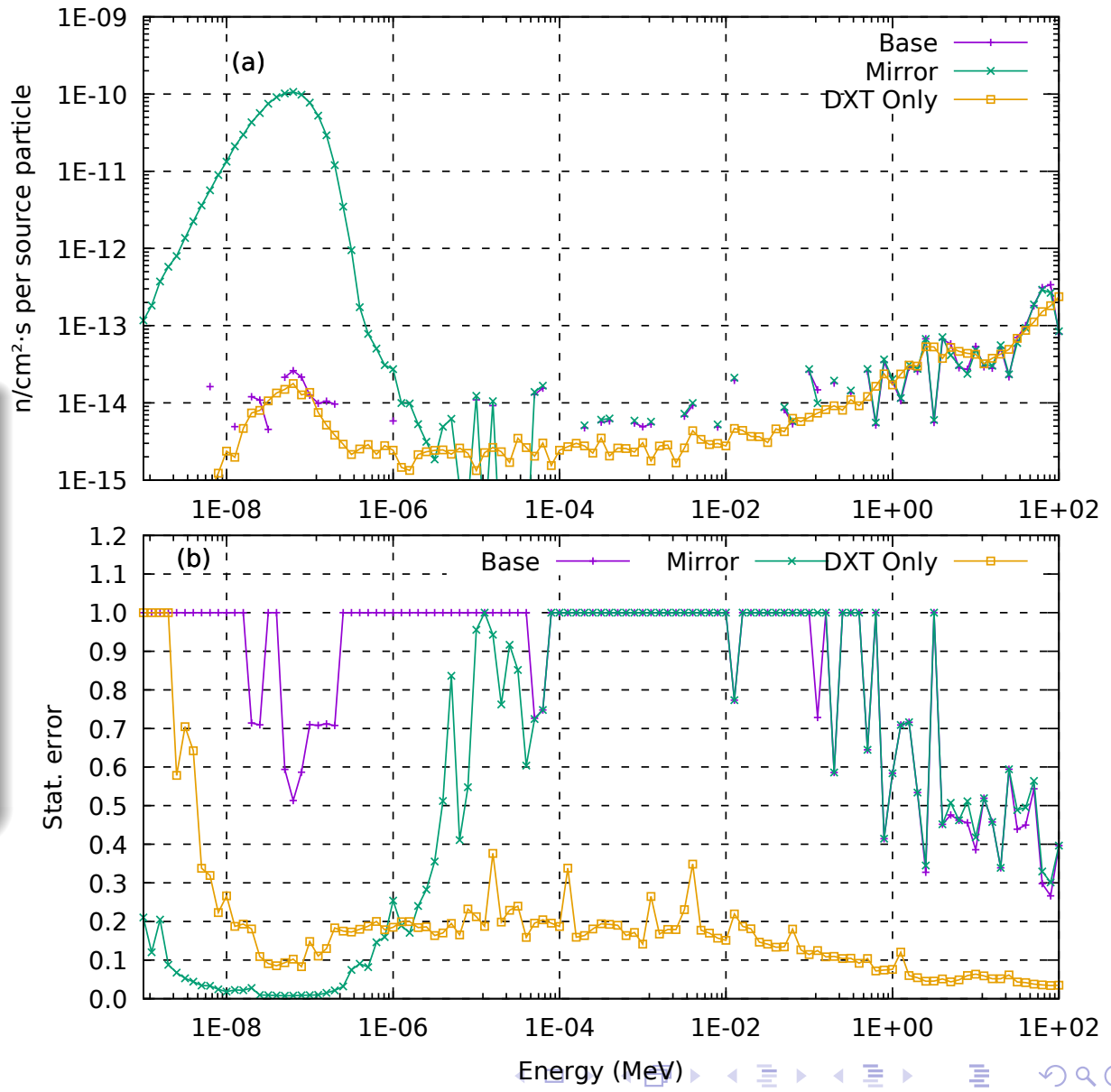
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(Neutron leak)

Effects of mirror card and DXT

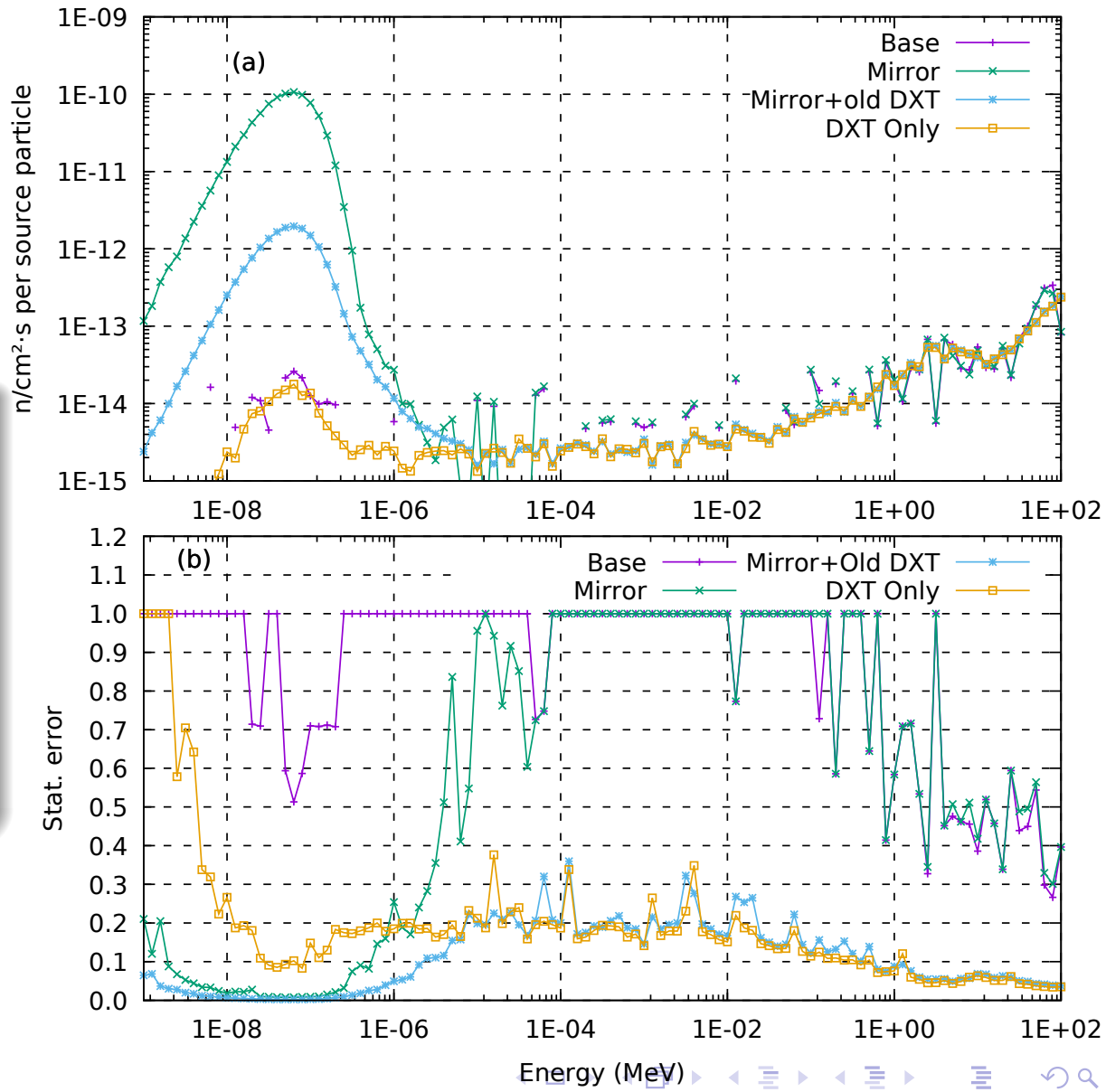
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(Neutron leak)

Effects of mirror card and DXT

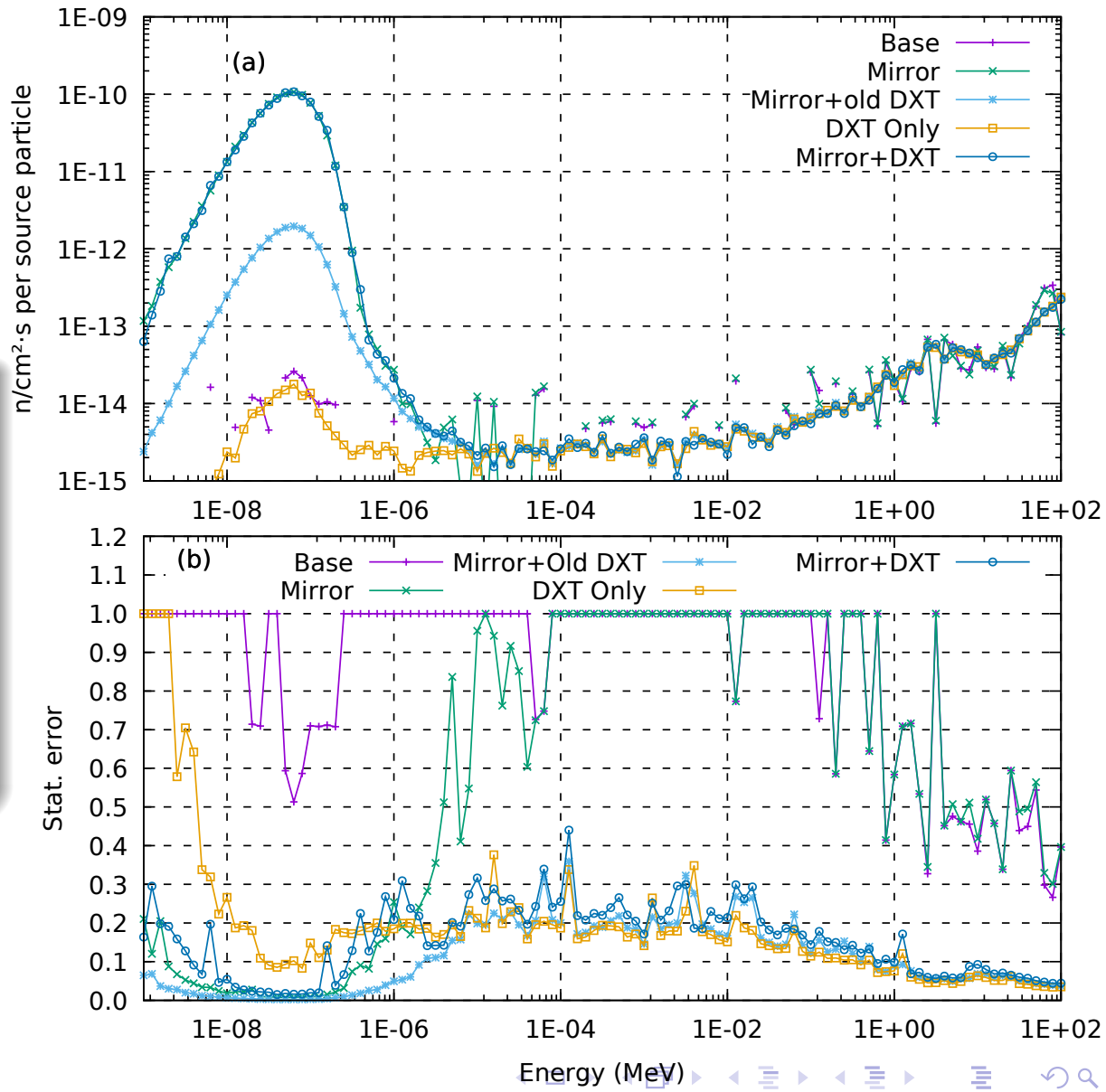
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(Neutron leak)

Effects of mirror card and DXT

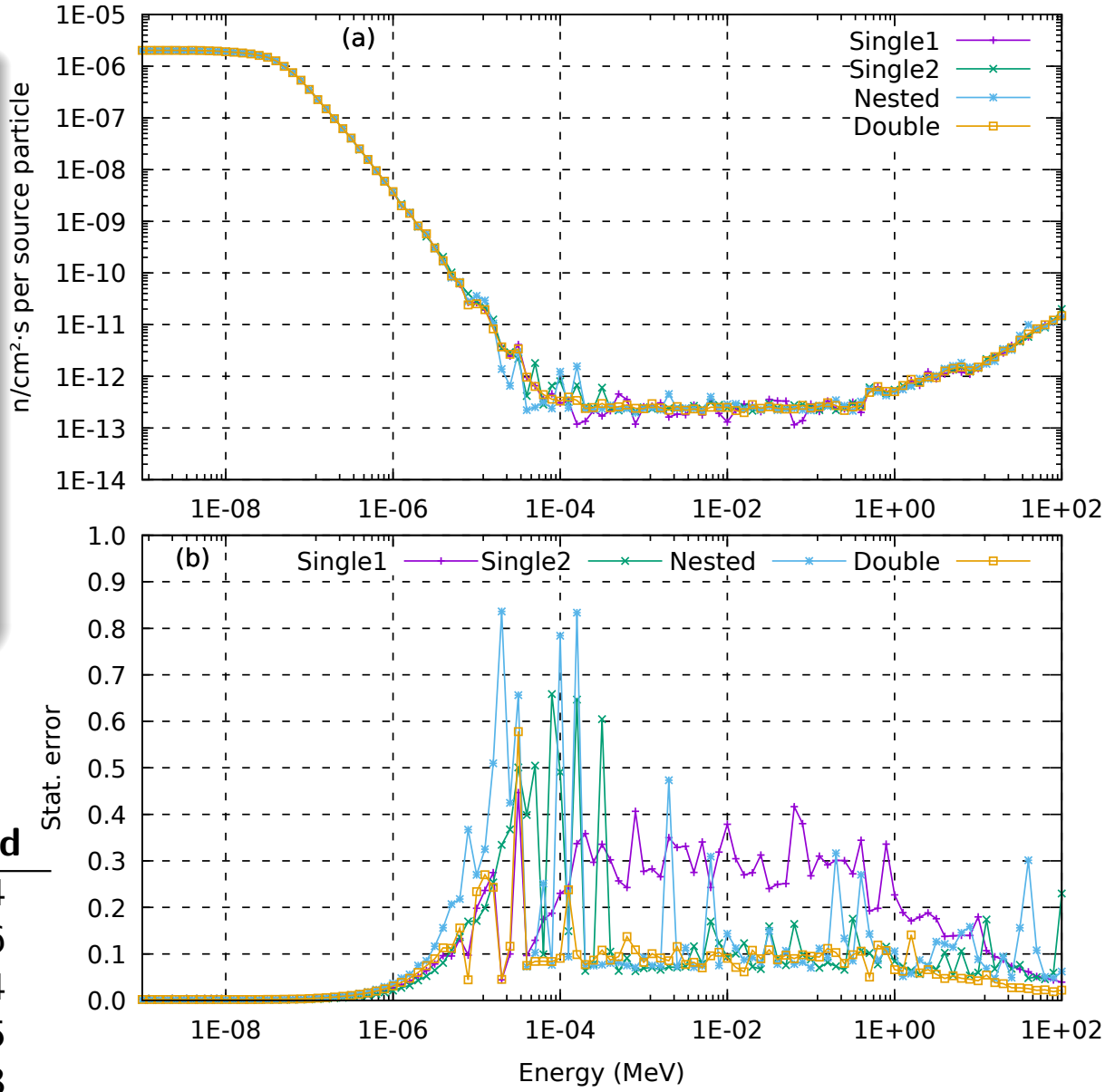
First set of results is the base case (no DXT and no mirror), only mirror card, only DXT, 'stock' DXT with mirror, and new DXT with mirrors. Runtime is roughly constant at 10.000 minutes-core in order to have more comparable results.



DXT results(Different configurations)

The second set of results compares the different DXT configurations, with also a nested option in order to check for possible unforeseen effects. All configuration give the same spectrum, with the difference being the statistical errors. This shows that the DXT is not skewing the results.

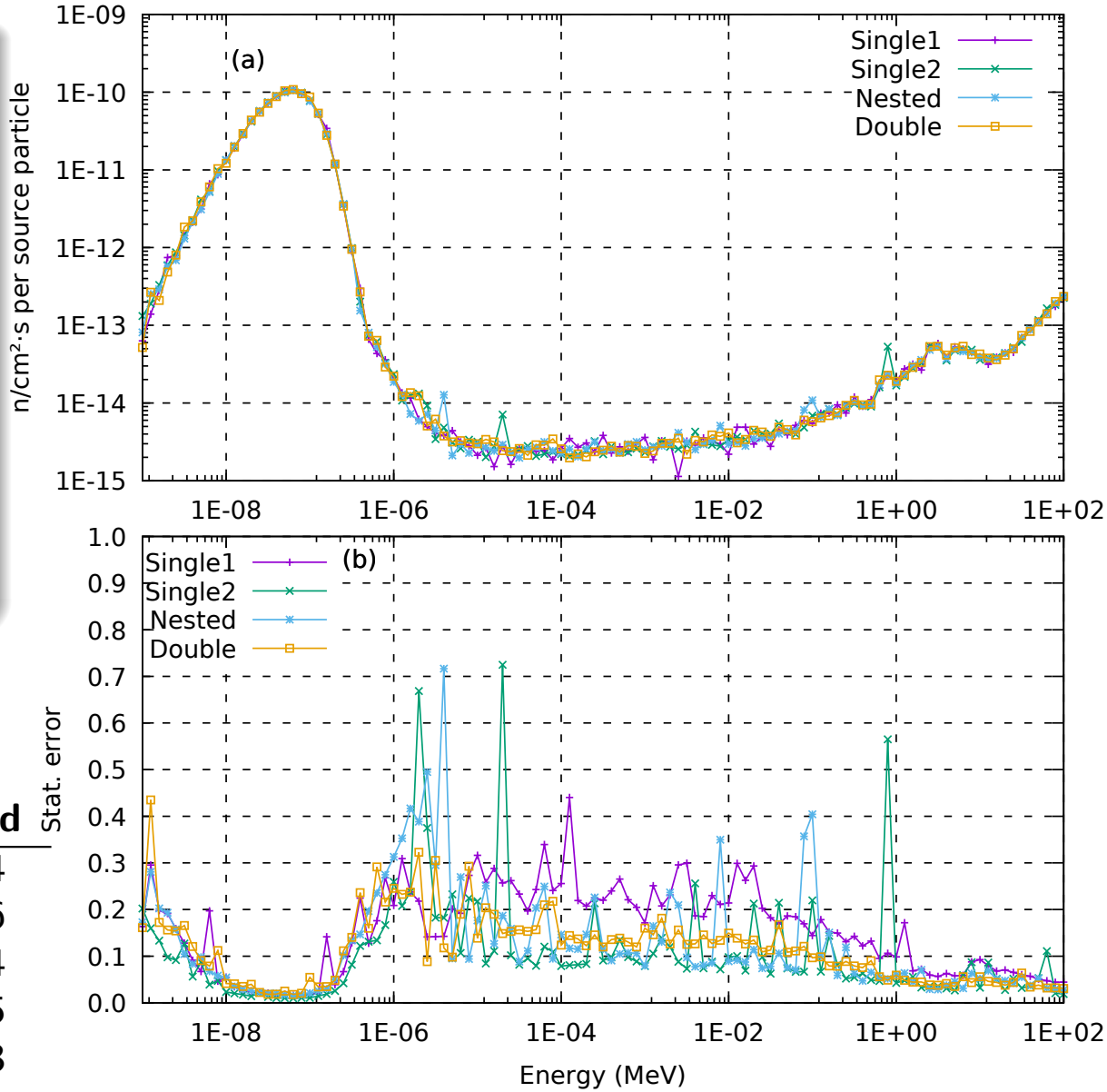
Case	Weight created	Weight destroyed
Single1	4.10E-04	4.10E-04
Single2	8.80E-06	8.76E-06
Double	4.18E-04	4.19E-04
Nested	1.56E-05	1.62E-05
Old DXT	4.09E-04	1.85E-03



DXT results(Different configurations)

The second set of results compares the different DXT configurations, with also a nested option in order to check for possible unforeseen effects. All configuration give the same spectrum, with the difference being the statistical errors. This shows that the DXT is not skewing the results.

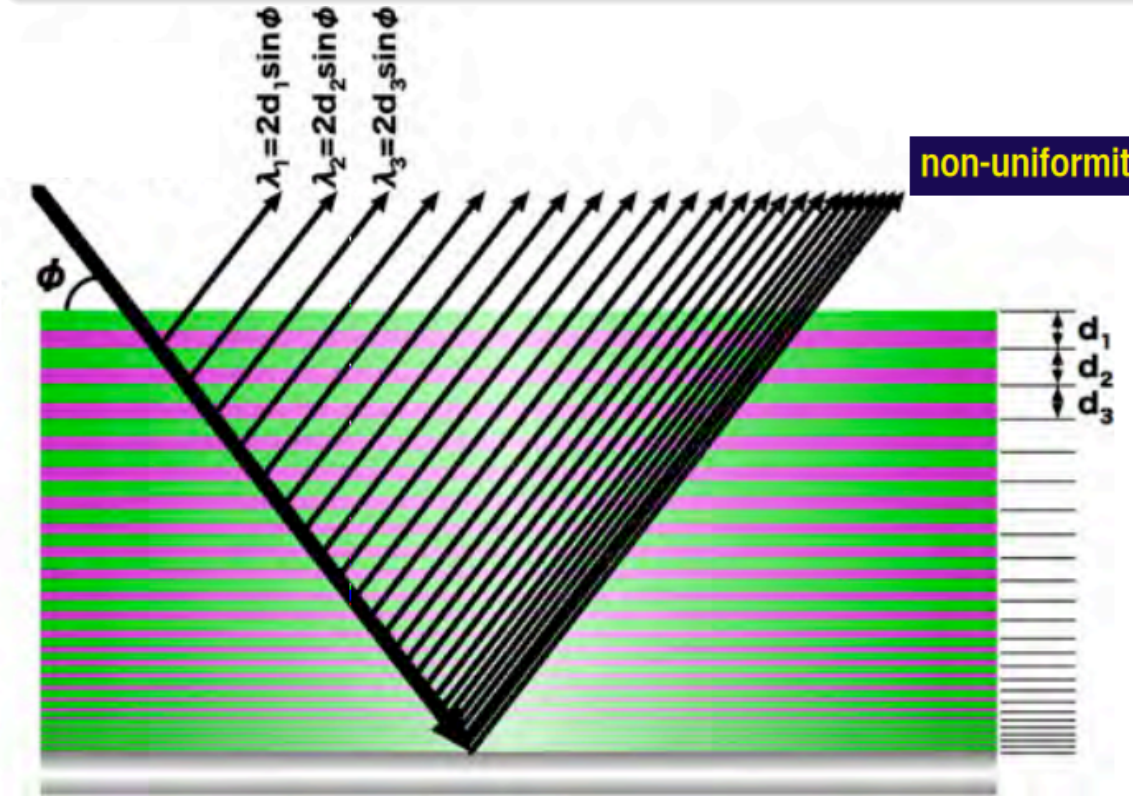
Case	Weight created	Weight destroyed
Single1	4.10E-04	4.10E-04
Single2	8.80E-06	8.76E-06
Double	4.18E-04	4.19E-04
Nested	1.56E-05	1.62E-05
Old DXT	4.09E-04	1.85E-03



Gamma generation in coatings

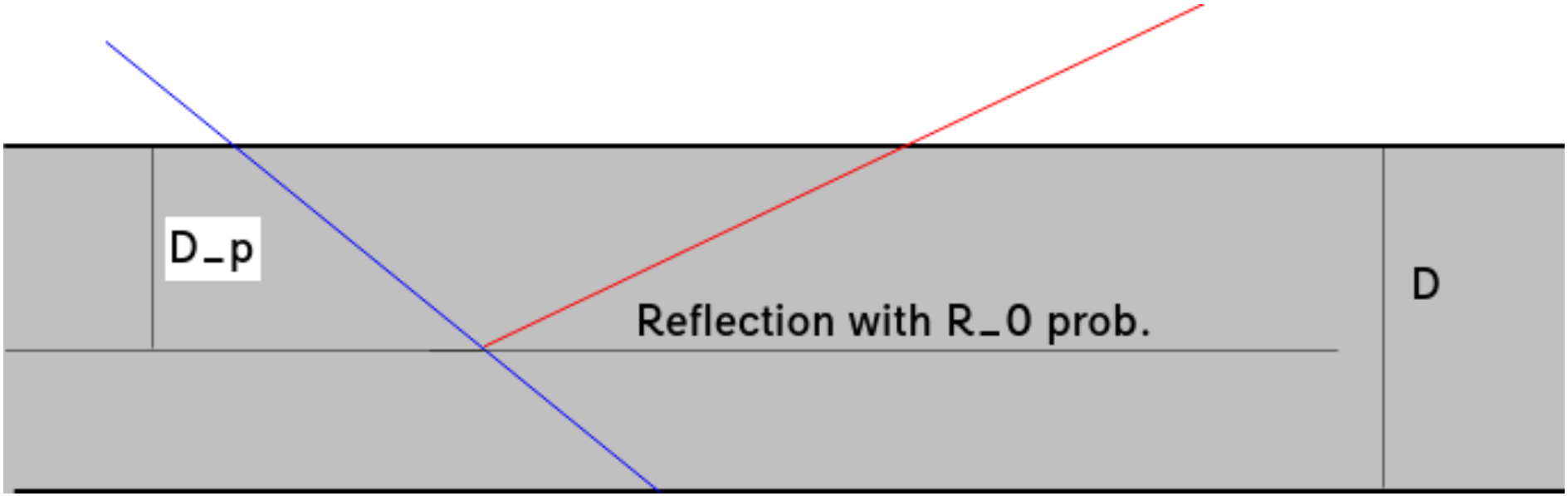
Issue and explanation

The Supermirror implementation of MCNP6 (and PHITS) assumes that reflection takes place on the outer surface of the mirror. However, in reality, this is only reasonably true for neutrons below the critical transfer momentum for the material of the outer layer (Typically Nickel) Q_c. Neutrons above that energy penetrate in the Ni/Ti layers of the supermirrors, and are reflected at a depth where the bilayer depth satisfies Bragg's condition.



Effects on gamma generation

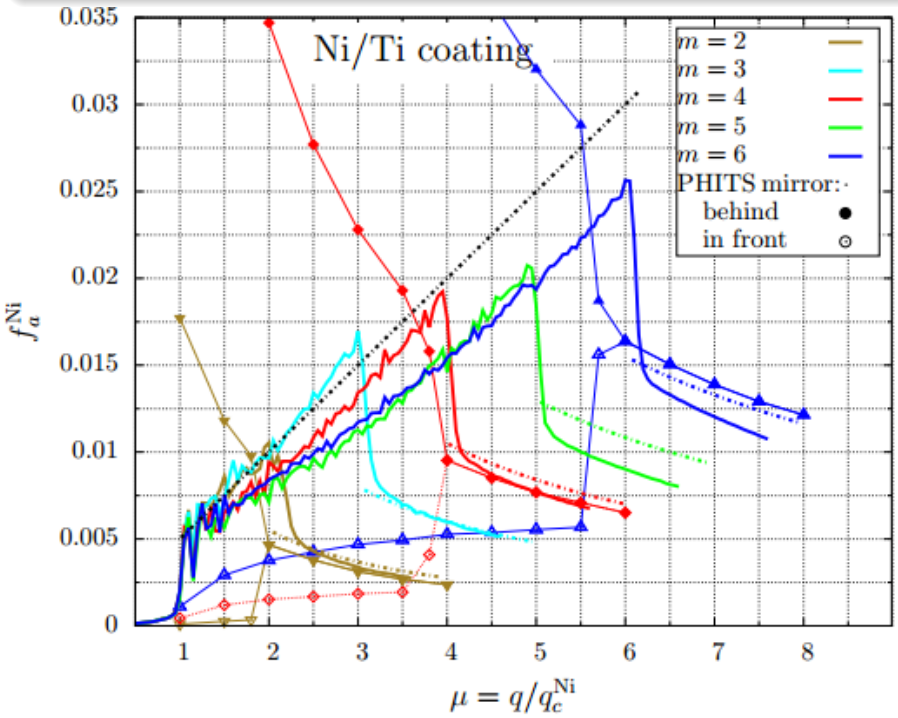
MCNP6 uses an empirical equation also used by McStas, which is an approximation to the actual reflectivity of the supermirror. Even if this equation is a reasonable aproximation, and the reflection of the neutrons is correctly calculated, the abpsortion of neutrons inside the coating is heavily understimated.
In MCNP, the abpsortion is, aproximately:
In a more realistic representation, however, the abpsortion has two parts, one caused by the transmitted wave, and one caused by the reflected wave.
With



Previously proposed solutions

Mirror displacement

It is possible to move the mirror surface to the back of the coating (i.e: Interface between the coating and the substrate), but doing so heavily overstimates the gamma generation, specially for lower Q/Qc values.



From R. Kolevatov
NIMA Volume 922, 1 April 2019, Pages 98-107

Our proposal

A more realistic approach to neutron reflection

In order to have a more realistic representation of reflected neutrons, the idea is having the neutrons make a walk inside the coating that (more or less) corresponds to the actual depth of reflection. Notice that, during that walk, it is possible for the neutron to be scattered, which will cause it to be transmitted since the reflection angles are so low. This effectively reduces reflectivity compared to the equation used.

A Implementation in the software

When a neutron has crossed a surface flagged as a mirror, a property of the particle is activated, which causes an additional track length (DRS) to be defined. This length is a fraction (dependant on Q/Q_c) of the distance to next surface. Because coatings are so thin, the next surface can be assumed to be the coating/substrate interface with a minimal fraction of errors. The walk proceeds as normal, and if DRS is the track length, the reflection subroutine is run. In any case, the flag is cleared for the next track length.

Penetration depth estimation

A critical parameter in this proposal is the fraction of the supermirror depth that the neutron will travel. Because MCNP has no way of simulating the physics at all, an analytical expression must be used. From the Hayter and Mook algorithm, it follows that said depth varies with $(Q/Q_c)^3$. Different algorithms such as Masalovich will have a different penetration depth. However, we believe our calculation will still be a reasonable approach.

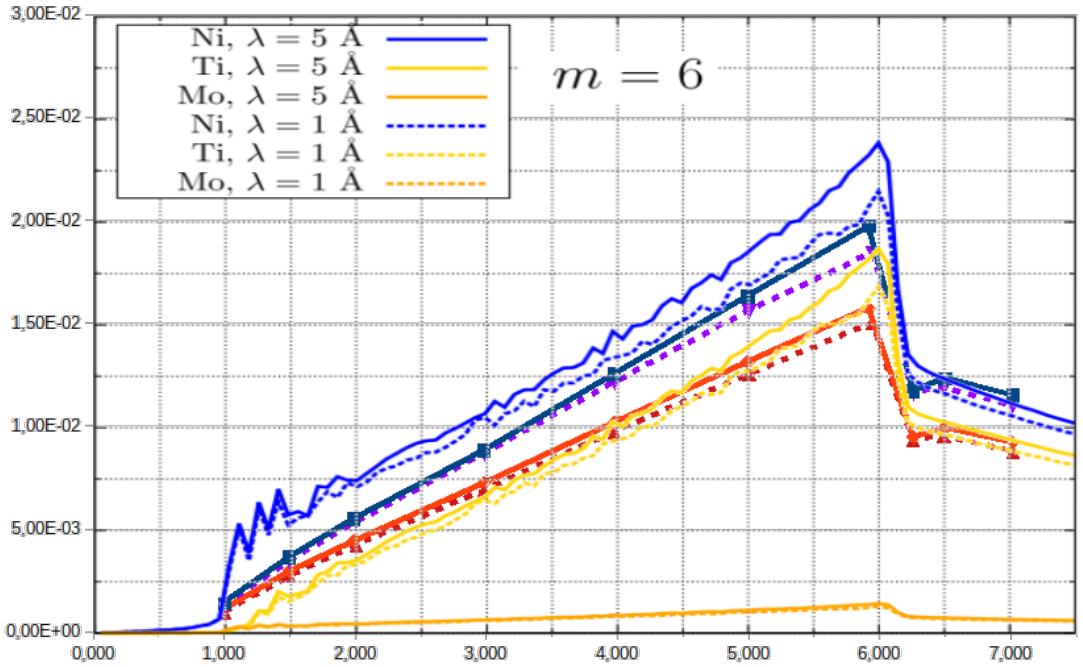
Change in reflectivity

Because we are now transporting the neutron inside the coating, there is an additional loss that reduces reflectivity compared to the McStas formula. Thus, the coefficients of the formula must be modified in order to compensate for this. A spreadsheet to help match provider data has been developed. The reflectivity is no longer Wavelength independent as it used to be the case. While this diverges from McStas, it is ultimately an improvement in the realism of the calculations.

Comparison between QM calculations and MCNP approximation

Graphical comparison

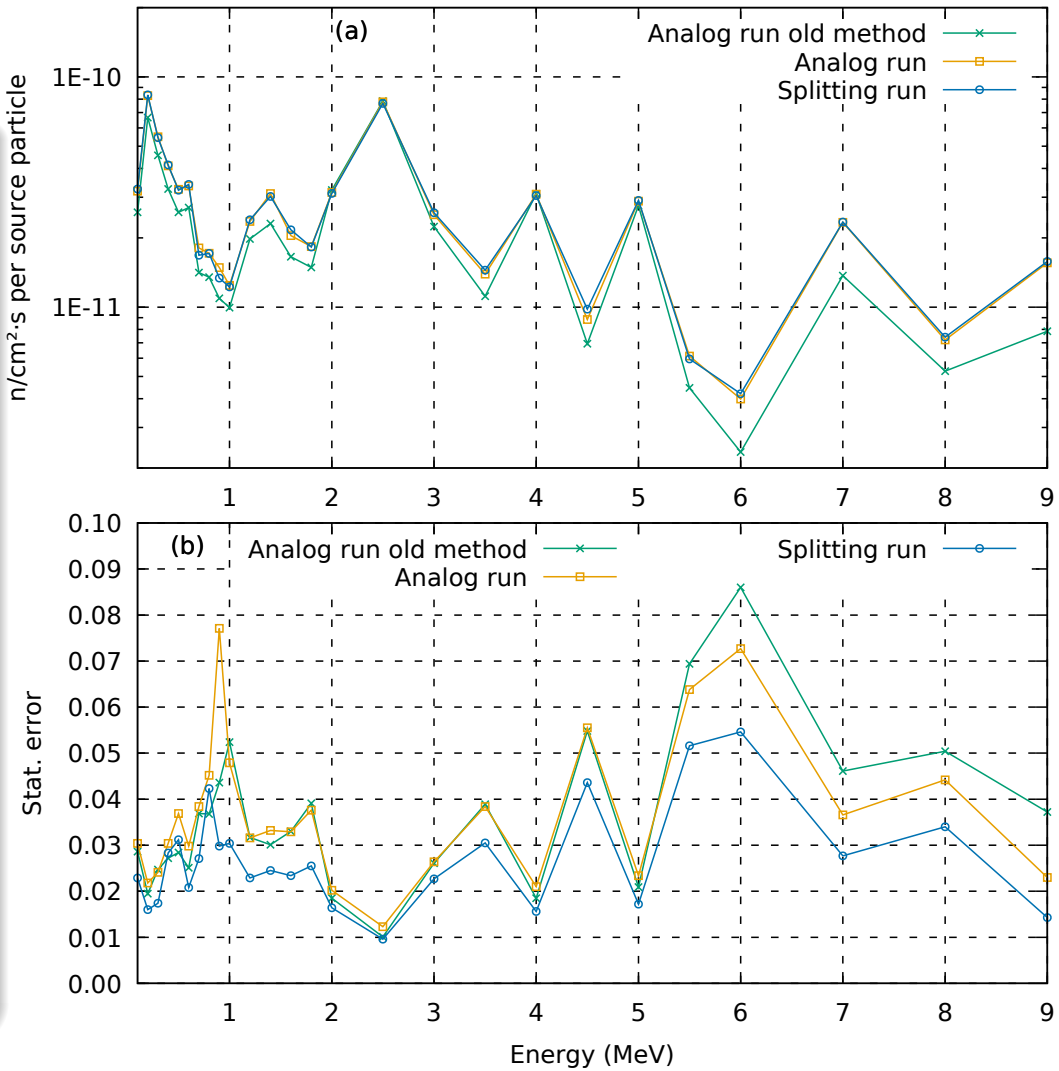
The results are fairly similar in the middle u range. At lower u range, the outer layer of Ni causes a significant difference. At u close to m, the reflectivity of the mirrors does not quite match, with the MCNP adjustment being somewhat lower, causing in turn slightly lower abpsortion. Coating thickness is 12 um.



Gamma generation in the testbench

RFLAG and method effects

This modification in the logic of the neutron reflection has a visible effect when we look at the gamma generation in the test bench specified before. The difference is specially noticeable at the higher gamma energies (over 5 MeV), which makes sense considering the emitted spectrum from Ti(n,γ) and Ni(n,γ) reactions. While the number of those gammas are few compared to those from absorption in the substrate, their higher energy more than make up for it in terms of dose relevance. Also, notice that RFLAG=2 does improve the statistic for a similar runtime.



Using the supermirror physics in MCNP6.2

The basics

The declaration of a reflective surface is the same as in MCNPX: **REFLEn M** $C_1 C_2 ... C_n$ Where n is the number of reflecting surface, M is the reflecting mode declared in REFF card, and C_1 to C_i are the cells the reflecting neutrons enter (positive sign) or leave (negative sign).

Duplicate surfaces and REFLE cards

Using a REFLE card in a duplicate card will result in said surface not reflecting anything at all. The results of the REFLE cards can be checked in the output, so the user can tell if it is what they desired. The desirable behavour would be imcn() handling this. This however, leaves an open question:

- What happens when a duplicated surface has different m-number in different parts?

RFLAG card

Using particle split at mirrors

The RFLAG card must be declared for each REFLE card if you wish the default behavour (RFLAG=0) to change. This is simply declared as **RFLAGn mode** Where n is the corresponding REFLE card and mode is 0 or A for analog behavour, 1 or R for keep only reflective part and 2 or S for splitting. Unlike, say, tally binning, there is no RFLAG0 to change the default, so you need to declare it for every surface.

Reflection coefficients and coating walk

Declaration and numbers used

The reflection conditions must be declared in a REFF card, featuring the coefficients that define the reflectivity. The equations is the same as McStas, but, as we will see, because of the physics, the coefficients are different: **REFFn R0 Qc m A W** Where n is the number indicating the reflecting conditions, R0 is the reflectivity below the critical momentum, Qc is said critical momentum, m is the m number, and A and W are coefficients that dictate the decline of the reflectivity.

McStas coefficients Vs. MCNP coefficients

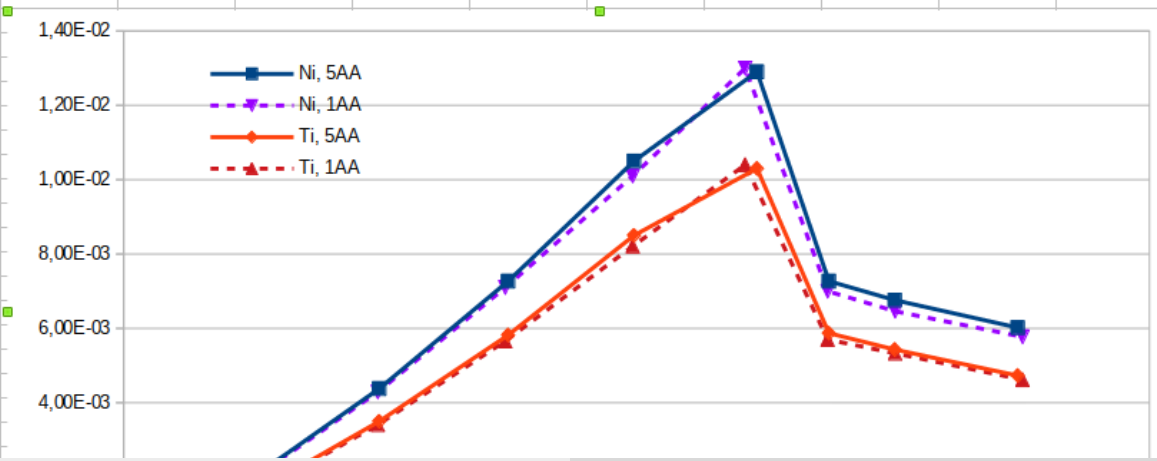
Originally, using the very same number as in McStas was the way to go, since it ensured that both codes would do the same to reflect a neutron. However, with the development of the detailed physics, this is no longer the case, as the interaction with the coating reduces the effective reflectivity. Thus, the coefficients must be adjusted accordingly

Work performed for adjusting coefficients

Quick reference spreadsheet

Using the abpsortion and total cross-section, the total reflectivity can be calculated in a spreadsheet, and therefore we can adjust the parameters to better match the curves. An additional benefit is that this method also captures the wavelength dependecy of reflectivity.

		Sin(alpha)	QQ	a	b	RR	u	D		I0	I1	R1	R2	T1
		8,56E-03	0,021516045	2,00	0,999951865	0,989952347	1,001	0,129920064		1	0,996888597	0,986872206	**0,983801648**	0,010016391
		1,28E-02	0,032173525	2,00	0,967979425	0,958299631	1,496	0,434395021		1	0,993042892	0,951632637	**0,945012026**	0,041410255
		1,71E-02	0,042981819	2,00	0,935554544	0,926198999	1,999	1,035722143		1	0,987583448	0,914698801	**0,903341395**	0,072884647
m	3	2,13E-02	0,053538757	2,00	0,90388373	0,894844893	2,490	2,001675809		1	0,980735045	0,877605746	**0,860698711**	0,103129299
R0	0,99	2,54E-02	0,063844339	1,58	0,872966984	0,680789441	2,970	3,394345768		1	0,97260469	0,662139003	**0,643999499**	0,310465687
a1	3	2,78E-02	0,069876875	0,00	0,854869376	1,80814E-05	3,250	4,450300304		1	0,967183037	1,7488E-05	**1,69141E-05**	0,967165549
ww	0,001	3,00E-02	0,075406699	0,00	0,838279902	2,78988E-10	3,507	5,592663563		1	0,961783466	2,68326E-10	**2,58072E-10**	0,961783465
Qc	0,0215	3,41E-02	0,085712282	0,00	0,807363155	0	3,987	8,213307205		1	0,950623813	0	**0**	0,950623813
E(meV)	3,27													
D_total(um)	3,5													
Sigma_total	2,05E-04													
Sigma_abps	1,07E-04													
Adjustment case		1,00E+00	2,531639785			1		100		1	0,9795	0,9795	**0,95942025**	0



	Penetration ideas			
	1			#¡REF!
	2			
	3			

Geometry of the coating

Microns thickness in a 100m+ model

A (rather undesirable) consequence of the modifications is that the coatings must now be precisely modelled in order to obtain consistent results. This means that not implementing the coatings will result in very wrong results. Of course, the older patch without this modification is still available, but we probably would want to be able to select behavour

- In which card?
- Situations where it is needed?

Dealing with curved surfaces

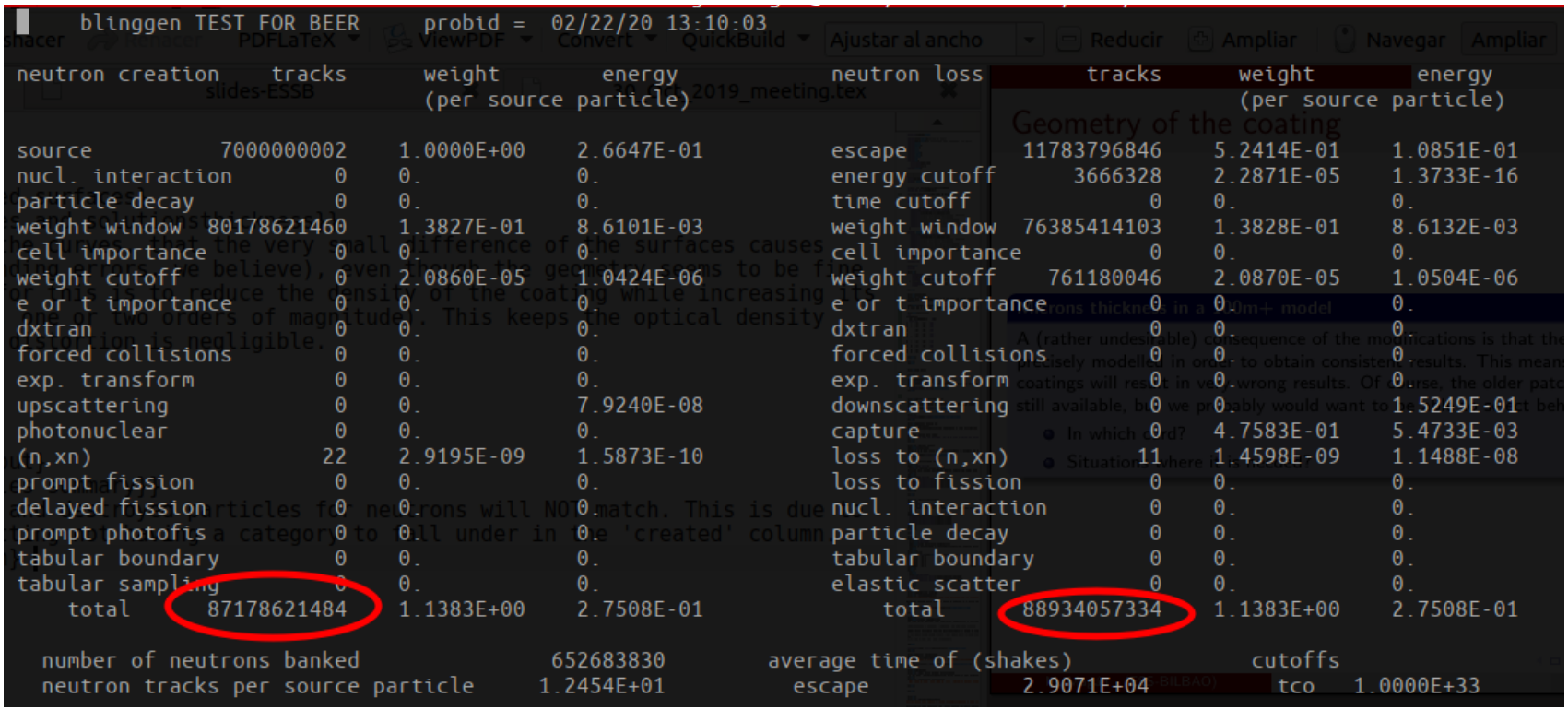
Lost particles and solutionsthickness

It is possible, at least for the curves, that the very small difference of the surfaces causes particle loss (Because of rounding errors, we believe), even though the geometry seems to be fine in the plotter. A workaround for this is to reduce the density of the coating while increasing its depth by the same factor (say, one or two orders of magnitude). This keeps the optical density constant, and the geometrical distortion is negligible.

Debugging and output

Split particles summary

If using RFLAG=2, the created and destroyed particles for neutrons will NOT match. This is due to the particles created by splitting not having a category to fall under in the 'created' column. **However, weight MUST match. Again, this is not desired behavour, but implementing the logging takes some time, and circular dependecy solving .**



blinggen TEST FOR BEER probid = 02/22/20 13:10:03

neutron creation	tracks	weight (per source particle)	energy	neutron loss	tracks	weight (per source particle)	energy
source	7000000002	1.0000E+00	2.6647E-01	escape	11783796846	5.2414E-01	1.0851E-01
nucl. interaction	0	0.	0.	energy cutoff	3666328	2.2871E-05	1.3733E-16
particle decay	0	0.	0.	time cutoff	0	0.	0.
weight window	80178621460	1.3827E-01	8.6101E-03	weight window	76385414103	1.3828E-01	8.6132E-03
cell importance	0	0.	0.	cell importance	0	0.	0.
weight cutoff	0	2.0860E-05	1.0424E-06	weight cutoff	761180046	2.0870E-05	1.0504E-06
e or t importance	0	0.	0.	e or t importance	0	0.	0.
dxtran	0	0.	0.	dxtran	0	0.	0.
forced collisions	0	0.	0.	forced collisions	0	0.	0.
exp. transform	0	0.	0.	exp. transform	0	0.	0.
upscattering	0	0.	7.9240E-08	downscattering	0	0.	1.5249E-01
photonuclear	0	0.	0.	capture	0	4.7583E-01	5.4733E-03
(n,xn)	22	2.9195E-09	1.5873E-10	loss to (n,xn)	11	1.4598E-09	1.1488E-08
prompt fission	0	0.	0.	loss to fission	0	0.	0.
delayed fission	0	0.	0.	nucl. interaction	0	0.	0.
prompt photofis	0	0.	0.	particle decay	0	0.	0.
tabular boundary	0	0.	0.	tabular boundary	0	0.	0.
tabular sampling	0	0.	0.	elastic scatter	0	0.	0.
total	87178621484	1.1383E+00	2.7508E-01	total	88934057334	1.1383E+00	2.7508E-01

number of neutrons banked 652683830 average time of (shakes) cutoffs
neutron tracks per source particle 1.2454E+01 escape 2.9071E+04 tco 1.0000E+33

DISCUSSION AND QUESTIONS