GPU Acceleration (and else)

By a non-specialist…

High Performance Monte-Carlo Radiation Simulations workshop
5–6 March 2026,
Sikyon Coast Hotel & Resort

Marc Verderi,
Laboratoire Leprince-Ringuet

Some preliminary remarks

- I am from HEP background, so I appreciated very much the note accompanying the meeting, it helps (me) realizing how different the HEP and space use-cases are:
  - Basically, for most space missions, **particle physics is a nuisance...** (somewhat ≠ than HEP case)
    - And space would live happily without particle physics…
  - Some (small) parts are critical, while details in others don't matter (somewhat ≠ than HEP case)
  - So, I will come with questions, I hope relevant
- I have to mention I never practiced GRAS
  - So excuse in advance naïve questions…
- I did not stick to "GPU acceleration" *per se*
  - This is one considered way to speed-up the simulation
  - But speeding-up can be done in other ways
    - Biasing, of course, but also "classical techniques or tunes"
- We always want the simulation to be "as fast as possible"
  - But what speed-up is looked for ? Can it be at the cost of physics quality, or not ?
  - And, at the opposite, are there aspects for which the physics quality need to be improved, even at the cost of computing speed ?
  - Being an adjustable trade-off between "physics quality" and "computing performance" is the most we can ask to a physics software
    - And so, we cannot talk about one without talking about the other.

High Energy Particle MC on GPU ?

- Running on GPU is ***not a "technical" problem*** of rewriting the code it is a ***"processing flow" problem***.
- GPU are designed, and super efficient, for problems with **« many very similar things »** **« behaving almost the same »** and that **« live long in memory »**
  - This leads to strong parallelization –there are **"no divergences"**– → **efficient processing**
  - Typical example and original motivation : **optical photons**
- But a high energy physics Monte Carlo like Geant4 is **very subject to divergences**:
  - **« ~~many very similar things~~ » → « many very different things » !**
    - Many type of particles; many type of solids
  - **« ~~behaving almost the same~~ » → « behaving not at all the same » !**
    - Particle & energy dependence of physics; geometry calculations strongly differ between solids
  - **« ~~live long in memory~~ » → « many particles die after one or a few steps »**

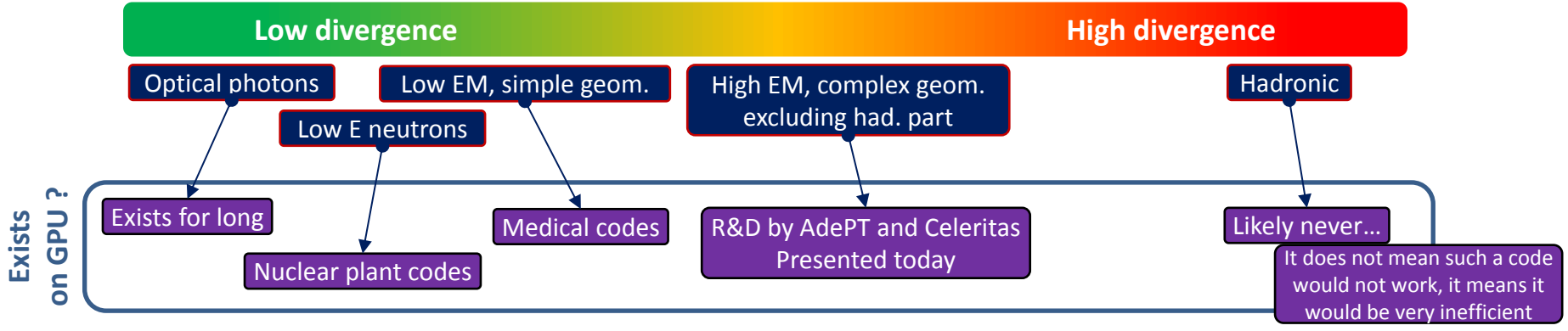


Table of issues from meeting note

	Physical Effect	Geometry	E range	Particle type	Phys. list	Particularity
TID	Total Ionizing Dose	Spacecraft scale down to small scales	keV – 500 MeV	e- , p	EM opt4 QGSP_BIC	Use mass calculation. Automated volume name is annoying
TNID & DDD	Total Non-Ionizing Dose & Displacement Damage Dose	Same	100 keV – 500 MeV	p, e-, n, heavy ions	QGSP_BIC_HP	Too few materials in GRAS for DDD: Si, GaAs, InP
Internal Charging	Charge accumulation in dielectric → discharge	Cables, connectors, PCB, multilayer dielectric structures	Low E – 2 MeV	e- , in GRAS	EM opt4	Significant CPU difference between Geant4 mesh and Gmsh
SEU	Single Event Upset	Micrometer scale of digital semiconductors	1 MeV/u – several GeV/u	p, heavy ions	EM opt4 QGSP_BIC	Small XS → biasing

Table of issues from meeting note

What about GPU acceleration ?

	Physical Effect	Geometry	E range	Particle type	Phys. list	Particularity
TID	Total Ionizing Dose	Spacecraft scale down to small scales	keV – 500 MeV	e-, p	EM opt4 QGSP_BIC	Use mass calculation. Automated volume name is annoying
TNID & DDD	Total Non-Ionizing Dose & Displacement Damage Dose	Same	100 keV – 500 MeV	p, e-, n, heavy ions	QGSP_BIC_HP	Too few materials in GRAS for DDD: Si, GaAs, InP
Internal Charging	Charge accumulation in dielectric → discharge	Cables, connectors, PCB, multilayer dielectric structures	Low E – 2 MeV	e-, in GRAS Is processing time a problem ?	EM opt4	Significant CPU difference between Geant4 mesh and Gmsh
SEU	Single Event Upset	Micrometer scale of digital semiconductors	1 MeV/u – several GeV/u	p, heavy ions	EM opt4 QGSP_BIC	Small XS → biasing : yes !

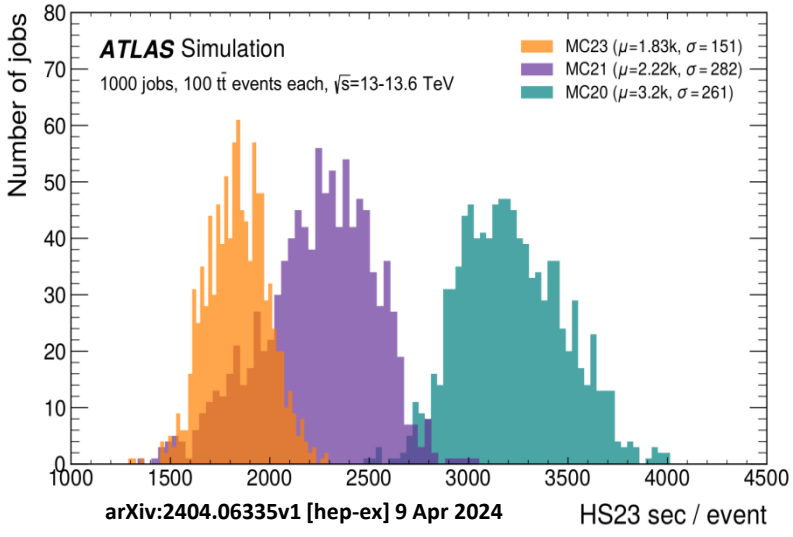
Candidates for GPU accelerators (many questions still)

Need the "speed of light": kill all EM particles as soon as created : what is the boost ?

Performances from user's tunes

- **"Classical" speed-up techniques may lead to large improvement**

- **ATLAS Experiment:** example of in depth examination for performance improvements
  - In close collaboration with Geant4 developers
  - **mc21 → mc20 improvements (mcXX = ATLAS softw.)**
    - EM range cuts, O(6%)
      - reduces the number of low energy electrons
    - Gamma & neutron Russian roulette, O(10%)
      - reduces the number of tracked gammas and neutrons
    - Geant4 built as a single library, O(5 – 7 %)
    - G4GammaGeneralProcess, O(3%)
      - improves the time spent to transport gammas
    - magnetic field optimization, O(3%)
    - optimized EM end-cap geometry, O(5 – 6 %)
  - **mc21 → mc23 improvements:**
    - **Woodcock tracking in the EM end-cap, 17.5%**
    - VecGeom for polycones, tubes, and cones O(2 – 7%)



  - **Overall : a 1.84 speed-up at nearly <u>no cost on the physics !</u>**

- (I repeat my ignorance on GRAS) **Would GRAS benefit from similar examination ?**

AdePT & Celeritas main findings

- In December 2023; an assessment of AdePT and Celeritas was made
  - (Followed by a delta-assessment in March 2025)
  - AdePT and Celeritas focus on porting EM physics on GPU, mainly for calorimeters

- The two GPU-projects came to a set of findings:
  - LHC scale simulation doable on GPU → **an achievement !**
  - Physics is not a bottleneck → **a good surprise**
  - Geometry is a bottleneck → **a bad surprise**
    - It is 99% of the GPU time
      - It is however ~90% on CPU for same HEP applications…
    - Diversity of geometry volumes makes GPU simulation to go serial too often
      - Ie: a box can not be processed in parallel of a cylinder (virtuality issue)
    - Change of volume representation was tried (VecGeom2), using simple surfaces
      - far better for parallelization of calculations
      - But many surfaces are needed → no gain wrt virtuality
  - CPU–GPU communication is not a bottleneck → **another good surprise**
    - Allows processing tracks in // on the GPU and to send back steps to the CPU
      - to be treated by usual sensitive detector codes

Interleaved Track Steps

- // processing of tracks on GPU leads → // production of steps for these tracks
- When sent back to the CPU, these are **not consecutive steps of a same track**
  - A feature that **can be a radical change for sensitive detectors logic**

- This is the “interleaved track steps” issue

- For calorimeters, this is not a big issue
  - To first order, a calorimeter sum-up steps
- For tracking systems, this is a “nightmare”
  - Reordering needed to re-create proper series of consecutive steps for each track
  - How much must be payed for this in terms of CPU is not established yet

- For space ?
  - The “interleaved track steps” is likely not an issue, to my understanding
  - Dose is a driving simulation observable
    - And is obtained by summing up contributions from steps, regardless of ordering

Some questions…

- Questions in the introduction:
  - What is the speed-up looked for ?
    - Likely depends on the applications
  - Is this at the cost of physics quality or not ?
  - Are there physics aspects which need to be improved ?
- Space applications have few critical parts, while other could be treated more grossly
  - They alternate in the geometry (to my understanding)
    - In contrast with HEP case, where "(Interaction Point) tracking → EM calorimeter → had. calorimeter → muon chambers" where quality need goes grossly from high to low
  - Would an "ATLAS-like" examination be relevant in GRAS applications ?
    - To implement a "quality per region" (with smooth & smart transition from low → high !)
    - And of the stepping action ? ;)
- Space is mainly interested in low/intermediate energies
  - An energy domain in which AdePT and Celeritas would have to be benchmarked (I believe)
  - We know GPU-based medical applications -low E, simple voxel geometries- are super efficient (speed-up 10 – 100 factors (*))
    - From experience with AdePT & Celeritas, we may interpret this is coming from the simple geometries in medical (only small boxes)
    - But space geometries are not simple, so this requires a careful look
    - (*) BTW, any plans for AdePT & Celeritas to compare on the same medical applications ?