

The background is a vibrant, abstract composition of overlapping, organic shapes in shades of light blue, lime green, lavender, and teal. Some shapes contain white dots or patterns, resembling a stylized floral or cellular structure. The overall effect is a dense, textured collage.

# GRAS on GPUs?

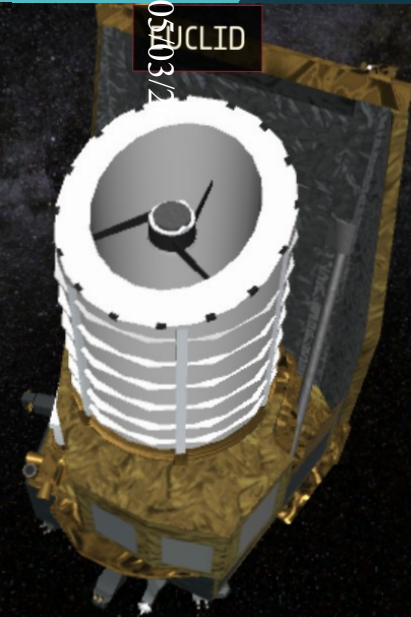
CELERITAS, ADEPT, NEURAL SURROGATES AND ALL THAT

# Space Radiation is Not a Controlled Physics Experiment (But Simulations are necessary)

2

- ▶ Thousands of different CAD models of entire satellites.
- ▶ Extremely important Deep-shielded rare events (Single Event Upsets SEUs)
- ▶ Need for standardized, repeatable analyses across different missions.
  - TID, Total Ionizing Dose,
  - NIEL, Non-Ionizing Energy Loss,
  - LET Linear Energy Transfer
- ▶ Yet we do need to minimize writhing C++ for each new geometry, mission, etc

D. Lenis, Sykies, 05/03/2



## What is GRAS?

- An ESA-validated, macro-driven framework wrapping Geant4.
  - \* Moves geometry parsing, scoring instantiation, and primary generation out of compiled C++ and entirely into runtime macros.
  - \* The result: Not a toolkit (like G4) but a generalized executable capable of arbitrary space simulations.
- More details, next talk by Fan Lei

D. Lenis, Sykies, 05/03/26



# The Specialized Scoring Machinery

- Automate hit collection without custom G4VPrimitiveScorer for each project
- Built-in scorers specialized for space: TID , NIEL and LET
- Dynamically attaching custom GRAS scorers via macro commands to any logical volume

## Dynamic Geometry Injection...

- ...Which means basically that we never use `G4VUserDetectorConstruction`
- strict reliance on `G4GDMLParser` for handling arbitrary spacecraft complexities
- GRAS dynamically maps requested materials to the ingested GDML logical volumes on the fly

## Small Statistics Problem: Adjoint MC

- Spaceship  $\sim 10 \text{ m}^3$ . Chip  $\sim 1 \text{ cm}^3$ . Result: Resource waste, not one hit scored, simulation fail.
- Configure reverse particle that starts from the chip going outward and then follow it back towards the sensitive volume
- Details by the author Laurent Desorgher later today

It's now a part of Geant4!  
examples/extended/biasing/ReverseMC01  
— called Reverse MC for marketing reasons



## Architectural Problems....

- ✓ GRAS works on the CPU:  
Parallelization on multi-core multi-threaded CPUs (à la Geant4's threading/tasking) is now complete (kudos M. Axiotis)
- ✓ Is this enough? NO!
  - More details @Benjamin Jeanty-Ruard 's presentation
  - Plus tomorrow detailed presentations on DNA, Microdosimetry etc

### SO How do we sent GRAS to the GPU?

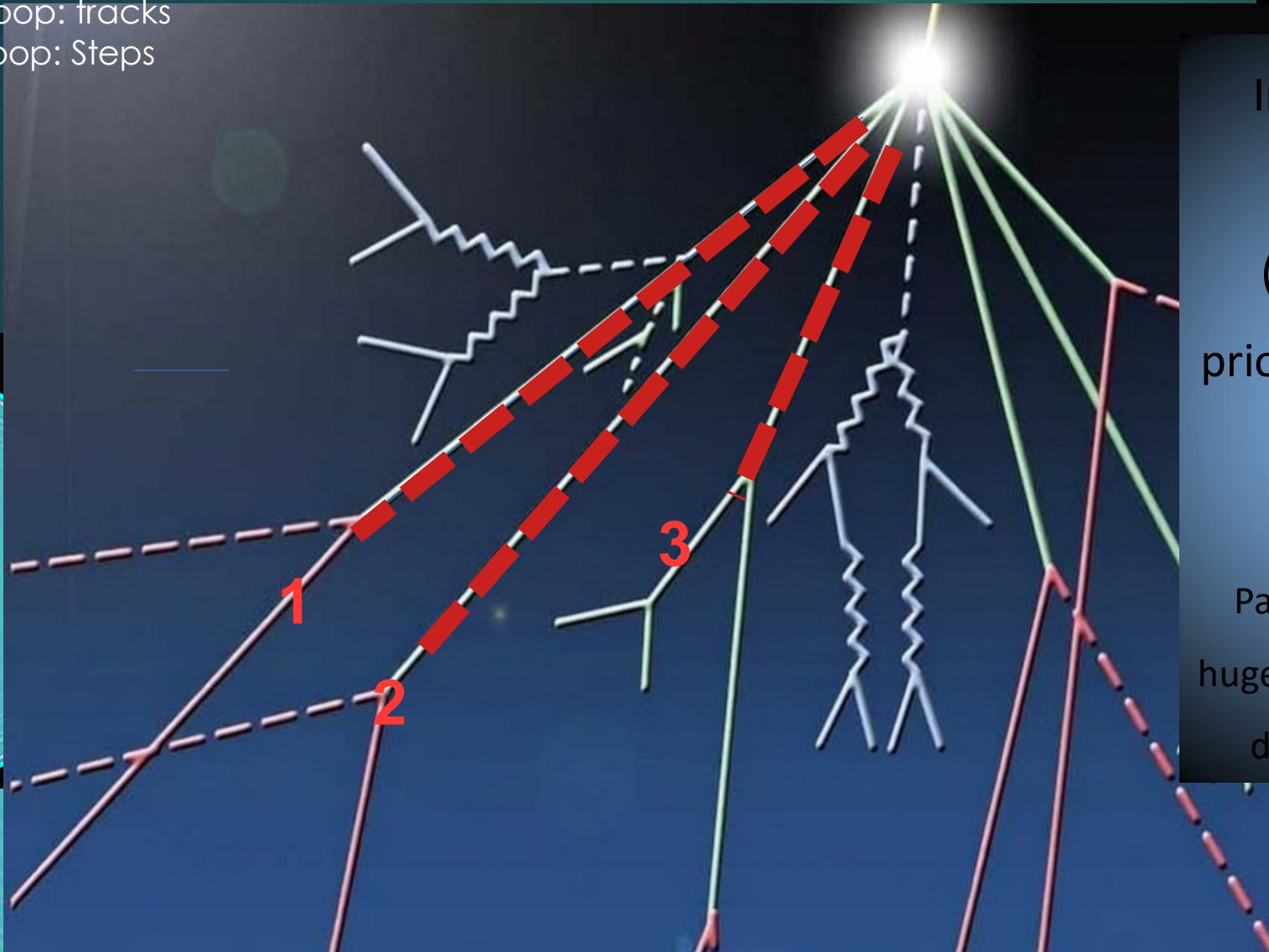
- GPU offloading (Celeritas/Adept) favors data-parallel buffers, not track-by-track C++ virtual calls
- Objective: Moving from CPU-bound per-track scoring to asynchronous GPU-batch processing

# GEANT4

(Explanation for the GRAS crowd)

Outer loop: tracks

Inner Loop: Steps



Inner loop cannot be parallelized  
(we don't know a priori the number of steps!)

Parallelize outer loop – huge calculations, can be done only on CPUs (ie clusters, MPI ...)

# Celeritas

Outer Loop: Current Step

Parallel Inner Loop: loop over tracks



Invert the loops!

We always know the

length of the vector we are

looping over –

straightforward to

parallelize



Caveat: Will this be efficient at low energies, ie when there is a small number of tracks?

## Scoring on GPUs

- GRAS scoring: TID (energy deposition), NIEL (non-ionizing loss), and LET (dE/dx histograms)
- Challenge: Accumulating sparse energy deposits across massive thread counts, on a possibly huge number of sensitive volumes
  - Example: millions of individual voxels in a radiation on human body simulation
- How do you solve this?
- Will it be efficient, esp. Considering we are at relatively low energies?

# Geometry on Accelerators

## VecGeom & ORANGE

- Current GRAS relies heavily on GDML

VecGeom VS ORANGE (Celeritas):

Hierarchical Constructive Solid Geometry CSG tree-traversal (VecGeom) vs.  
Surface-boundary implicit math (ORANGE)

(Mental note to myself: Remember to admit you have no idea what this means)

- Are they mature and ready for production code?
- Is there 100% geometry consistency between the G4-CPU host and the GPU-accelerator?

## Adjoint MC: How to port?

- GRAS Reverse MC is essential for small sensitive volumes in large spacecraft
- Adjoint physics (adjoint-eBrems, adjoint-Compton, adjoint-Whatnot) is not implemented (for the time being!) in Celeritas/Adept kernels
- **Problem:** Adjoint kernels require different weight-management and importance-sampling logic than forward EM kernels
- Laurent will explain

## • NNs as Physics Surrogates

- Integrating NN-based 'Fast Sims' into the GRAS modular analysis framework
- Use Case as in [examples/extended/parameterisations/Par04](#):
  - Using GPU sim to produce a large training set
  - Using trained GNNs to predict dose in shielding voxels without full tracking
  - In GRAS this produces an 'Interface' Problem: How do we create a standard API for NN inference (ONNX/pytorch) within the G4 event loop which is controllable through macros?