Geometry Modeling and Navigation for Simulation on GPU

Ben Morgan

WARWICK
THE UNIVERSITY OF WARWICK

VecGeom in AdePT and Celeritas

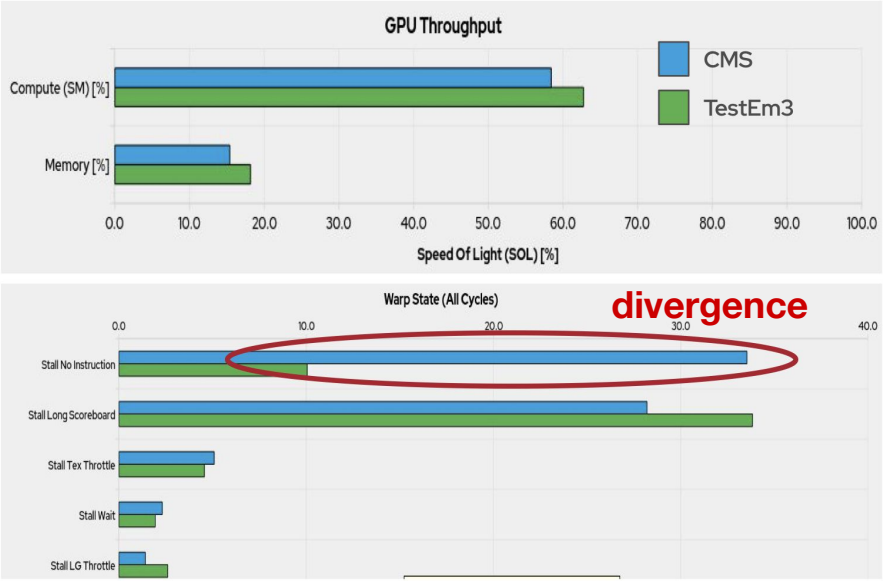
- Library for geometry modelling and navigation developed in GeantV SIMD R&D
  - *https://gitlab.cern.ch/VecGeom/VecGeom*
  - *Constructive Solid Geometry (CSG), shapes can be used in Geant4*
  - *CUDA support also available from original development work*
- AdePT and Celeritas adopted it as pre-existing solution for GPU, also allowing comparison with Geant4 CPU, gradually improving GPU support
  - *Custom optimised navigation state, single-precision support*
  - *Moved from a simple "loop" navigator to an optimized BVH navigator*
  - *Simplified build using modern CMake CUDA support*
  - *Improved GDML support, allowing import/use of almost any geometry*
- **Fine first solution, but detailed use and profiling have revealed limitations**
  - ***NB: I am not a geometry expert, so will only give a very high level overview!***

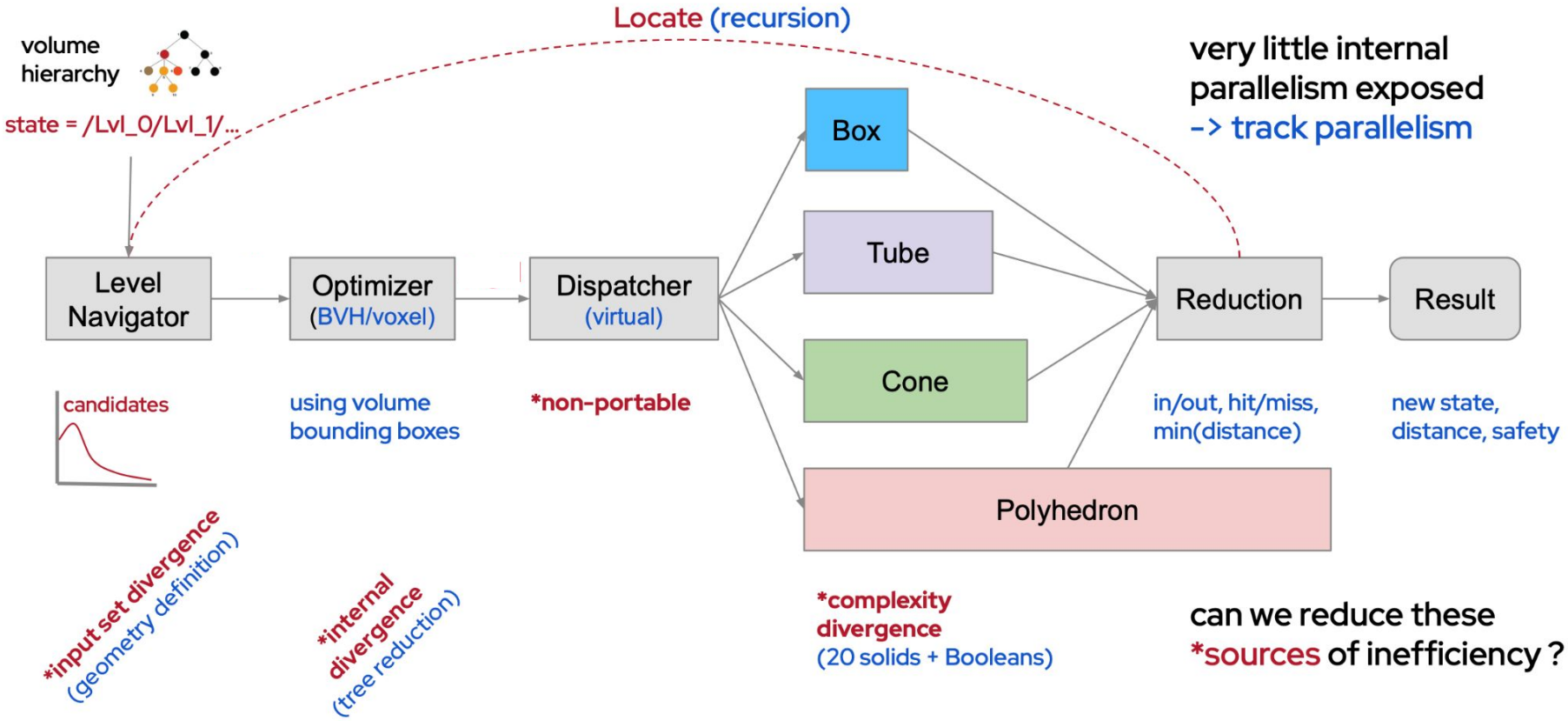
Identified Limitations of VecGeom

- A translation of the C++ CPU object model to GPU via compilation with nvcc to a separate namespace in a second library
  - *Implementation is CUDA specific, use requires careful device linking, use of appropriate CUDA runtime*
  - *Use of virtual functions impedes portability to non-NVIDIA systems*
- Performance bottleneck in stepping workflow on GPU from
  - *Virtual function calls*
  - ***Code complexity** results in kernels being register hungry, limiting achievable **warp concurrency***
  - ***Branch-y algorithms/code** leading to many **divergent paths in kernels**, limiting low level **thread concurrency***

Divergence increases with geometric complexity

- Moving from simple to complex geometry : longer stalls within warps for the same SM compute
- *Running 10^6 10GeV electrons in TestEM3 vs CMS problems with AdePT shows CMS 2.6x slower*
- One of the potential show stoppers for GPU vs. CPU simulation efficiency in complex setups





Navigation workflow per track query in VecGeom

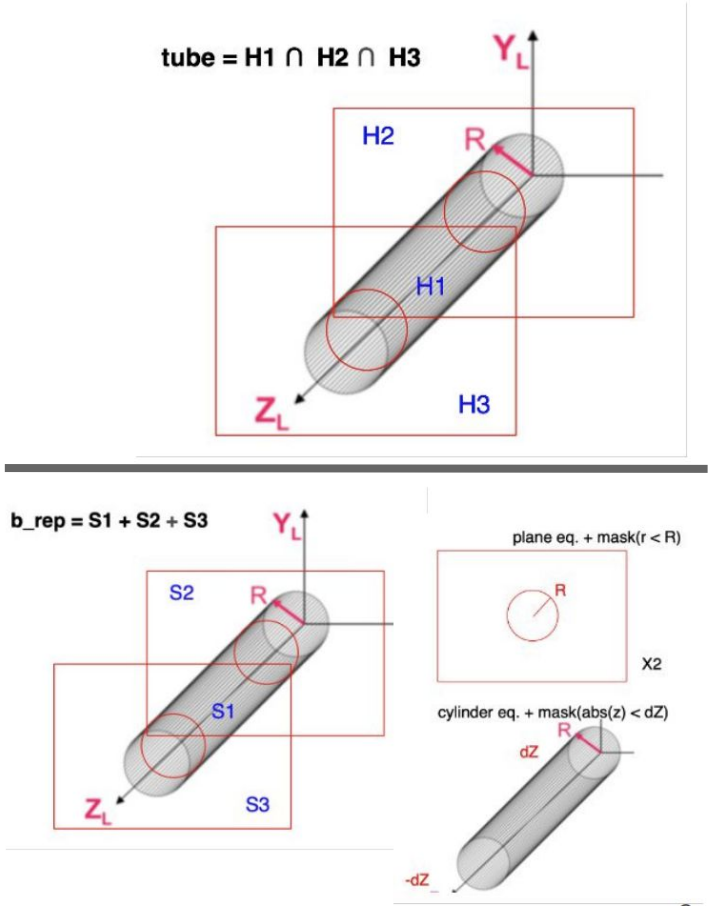
Focus here on divergence resulting from CSG solids

Addressing the problem: Surface-based Models

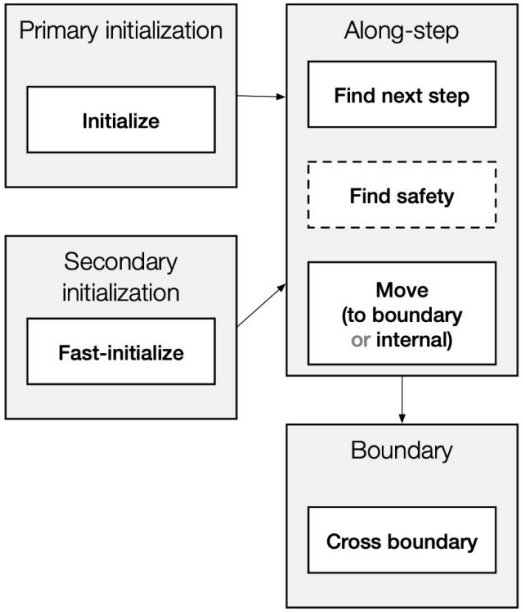
- VecGeom Evolution Meeting in January discussed various options, with surface-based models showing the most promise
- Valuable experience and ongoing development at ORNL from Scale/Shift Nuclear Systems codes
  - *New ORANGE surface-based library for GPU*
- Motivated explorations at CERN and collaboration with Celeritas on requirements for surface models for the HEP use case
  - *Potential for implementation in VecGeom, and prototyping*
- As with AdePT/Celeritas, different lines of development enabling broad exploration, but ultimately aim for common solutions

Surface model benefits

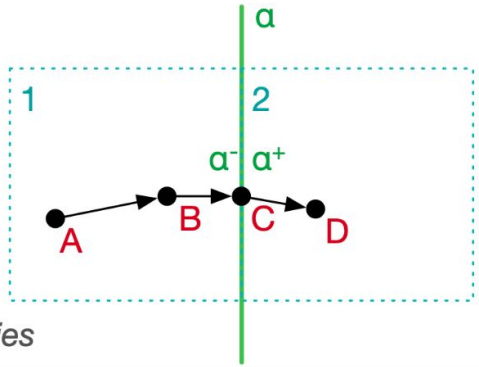
- Factors the navigation problem at lower level
  - *More simple and uniform code, even if code path is sometimes longer*
  - ***Less branching for primitive surfaces than for primitive solids***
  - *Allow reducing the number and size of divergent critical sections*
- **Unbounded CSG model** (example: ORANGE)
  - *Volume = intersection of half-spaces*
  - *Supporting Boolean operations*
- **Bounded surface models**
  - *Similar to tessellations but supporting more complex surfaces (second order)*
  - *Face = Half-space + frame*
  - *Example: [detray model in ACTS]*
- [Both have advantages/disadvantages]



Celeritas geometry interface



	Position	Volume	Surface+Sense
(input)	A	–	
Initialize	A	1	–
Find step	A	1	–
Move internal	B	1	–
Move to bdy	C	1	α inside
Cross bdy	C	2	α outside
Move internal	D	2	–



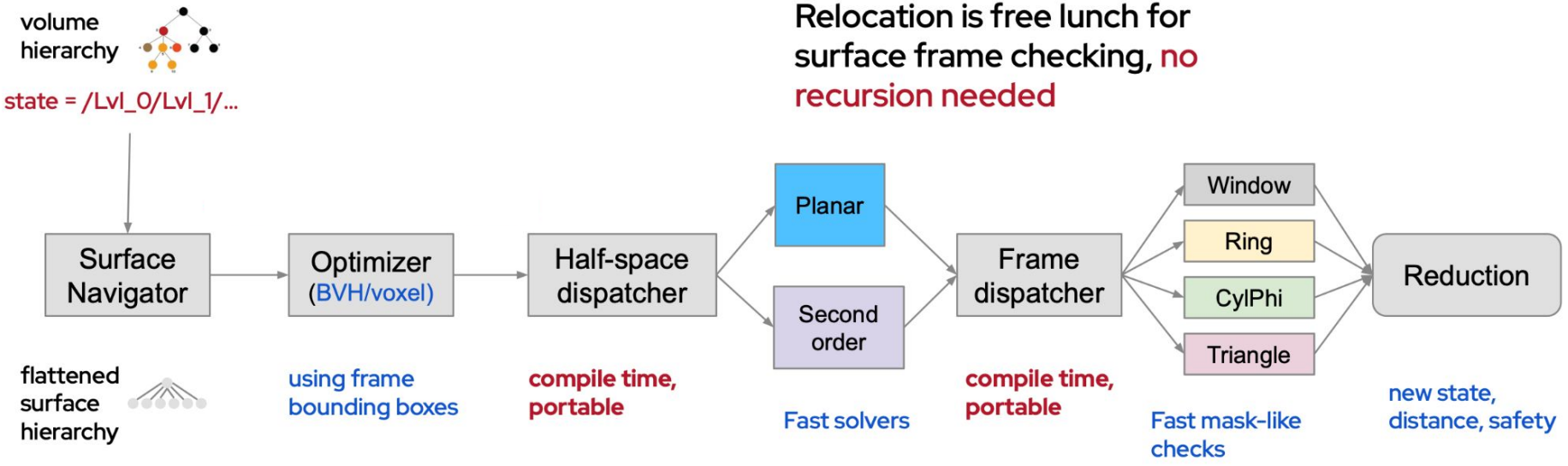
Key feature: Error mitigation via Surface-Volume connection (avoiding kernel slowdown)

OAK RIDGE National Laboratory

**exact handling of direction changes on boundaries*

Tracking workflow in Celeritas/ORANGE

See Seth Johnson's presentation for full details

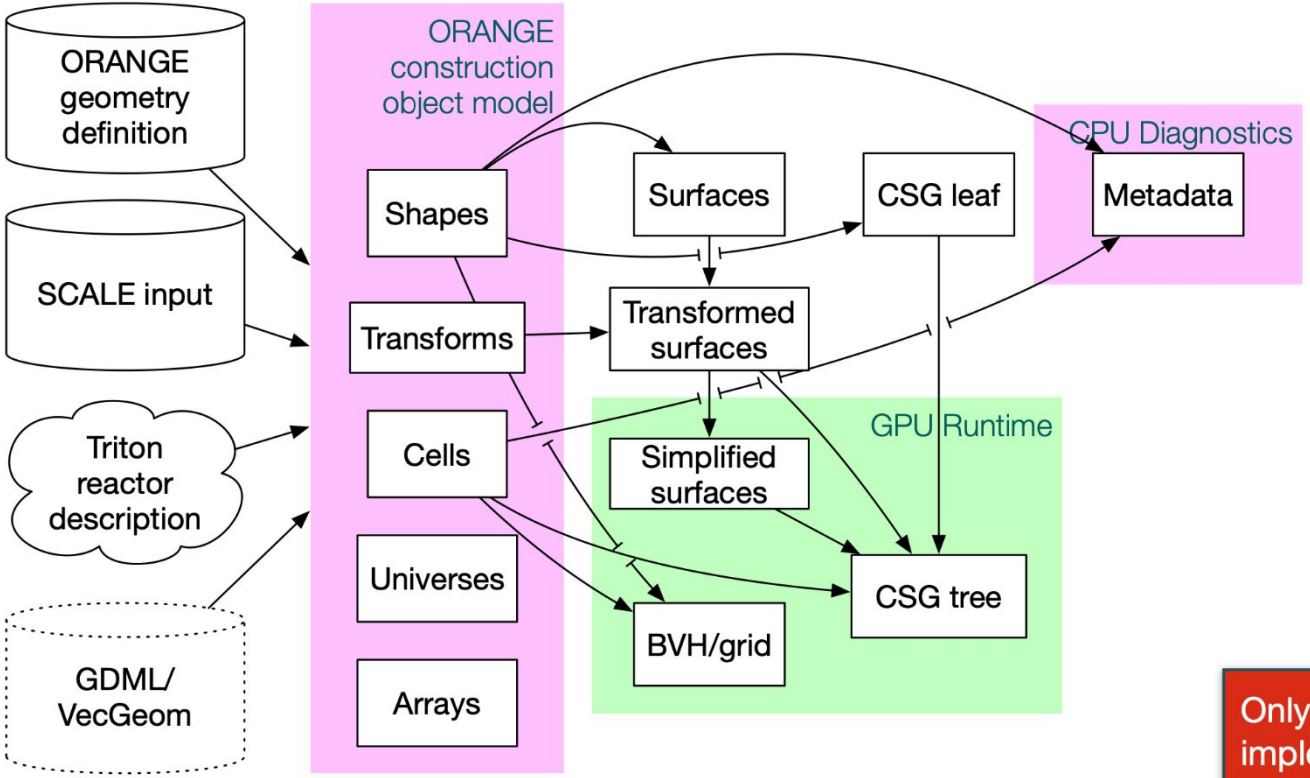


better balanced input per particle due to flattening and mixing surfaces from different volumes

faster divergent sections with fewer branches (2 half-space types and 6-8 masks) (?)

Navigation workflow per track query for VecGeom Bounded Surface Model

See Andrei Gheta's presentation for full details



Only **partially** implemented in Celeritas ORANGE

OAK RIDGE National Laboratory

Surface Model Construction from GDML et al

Shown for ORANGE, but key for VecGeom as well for comparison with standard CSG implementations!

Current Status

- ORANGE under active development, testing in Celeritas
  - *Potential for additional optimization using inter-thread co-operation on device (i.e. more than one thread per track)*
- VecGeom has implemented core features for a bounded surface model
  - *Adding more solids, GPU portability*
  - *Gradual integration, testing in AdePT*
- For full details and additional technical information, see material from
  - *Geometry session at HSF Detector Simulation on GPU Meeting*
  - *Geometry WG Reports at Geant4 Collaboration Workshop*

Upcoming UK Contribution through ExaTEPP

- Awarded 0.5FTE x 2yrs RSE time at each of Sheffield and Warwick to contribute to this area through ExaTEPP bid to UKRI ExCALIBUR Phase-2
  - *Close coordination with SWIFT-HEP and AdePT/Celeritas teams*
  - *Kickstart meeting planned for early December*
- Early days of course, but should maybe think about potential cross-cutting topics with other work packages in SWIFT-HEP?
  - *Reconstruction? VecGeom discussion with ACTS for example, which also included tracking with magnetic fields.*
  - *Opticks/OptiX work in Simulation? Use of ray tracing/shaders for charged particle tracking is a possible area for R&D*
  - *Any others? E.g. portability APIs like Alpaka, oneAPI, testing on Intel/AMD?*

Summary

- VecGeom implementation of CSG solids/navigation seen to be key bottleneck for AdePT/Celeritas simulation on GPU
- Investigation of Bounded/Unbounded Surface Model geometry underway to address limitations
- UK contribution to these efforts as part of ExaTEPP grant from UKRI ExCALIBUR program