

Progress in adapting the QSS Stepper to the current version of Geant4

Testing and benchmark results

Rodrigo Castro, Lucio Santi, Alejandro Mignanelli

University of Buenos Aires and
ICC-CONICET, Argentina.
rcastro@dc.uba.ar

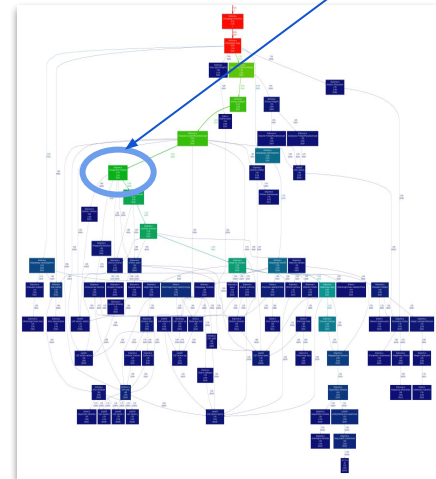
Agenda

1. Main Motivation
2. Quick refresher on QSS (more in the backup slides)
3. Test of the QSS Stepper in a relevant subset of typical examples
4. Verify the correct functioning of the examples
5. Benchmark the QSS Stepper against the default Geant4 Stepper
6. Error assessment for different QSS accuracies

Motivation

- **Simulation domain:** Tracking of subatomic particles
 - Undergoing physics processes within **complex detector geometries**
- **Key issue:** Handling of boundary crossings across discrete volumes
 - Can require **CPU-intensive, ad-hoc iterative algorithms**
 - **Can we do better?**
- **Approach:** Family of hybrid (continuous/discrete-event) integrators
 - **Quantized State System (QSS) numerical methods**
 - **Attractive performance features** for HEP applications

G4PropagatorInField::ComputeStep



callgrind performance analysis

ComputeStep: ~47%

case **without**
iterative location
of intersection
points
(less dense geometry)

libG4geometry.so
G4PropagatorInField::ComputeStep
47.67%
(1.12%)
500000x

libG4geometry.so
G4ChordFinder::AdvanceChordLimited
39.14%
(0.30%)
500000x

2.4
5000

ComputeStep: ~81%

case **with**
iterative location
of intersection
points
(more dense geometry)

libG4geometry.so
G4PropagatorInField::ComputeStep
81.75%
(0.63%)
50000x

75.59%
150000x

libG4geometry.so
G4ChordFinder::AdvanceChordLimited
75.59%
(0.30%)
150000x

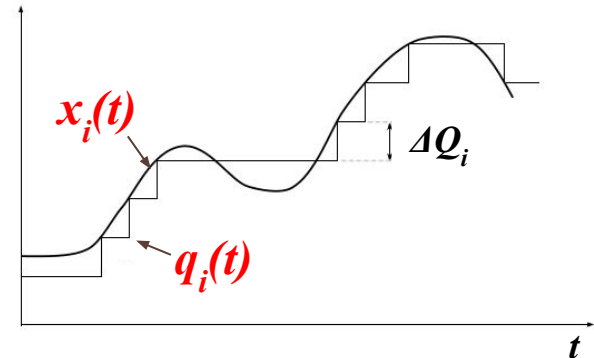
2.19%
150000x

Synthetic benchmark
Perfect 2D particle oscillator
radius: 45 mm
Geometry: Parallel planes
G4 params:
epsilon = 1E-7
deltaChord = 0.25 mm
stepMax = 20 mm
trackLength = 1000 m

Quantized State System (QSS) numerical methods

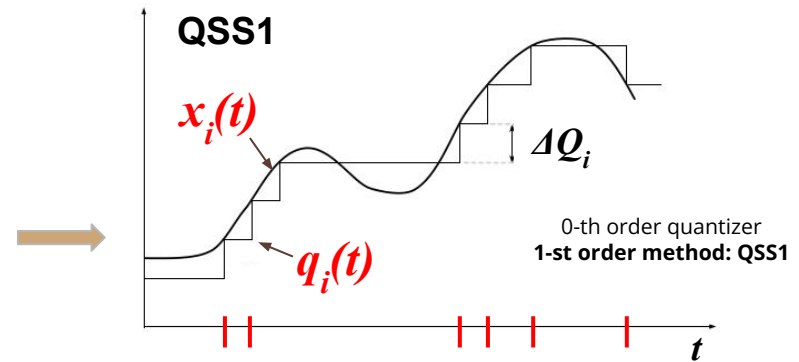
- Based on **state variables quantization**
- QSS methods **discretize the system state variables** as opposed to classical solvers which **discretize the time** (e.g. family of Runge-Kutta methods)
- **Continuous state variables** approximated by **Quantized state variables**
 - A **quantization function** is in charge of controlling **error and accuracy** throughout the simulation

$$\underbrace{\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))}_{\text{ODE system}} \Rightarrow \underbrace{\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t))}_{\text{ODE quantized system}}$$

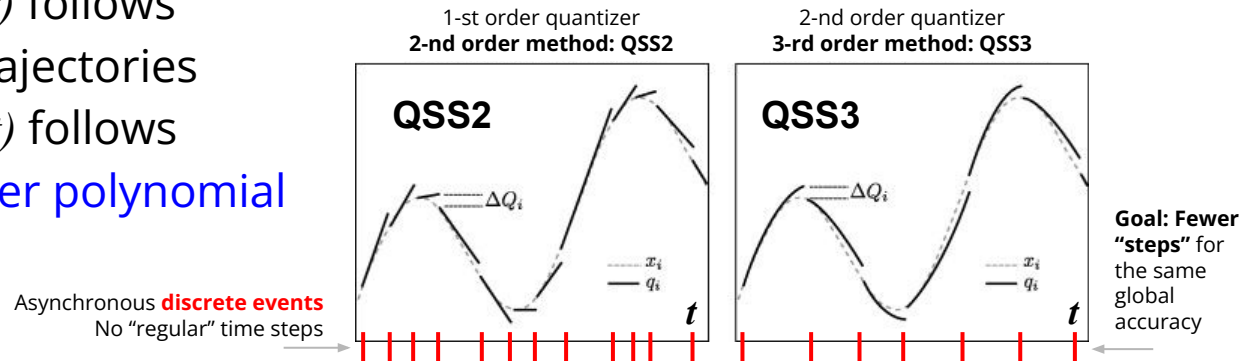


Higher order QSS

$$q_i(t) = \begin{cases} x_i(t) & \text{if } |q_i(t^-) - x_i(t)| \geq \Delta Q_i \\ q_i(t^-) & \text{otherwise} \end{cases}$$

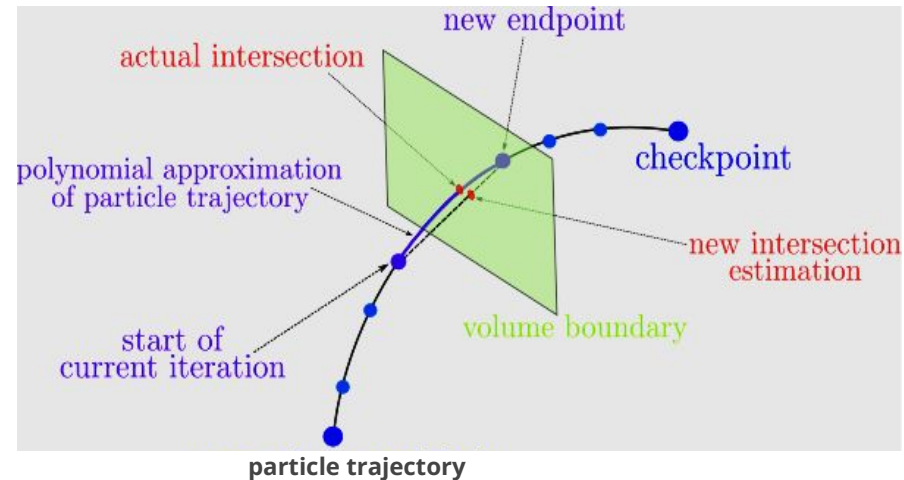
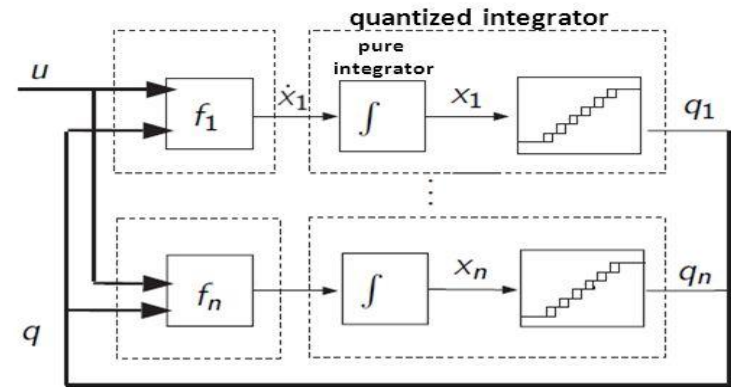


- ΔQ_i is the **quantum**
 - **Maximum deviation allowed** between x_i and q_i (error control)
 - Derived from the **accuracy** demanded by the user
- **Higher order QSS methods (QSSn)** follow a similar principle
 - In a **QSS1** method, $q(t)$ follows **piecewise constant** trajectories
 - In a **QSSn** method, $q(t)$ follows **piecewise (n-1)-th order polynomial** trajectories



Main QSS features for HEP problems

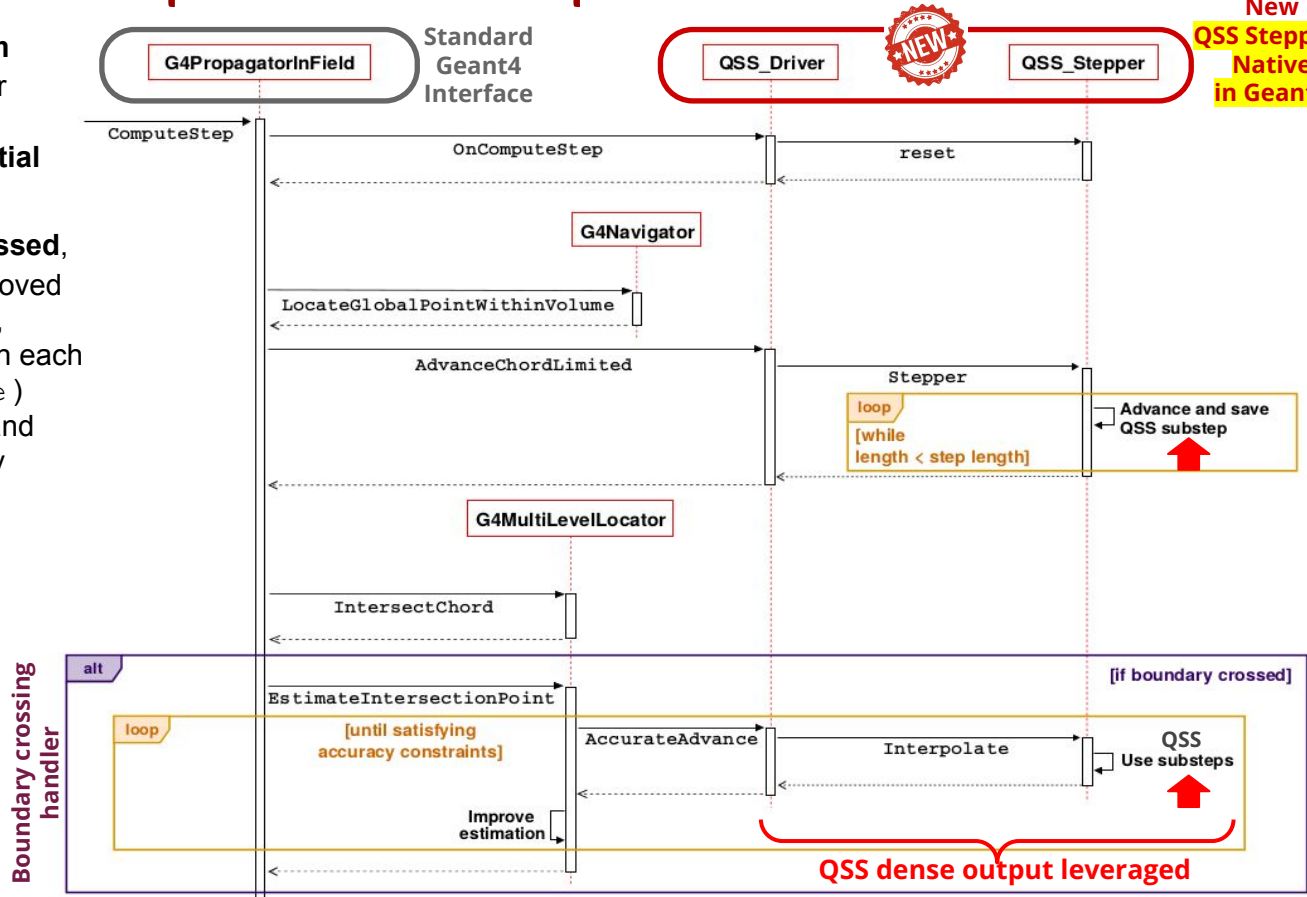
- **Inherent asynchronicity**
 - Decoupled, independent computation of changes in each state variable (no “global clock”)
- **Dense trajectory output**
 - Supported by **piecewise polynomial** approximations of trajectories
- **Lightweight discontinuity handling**
 - **Boundary crossings detected by finding roots of simple polynomials**



QSS-based step computation sequence in Geant4

- The **intersection-finding algorithm** starts with a quick test using a linear segment joining the step endpoints (`IntersectChord`) yielding an **initial estimation** of the intersection point
- In case a **volume boundary is crossed**, this estimation is progressively improved (`EstimateIntersectionPoint`, that queries the Integration Driver on each of its iterations (`AccurateAdvance`) in order to advance a given length and then test which side of the boundary the particle lies in
- The **QSS Driver**, by means of the Interpolation Driver's custom behavior, issues an `Interpolate` call to the **QSS Stepper**
- `Interpolate` is handled very **efficiently** leveraging the **polynomial QSS Substeps** previously computed and saved

New
QSS Stepping
Native
in Geant4



Summary of results: QSS vs. DOPRI

Example	Method	QSS accuracy parameters		% of Intersections per G4 Step	QSS Substeps per G4 Step	User Time (seg)	System Time (seg)	Real Time (seg)	Average Time per G4 Step (seg)	Speedup (QSS vs. DOPRI) Real Time
		dQrel	dQmin							
B2a	DOPRI	N/A	N/A	3.79%	N/A	2.052	0.175	2.614	1.3E-04	N/A
B2a	QSS	1.0E-02	1.0E-03	3.75%	10.191	2.067	0.176	2.654	1.3E-04	-1.53%
B2b	DOPRI	N/A	N/A	3.73%	N/A	2.081	0.178	2.651	1.3E-04	N/A
B2b	QSS	1.0E-02	1.0E-03	3.77%	10.209	2.107	0.178	2.680	1.3E-04	-1.09%
B4c	DOPRI	N/A	N/A	4.31%	N/A	1.623	0.180	2.202	1.1E-03	N/A
B4c	QSS	1.0E-02	1.0E-03	4.02%	2.517	1.603	0.182	2.170	2.1E-03	1.43%
B4d	DOPRI	N/A	N/A	4.31%	N/A	1.637	0.183	2.217	1.1E-03	N/A
B4d	QSS	1.0E-03	1.0E-04	4.19%	5.026	1.605	0.178	2.164	1.1E-03	2.39%
B5 SingleBeam	DOPRI	N/A	N/A	2.78%	N/A	3.442	0.257	4.004	1.1E-01	N/A
B5 SingleBeam	QSS	1.0E-03	1.0E-04	2.78%	1,494.940	3.259	0.245	3.841	1.1E-01	4.06%
Extended Field 01	DOPRI	N/A	N/A	6.51%	N/A	1.020	0.096	1.347	7.4E-04	N/A
Extended Field 01	QSS	1.0E-02	1.0E-03	5.99%	37.787	1.014	0.096	1.333	6.7E-04	1.03%
Extended Field 02	DOPRI	N/A	N/A	19.17%	N/A	1.270	0.124	1.612	9.7E-04	N/A
Extended Field 02	QSS	1.0E-02	1.0E-03	19.17%	3.056	1.265	0.128	1.610	9.7E-04	0.07%
Extended Field 03	DOPRI	N/A	N/A	14.76%	N/A	1.375	0.186	1.783	1.9E-04	N/A
Extended Field 03	QSS	1.0E-02	1.0E-03	9.99%	62.279	2.608	0.451	3.281	8.2E-05	-83.95%
Extended Field 06	DOPRI	N/A	N/A	0.08%	N/A	0.030	0.010	0.037	3.1E-05	N/A
Extended Field 06	QSS	1.0E-02	1.0E-03	0.08%	1.190	0.032	0.012	0.040	3.3E-05	-7.27%

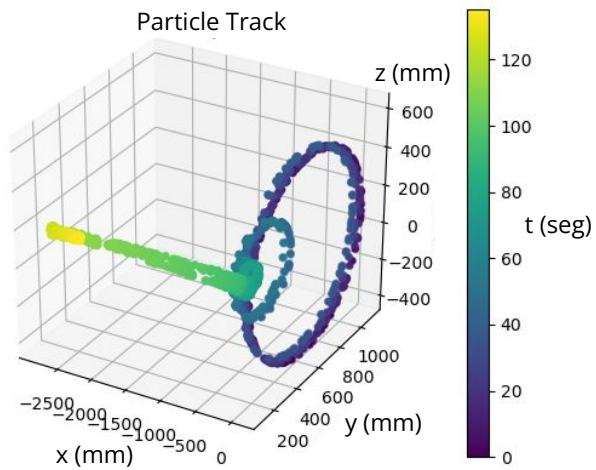
(*)

Note: In all cases (*) where QSS is able to outperform DOPRI, only the best combination of QSS accuracy parameters is shown (relative and minimum Quantum delta sizes, dQrel and dQmin). Other combinations may exist that could even perform worse than DOPRI.

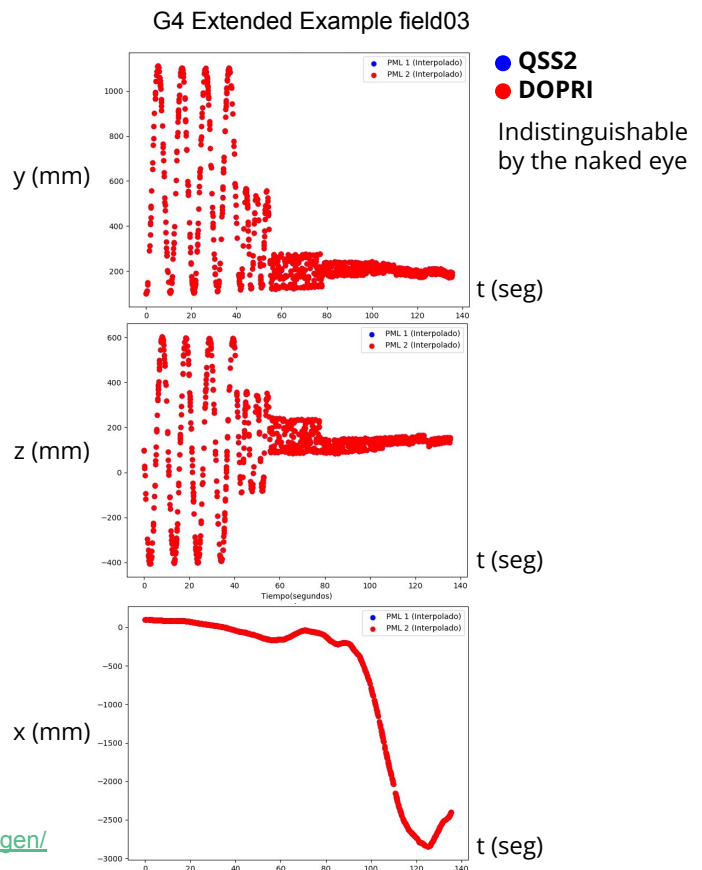
New: Logging of time series for error assessment

- Calculation of the **Mean Square Error (MSE)** for $x(t), y(t), z(t)$ and the Track Length $L(t)$
- Thorough **systematic comparison** of deviation between methods for **different accuracies**
- **Interpolation** of asynchronous time series
- E.g.: QSS2 vs DOPRI

$dQ_{Rel}=1e-5$, $dQ_{Min}=1e-6$
X_MSE = 1.64
Y_MSE = 0.00072
Z_MSE = 0.0014
L_MSE = 0.0



https://geant4-userdoc.web.cern.ch/Doxygen/examples_doc/html/Examplefield03.html



New: Logging of time series for error assessment

- Calculation of the **Mean Square Error (MSE)** for $x(t), y(t), z(t)$ and the Track Length $L(t)$
- Thorough **systematic comparison** of deviation between methods for **different accuracies**
- **Interpolation** of asynchronous time series
- E.g.: QSS2 vs DOPRI

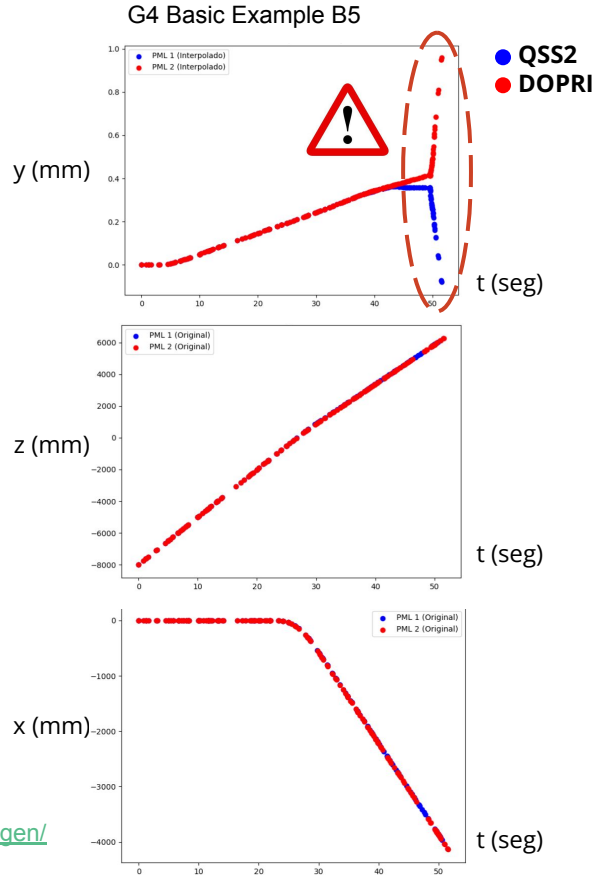
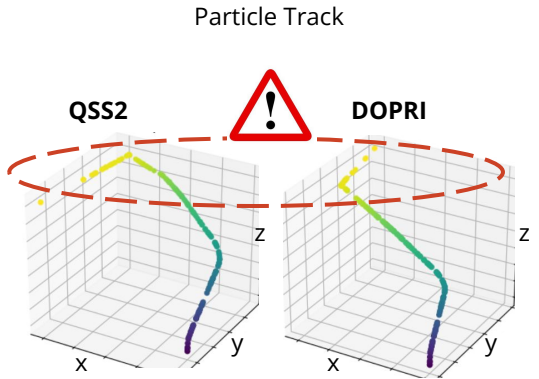
$dQ_{Rel}=1e-1, dQ_{Min}=1e-2$

Z_MSE = 33.35

X_MSE = 0.026

Y_MSE = 182.28

L_MSE = 5.9e-13



https://geant4-userdoc.web.cern.ch/Doxygen/examples_doc/html/ExampleB5.html

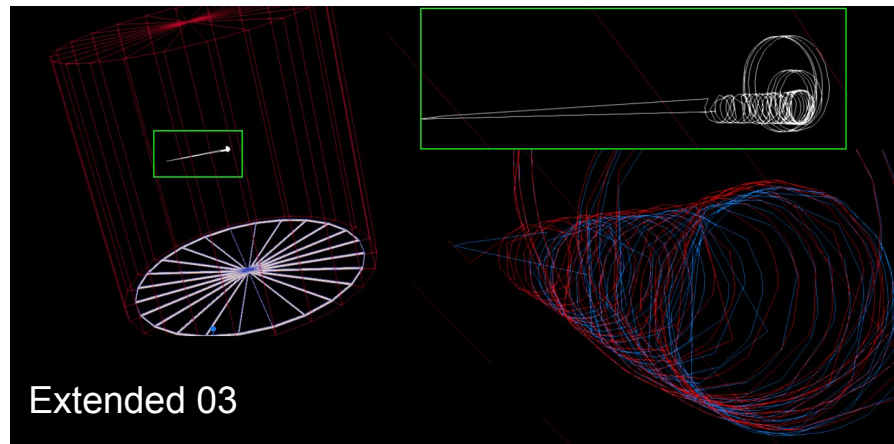
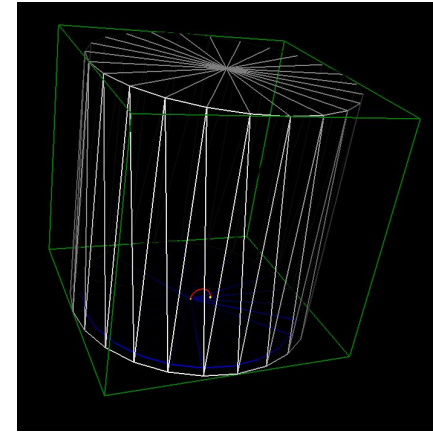
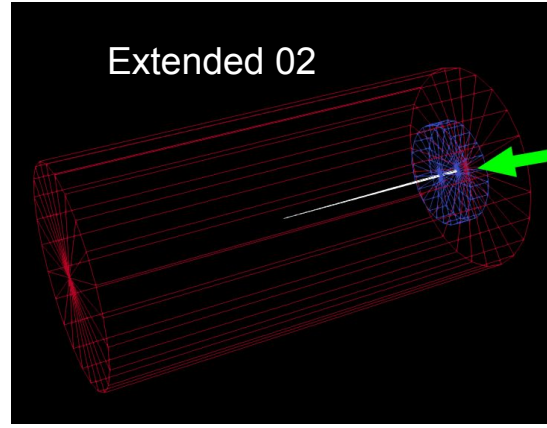
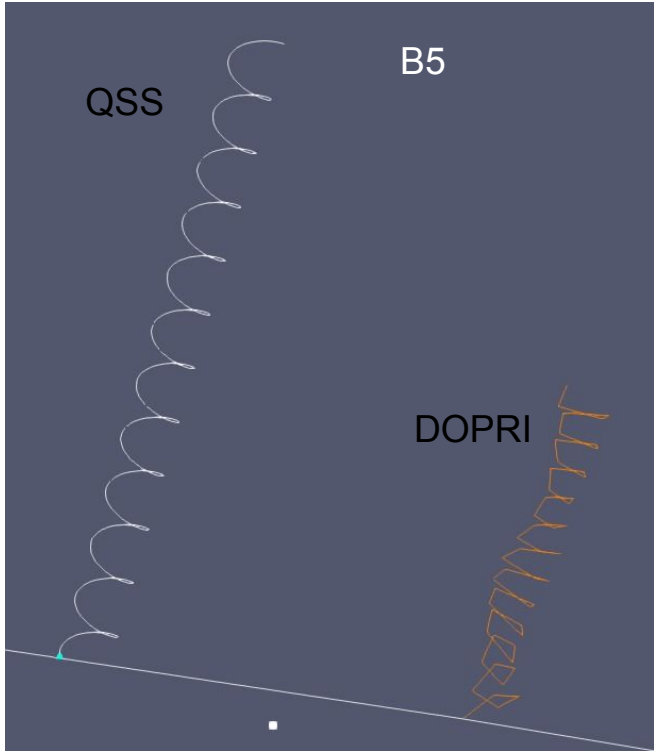
Results highlights

- 11 examples tested and verified successfully:
 - **Basic** (B2a, B2b, B4c, B4d, B5), **Extended** (with magnetic field: 01, 02, 03, 06), **Advanced** (ams_ECAL)
 - **FullSimLight**, a lightweight standalone Geant4 simulation tool that supports the full ATLAS geometry and the ATLAS magnetic field map
- Benchmarks made against G4 (ver. 11.0.0-ref-02) default stepper (DOPRI with Interpolation Driver)
- In 5 cases *there exist* **QSS accuracy parameters** that can outperform DOPRI
 - **However**, the ratio of geometry intersections per G4 step remains below 19% in all tested examples (typically around 5%) => these are **not** “QSS-friendly” scenarios (not too many intersections per step)
- Particle trajectories were compared visually using Paraview and VTK output files
- Benchmarking software: we continue developing a **toolset for repeatable benchmarking** that can be parameterized to produce systematic performance comparisons across G4 Steppers

Benchmark computing platform

- All experimentations carried out in CERN’s OpenLab (controlled environment)
- Hardware specs: Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz (64 CPUs) 64 GB RAM

Example visualizations



Thank you

Backup Slides

QSS solvers for HEP applications

- Started as a collaboration with the Detector Simulation Group in Fermilab
 - w/Daniel Elvira & Team, Software for Physics Applications Dept., Scientific Computing Div.
- Since 2015 - Research on Efficient simulation of particle systems (HEP and other apps.)
- 2 completed Master's Thesis
- 1 PhD Thesis
- 5 peer-reviewed publications
- Successful case of a HEP/Computer Science interdisciplinary collaboration
 - Results relevant and innovative both for the Physics and the Computer communities

Latest manuscript:

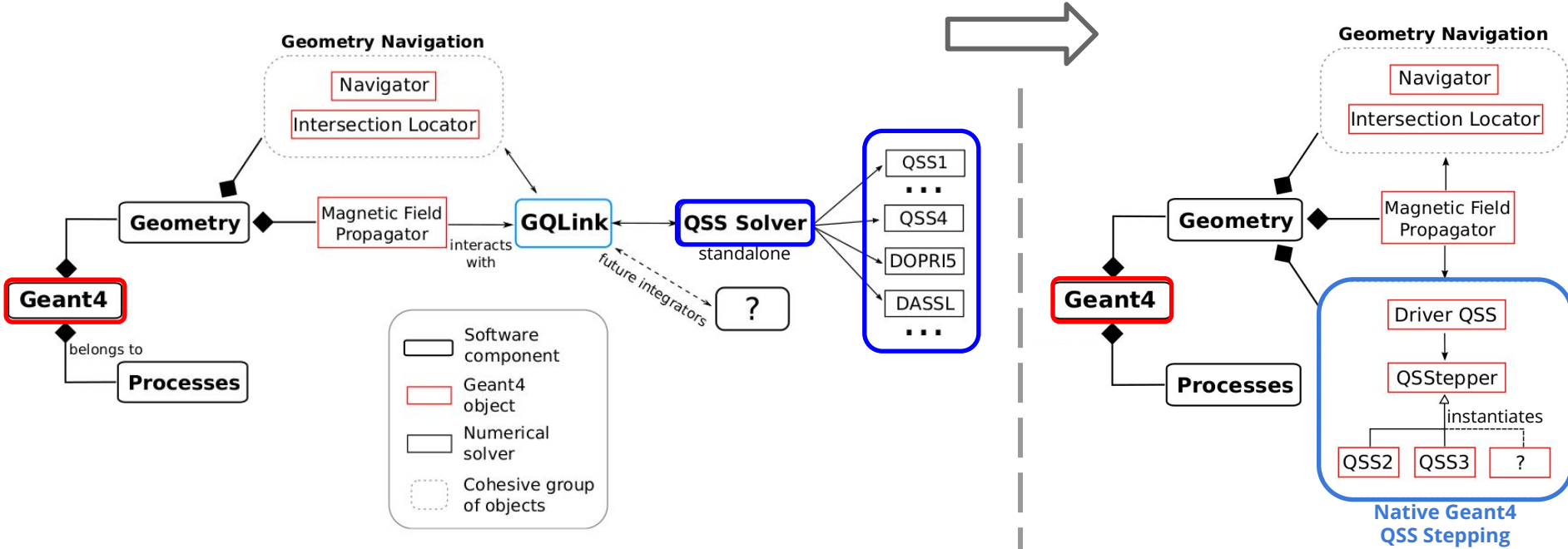
Efficient discrete-event based particle tracking simulation for high energy physics

L. Santi, L. Rossi, and R. Castro

<https://doi.org/10.1016/j.cpc.2020.107619>



Integration with G4: High Level architectures



Strategy 2: "Embedded QSS"

- **QSSStepper** for **Geant4**
- New native G4 Steppers

CMS Benchmark results

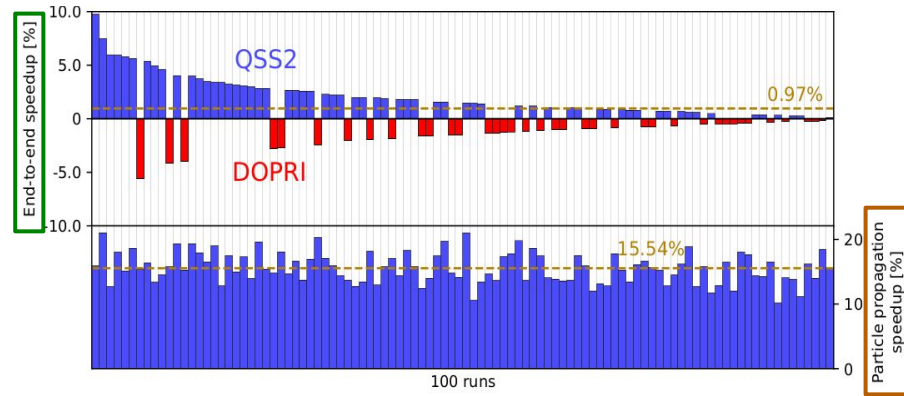
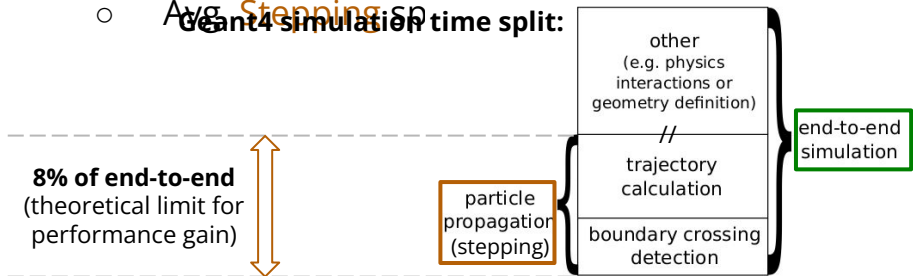
- Experimental results
 - CMS full Run 1 geometry
 - Single π^- particles, Physics list FTFP_BERT
 - 100 independent runs, 2000 particle gun events

- **QSS2 vs. DOPRI**

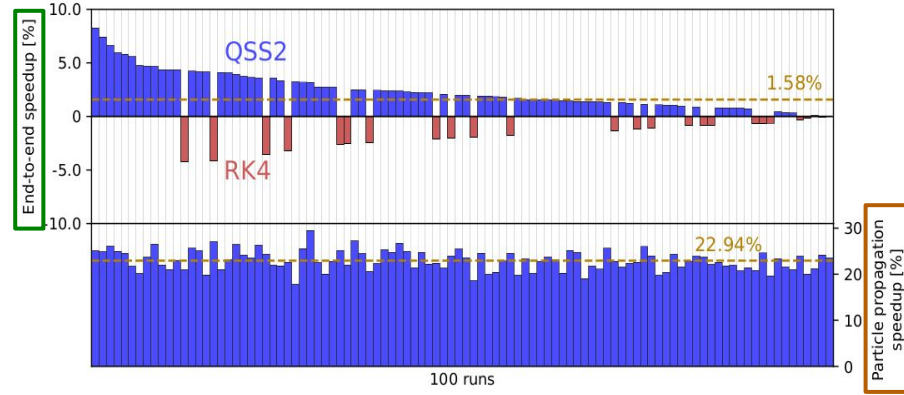
- 62 runs favorable for QSS; 38 for Geant4
- Avg. **End to End** speedup: **~1%** (max. ~10%)
- Avg. **Stepping** speedup: **~15%** (max. ~20%)

- **QSS2 vs. RK4**

- 77 runs favorable for QSS; 23 for Geant4
- Avg. **End to End** speedup: **~1.5%** (max. ~8%)
- Avg. **Stepping** speedup: **~23%**



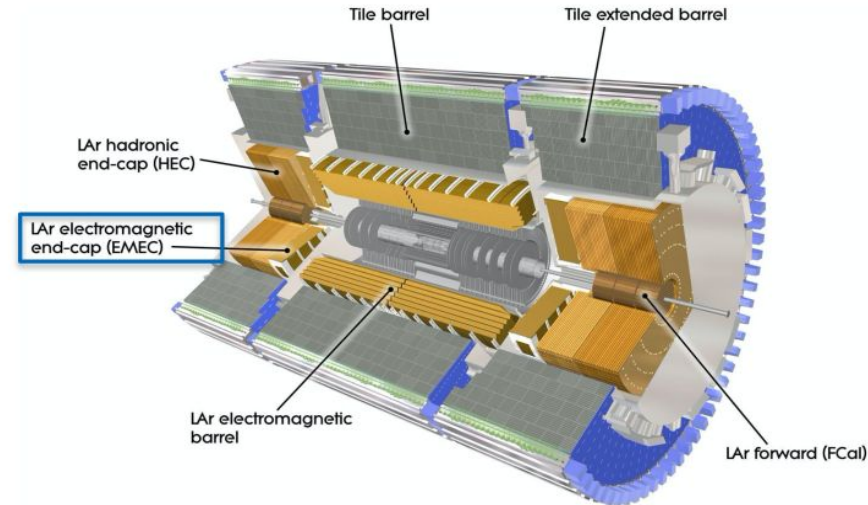
(a) QSS2 vs. DOPRI
(comparison with "DOPRI with Interpolation" is work in progress)



(b) QSS2 vs. RK4

ATLAS as a next reference model

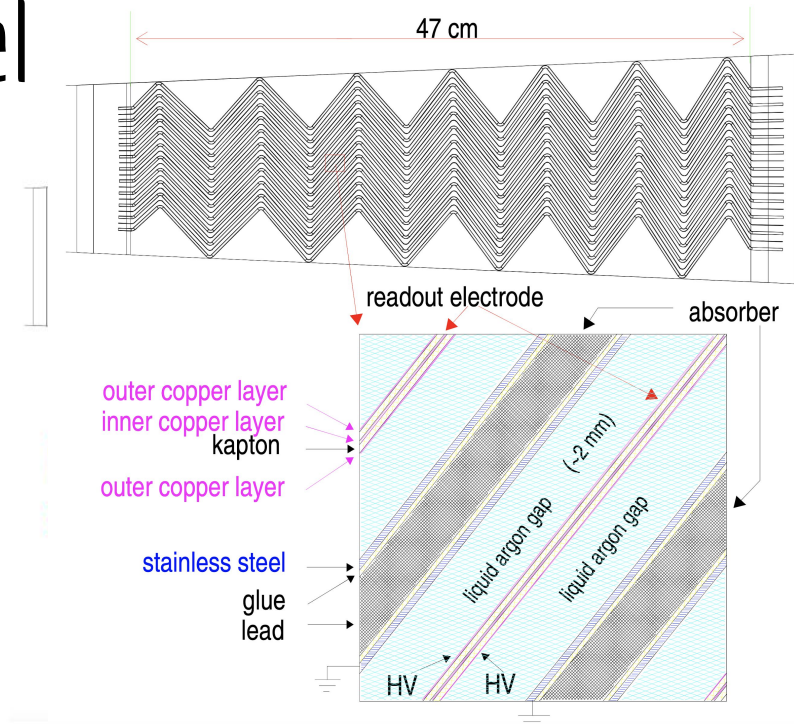
- **ATLAS experiment** as next reference model for QSS methods
- Explore simulation scenarios that use QSS methods integrated into the recently developed **FullSimLight** simulation prototype [1][2], a lightweight standalone Geant4 simulation tool that supports the full ATLAS geometry and the ATLAS magnetic field map
- Of particular interest is the simulation of the **EMEC** detector
- The **EMEC (ElectroMagnetic End-Cap) [3]** is a lead-liquid argon sampling calorimeter with interleaved accordion-shaped absorbers and electrodes.
 - The accordion fold amplitude varies with the radius from the wheel center leading to a varying gap size
 - It has been implemented as a **custom solid (GeoSpecialShape)** in GeoModel and in Geant4



Picture of an electromagnetic end-cap module during stacking, showing the accordion structure of the ATLAS EM calorimeters.

ATLAS as a next reference model

- The EM calorimeters comprise accordion-shaped copper-kapton electrodes positioned between lead absorber plates and kept in position by honeycomb spacers while the system is immersed in LAr [1]
- The **EMEC special shape** is a well-known hotspot in the ATLAS simulation:
 - takes a significant amount of the total full Geant4 simulation CPU time: **~11.5%**
- The research hypothesis is that the **EMEC's densely layered geometry** is a very **suitable** application case where the efficient discontinuity handling property of QSS can be effectively leveraged.



Module / Class / Source Function / Call Stack	CPU Time ▾		Instructions Retired		Microarchitecture Usage	
					Microarchitecture Usage	CPI Rate
▶ libG4geometry.so	27.0%		26.9%		40.2%	0.665
▶ libG4processes.so	24.2%		21.4%		32.2%	0.693
▼ libGeoSpecialShapes.so	11.5%		14.2%		53.9%	0.571
▶ LArWheelCalculator_Impl::DistanceCalculatorSaggingOff	6.7%		8.9%		50.1%	0.563
▶ LArWheelCalculator	2.4%		3.3%		66.7%	0.570
▶ LArWheelCalculator_Impl::WheelFanCalculator<LArWheelCalculator_Impl::SaggingOff_t>	2.1%		1.8%		45.5%	0.610

Plans for QSS integration into the Geant4 release

- March/April
 - Goal: Integrate the already developed QSS capabilities (last integration: v10.5)
 - Incorporate members of the UBA Team (Simulation Lab, CS Dept.) to the Geometry and Transport WG
 - Initial tests, code housekeeping, documentation for final users.
- June/July/August
 - Goal: Include QSStepper into the Geant4 Quality Assurance regular procedures (collab. with Soon Yung Jun, Fermilab)
 - Reproduce benchmarks already run by the UBA Team in Argentina
 - Start adding more applications (based on the success of previous benchmarks)
- September/October
 - Goal: QSStepper in the next *development version*
 - Assess performance, identify bottlenecks and opportunities for improvements
 - Design/start new projects for extensions/refinements/enhancements
 - Typically advanced undergrad students, Master's Thesis, 6mo-1yr. Potentially a new PhD student
- November/December
 - Goal: QSStepper in the next *release version*
 - Design/start maintenance procedures/plan
 - More goals TBD according to the progresses made so far

Summary

- Performance gains in Geant4 achievable by QSS methods are a fact
 - But also largely application-dependent
- We are entering a new phase of more comprehensive benchmark-based performance characterizations
 - CMS continues to serve as a reference model (add more test cases)
 - ATLAS to be soon added to the list, bringing in new particular challenges
- HEP as a provider of challenging applications for continued Simulation-specific R&D