Quantum Simulation Modules for HEP Event Generation

Recent studies indicate that quantum simulators, both analog and digital, can probe non-perturbative phenomena and real-time dynamics that are difficult to access with classical lattice techniques, with clear relevance for high-energy physics. This project explores how such advances can be translated into practical components for HEP event generation and detector simulation workflows. The work will focus on identifying a limited, demonstrable use case motivated by LHC physics, developing appropriate data representations, and designing robust software interfaces that allow targeted quantum simulation modules to be integrated into existing HEP generators.

The primary software deliverable will be a reproducible Python/Qiskit based prototype module for Schwinger model simulation, including benchmark observables, current-current correlation routines, validation plots, and example config files. The code will be organized clearly and will serve as a demonstration for future integration into HEP event generation or detector simulation workflows.

The proposed start date is June 8, 2026, with a planned project duration of 12 weeks.

Timeline:

Weeks	Planned Work	Deliverables
Weeks 1–2	Learn the basics of the Schwinger model and quantum simulation. Install the required software (Python, Qiskit) and run example notebooks.	Working software setup and short summary of what was learned.
Weeks 3–4	Implement a simple Schwinger model simulation and reproduce reference results from provided examples. Generate basic plots of simulation outputs.	Initial simulation code and validation plots.
Weeks 5–6	Add calculations of benchmark observables and study how results change when simulation parameters are varied.	Benchmark plots and documented analysis.
Weeks 7–8	Implement routines to calculate current-current correlators and generate plots for short-time evolution.	Correlator calculation code and example results.
Weeks 9–10	Compare results from different simulation settings and identify the main sources of numerical uncertainty. Improve code organization and documentation.	Comparison plots, cleaned-up code, and user documentation.

Weeks 11–12	Prepare final figures, summarize results, and contribute to the final report and presentation. Package the code and examples for future use.	Final report, presentation slides, and reproducible code package.