CERN Summer Student 2022

Project:
Integration of
ACTS into Gaudi

Doğa Elitez
Summer Student in EP-ADP-OS
M.Sc. in JGU Mainz

Supervisor:
Paul Gessinger-Befurt
Motivation

- Event reconstruction in high energy physics experiments starts with reconstruction of the trajectory of the charged particles.
- Various software and toolkits are used to describe the tracks in the detectors.
- High computational performance needed for future projects
- Increase of data with HL-LHC.
Theory: Track reconstruction in a nutshell

Track Reconstruction

- "Finding sets of measurements coming from one charged particle and building the associated trajectory through the detector"
- By using tracks, one can create higher level reconstruction objects.
- Trajectory of a charged particle in a magnetic field is defined by track parameters:

\[ \vec{q} = (l_1, l_2, \phi, \theta, q/p) \]

Track Propagation

- Detectors consist of many different components arranged in various structures.
- Track reconstruction uses the expression of these tracks at different surfaces in the detector.
- Different pattern recognition algorithms are used to recognize the tracks, then they are fitted through i.e. statistical methods and filters.
Toolkits: ACTS and Gaudi

ACTS: A Common Tracking Software

- Experiment-independent toolkit for track reconstruction applications
- Individual and configurable components are provided for the assembly of the experiment
- Main components:
  - Tracking geometry for the navigation and particle propagation
  - Magnetic field description
  - Seed finding algorithms
  - Track find and fitting chains

Gaudi

- A framework software package used to build data processing applications, developed mainly for ATLAS and LHCb experiments
- It is based on C++ with some key elements:
  - Boolean functions called "StatusCode" for initialization, execution and finalization
  - Algorithms and services are configured through python scripts
  - 'Gaudi Properties' allows easy configuration

Fun fact:
Gaudi means "big party" in southern German

ACTS integration to key4hep requires (among others) Gaudi integration

(ACTS Documentation: https://acts.readthedocs.io/en/latest/
Gaudi Documentation: https://gaudi-framework.readthedocs.io/en/latest/)
What have I done?:
Development of a service for integrating ACTS into Gaudi

Development of a service for the tracking geometry (GeoSvc)
Build detector geometry via DD4Hep from .XML files
Convert tracking geometry using ACTS
Visualize the geometry in a .OBJ file
What have I done?:
Development of a service for integrating ACTS into Gaudi

```python
from Configurables import PropagatorAlg
from Configurables import GeoSvc
sys.path.append('/home/deliter/ACTS/acts/services/acts/ActTracking')
import actsunits

alglist = []

b = PropagatorAlg("PropagatorAlg")
b.node = 0
b.stereoLogger = False
b.debugOutput = False
b.energyLoss = True
b.energyLoss = True
b.useScattering = False
b.recordMaterialInteractions = True
b.runTests = 100
b.omega = 0.03
b.omegam = 0.53
b.phisigma = 0.001
b.thermSigma = 0.001
b.covarianceTransport = False
b.omegam = 0.0001 / 1 * actsunits.mm
b.tSigma = 1 * actsunits.mm
b.ptoctors = 5000 * actsunits.mg
b.maxStepSize = 3 * actsunits.mm
b.sensitiveIdpt = 0

alglist.append(b)

a = GeoSvc("GeoSvc")
a.detectors = ['/home/deliter/ACTS/acts/thirdpartr/openDataDetector/xml/openDataDetector.xml']
a.debugGeometry = True
a.outputFileName = "MyDataFile"

from Configurables import ApplicationMgr

from Configurables import THistSvc:
THistSvc().Output = "[\"data file\"="propagatoralgoutput.root\", \"TYPE\"="ROOT\", \"OPT\"="RECREATE\"]"
THistSvc().PrintAll = True
THistSvc().AutoSave = True
THistSvc().AutoFlush = True

ApplicationMgr(TopAlg=alglist, Event='"HOMEP', EventMax=100, ExtSvc=[a], OutputLevel=DEBUG)
```

Development of a propagation testing algorithm (PropagatorAlg)

Inherited from the Gaudi Algorithm class
Use GeoSvc to extract the tracking geometry
For each particle, extract the track parameters, describe their propagation along a magnetic field
Conclusion: What is next? Creating a complete example

- Installing libraries
- Building dependencies
- Acts tracking geometry
- Propagation algorithm based on Acts example
- Integration of the particle gun example
- Integration of FATRAS
- .ROOT, .JSON output files
- Edm4hep integration

Thank you for your attention!