ACTS Developers Workshop 2022

GSF overview & status update

26.09.2022

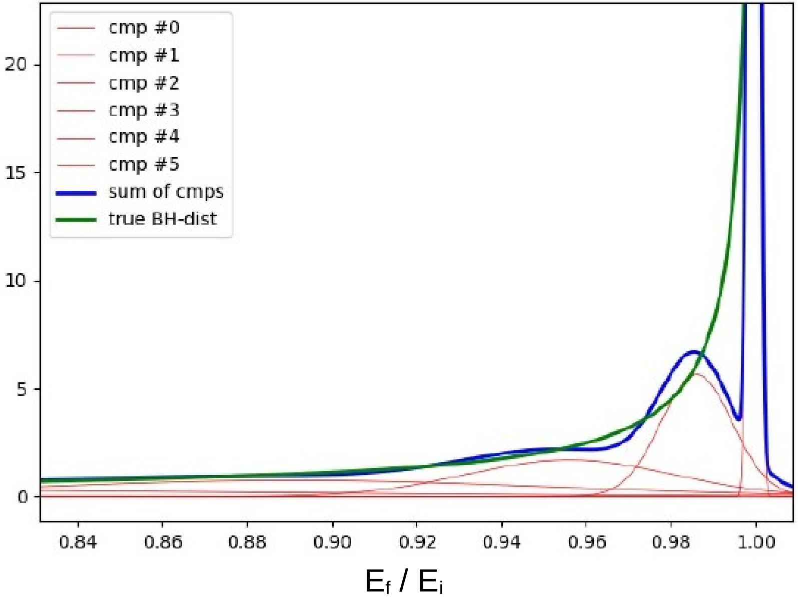
Benjamin Huth
Universität Regensburg

Overview

- Track Fitter suitable for handling non-gaussian effects
  - Fit multiple gaussian components in parallel with Kalman algorithm
  - Useful for electron (re)fitting
    - bremsstrahlung energy loss highly non gaussian
  - Describe track state as gaussian mixture:

$$p(\vec{x}) = \sum_i w_i \varphi(\vec{x}; \mu_i, \Sigma_i), \quad \sum_i w_i = 1$$

Probability distribution E_f / E_i energy loss for x=10mm



GSF update step

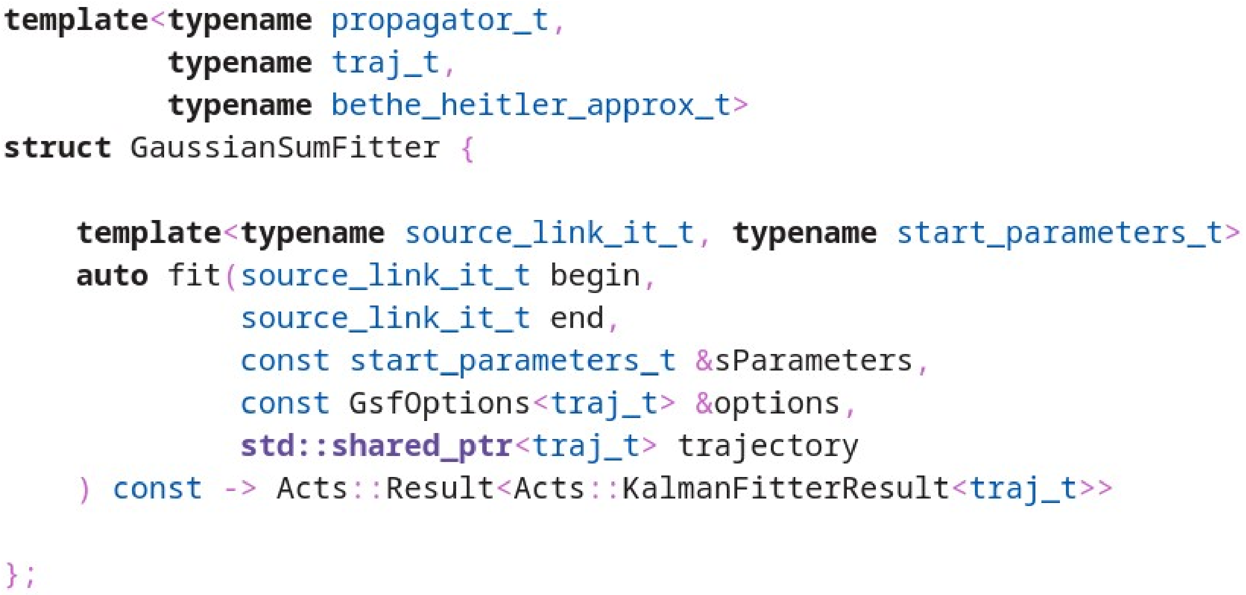
- When finding a measurement:
  - Kalman update for every surface as with KF
  - Reweight components based on compatability with the measurement
- When finding material on a surface:
  - Model bremsstrahlung for each component
  - Combine close components to keep # of components fix
  - Handle multiple scattering as in KF

Implementation strategy

- Need special Stepper to handle multiple components (`Acts::MultiEigenStepperLoop`)
  - Internally propagate multiple components
  - Interface as one “average state” to Navigator
    - Minimize navigation overhead
  - Possible optimization (not yet merged): `Acts::MultiEigenStepperSIMD`
- As much shared code as possible
  - `MultiEigenStepperLoop` inherits `EigenStepper`
  - Kalman update, Extensions

Interface

- Interface similar the `Acts::KalmanFitter`
  - Returns a `Acts::KalmanFitterResult`



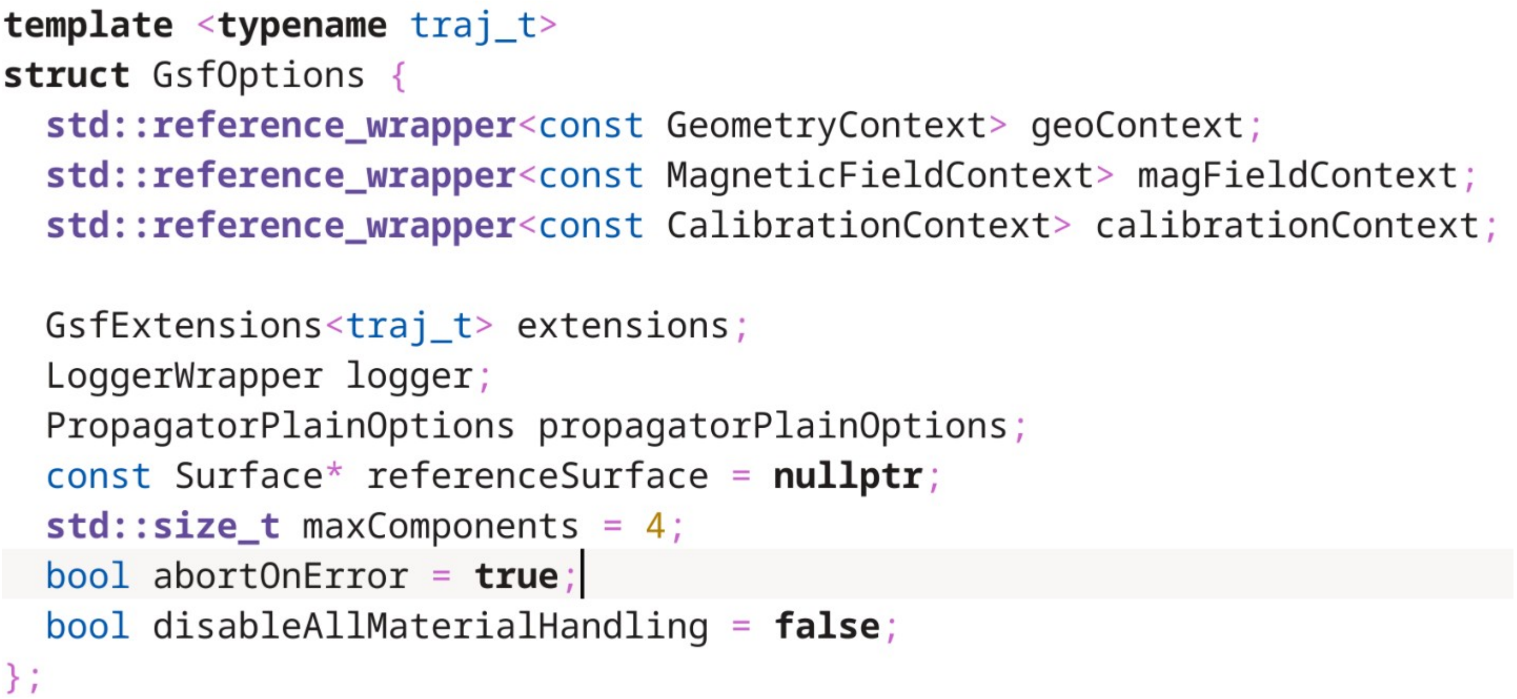
```
template<typename propagator_t,
         typename traj_t,
         typename bethe_heitler_approx_t>
struct GaussianSumFitter {

    template<typename source_link_it_t, typename start_parameters_t>
    auto fit(source_link_it_t begin,
             source_link_it_t end,
             const start_parameters_t &sParameters,
             const GsfOptions<traj_t> &options,
             std::shared_ptr<traj_t> trajectory
    ) const -> Acts::Result<Acts::KalmanFitterResult<traj_t>>

};
```

Configuration

- Various options (but there will come more...)

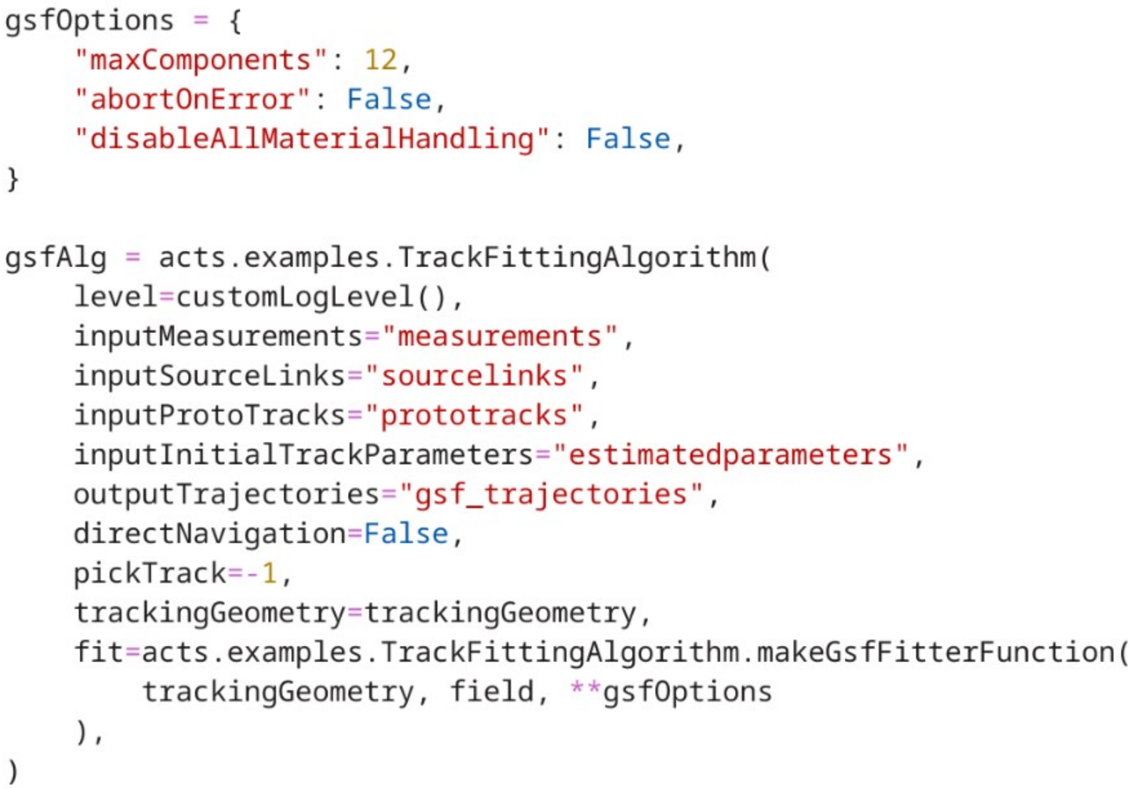


```
template <typename traj_t>
struct GsfOptions {
  std::reference_wrapper<const GeometryContext> geoContext;
  std::reference_wrapper<const MagneticFieldContext> magFieldContext;
  std::reference_wrapper<const CalibrationContext> calibrationContext;

  GsfExtensions<traj_t> extensions;
  LoggerWrapper logger;
  PropagatorPlainOptions propagatorPlainOptions;
  const Surface* referenceSurface = nullptr;
  std::size_t maxComponents = 4;
  bool abortOnError = true;
  bool disableAllMaterialHandling = false;
};
```

ACTS integration

- Available in the ACTS python bindings
- Stability check in CI
  - For runtime errors in generic detector and ODD
  - No physics performance monitored yet



```
gsfOptions = {
    "maxComponents": 12,
    "abortOnError": False,
    "disableAllMaterialHandling": False,
}

gsfAlg = acts.examples.TrackFittingAlgorithm(
    level=customLogLevel(),
    inputMeasurements="measurements",
    inputSourceLinks="sourcelinks",
    inputProtoTracks="prototracks",
    inputInitialTrackParameters="estimatedparameters",
    outputTrajectories="gsf_trajectories",
    directNavigation=False,
    pickTrack=-1,
    trackingGeometry=trackingGeometry,
    fit=acts.examples.TrackFittingAlgorithm.makeGsfFitterFunction(
        trackingGeometry, field, **gsfOptions
    ),
)
```

Status & Future

- First version merged early 2022
  - Not yet production ready
  - Still ongoing bugfixing
- Plans for validation and improvement:
  - Validation with Full simulation in OpenDataDetector
  - Cooperation with LDMX