

---

# global $\chi^2$ fitter

---

31 May 2021

Acts fitter discussion

Ralf Farkas

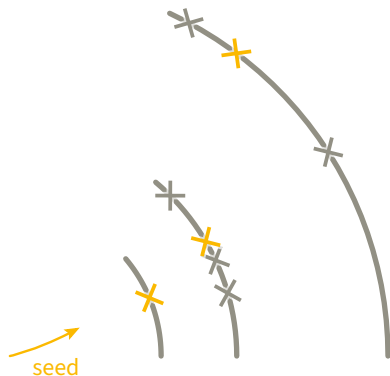
ralf@uni-bonn.de

Part 1

# approach

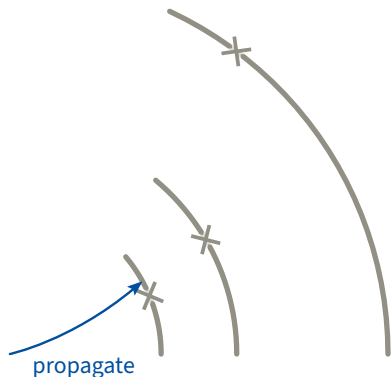
# $\chi^2$ fitter – approach

- starting track parameters  $\vec{x}_0$  (6-dim)



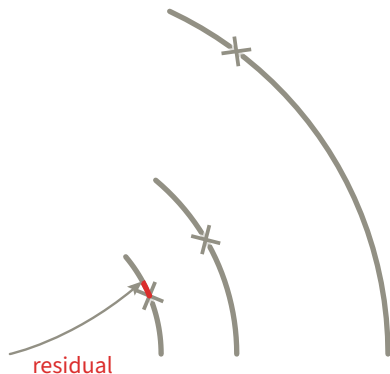
# $\chi^2$ fitter – approach

- starting track parameters  $\vec{x}_0$  (6-dim)
- propagation with transport matrix  $H$



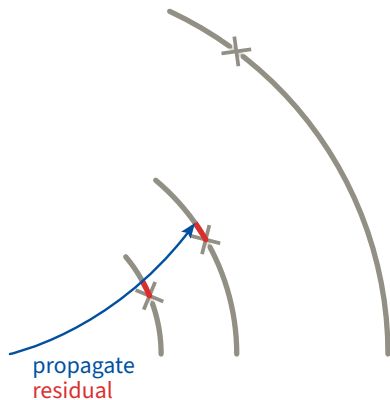
## $\chi^2$ fitter – approach

- starting track parameters  $\vec{x}_0$  (6-dim)
- propagation with transport matrix  $H$
- residuals  $r = m - H\vec{x}_0$



## $\chi^2$ fitter – approach

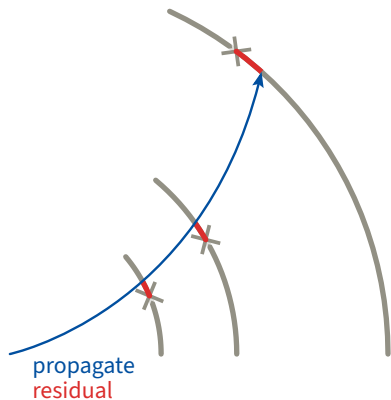
- starting track parameters  $\vec{x}_0$  (6-dim)
- propagation with transport matrix  $H$
- residuals  $r = m - H\vec{x}_0$



## $\chi^2$ fitter – approach

- starting track parameters  $\vec{x}_0$  (6-dim)
- propagation with transport matrix  $H$
- residuals  $r = m - H\vec{x}_0$

$$\chi^2 = r^T \text{cov}^{-1} r$$



# $\chi^2$ fitter – approach

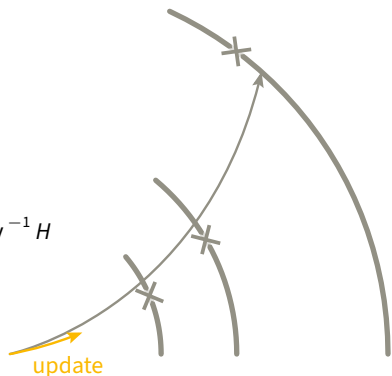
- starting track parameters  $\vec{x}_0$  (6-dim)
- propagation with transport matrix  $H$
- residuals  $r = m - H\vec{x}_0$

$$\chi^2 = r^T \text{cov}^{-1} r$$

- update

$$\frac{d\chi^2}{dx} = -2H^T \text{cov}^{-1} r \quad \frac{d^2\chi^2}{dx^2} = 2H^T \text{cov}^{-1} H$$

$$\vec{x}_1 = \vec{x}_0 - \left( \frac{d^2\chi^2}{dx^2} \right)^{-1} \frac{d\chi^2}{dx}$$





## $\chi^2$ fitter – approach

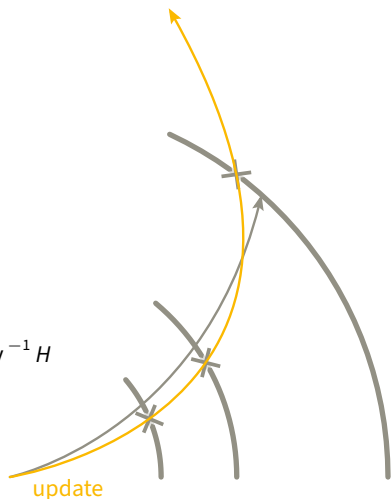
- starting track parameters  $\vec{x}_0$  (6-dim)
- propagation with transport matrix  $H$
- residuals  $r = m - H\vec{x}_0$

$$\chi^2 = r^T \text{cov}^{-1} r$$

- update

$$\frac{d\chi^2}{dx} = -2H^T \text{cov}^{-1} r \quad \frac{d^2\chi^2}{dx^2} = 2H^T \text{cov}^{-1} H$$

$$\vec{x}_1 = \vec{x}_0 - \left( \frac{d^2\chi^2}{dx^2} \right)^{-1} \frac{d\chi^2}{dx}$$



Part 2

# implementation

## implementation – Acts::Chi2Fitter

---

- `.fit()` function is entry point, runs propagation with `Chi2Actor`, calculates  $\chi^2$ , and delta on parameters
- iteration?
- result object `Acts::Chi2FitterResult` holds the *collectors* for  $\partial\chi^2$ ,  $\partial^2\chi^2$ ,  $r$ , ...
- *collectors* are filled during propagation in `Chi2Actor`
- » cannot use the same result class as KF
- » compatibility with e.g. `RootTrajectoryStatesWriter`?

## algorithm – ActsExamples::TrackFittingAlgorithm

---

- ActsExamples::TrackFittingAlgorithm is for the KF
- » re-use same TrackFittingAlgorithm, with option for  $\chi^2$ ?

Part 3

# configuration

## configuration – Chi2FitterOptions vs. TrackFittingAlgorithm::Config

---

- Acts::Chi2Fitter has it's own Acts::Chi2FitterOptions
  - for nIterations, eLoss, ...
- Acts::TrackFittingAlgorithm has TrackFittingAlgorithm::Config
  - is configured when adding Algorithm to the Sequencer
  - Chi2FitterOptions are initialised in TrackFittingAlgorithm::execute()
    - which then calls fitTrack()
      - which calls the fitter function m\_cfg.fit()
        - which calls Acts::Chi2Fitter.fit()

## configuration – Examples/Reconstruction/RecCKFTracks.cpp

```
126 TrackFittingAlgorithm::Config fitter;
127 fitter.inputMeasurements = hitSmearingCfg.outputMeasurements;
128 fitter.inputSourceLinks = hitSmearingCfg.outputSourceLinks;
129 fitter.inputProtoTracks = trackFinderCfg.outputProtoTracks;
130 fitter.inputInitialTrackParameters = particleSmearingCfg.outputTrackParameters;
131 fitter.outputTrajectories = "trajectories";
132 fitter.trackingGeometry = trackingGeometry;
133 fitter.useChi2Fitter = true;
134 fitter.chi2numberIterations = 2;
135 // fitter.fit = TrackFittingAlgorithm::makeTrackFitterFunction(...);
136 fitter.chi2Fit = TrackFittingAlgorithm::makeChi2TrackFitterFunction(
137     trackingGeometry, magneticField);
138 sequencer.addAlgorithm(
139     std::make_shared<TrackFittingAlgorithm>(fitter, logLevel));
```

- » clutter TrackFittingAlgorithm::Config with  $\chi^2$  specific options?
- » Add Acts::Chi2FitterOptions object instead of using makeChi2TrackFitterFunction?

Part 4

# fitter function



## fitter function 1/2 — Examples/Algorithms/TrackFitting/src/TrackFittingAlgorithmFunction.cpp

```
27 template <typename track_fitter_t>
28 struct TrackFitterFunctionImpl {
29     track_fitter_t trackFitter;
30
31     TrackFitterFunctionImpl(track_fitter_t&& f) : trackFitter(std::move(f)) {}
32
33     ActsExamples::TrackFittingAlgorithm::TrackFitterResult operator()(
34         const std::vector<ActsExamples::IndexSourceLink>& sourceLinks,
35         const ActsExamples::TrackParameters& initialParameters,
36         const ActsExamples::TrackFittingAlgorithm::TrackFitterOptions& options)
37         const {
38         return trackFitter.fit(sourceLinks, initialParameters, options);
39     };
40 };
```

## fitter function 2/2 — Examples/Algorithms/TrackFitting/src/TrackFittingAlgorithmFunction.cpp

```
143 ActsExamples::TrackFittingAlgorithm::Chi2TrackFitterFunction
144 ActsExamples::TrackFittingAlgorithm::makeChi2TrackFitterFunction(
145     std::shared_ptr<const Acts::TrackingGeometry> trackingGeometry,
146     Options::BFieldVariant magneticField) {
147
148     // unpack the magnetic field variant and instantiate the corresponding fitter.
149     return std::visit(
150         [trackingGeometry](auto&& inputField) -> Chi2TrackFitterFunction {
151             // ...
152             using Propagator = Acts::Propagator<Stepper, Navigator>;
153             using Fitter = Acts::Chi2Fitter<Propagator>;
154             // ...
155             Propagator propagator(std::move(stepper), std::move(navigator));
156             Fitter trackFitter(std::move(propagator));
157
158             return Chi2TrackFitterFunctionImpl<Fitter>(std::move(trackFitter));
159         },
160         std::move(magneticField));
161 }
```