Overview of current tracking output from EICRecon

Barak Schmookler

Data flow and output

Truth-seeded track seed parameters

```
28        app->Add(new JChainFactoryGeneratorT<TrackParamTruthInit_factory>(
29                {"MCParticles"}, "InitTrackParams"));
```

Real-seeded track seed parameters

```
68        app->Add(new JChainFactoryGeneratorT<TrackSeeding_factory>(
69                {"CentralTrackingRecHits"}, "CentralTrackSeedingResults"));
```

EICRecon tracking.cc file

```
350       edm4eic::TrackParameters:
351         Description: "ACTS Bound Track parameters"
352         Author: "W. Armstrong, S. Joosten"
353         Members:
354           - int32_t           type            // Type of track parameters (-1/seed, 0/head, ...)
355           - edm4hep::Vector2f loc             // 2D location on surface
356           - edm4eic::Cov2f    locError        // Covariance on loc
357           - float             theta           // Track polar angle [rad]
358           - float             phi             // Track azimuthal angle [rad]
359           - float             qOverP          // [e/GeV]
360           - edm4eic::Cov3f    momentumError   // Covariance on theta, phi and qOverP
361           - float             time            // Track time [ns]
362           - float             timeError       // Error on the time
363           - float             charge          // Particle charge
364         OneToOneRelations:
365           - edm4eic::Trajectory trajectory    // Trajectory associated with these track parameters
```

edm4eic.yaml file

In EICRecon, each seed will produce a single trajectory and a single set of track parameters. This is because we only save trackTips.front() right now.

Data flow and output

Truth-seeded tracking output

```
54  app->Add(new JChainMultifactoryGeneratorT<CKFTracking_factory>(
55      "CentralCKFTrajectories",
56      {
57          "InitTrackParams",
58          "CentralTrackerSourceLinker"
59      },
60      {
61          "CentralCKFTrajectories",
62          "CentralCKFTrackParameters",
63          "CentralCKFActsTrajectories",
64      },
65      app
66  ));
```

Real-seeded tracking output

```
71  app->Add(new JChainMultifactoryGeneratorT<CKFTracking_factory>(
72      "CentralCKFSeededTrajectories",
73      {
74          "CentralTrackSeedingResults",
75          "CentralTrackerSourceLinker"
76      },
77      {
78          "CentralCKFSeededTrajectories",
79          "CentralCKFSeededTrackParameters",
80          "CentralCKFSeededActsTrajectories",
81      },
82      app
83  ));
```

EICRecon tracking.cc file

```
330  edm4eic::Trajectory:
331    Description: "Raw trajectory from the tracking algorithm"
332    Author: "S. Joosten, S. Li"
333    Members:
334      - uint32_t          type             // 0 (does not have good track fit), 1 (has good track fit)
335      - uint32_t          nStates          // Number of tracking steps
336      - uint32_t          nMeasurements    // Number of hits used
337      - uint32_t          nOutliers        // Number of hits not considered
338      - uint32_t          nHoles           // Number of missing hits
339      - float             chi2             // Total chi2
340      - uint32_t          ndf              // Number of degrees of freedom
341      - uint32_t          nSharedHits      // Number of shared hits with other trajectories
342    VectorMembers:
343      - float             measurementChi2  // Chi2 for each of the measurements
344      - float             outlierChi2      // Chi2 for each of the outliers
345    OneToManyRelations:
346      - edm4eic::TrackParameters trackParameters // Associated track parameters, if any
347      - edm4eic::TrackerHit  measurementHits // Measurement hits used in this trajectory
348      - edm4eic::TrackerHit  outlierHits     // Outlier hits not used in this trajectory
```

```
350  edm4eic::TrackParameters:
351    Description: "ACTS Bound Track parameters"
352    Author: "W. Armstrong, S. Joosten"
353    Members:
354      - int32_t           type             // Type of track parameters (-1/seed, 0/head, ...)
355      - edm4hep::Vector2f loc              // 2D location on surface
356      - edm4eic::Cov2f    locError         // Covariance on loc
357      - float             theta            // Track polar angle [rad]
358      - float             phi              // Track azimuthal angle [rad]
359      - float             qOverP           // [e/GeV]
360      - edm4eic::Cov3f    momentumError    // Covariance on theta, phi and qOverP
361      - float             time             // Track time [ns]
362      - float             timeError        // Error on the time
363      - float             charge           // Particle charge
364    OneToOneRelations:
365      - edm4eic::Trajectory trajectory     // Trajectory associated with these track parameters
```

edm4eic.yaml file

Data flow and output

Truth-seeded tracking output

```
// link charged particles to PID and to MC truth
app->Add(new JChainMultifactoryGeneratorT<ParticlesWithPID_factory>(
      "ChargedParticlesWithAssociations",
      {
        "MCParticles",                    // edm4hep::MCParticle
        "CentralCKFTrajectories",         // edm4eic::Trajectory
        "DRICHMergedIrtCherenkovParticleID" // edm4eic::CherenkovParticleID
      },
      {
        "ReconstructedChargedParticles",           // edm4eic::ReconstructedParticle
        "ReconstructedChargedParticleAssociations", // edm4eic::MCRecoParticleAssociation
        "ReconstructedChargedParticleIDs"          // edm4hep::ParticleID
      },
      link_cfg,
      app
      ));
```

Real-seeded tracking output

```
app->Add(new JChainMultifactoryGeneratorT<ParticlesWithPID_factory>(
      "ChargedParticlesWithAssociations",
      {
        "MCParticles",                    // edm4hep::MCParticle
        "CentralCKFSeededTrajectories",   // edm4eic::Trajectory
        "DRICHMergedIrtCherenkovParticleID" // edm4eic::CherenkovParticleID
      },
      {
        "ReconstructedSeededChargedParticles",           // edm4eic::ReconstructedParticle
        "ReconstructedSeededChargedParticleAssociations", // edm4eic::MCRecoParticleAssociation
        "ReconstructedSeededChargedParticleIDs"          // edm4hep::ParticleID
      },
      link_cfg,
      app
      ));
```

EICRecon pid.cc file

```
edm4eic::ReconstructedParticle:
  Description: "EIC Reconstructed Particle"
  Author: "W. Armstrong, S. Joosten, F. Gaede"
  Members:
    - int32_t          type             // type of reconstructed particle. Check/set collection parameters ReconstructedParticleTypeNames and ReconstructedParticleTypeValues.
    - float            energy           // [GeV] energy of the reconstructed particle. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f momentum        // [GeV] particle momentum. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f referencePoint  // [mm] reference, i.e. where the particle has been measured
    - float            charge           // charge of the reconstructed particle.
    - float            mass             // [GeV] mass of the reconstructed particle, set independently from four vector. Four momentum state is not kept consistent internally.
    - float            goodnessOfPID    // overall goodness of the PID on a scale of [0;1]
    - edm4eic::Cov4f   covMatrix        // covariance matrix of the reconstructed particle 4vector (10 parameters).
    ##@TODO: deviation from EDM4hep: store explicit PDG ID here. Needs to be discussed how we
    ##       move forward as this could easiliy become unwieldy without this information here.
    ##       The only acceptable alternative would be to store reconstructed identified
    ##       particles in separate collections for the different particle types (which would
    ##       require some algorithmic changes but might work. Doing both might even make
    ##       sense. Needs some discussion, note that PID is more emphasized in NP than
    ##       HEP).
    - int32_t          PDG              // PDG code for this particle
    ## @TODO: Do we need timing info? Or do we rely on the start vertex time?
  OneToOneRelations:
    - edm4eic::Vertex      startVertex    // Start vertex associated to this particle
    - edm4hep::ParticleID  particleIDUsed // particle ID used for the kinematics of this particle
  OneToManyRelations:
    - edm4eic::Cluster     clusters       // Clusters used for this particle
    - edm4eic::Track       tracks         // Tracks used for this particle
    - edm4eic::ReconstructedParticle particles // Reconstructed particles that have been combined to this particle
    - edm4hep::ParticleID  particleIDs    // All associated particle IDs for this particle (not sorted by likelihood)
  ExtraCode:
    declaration: "
      bool isCompound() const {return particles_size() > 0;}\n
```

edm4eic.yaml file

Data flow and output

Currently, uses truth-seeded tracking output only

```
85      app->Add(new JChainFactoryGeneratorT<TrackProjector_factory>(
86              {"CentralCKFActsTrajectories"}, "CentralTrackSegments"));
87
88      app->Add(new JChainFactoryGeneratorT<IterativeVertexFinder_factory>(
89              {"CentralCKFActsTrajectories"}, "CentralTrackVertices"));
90
91      app->Add(new JChainMultifactoryGeneratorT<TrackPropagation_factory>(
92              "CalorimeterTrackPropagator",
93              {"CentralCKFActsTrajectories"},
94              {"CalorimeterTrackProjections"},
95              app
96              ));
```

Track parameters at various tracking layers

Reconstructed primary vertex

Track projections to calorimeters

EICRecon tracking.cc file

Looking at the results with single muons

```
root [1] events->SetAlias("P_gen","sqrt(MCParticles.momentum.x*MCParticles.momentum.x+MCParticles.momentum.y*MCParticles.momentum.y+MCP
articles.momentum.z*MCParticles.momentum.z)")
(bool) true
root [2]
root [2] events->SetAlias("Theta_gen","acos(MCParticles.momentum.z/P_gen)")
(bool) true
root [3]
root [3]
root [3]
root [3] events->Scan("P_gen:Theta_gen","MCParticles.generatorStatus==1")
************************************************
*    Row    * Instance *     P_gen * Theta_gen *
************************************************
*        0 *        2 * 3.5088528 * 0.1949865 *
*        1 *        2 * 5.3045405 * 2.9998522 *
*        2 *        2 * 15.631042 * 0.1103796 *
*        3 *        2 * 12.187963 * 0.0852989 *
*        4 *        2 * 7.9803351 * 0.2160943 *
*        5 *        2 * 1.3676079 * 2.3534648 *
*        6 *        2 * 1.0619225 * 0.1171895 *
*        7 *        2 * 10.641024 * 2.6204696 *
*        8 *        2 * 18.591475 * 3.0206306 *
*        9 *        2 * 0.9774184 * 0.0614766 *
*       10 *        2 * 15.670197 * 0.9256906 *
```

Event number

Muon momentum

Muon theta

Looking at (real-seeded) results with single muons

```
root [9] events->SetAlias("P_rec","sqrt(ReconstructedSeededChargedParticles.momentum.x*ReconstructedSeededChargedParticles.momentum.x+R
econstructedSeededChargedParticles.momentum.y*ReconstructedSeededChargedParticles.momentum.y+ReconstructedSeededChargedParticles.momen
um.z*ReconstructedSeededChargedParticles.momentum.z)")
(bool) true
root [10]
root [10] events->SetAlias("theta_rec","acos(ReconstructedSeededChargedParticles.momentum.z/P_rec)")
(bool) true
root [11]
root [11] events->Scan("CentralTrackSeedingResults.theta:CentralCKFSeededTrackParameters.theta:theta_rec:CentralCKFSeededTrajectories.n
************************************************************************
*    Row    * Instance * CentralTr * CentralCK * theta_rec * CentralCK *
************************************************************************
*        0 *        0 * 0.1937180 * 0.1945763 * 0.1945763 *         4 *
*        0 *        1 * 0.1940771 * 0.1945463 * 0.1945463 *         4 *
*        0 *        2 * 0.1938095 * 0.1947055 * 0.1947055 *         4 *
*        1 *        0 * 2.9997701 * 2.9994049 * 2.9994049 *         2 *
*        1 *        1 * 2.9998300 * 2.9996504 * 2.9996504 *         2 *
*        2 *        0 * 0.1104461 * 0.1103764 * 0.1103764 *         4 *
*        2 *        1 * 0.1103586 * 0.1102905 * 0.1102905 *         4 *
*        3 *        0 * 0.0850231 * 0.0851216 * 0.0851216 *         4 *
*        3 *        1 * 0.0850503 * 0.0851085 * 0.0851085 *         4 *
*        4 *        0 * 0.2157898 * 0.2161549 * 0.2161550 *         4 *
*        4 *        1 * 0.2160625 * 0.2161960 * 0.2161960 *         4 *
*        4 *        2 * 0.2157084 * 0.2162159 * 0.2162160 *         4 *
*        5 *        0 * 2.3530783 * 2.3530213 * 2.3530213 *         3 *
*        6 *        0 * 0.1159824 * 0.1192474 * 0.1192474 *         2 *
*        6 *        1 * 0.1127062 * 0.1198665 * 0.1198665 *         2 *
```

Event number

Seed theta

Track parameters theta

Reconstructed Charged particles theta

Trajectory nMeasurements

Looking at (real-seeded) results with single muons

```
root [9] events->SetAlias("P_rec","sqrt(ReconstructedSeededChargedParticles.momentum.x*ReconstructedSeededChargedParticles.momentum.x+R
econstructedSeededChargedParticles.momentum.y*ReconstructedSeededChargedParticles.momentum.y+ReconstructedSeededChargedParticles.momen
um.z*ReconstructedSeededChargedParticles.momentum.z)")
(bool) true
root [10]
root [10] events->SetAlias("theta_rec","acos(ReconstructedSeededChargedParticles.momentum.z/P_rec)")
(bool) true
root [11]
root [11] events->Scan("CentralTrackSeedingResults.theta:CentralCKFSeededTrackParameters.theta:theta_rec:CentralCKFSeededTrajectories.
************************************************************************
*    Row   * Instance * CentralTr * CentralCK * theta_rec * CentralCK *
************************************************************************
*        0 *        0 * 0.1937180 * 0.1945763 * 0.1945763 *         4 *
*        0 *        1 * 0.1940771 * 0.1945463 * 0.1945463 *         4 *
*        0 *        2 * 0.1938095 * 0.1947055 * 0.1947055 *         4 *
*        1 *        0 * 2.9997701 * 2.9994049 * 2.9994049 *         2 *
*        1 *        1 * 2.9998300 * 2.9996504 * 2.9996504 *         2 *
*        2 *        0 * 0.1104461 * 0.1103764 * 0.1103764 *         4 *
*        2 *        1 * 0.1103586 * 0.1102905 * 0.1102905 *         4 *
*        3 *        0 * 0.0850231 * 0.0851216 * 0.0851216 *         4 *
*        3 *        1 * 0.0850503 * 0.0851085 * 0.0851085 *         4 *
*        4 *        0 * 0.2157898 * 0.2161549 * 0.2161550 *         4 *
*        4 *        1 * 0.2160625 * 0.2161960 * 0.2161960 *         4 *
*        4 *        2 * 0.2157084 * 0.2162159 * 0.2162160 *         4 *
*        5 *        0 * 2.3530783 * 2.3530213 * 2.3530213 *         3 *
*        6 *        0 * 0.1159824 * 0.1192474 * 0.1192474 *         2 *
*        6 *        1 * 0.1127062 * 0.1198665 * 0.1198665 *         2 *
```

Looking at the 1st event, we see 3 seeds. Each seed produces a single trajectory and a single set of track parameters. This is because we only save trackTips.front() right now.

The ReconstructedChargedParticles copies the information from the track parameters.

Looking at (real-seeded) results with single muons

```
root [9] events->SetAlias("P_rec","sqrt(ReconstructedSeededChargedParticles.momentum.x*ReconstructedSeededChargedParticles.momentum.x+R
econstructedSeededChargedParticles.momentum.y*ReconstructedSeededChargedParticles.momentum.y+ReconstructedSeededChargedParticles.momen
um.z*ReconstructedSeededChargedParticles.momentum.z)")
(bool) true
root [10]
root [10] events->SetAlias("theta_rec","acos(ReconstructedSeededChargedParticles.momentum.z/P_rec)")
(bool) true
root [11]
root [11] events->Scan("CentralTrackSeedingResults.theta:CentralCKFSeededTrackParameters.theta:theta_rec:CentralCKFSeededTrajectories.
************************************************************************
*    Row   * Instance * CentralTr * CentralCK * theta_rec * CentralCK *
************************************************************************
*        0 *        0 * 0.1937180 * 0.1945763 * 0.1945763 *         4 *
*        0 *        1 * 0.1940771 * 0.1945463 * 0.1945463 *         4 *
*        0 *        2 * 0.1938095 * 0.1947055 * 0.1947055 *         4 *
*        1 *        0 * 2.9997701 * 2.9994049 * 2.9994049 *         2 *
*        1 *        1 * 2.9998300 * 2.9996504 * 2.9996504 *         2 *
*        2 *        0 * 0.1104461 * 0.1103764 * 0.1103764 *         4 *
*        2 *        1 * 0.1103586 * 0.1102905 * 0.1102905 *         4 *
*        3 *        0 * 0.0850231 * 0.0851216 * 0.0851216 *         4 *
*        3 *        1 * 0.0850503 * 0.0851085 * 0.0851085 *         4 *
*        4 *        0 * 0.2157898 * 0.2161549 * 0.2161550 *         4 *
*        4 *        1 * 0.2160625 * 0.2161960 * 0.2161960 *         4 *
*        4 *        2 * 0.2157084 * 0.2162159 * 0.2162160 *         4 *
*        5 *        0 * 2.3530783 * 2.3530213 * 2.3530213 *         3 *
*        6 *        0 * 0.1159824 * 0.1192474 * 0.1192474 *         2 *
*        6 *        1 * 0.1127062 * 0.1198665 * 0.1198665 *         2 *
```

The 3 seeds/tracks look like duplicates. The reconstructed theta angle for the tracks is close to the generated muon theta angle of 0.1950 Radians.

Association between trajectories and track parameters

```
CentralCKFSeededTrajectories = (vector<edm4eic::TrajectoryData>*)0x4e52290
CentralCKFSeededTrajectories.type = 0, 0, 0
CentralCKFSeededTrajectories.nStates = 10, 10, 10
CentralCKFSeededTrajectories.nMeasurements = 4, 4, 4
CentralCKFSeededTrajectories.nOutliers = 3, 3, 3
CentralCKFSeededTrajectories.nHoles = 1, 1, 1
CentralCKFSeededTrajectories.chi2 = 2.466630, 2.439069, 2.992647
CentralCKFSeededTrajectories.ndf = 14, 14, 14
CentralCKFSeededTrajectories.nSharedHits = 0, 0, 0
CentralCKFSeededTrajectories.measurementChi2_begin = 0, 4, 8
CentralCKFSeededTrajectories.measurementChi2_end = 4, 8, 12
CentralCKFSeededTrajectories.outlierChi2_begin = 0, 3, 6
CentralCKFSeededTrajectories.outlierChi2_end = 3, 6, 9
CentralCKFSeededTrajectories.trackParameters_begin = 0, 1, 2
CentralCKFSeededTrajectories.trackParameters_end = 1, 2, 3
CentralCKFSeededTrajectories.measurementHits_begin = 0, 0, 0
CentralCKFSeededTrajectories.measurementHits_end = 0, 0, 0
CentralCKFSeededTrajectories.outlierHits_begin = 0, 0, 0
CentralCKFSeededTrajectories.outlierHits_end = 0, 0, 0
CentralCKFSeededTrajectories#0 = (vector<podio::ObjectID>*)0x4f2ea10
CentralCKFSeededTrajectories#0.index = 0, 1, 2
CentralCKFSeededTrajectories#0.collectionID = 84, 84, 84
CentralCKFSeededTrajectories_0 = (vector<float>*)0x565b240
CentralCKFSeededTrajectories_1 = (vector<float>*)0x565bef0
```

```
edm4eic::Trajectory:
  Description: "Raw trajectory from the tracking algorithm"
  Author: "S. Joosten, S. Li"
  Members:
    - uint32_t          type              // 0 (does not have good track fit), 1 (has good track fit)
    - uint32_t          nStates           // Number of tracking steps
    - uint32_t          nMeasurements     // Number of hits used
    - uint32_t          nOutliers         // Number of hits not considered
    - uint32_t          nHoles            // Number of missing hits
    - float             chi2              // Total chi2
    - uint32_t          ndf               // Number of degrees of freedom
    - uint32_t          nSharedHits       // Number of shared hits with other trajectories
  VectorMembers:
    - float             measurementChi2   // Chi2 for each of the measurements
    - float             outlierChi2       // Chi2 for each of the outliers
  OneToManyRelations:
    - edm4eic::TrackParameters trackParameters // Associated track parameters, if any
    - edm4eic::TrackerHit  measurementHits // Measurement hits used in this trajectory
    - edm4eic::TrackerHit  outlierHits     // Outlier hits not used in this trajectory
```

If in the future we allow the (Multi)Trajectory to have multiple sets of track parameters, we can use this association to link the data types.

A couple issues with trajectory information

```
CentralCKFSeededTrajectories = (vector<edm4eic::TrajectoryData>*)0x4e52290
CentralCKFSeededTrajectories.type = 0, 0, 0
CentralCKFSeededTrajectories.nStates = 10, 10, 10
CentralCKFSeededTrajectories.nMeasurements = 4, 4, 4
CentralCKFSeededTrajectories.nOutliers = 3, 3, 3
CentralCKFSeededTrajectories.nHoles = 1, 1, 1
CentralCKFSeededTrajectories.chi2 = 2.466630, 2.439069, 2.992647
CentralCKFSeededTrajectories.ndf = 14, 14, 14
CentralCKFSeededTrajectories.nSharedHits = 0, 0, 0
CentralCKFSeededTrajectories.measurementChi2_begin = 0, 4, 8
CentralCKFSeededTrajectories.measurementChi2_end = 4, 8, 12
CentralCKFSeededTrajectories.outlierChi2_begin = 0, 3, 6
CentralCKFSeededTrajectories.outlierChi2_end = 3, 6, 9
CentralCKFSeededTrajectories.trackParameters_begin = 0, 1, 2
CentralCKFSeededTrajectories.trackParameters_end = 1, 2, 3
CentralCKFSeededTrajectories.measurementHits_begin = 0, 0, 0
CentralCKFSeededTrajectories.measurementHits_end = 0, 0, 0
CentralCKFSeededTrajectories.outlierHits_begin = 0, 0, 0
CentralCKFSeededTrajectories.outlierHits_end = 0, 0, 0
CentralCKFSeededTrajectories#0 = (vector<podio::ObjectID>*)0x4f2ea10
CentralCKFSeededTrajectories#0.index = 0, 1, 2
CentralCKFSeededTrajectories#0.collectionID = 84, 84, 84
CentralCKFSeededTrajectories_0 = (vector<float>*)0x565b240
CentralCKFSeededTrajectories_1 = (vector<float>*)0x565bef0
```

```
edm4eic::Trajectory:
  Description: "Raw trajectory from the tracking algorithm"
  Author: "S. Joosten, S. Li"
  Members:
    - uint32_t          type              // 0 (does not have good track fit), 1 (has good track fit)
    - uint32_t          nStates           // Number of tracking steps
    - uint32_t          nMeasurements     // Number of hits used
    - uint32_t          nOutliers         // Number of hits not considered
    - uint32_t          nHoles            // Number of missing hits
    - float             chi2              // Total chi2
    - uint32_t          ndf               // Number of degrees of freedom
    - uint32_t          nSharedHits       // Number of shared hits with other trajectories
  VectorMembers:
    - float             measurementChi2   // Chi2 for each of the measurements
    - float             outlierChi2       // Chi2 for each of the outliers
  OneToManyRelations:
    - edm4eic::TrackParameters trackParameters // Associated track parameters, if any
    - edm4eic::TrackerHit  measurementHits // Measurement hits used in this trajectory
    - edm4eic::TrackerHit  outlierHits     // Outlier hits not used in this trajectory
```

We have a list of indices for the individual hit chi-squares, but we can't find the actual values in the file.

A couple issues with trajectory information

```
root [34] events->Scan("CentralCKFSeededTrajectories.nSharedHits")
************************************
*    Row   * Instance * CentralCK *
************************************
*        0 *        0 *         0 *
*        0 *        1 *         0 *
*        0 *        2 *         0 *
*        1 *        0 *         0 *
*        1 *        1 *         0 *
*        2 *        0 *         0 *
*        2 *        1 *         0 *
*        3 *        0 *         0 *
*        3 *        1 *         0 *
*        4 *        0 *         0 *
*        4 *        1 *         0 *
*        4 *        2 *         0 *
*        5 *        0 *         0 *
*        6 *        0 *         0 *
*        6 *        1 *         0 *
```

```
edm4eic::Trajectory:
  Description: "Raw trajectory from the tracking algorithm"
  Author: "S. Joosten, S. Li"
  Members:
    - uint32_t          type             // 0 (does not have good track fit), 1 (has good track fit)
    - uint32_t          nStates          // Number of tracking steps
    - uint32_t          nMeasurements    // Number of hits used
    - uint32_t          nOutliers        // Number of hits not considered
    - uint32_t          nHoles           // Number of missing hits
    - float             chi2             // Total chi2
    - uint32_t          ndf              // Number of degrees of freedom
    - uint32_t          nSharedHits      // Number of shared hits with other trajectories
  VectorMembers:
    - float             measurementChi2  // Chi2 for each of the measurements
    - float             outlierChi2      // Chi2 for each of the outliers
  OneToManyRelations:
    - edm4eic::TrackParameters trackParameters // Associated track parameters, if any
    - edm4eic::TrackerHit  measurementHits // Measurement hits used in this trajectory
    - edm4eic::TrackerHit  outlierHits     // Outlier hits not used in this trajectory
```

The nSharedHits for all the trajectories seems to always be zero, even when we have the duplicated tracks.

A couple issues with trajectory information

```
root [33] events->Scan("CentralCKFSeededTrajectories#0.collectionID:CentralCKFSeededTrajectories#1.collectionID:CentralCKFSeededTrajectories#2.collectionID")
************************************************************
*    Row   * Instance * CentralCK * CentralCK * CentralCK *
************************************************************
*        0 *        0 *        84 *           *           *
*        0 *        1 *        84 *           *           *
*        0 *        2 *        84 *           *           *
*        1 *        0 *        84 *           *           *
*        1 *        1 *        84 *           *           *
*        2 *        0 *        84 *           *           *
*        2 *        1 *        84 *           *           *
*        3 *        0 *        84 *           *           *
*        3 *        1 *        84 *           *           *
*        4 *        0 *        84 *           *           *
*        4 *        1 *        84 *           *           *
*        4 *        2 *        84 *           *           *
*        5 *        0 *        84 *           *           *
*        6 *        0 *        84 *           *           *
*        6 *        1 *        84 *           *           *
*        7 *        0 *        84 *           *           *
*        7 *        1 *        84 *           *           *
*        7 *        2 *        84 *           *           *
*        8 *        0 *        84 *           *           *
*        8 *        1 *        84 *           *           *
*        9 *        0 *        84 *           *           *
*       10 *        0 *        84 *           *           *
*       10 *        1 *        84 *           *           *
*       10 *        2 *        84 *           *           *
*       11 *        0 *        84 *           *           *
```

```
edm4eic::Trajectory:
  Description: "Raw trajectory from the tracking algorithm"
  Author: "S. Joosten, S. Li"
  Members:
    - uint32_t          type              // 0 (does not have good track fit), 1 (has good track fit)
    - uint32_t          nStates           // Number of tracking steps
    - uint32_t          nMeasurements     // Number of hits used
    - uint32_t          nOutliers         // Number of hits not considered
    - uint32_t          nHoles            // Number of missing hits
    - float             chi2              // Total chi2
    - uint32_t          ndf               // Number of degrees of freedom
    - uint32_t          nSharedHits       // Number of shared hits with other trajectories
  VectorMembers:
    - float             measurementChi2   // Chi2 for each of the measurements
    - float             outlierChi2       // Chi2 for each of the outliers
  OneToManyRelations:
    - edm4eic::TrackParameters trackParameters // Associated track parameters, if any
    - edm4eic::TrackerHit  measurementHits // Measurement hits used in this trajectory
    - edm4eic::TrackerHit  outlierHits     // Outlier hits not used in this trajectory
```

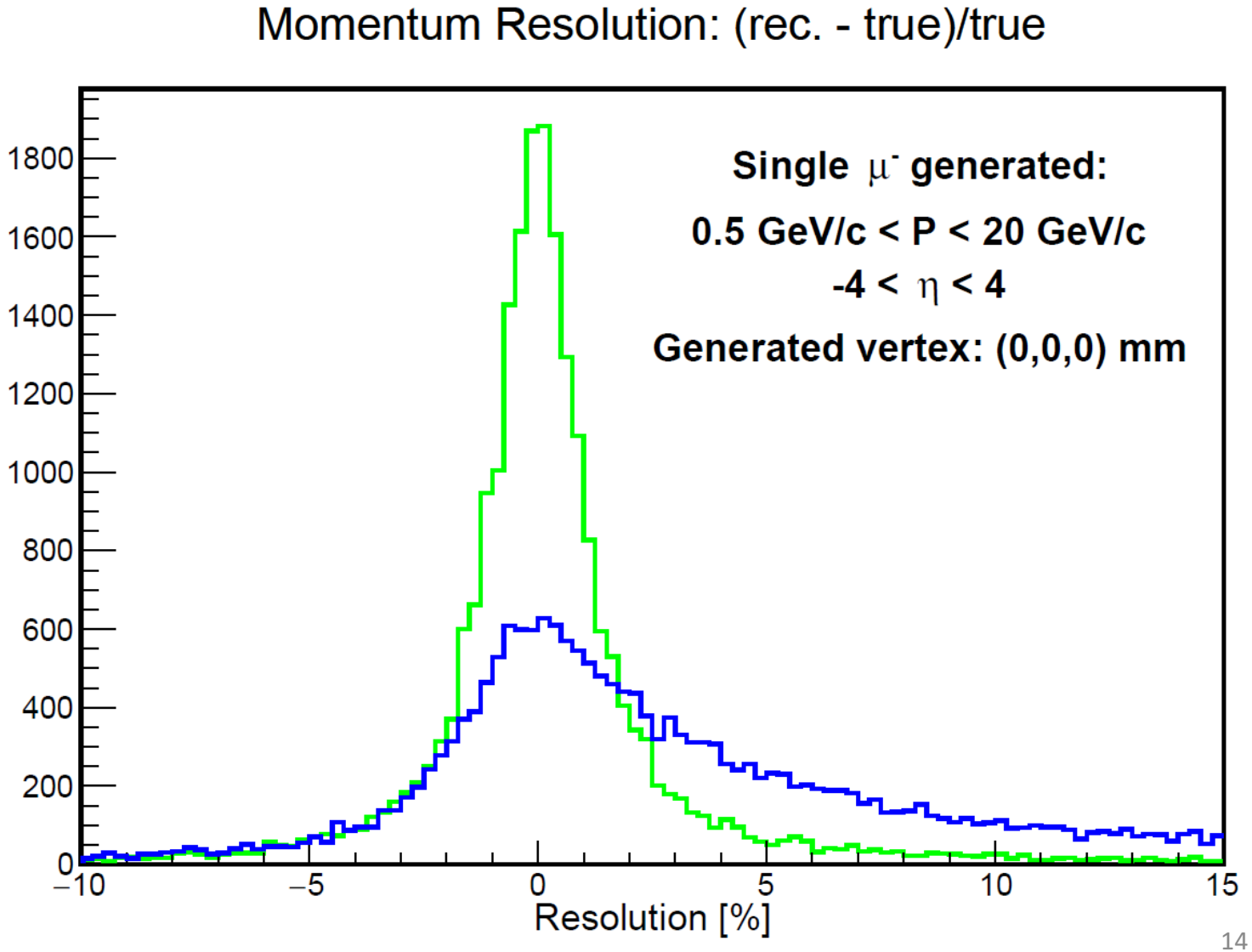
The associations to the digitized hits are missing. This is expected right now, since we are not extracting the used hits after the CKF fit.

We may need to add an index to the source linker to keep track of the hits as we convert back and forth from edm4eic to ACTS format. This will require some coordination with the software group.

Resolution comparison at seed and track parameter level

Seed level

Track parameter level



One recent update to truth seeding that should be checked (prior code)

- ACTS requires us to pass the seed local coordinates and the reference point (perigee surface). Previously for the truth seeding, we defined the reference point to be the particle's generation vertex and set the local coordinates to zero.

```
Acts::BoundVector  params;
params(Acts::eBoundLoc0)    = 0.0 * mm ;  // cylinder radius
params(Acts::eBoundLoc1)    = 0.0 * mm ;  // cylinder length
params(Acts::eBoundPhi)     = phi;
params(Acts::eBoundTheta)   = theta;
params(Acts::eBoundQOverP)  = charge / (pinit * GeV);
params(Acts::eBoundTime)    = part->getTime() * ns;

//// Construct a perigee surface as the target surface
auto pSurface = Acts::Surface::makeShared<Acts::PerigeeSurface>(
        Acts::Vector3{part->getVertex().x * mm, part->getVertex().y * mm, part->getVertex().z * mm});

auto result = new eicrecon::TrackParameters({pSurface, params, charge, cov});
```

All this is done in the track seeding code. Without smearing, this should give the ideal track seed for a given particle.

One recent update to truth seeding that should be checked (current code)

- Now, we set the reference point to (0,0,0) and use the particle's generation vertex as the local coordinates.

```
auto v = mcparticle.getVertex();
track_parameter.setLoc({static_cast<float>(std::hypot(v.x, v.y)), static_cast<float>(v.z)}); // 2d location on surface [mm]
```

In track seeding code. The Perigee surface cannot be saved into the *edm4eic::TrackParameters* data container.

```
// Construct a perigee surface as the target surface
auto pSurface = Acts::Surface::makeShared<const Acts::PerigeeSurface>(Acts::Vector3(0,0,0));
```

In CKF factory which is used for both truth and real seeded tracking. The reference point is set to the origin. Note that for the real seeds, we fit 3 points and find the point of closest approach in the (x,y) plane, and extract the z value when r=0 from a linear fit. So, using the origin as the reference point makes sense.

One recent update to truth seeding that should be checked

- For particles with a generation vertex on the z-axis, the results should be equivalent for the prior and current versions.
- However, for non-zero (x,y) generated vertex coordinates, note that the old version gives a specific point in space, while the new version gives a specific z-value but only specifics a circle in the (x,y) plane.
- So, for secondary particles where the particle's generation vertex is far from the z-axis, there may be differences between the old and new versions based on how the CKF treats the input.

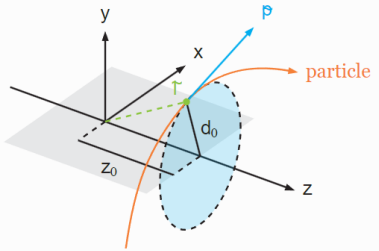
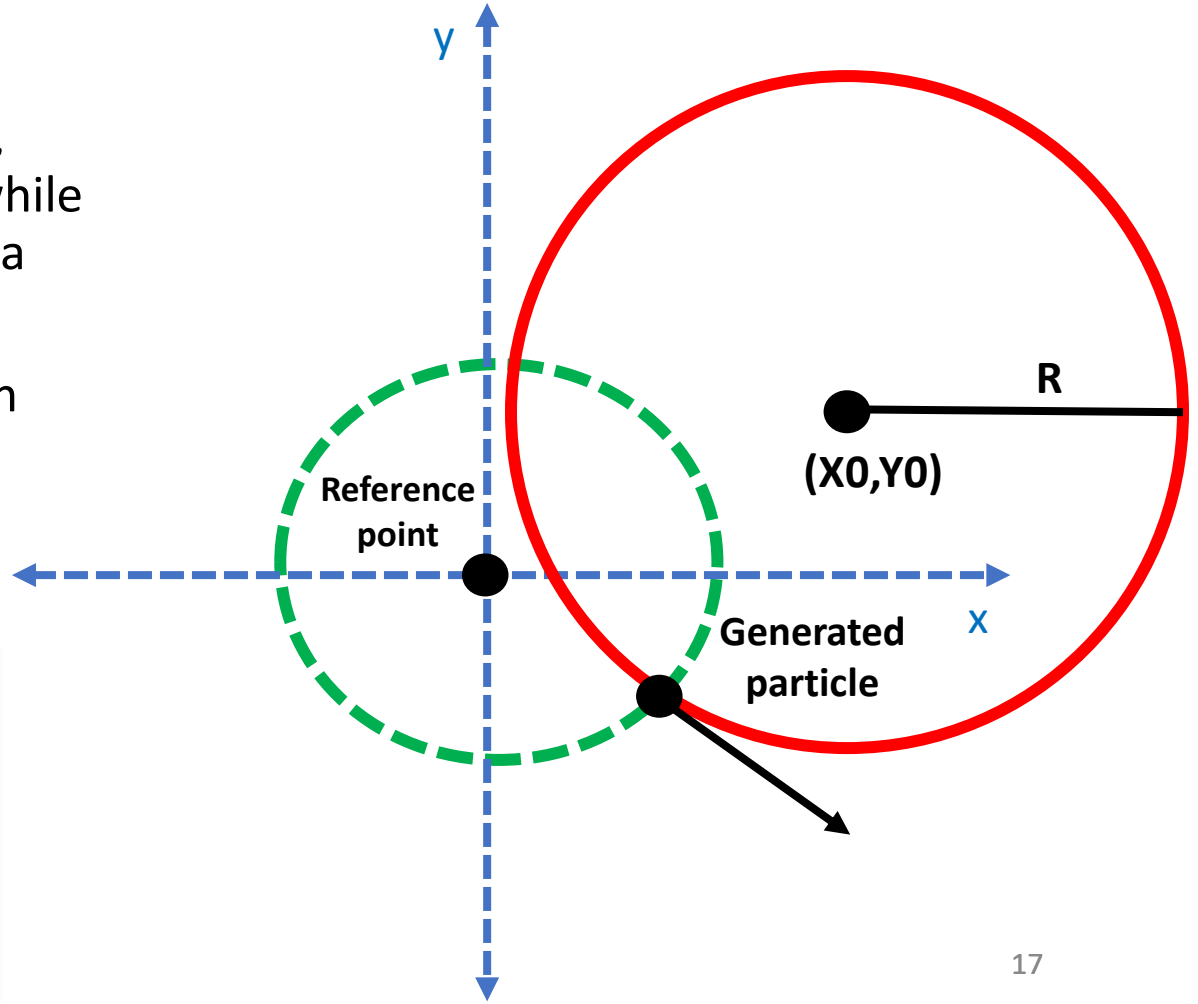


Fig. 4 *Illustration of the perigee parametrization which uses the point of closest approach relative to a reference point. The impact parameter d_0, the position l and the momentum vector $\vec{p}$ are shown.*



Summary

- We have discussed the current tracking-related output that is saved to the standard EICRecon output ROOT file.
- We are currently trying to understand a few issues with the track QA information, as well as add some additional associations to the hit containers.
- For the truth-seeding, we should check if our current implementation works well for particles generated away from the beamline.

Backup

Seed multiplicity – why do we see many events with 3 seeds?

If we have a particle at mid-rapidity which hits layers L0, L1, L2, L3, and L4, then we can make the following combinations:

1. L0,L1,L2
2. L0,L2,L3
3. L0,L3,L4
4. ✖ L0,L1,L3
5. ✖ L0,L1,L4
6. ✖ L0,L2,L4
7. ✖ L1,L2,L3
8. ✖ L1,L2,L4
9. ✖ L1,L3,L4
10. ✖ L2,L3,L4

ACTS seed finder and filter parameters

Parameter	Description	My New Default
bFieldInZ	z component of magnetic field	1.7 T
rMax	Maximum r value to look for seeds	440 mm
rMin	Minimum r value to look for seeds	33 mm
zMin	Minimum z value to look for seeds	-1500 mm
zMax	Maximum z value to look for seeds	1700 mm
beamPosX	Beam offset in x	0
beamPosY	Beam offset in y	0
deltaRMinTopSP	Min distance in r between middle and top SP in one seed	**10 mm**
deltaRMinBottomSP	Min distance in r between middle and bottom SP in one seed	**10 mm**
deltaRMaxTopSP	Max distance in r between middle and top SP in one seed	**200 mm**
deltaRMaxBottomSP	Max distance in r between middle and top SP in one seed	**200 mm**
collisionRegionMin	Min z for primary vertex	-250 mm
collisionRegionMax	Max z for primary vertex	250 mm
cotThetaMax	Cotangent of max theta angle	**27.29**
minPt	Min transverse momentum	100 MeV/cotThetaMax
maxSeedsPerSpM	Max number of seeds a single middle space point can belong to - 1	0
sigmaScattering	How many standard devs of scattering angles to consider	5
radLengthPerSeed	Average radiation lengths of material on the length of a seed	0.1
impactMax	Max transverse PCA allowed	3 mm
rMinMiddle	Min R for middle space point	20 mm
rMaxMiddle	Max R for middle space point	400 mm
bFieldMin	min B field	0.1

Issue with total chi-square

```
root [17] events->Scan("CentralTrackSeedingResults.theta:CentralCKFSeededTrackParameters.theta:theta_rec:CentralCKFSeededTrajectories.n
Measurements:CentralCKFSeededTrajectories.chi2")
************************************************************************************************
*    Row * Instance * CentralTr * CentralCK * theta_rec * CentralCK * CentralCK *
************************************************************************************************
*        0 *        0 * 0.1937180 * 0.1945763 * 0.1945763 *        4 * 2.4666304 *
*        0 *        1 * 0.1940771 * 0.1945463 * 0.1945463 *        4 * 2.4390687 *
*        0 *        2 * 0.1938095 * 0.1947055 * 0.1947055 *        4 * 2.9926469 *
*        1 *        0 * 2.9997701 * 2.9994049 * 2.9994049 *        2 * 0.4067698 *
*        1 *        1 * 2.9998300 * 2.9996504 * 2.9996504 *        2 * 0.0245701 *
*        2 *        0 * 0.1104461 * 0.1103764 * 0.1103764 *        4 * 6.1610455 *
*        2 *        1 * 0.1103586 * 0.1102905 * 0.1102905 *        4 * 5.3399877 *
*        3 *        0 * 0.0850231 * 0.0851216 * 0.0851216 *        4 * 8.0655946 *
*        3 *        1 * 0.0850503 * 0.0851085 * 0.0851085 *        4 * 9.4841909 *
*        4 *        0 * 0.2157898 * 0.2161549 * 0.2161550 *        4 * 5.5669741 *
*        4 *        1 * 0.2160625 * 0.2161960 * 0.2161960 *        4 * 4.2218980 *
*        4 *        2 * 0.2157084 * 0.2162159 * 0.2162160 *        4 * 3.5914709 *
*        5 *        0 * 2.3530783 * 2.3530213 * 2.3530213 *        3 * 2.5779833 *
*        6 *        0 * 0.1159824 * 0.1192474 * 0.1192474 *        2 * 0.6670103 *
*        6 *        1 * 0.1127062 * 0.1198665 * 0.1198665 *        2 * 0.7781642 *
*        7 *        0 * 2.6202621 * 2.6202397 * 2.6202397 *        7 * 13.872473 *
*        7 *        1 * 2.6200094 * 2.6202220 * 2.6202221 *        7 * 16.563919 *
*        7 *        2 * 2.6201334 * 2.6202380 * 2.6202380 *        7 * 13.850541 *
```

Event number

Seed theta

Track parameters theta

Reconstructed Charged particles theta

Trajectory nMeasurements

Total chi-square

One strange thing is that even though the tracks all have the about same parameters as the generated particles and the same number of measurement, there chi-suare can differ substantially.

Issue with total chi-square

- A guess for why we may see this chi-square difference is that our initial covariance matrix has too small uncertainties.
- We can adjust this and check the effect.

```
  edm4eic::TrackParameters *params = new edm4eic::TrackParameters{
    -1, // type --> seed(-1)
    {(float)localpos(0), (float)localpos(1)}, // 2d location on surface
    {0.1,0.1}, //covariance of location
    theta, //theta [rad]
    (float)phi, // phi [rad]
    qOverP, // Q/p [e/GeV]
    {0.05,0.05,0.05}, // covariance on theta/phi/q/p
    10, // time in ns
    0.1, // error on time
    (float)charge // charge
  };

  trackparams.push_back(params);
}
```

TrackSeeding.cc file