



https://cern.ch/acts

Acts Concept, Status & Plans



A. Salzburger (CERN)



https://gitlab.cern.ch/acts/acts-framework/merge_requests/193

Chapter One Yesterday & Today

Motivation ATLAS Tracking SW

Common set of Tools and interfaces



Detector specific extension

InnerDetector			Calorimeter	Mu	MuonSpectrometer			
EDM (InDetRecEvent)	Geometry (InDetTrackingGeom etry)	Tools (InDetRecTools)	Geometry (CaloTrackingGeometry)	EDM (MuonRecEvent)	Geometry (MuonTrackingGeom etry)	Tools (MuonRecTools)		

Fast track simulation extension

ISF_Fatras						
EDM	Geometry	Tools				
(ISF_FatrasEvent)	(ISF_FatrasDetDescrTools)	(ISF_FatrasRecoTools)				

Evolution From ATLAS to Acts

Review

- code usage, code quality, memory usage and execution speed
- check for readiness for concurrent code execution

Update, documentation, integration & testing

- update to C++17 standard,
- simplify, documentation
- Integration in Acts
- Unit tests and regression tests against ATLAS code



Review ATLAS Tracking SW

ATLAS Tracking modules

Tracking								
Geometry (TrkDetDescr)	Event Data Model (TrkEvent)	Extrapolation (TrkExtrapolation)	Fitting (TrkFitter)	Calibration, general (TrkTools)	Alignment (TrkAlignment)			

- Type safety throughout the code
- Top level agnostic code for EDM, tools and conditions
- Fast simulation capability
- Highly configurable
- Partly dynamic size EDM
- Deep level of virtual interfacing
- □ Lazily initialised (mostly removed in LS-1)

Review ATLAS Tracking SW

ATLAS Tracking modules



V	Type safety throughout the code			
Ø	Top level agnostic code for EDM, tools and conditions			
Ø	Fast simulation capability]	
	Highly configurable			
	Partly dynamic size EDM			
	Deep level of virtual interfacing			
	Lazily initialised (mostly removed in LS-1)			
		Acts		

Code from ATLAS to Acts

ATLAS Tracking modules



Renaming from ATLAS to Acts

ATLAS Tracking modules to Acts modules



Chapter Two design choices & modules

Design choices

Compiling vs. Interfacing

- Eigen is a header-only library
- shifted some of the lifting from interfaces to compile-time checks *Interface is checked via C++ concept classes & type trait asserts*
- time critical components are resolved compile-time full stack compiles in a few minutes not moved are single-call modules, e.g. geometry building

template <typename T> using step_size_t = decltype(std::declval<T>().stepSize);

// clang-format off

// clang-format off

```
template <typename S, typename state = typename S::State>
  struct StepperConcept {
    constexpr static bool state_exists = exists<state_t, S>;
    static_assert(state_exists, "State type not found");
    constexpr static bool jacobian_exists = exists<jacobian_t, S>;
    static_assert(jacobian_exists, "Jacobian type not found");
    constexpr static bool covariance_exists = exists<covariance_t, S>;
    static_assert(covariance_exists, "Covariance type not found");
    constexpr static bool bound_state_exists = exists<bound_state_t, S>;
    static_assert(bound_state_exists, "BoundState type not found");
```

Some level of dispatching will be necessary - Pre-compiled modules ready for usage

Design & development choices



- Configuration is done by a nested **Config** struct
- Re-entrance is done by a nested **State** struct
- Runtime options are done by a **Options** struct

```
namespace Acts {
 /// doxygen documentation
  class WorkHorse {
   /// @struct Config of this Horse
    struct Config {
       int horseNumber = 0; ///< the passed path so far</pre>
    };
    /// @struct Cache for the WorkHorse
    struct State {
       float accumulatedPath = 0.; ///< the passed path so far</pre>
    };
    /// @struct Cache for the WorkHorse
    struct Options {
       bool runBackwards = true; ///< switch horse direction per run</pre>
    };
    /// method to make the horse run
    /// @param hState - cache tracker for this horse
    /// @param coords - place where the horse should run to
    /// @return a result, horse may drop dead if max path is reached
    const RunResult run(State& hState, const Vector3D& coords, const Options& opts) const;
};
```

convention



Tracking



- Proxy mechanism for GeoModel sensitive elements (allows for alignment following)
- ☑ TrackingGeometry with intrinsic navigation
- Detector agnostic geometry description
- ☑ Surface based description for Propagation
- Surface and Volume based material description
- Distinction between free and non-free surfaces (unclear ownership)

Proxy mechanism

Geometry binding via DetectorElementBase



class MyDetectorElement {
 /// @copydoc DetectorElementBase::asscociatedSurface
 const PlaneSurface& associatedSurface() const;
 };



Geometry data locality

Geometry building ATLAS:



Tests showed that many cache misses in transform() lookup

- as surfaces are built as proxies and forward the positioning information the transform() call points to different memory
- In a multi-threaded application this is particularly tricky (see later)

Acts Surface classes



Surface classes are largely transcribed from ATLAS SW

- code simplified
- memory structure updated

Distinction between free and non-free surfaces



Geometry memory management

<pre>namespace Acts {</pre>	Acts	C++ Utilities library Dyn	<pre>mamic memory management std::enable_shared_from_this</pre>			
	7,613	std::enable_shared_from_this				
<pre>class DetectorElementBase;</pre>		Defined in header <memory></memory>				
class SurfaceBounds;		<pre>template< class T > class enable_shared_from_this; (since C++11)</pre>				
<pre>class ISurfaceMaterial; class Laver;</pre>		<pre>std::enable_shared_fr safely generate additional</pre>	<pre>std::enable_shared_from_this allows an object t that is currently managed by a std::shared_ptr named pt to</pre>			
class TrackingVolume:		Publicly inheriting from st	at states per instances per, per, p_{12} , that all share ownership of t with per			
ctuss mackingvotume,		shared_from_this. If an	n object t of type T is managed by a std::shared_ptr <t> named pt, then calling</t>			
/// Aclass Surface		T::shared_from_this wi	vill return a new <pre>std::shared_ptr<t></t></pre> that shares ownership of t with pt.			
///						
/// Obrief Abstract Base Class fo	r tracking surfaces	Member functio	ons			
/// @DITEL ADSTLACT DASE CLASS TO	I LIACKING SUITACES	(constructor)	constructs an enable_shared_from_this object (protected member function)			
/// The Surface class builds the	core of the Acts Tracking Geom	(destructor)	destroys an enable_shared_from_this object (protected member function)			
<pre>/// All other geometrical objects /// are built from it.</pre>	are either extending the surf	operator=	(protected member function)			
///		<pre>shared_from_this</pre>	returns a <pre>shared_ptr</pre> which shares ownership of *this <pre>(public member function)</pre>			
/// Surfaces are either owned by I	Detector elements or the Track	<pre>weak_from_this(C++17)</pre>	returns the weak_ptr which shares ownership of *this (public member function)			
/// IN WHICH Case they are not co	pieu within the uata model obj					
class Surface , public virtual Co	amatryObject					
public std::enable	e shared from this <surface> {</surface>		C++11 feature allowing			
public:						
/// @enum SurfaceType			shared ptr access from			
///						
<pre>/// This enumerator simplifies</pre>	the persistency & calculations	,	plain pointers, solved			
<pre>/// by saving a dynamic_cast, e</pre>	.g. for persistency	-	acometry/exect dete			
enum SurfaceType {			geometry/event data			
Cone = 0 ,			binding			
Cylinder = 1 ,			binding			
Disc = 2,						
Perigee = 3,						
Plane = 4,			Access for calculations via			
Straw = 5,			const objects			
Curvilinear = 6,			Const objecta			
Other = 7			In order to ontimise access			
};						
			speed			

TrackingGeometry

Most concepts from ATLAS TrackingGeometry adopted in ACTS

- Volumes with static layer configuration
- Volumes with dense material - Volumes with floating objects Solenoid Toroid Toroid Inner Detector Calorimeter Not fully implemented yet Toroid Muon System

ATLAS ID setup with very similar navigation support for a STEP extension (see later) and new cell navigation

Geometry Customise





Geometry Customise





Geometry Material description

ACTS comes with a built-in material description

- material on static layer configuration
- material with dense material

Automated procedure to map material from simulation input



Material map to DD4HEP surfaces



Material maps

Are stored in root or json format



Material Customize



Running a dedicated Geant4 job records G4 material information



Material Customize



Running a dedicated Geant4 job records G4 material information

ROOT File (TGeo description) -> GDML



Event Data Model



Review ATLAS Tracking Linear Algebra library

Initial ATLAS (Tracking) code was based on CLHEP

- change during LS1 to Eigen after extensive tests

Achieved speed-up w.r.t. CLHEP in 5x5 matrix multiplication testbed



Starting from LS1 EDM

- Eigen dependency fully extended to geometry model as well
- ATLAS GeoModel based on Eigen to be integrated easily

Projects using Eigen

Feel free to add yourself! If you don't have access to the wiki or if you are not sure about the relevance of your project, ask at the #Mailing list.

Extensions, numerical computation

- Google's Ceres 🗗 solver is a portable C++ library that allows for modeling and solving large complicated nonlinear least squares problems.

Review ATLAS Tracking Event Data Model (1)

TrackParameters



Review ATLAS Tracking Event Data Model (2)

TrackParameters

LS1 cleanup reduced number of lines massively - keeping the same functionality & type safety

					25.000	Contributed Lines of Code	
Package	C++	C/C++	C++	C/C++	35000		
		Header		Header	30000		
TrkParameterBase	63	561	11	214	25000		
TrkParameters	1715	602	0	52	20000	cleanup	
TrkNeutralParameters	1425	663	0	48	SP 15000		Ln/
ExtendedTrkParameterBase	0	295	0	0	<u>ا</u> المراجع	Run-1	
ExtendedTrkParameters	1412	514	0	0	10000		
ExtendedTrkNeutralParameters	1416	514	0	0	5000		<u></u>
					0		- <u>1</u>
Total	6031	3149	11	266	-5000		
					AUG 2009 FED 2010 AUG 201	60 Feb 2012 AUG 2012 Feb 2012 AUG 2012 Feb 2013 AUG	2013 Feb 201

Event Data Model Track Parameters



If you want the 5-parameterisation, you just write
auto pars = perigee.parameters().block<5,1>();

Event Data Model Track Parameters

TrackParameters

SingleBoundParameters

SingleCurvilinearParameters

Extension for Multi Component representation

- avoid copying of Extrapolator (as done in ATLAS) and Fitter infrastructure for multi-variant fitters (MultiTrackFitter, GSF) *act as single track parameters in navigation, but will be propagated as multiple components in between*

MultivariantTrackParameters

MultivariantBoundParameters

MultivariantCurvilinearParameters

Event Data Model Measurements & calibration





Event Data Model Customise





Interface needed by acts tools:



Track and TrackState description



Extrapolation



Acts

Magnetic field field caching

Magnetic field caching found to reduce CPU time in

- Simulation (up to 20%)
- Reconstruction (around few %)

Tracking locks the field cell in the

magnetic field service

- not ideal for concurrent usage

(field cell is not exposed to propagator, needs to be secured within FieldSvc)

Acts field service provides a field cell to be cached by the caller (see propagation) - C++ concept for field cell



Field look up in Runge-Kutta integration
Magnetic field

Tests using different magnetic field inputs within Acts

ATLAS map (currently converted from ATLAS root file),
direct use of ATLAS MagneticFieldSvc possible (template parameter)
FCC-hh field map



ATLAS magnetic field map in ACTS

Propagation | Extrapolation

ATLAS Tracking SW :

- distinction between propagation (transport)

and extrapolation (transport, navigation & material effects integration)



Propagation | Extrapolation ATLAS

Extrapolator Tool interface was expanded to do more & more

- hole search, jacobian collection, material collection
- fast simulation, track-to-cal, track-through-calo, etc. ...

Line	85	/** N 0) />Neutral parameters method	171	
1 /	86	- returns a ParametersBase object as well, 0 if the extrapolation did not succeed	172	25b 257 /++ C 7) doConfigured Aldfool extrapolation methods//b of 8 71/+/
2 /	IExtrapolator.h. (c) ATLAS Detector software	-/	173	237 /····································
2/	88		174	258 virtual startpart-const rater assort track bareness parm
4	89	virtual const NeutralParameters* extrapolate(const NeutralParameters& parameters,	175	209 Contraction of a party provide the second party provide the second party party provide the second party
5 8	fodef TREEXINTERFACES IEXTRAPOLATOR H 90	const Surface4 sf,	176 /** S 8) Strategy Pattern extrapolation method	200 Boundarytherk beherk
6 1	efine TREEXINTERFACES IEXTRAPOLATOR H 91	PropDirection dir=anyDirection,	177 - Extrapolation using specific intermediate surfaces and energy loss effects to be accounted for at	201 262
7	92	BoundaryCheck bcheck = true) const = 0;	178 each surface as specified by the corresponding MaterialEffectsCaTrack. The last propagation ends	202 Part (c) eBrothesis part (c)
8/	(Gaudi 93		170 at the last surface given, applying the corresponding NaterialEffectsOnTrack to the track parameters	2003 264 MaterialDodateMode mature
0 #	94	/** [TrackParameters] */	190 before returning.	20%
10 /	95		100 e//	203 265 /t+ C 81 shownfigured &leftnol extrapolation methods/bb of 8 811+/
11	Include "TrikEventPrisitives/Proplimetion.b"		101 virtual cost TrackParameterst extrapolate(cost IPropagator), prop.	200 vital stduation TrackBaraneters, one Lawry extraplateTolertStation
12 8	nclude "TrisventPrimitives/ParticlePunthesis.b" 97	/** S 1) db>Strategy Pattern extrapolation method	102 cosst TrackBarandersk parn.	207 const TrackParameterse parm.
12	include "TribleterialOutras/AntarialPfeataDutrack.b"	- returns the TrackParameters at the Destination Surface (if extrapolation succeeds),	103 const stdumenter print	200 Proplinection direatylinection.
14	include "TriExTIT (1//MaterialDudateMode.b"	0 if extrapolation to destination surface does not succeed */	104 const Triu Tackinshi und tuni	205 BoundaryCheck beheck = true,
19 1	include induction international sciences by 100	virtual const TrackParameters* extrapolate(const IPropagators prop.	185 Const Intrinckingvilles (VI)	270 ParticleProthesis particlemion.
15	Include TRANUCLE/ACTIONATION 100	const TrackBaranetersk parm.	186 Propurection air,	2/1 State and Understand Picture (Molece) constants
16	Include Takobrieces/PoundaryLinetx.n 101	const Surfacek sf.	187 Particlesypthesis particlesypthesis particlesypthesis and a	272
10	Include Tarking the factor of	PropDirection direction.	188 Raterialopatende Ratepide=adixise) const = 0;	274 /** C 9) <>>Configured AlgTool extrapolation method
10	include "Tr/Daymeter/Daymeter/Daymeter/Days	RoundaryCheck behave = true.		275 virtual const TrackParameters* extrapolateToWolume(const TrackParameters* parm,
19 .	Include Introductobase/ratameterboase.ii	Destication for a still amion	190 /** S 9) <pre>costrategy Pattern extrapolation method</pre>	275 const Trk::TrackindVolume& vol.
20 /	ANDU 103	The second point of the second s	191 - extrapolation to the next active layer, based on the extrapolation to the next layer	277 PropDirection directory
21	and use a second and a second se	material updatement in the material cost = 0;	192 and layer identification	278 Particle#ypothesis particle#pion) const=0;
22	107. 107. 107. 107. 107. 107. 107. 107.	/tt 5 21 dbaStrategy Battern extrapolation methods/ba	193 7	279
23	108 109 100 100 100 100 100 100 100 100 100		194 virtual std::pair <const const="" layer*="" trackparameters*,=""> extrapolateToNextStation(const IPropagator& prop,</const>	280 /** C 10)
24 8	109	- arrowing a version of its according to the tracking detector elements he is between and the TrackParameters at the destination Surface (if final actuation in the	195	281 - Extrapolate to a destination surface, while collecting all the material layers in between.
25	110	As in severe un se incorporameters at the destination surface (if final extrapolation succeds), A if the astropolation to the destination surface devices in the surface (if a surface several surface), and the surface devices in the surface devices of t	196	282 */
26	111	v it the extrapolation to the destination surface does not succeed*/	197	283 virtual const std::vector <const trackstateonsurface*="">* extrapolateM(const TrackParameters) parameters,</const>
27 h	112	virtual const Stdiivector <const extrapolatestepwise(<="" td="" trackfarameters**=""><td>198</td><td>284 const Surfaces sf,</td></const>	198	284 const Surfaces sf,
28	113	const IPropagators prop,	199	285 PropDirection dir,
29	static const interlater interlater ('instrapolator', 1, 0);	const TrackParameters& parm,	200	286 BoundaryCheck beheck,
30	115	const Surfaces sf,	201 /** S 10) <>>Strategy Pattern extrapolation method	287 Particle#ppothesis particle=pion,
31	116	PropDirection dirmanyDirection,	202 - extrapolation to a volume boundary of an arbitrary tracking volume (not necessarily part of a tracking geometric structure)	7 288 Trk: (ExtrapolationCache* cache = 0) const = 0;
32	Transportation 117	BoundaryCheck beheck = true,	203 */	289
33	Interrace class for the exchapolation Algrool, it inherits from JALgrool 118 118	ParticleHypothesis particlempion) const = 0;	204 virtual const TrackParameters* extrapolateToVolume(const IPropagators prop,	290 /** C 11)
34	Detailed information about private memoers and memoer functions can be found in the actual implementation c. 119		205 const TrackParameters# parm,	291 - Extrapolate to a destination surface, while collecting all the material layers and transport jacobians in between.
35	Extrapolator which inherits from this one. 120	/** S 3) Strategy Pattern extrapolation method :	206 const Trk::TrackingVolume& vol,	292 */
36	121	- searches the closest TrackParameters of the Track to the destination Surface	207 PropDirection dirmanyDirection,	293 virtual const std::vector <const trackparameters*="">* extrapolateM(const TrackParameters& parameters,</const>
37	The Parametersbase and NeutralParameters methods only exist in conrigured mode, 122	- returns the TrackParameters at the Destination Surface (if extrapolation succeeds),	208 ParticleHypothesis particle=pion) const=0;	294 const Surface4 sf,
38	the strategy pattern is not really useful for straight line propagation. 123	0 if extrapolation to destination surface does not succeed */	209	295 PropDirection dir,
39	124	virtual const TrackParameters* extrapolate(const IPropagators prop,	210 /** C 1)	296 BoundaryCheck beheck,
40	(author Andreas.Salzburger(cern.ch 125	const Tracké trk,	211 virtual const TrackParameters* extrapolate(const TrackParametersš parm,	297 std::vector <materialeffectsomtrack> &material,</materialeffectsomtrack>
41	*/ 126	const Surfaces sf,	212 const Surfaces sf,	298 std::vector <trk::transportjacobian*> &jacs,</trk::transportjacobian*>
42	127	PropDirection dirmanyDirection,	213 PropDirection dirmanyDirection,	299 ParticleHypothesis particle=pion,
43	class TPropagator;	BoundaryCheck beheck = true,	214 BoundaryCheck beheck = true,	300 Trk::ExtrapolationCache* cache = 0; const = 0;
44	class INavigator;	ParticleRypothesis particle=pion,	215 ParticleBypothesis particlempion,	301
45	class IMaterialUpdater;	MaterialUpdateMode maturmode=addNoise) const = 0;	216 MaterialDodateMode matuppodemaddNoise.	302
46	class Track;		217 Trk::ExtrapolationCache* cache = 0 ; const = 0;	303 virtual const Trk::TrackParameters* extrapolateWithPathLimit(
47	class Surface;		218	304 const Trk::TrackParameters& parm,
48	class TrackingVolume; 132	/** 5 4) <>>trategy Pattern extrapolation method:	210 /** C 2) <>Configured AlgTool extrapolation method	305 double& pathLim,
49	class TrackingGeometry; 133	- direct extrapolation to the destination surface, no material effects	220 virtual cost std:/vector <cost trackparameters*="">* extrapolateStepwise(cost TrackParametersE nam.</cost>	306 Trk::PropDirection dir,
50	class TransportJacobian;	or intermediate steps are taken into account	221 Const Surface4 of	307 Trk: Particle Bypothesis particle,
51	class TrackStateOnSurface;	*/	222 Problemation	308 std::vector <const trk::trackparameters*="">*& parmOnSf,</const>
52	class Layer;	"/ minters and Trank Trank Trank and and the Trank to the Trank and the	ZZZ Proprietan ar any process,	309 std::vector <const trk::trackstateonsurface+="">+4 material,</const>
53	class Volume;	onet readaute provide a second	223 Boundary-DOCK DEROCK = LTUB, 234 BarticleBurtharis public Longt -	on 310 const Trk::TrackingVolume* boundaryVol=0,
54	138	const lackreisstore pain,	227 Particlesypotnesis particlespion) const =	311 MaterialUpdateMode matupmod = Trk::addNoise) const = 0;
55	class IExtrapolator : virtual public IAlgTool (CONSE SUFICION SI,	222 227 (tt C 2) children attranslation methods (b) of S 2111/	312
56	public:	Propurection direnyurection,	220 /	313 /** extrapolation method collecting intersections with subdetector boundaries and active volumes/layers.
57	141	Boundaryuneck Deneck = true,	22/ ***********************************	314 A primitive identification is provided - to be replaced with appropriate identifiar, and possibly merged
58	/** Virtual destructor */ 142	ParticleHypothesis particle=pion) const = 0;	220 Come Suffaces 52,	315 with TrackParameters. Material collection in option. Destination (subdetector boundary) : geoID (exit)
59	virtual -IExtrapolator()() 143		223 Proportection of Apple Section,	316 */
60	144	i a aj -wrasawayy attain axtrapolation methods (br)	230 Boarding pulses A second pulses and a second pulses	317 virtual const std::vector< std::pair< const Trk::TrackParameters*, int > >* extrapolate(
61	/** AlgTool interface methods */	- weaker availed the given tracking volume (boundaryvol),	231 rescalenypotensis participepton,	318 const TrkiiTrackParametersé parm,
62	static const interfaceID() { return IID_IExtrapolator; }	AA MAR AR YAYMA GAR (BIGIGHOOD BUTIACE FOR GESCHARION 18 USED	and Trivity and a state of the	319 Trk: PropDirection dir,
63	147	and a second and a second second many second s		320 Trk: ParticleRypothesis particle,
64	/** [XAOD Interface] */	virtual const sturryector <const extrapolatesiindiy(const="" prop,<="" propagators="" td="" trackyarameters**=""><td>237 /tt C 41 ch0mfigured AlcTeol extrapolation methods/bb of 5 4111/</td><td>321 std:vector<const trackstateonsurface*="" trk:="">*6 material,</const></td></const>	237 /tt C 41 ch0mfigured AlcTeol extrapolation methods/bb of 5 4111/	321 std:vector <const trackstateonsurface*="" trk:="">*6 material,</const>
65	149	const trackwarameters& parm,	233 / · · · · · / · · · · · · · · · · · ·	322 int destination = 3) const = 0;
66	/** xAOD 0) neutral xAOD particle */	PropDirection direanyDirection,	23b Vituel come inackrameters* extrapolateurectly(const inackrameters# parm,	
67	virtual const NeutralParameters* extrapolate(const xAOD::NeutralParticle& xnParticle, 151	BoundaryCheck bcheck = true,	23/ CODE SUITACE BI,	324 /** Meturn the irackingGeometry used by the Extrapolator (forwards information from Navigator) */
68	const Surface& sf, 152	ParticleBypothesis particlempion,	238 Propurection dirempurection,	325 virtual const flacking/sconetry() const = 0;
69	PropDirection dir=anyDirection, 153	<pre>const Volume* boundaryVol=0;</pre>	Z39 Boundarycheck Deneck = true,	326
	BoundaryCheck bcheck = true) const = 0; 154		240 Particlesypotnesis particlempion) const = 0;	327 /** Valuation ADLION 200 Can be implemented optionally, outside access to interval validation stars
70	155	/** S 6) AD>Strategy Pattern extrapolation method1	241	320 tem be advantation officially, Outside access to internal valuation steps
70 71		 extrapolation to the next active layer, based on the extrapolation to the next layer 	242 /** C 5) c>conjured AlgTool extrapolation method	329
70 71 72	156	and layer identification	243 virtual const stdiivector <const trackparameters*="">* extrapolateBlindly(const TrackParameters& parm,</const>	330 ***********************************
70 71 72 73	156 157		244 PropDirection dirmanyDirection,	331
70 71 72 73 74	155 /** xADO 0) neutral xADD particle */ 150	*/		111 (II ADDRES THE SUDWIDDRAGETOR TO THE GIVEN
70 71 72 73 74 75	/** sACO 0) seutral sACO particle */ virtual cosst TrachFurgmeters* extrapolate(cosst bACD::TrachFarticle& particleBase, 155	*/ virtual std::pair <const layer*="" trackparameters*,const=""> extrapolateToNextActiveLayer(</const>	245 BoundaryCheck beheck = true,	332 /** Access the subropagator to the given volume/
70 71 72 73 74 75 76	/** xACO 0) neutral xACO particle */ virtual const TrackParameters* extrapolate(const xACO:;TrackParticleS particleBase, const Sufface sf, 159	*/ virtual std::pair≺const TrackParameters*,const Layer*> extrapolateToNextActiveLayer{ const TPropagators prop,	245 BoundaryCheck beheck = true, 246 ParticleBypothesis particlempion,	332 /** Access the subrogagator to the given Volume*/ 333 virtual const IPropagator* subPropagator(const TrackingVolume\$ tvol) const = 0;
70 71 72 73 74 75 76 77	/** sAOD 0) neutral sAOD particle */ virtual const TrackParameters* extrapolate(const tAOD::TrackParticle& particleBase, const Earlange& df, PropUncetion dirangOirection, 160	<pre>*/ virtual std:spair<comst layer*="" trackfarameters*.comst=""> extrapolateToNextActiveLayer{ comst IPropagatort prop,</comst></pre>	245 BoundaryCheck beheek = trum, 246 ParticleBypothesis particlempion, 247 cosst Voisme boundaryChe(s) cosst = 0;	332 /** access the supercognitor to the given volume*/ 333 v*trail cosst lPropagator* subPropagator(cosst TrackingVolumes tvol) cosst = 0; 334 335 v
70 71 72 73 74 75 76 77 78	/** sado 0) smutral add0 particle */ 155 virtual cost TrackBarsmeters* entrapolate(cost sdd0);TrackBarticles particleBase, 163 PropDirection dirangeTerrotion, 161 Boundarytheck becket = trme, 162	*/ virtual std::pair4const TrackParameters*.const Layer*> extrapolateToNextActiveLayer(const IPropagatori prop,	245 Boundarythack behack = trum, 246 Particle#pytomis particlemptom, 247 const Volume* boundaryVol=0) const = 0; 248 Const Volume* boundaryVol=0) const = 0;	332 /** access the superpayator to the given volume*/ 333 virtual const Tropagator* subropagator(const TrackingVolumes tvol) const = 0; 334 335 335 1;
70 71 72 73 74 75 76 77 78 79	/** sAOD 0) neutral sAOD particle */ virtual const TrackParameters* extrapolate(const tAOD::TrackParticles particleBase, const Earlack Sf, PropEirention dirangOirection, BoundaryTheck beheck = tume, ParticleBayDetheck particleBayConst, 160	<pre>*/ virtual std::pair4coast TrackParameters*,coast Layer*> extrapolate708extActiveLayer(coast IPropagator& prop, coast IPropagator& prop,</pre>	245 BoundaryCheck beheck = true, 246 ParticleMypothesis particlempion, 247 Const Volume' boundaryVol=0) const = 0; 248 /** C 6) dxConfigured AlgTool extrapolation method√bb of g 6):*/	332 /** access the superpagator to the given volume*/ 333 virtual cost thropagator* subbropagator(cost TrackingVolumes tvol) cost = 0; 334
70 71 72 73 74 75 76 77 78 79 80	/** akon 0) mentral akon particle */ virtual const TracMurameters* extrapolac(const akon, TracMurticles particleBase, 159 virtual const TracMurameters* extrapolac(const Surfaces af, 159 PropDirection dirangOterntion, 160 BoundaryCheck becket = true, 162 ParticleMyothesis particlempica, 163 Natorialization antignode-macBoline) const = 0; 163	*/ virtual std::pair <comst layer*="" trackparameters*.comst=""> extrapolateToRextActiveLayer(comst IPropagator& prop,</comst>	245 Boundarytheck beheck = true, 246 ParticleStypticstis particlemytics, 247 cosst Volume* boundaryVol=0) cosst = 0; 248 /** C 6) 'ds/dsrfigured Algfhol extrapolation methods/be of \$ 6 j; */ 250 virtual sidi splrecosst TrackFarmeters*, cosst Layer> extrapolatoDestActiveLayer(332 /** access the incorposator to the given volume*/ 333 virtual cost Trachingfolders trachingfolders tvol; cost = 0; 334 335 335 j; 336 iaise cost Trichinfrachinesters* Trikinfictuation (extrapolaterithinthinit)
70 71 72 73 74 75 76 77 78 79 80 81	/** sAOD 0) neutral sAOD particle */ 155 virtual const TrackParameters* extrapolate(const %AOD::TrackParticles particleBase, 159 const Sardaces df, 160 PropDirection dirangOirection, 160 BoundaryThock beheck = tume, 162 ParticleBystenis particlePine, 162 NaterialtydateMode natupmoderaddBoise) const = 0; 164	<pre>*/ virtual std::pair<comst layer*="" trackparameters*,comst=""> extrapolate708extActiveLayer(comst IPropagator& prop, comst IPropagator& prop,</comst></pre>	245 BoundaryCheck behock = true, 246 Particle/spine, 247 const Volume' boundaryVole) const = 0; 248 /** C &) <0>Const/jurned AlgTool extrapolation method of g &):*/ 249 virtual std:spine-const trackfarameters*,const Layer* extrapolationStathLetivelayer(251 const trackfarameters*,const Layer* extrapolationStathLetivelayer(333 /** access the interpopulation to the given volume// 334 /** access the interpopulation is addropopulation (const Trackingfolumes two)) const = 0; 335 /* 336 /* 337 islice const Trivitanteters* 338 /* 339 /* 330 /* 331 /* 332 /* 333 (const Trivitanteters* /*parm*/, 334 (const Trivitanteters* /*parm*/,
70 71 72 73 74 75 76 77 78 79 80 81 82	/** add0 0) newtral add0 particle */ 155 virtual const TrachFuriameters* extrapolate(const add0);07mcMPerticle particleBase, 159 virtual const TrachFuriameters* extrapolate(const Berlines f. 150 PropDirection diramyDirection, 160 BoundaryCheck blocks = trum, 161 ParticleByothesis particlepice, 162 AttarialUpdateHode matugenderaddNoise) const = 0; 164 155 165	<pre>*/ virtual stds:pair4const TrackParameters*,const Layer*> extrapolato7050xxtActivuLayer{ const IPropagatort prop, const IPropagatort prop,</pre>	245 Boundarytheck thether,* trum, 246 ParticleSystemis particlemyton, 247 cosst Volume* boundaryVole() cosst = 0; 248 /** C 6; dwConfigured AlgOrol extrapolation method*/b> of 8 6):*/ 250 virtual std:spl=*cosst TrackFarameters*,cosst Layer*> extrapolationDestActiveLayer(251 cosst TrackFarameters* parm, 252 PropDirection dir*anyDirection,	332 /** access the superpayater to the given volume// 333 virtual cost Tropagator* subropagator(cost Trackingfolumes tvol) cost = 0; 334 illis 335 illis 336 illis 337 ialise cost Trk::TrackFarmeters* Trk::Estrapolator::estrapolator::estrapolator:// 338 deskim //public/// 339 deskim //public///
70 71 72 73 74 75 76 77 78 79 80 81 82 83	<pre>/** sAOD 0) neutral sAOD particle */ Virtual const TrackParameters* extrapolate(const tAOD::TrackParticles particleBase,</pre>	<pre>*/ virtual std::pair<comst layer*="" trackparameters*,comst=""> extrapolate708extActiveLayer(</comst></pre>	245 Boundarytheck behock = trum, 246 ParticleStyptick behock = trum, 247 coset Volume* boundarytole, 248 /** C 6) doldowflymmed Algtbol extrapolation methody/be of 2 8(jrt) 249 virtual stdispatie 240 virtual stdispatie 241 coset TrackParameters*, coset Layer* extrapolate/doinet.Methodset. 251 virtual stdispatie 252 PropDirection direasplicetion,	333 /** access the interpopulate to the given volume// 334 /** access the interpopulate to the given volume// 335 /* 336 /* 336 /* 337 islise cosst Trinification restrapolater(istrapolater) 338 /* 339 dealise //pathia/*/ 330 /* 330 /* 330 /*
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84	/** aNGO 0) neutral sNGO particle */ virtual const TrachFarameters* extrapolate(const SOD);TrachFarticleS particleBase, roppliarcian dis-splits(const Sories, 159 roppliarcian dis-splits(const Sories, 160 Bordiaryteck tokock = trum, ParticleMyoothesis particlempion, 162 MaterialDyintedtod matugende=addMoise) const = 0; /** (NeutralParameters) */ 167 /** (NeutralParameters) */ 167	<pre>*/ */ */ */ */ */ */ */ */ */ */ */ */ *</pre>	245 Boundarytheck behock = trum, 246 ParticleSystemis particlemyton, 247 cosst Volume* boundaryVole() cosst = 0; 248 /** C 6; d=Constigured AlgStol extrapolation method//b> of \$ 6; !>*/ 250 vistual stdi spli*cosst TrackParameters*,cosst Layer*> extrapolationStdexthetiveLayer(251 cosst TrackParameters* parm, 252 PropDirection dir*argOirection, 253 254	332 /** access the subcregalater to the given volume// 333 virtual cost Tropagator* subrogagator(cost Trackingfolumes tvol) cost = 0; 334 i 335 i 336 i 337 i 338 i 339 i 330 i 331 i 332 i 333 i 334 i 335 i 336 i 337 i 338 i 339 i 330 i 330 i 331 i 332 i 333 i 334 i 335 i 336 i 337 i 338 i 339 i 340 i 341 i

- ~ 360 lines interface in IExtrapolator.h
- ~ 4700 lines of code in Extrapolator.cxx

Extrapolator to Propagator



Propagator in Acts



Stepper Timing | Examples

Stepper comparison: Constant Field



Extrapolator to Propagator



Propagation | Extrapolation



ATLAS: at every step new dynamic memory allocation (TrackParameters)



Propagation Time component

Internal representation expanded from 7x7 description to 8x8

- full time covariance transport developed (and numerically tested)
- positive impact on execution speed



Comparison in EigenStepperBenchmark between the master branch at the end of the 15.05.2019 (commit: 05fe134a) in black and this branch with PropagatorOptions::propagateTime = true; in red. This is just the result, but I have no clue how it is possible



Propagation Interface

/// @brief Propagate track parameters		
///		
/// This function performs the propagation of the track parameters using the		
<pre>/// internal stepper implementation, until at least one abort condition is</pre>		
/// fulfilled or the maximum number of steps/path length provided in the		
<pre>/// propagation options is reached.</pre>		
///		
<pre>/// @tparam parameters_t Type of initial track parameters to propagate</pre>		
<pre>/// @tparam action_list_t Type list of actions, type ActionList<></pre>		
/// @tparam aborter_list_t Type list of abort conditions, type AbortList<>		
<pre>/// @tparam propagator_options_t Type of the propagator options</pre>		
///		
<pre>/// @param [in] start initial track parameters to propagate</pre>		
<pre>/// @param [in] options Propagation options, type Options<,></pre>		
///		
/// @return Propagation result containing the propagation status, final		
<pre>/// track parameters, and output of actions (if they produce any)</pre>		
///		
<pre>template <typename action_list_t,<="" parameters_t,="" pre="" typename=""></typename></pre>		
<pre>typename aborter_list_t,</pre>		
<pre>template <typename, typename=""> class propagator_options_t,</typename,></pre>		
<pre>typename path_aborter_t = detail::PathLimitReached></pre>	defines resul	It
Result <action_list_t_result_t< <<="" th=""><th></th><th></th></action_list_t_result_t<>		
<pre>typename stepper_t::template return_parameter_type<parameters_t>,</parameters_t></pre>	Input	
action_list_t>>		
propagate(
<pre>const parameters_t& start,</pre>		
<pre>const propagator_options_t<action_list_t, aborter_list_t="">& options) const;</action_list_t,></pre>		
	J Options	

Propagation Customize





Propagation Timing



	16:36:21	Sequencer	INF0	Added algorithm 'PropagationAlgorithm'
	16:36:21	Sequencer	INF0	Added writer 'RootPropagationStepsWriter'
	16:36:21	Sequencer	INF0	Processing events [0, 100)
	16:36:21	Sequencer	INFO	Starting event loop with 1 threads
	16:36:21	Sequencer	INF0	Ø services
	16:36:21	Sequencer	INF0	0 context decorators
	16:36:21	Sequencer	INF0	0 readers
	16:36:21	Sequencer	INF0	1 algorithms
	16:36:21	Sequencer	INF0	1 writers
	16:36:21	Sequencer	INF0	finished event 0
	[]			
	16:36:24	Sequencer	INF0	finished event 99
	16:36:25	Sequencer	INF0	Processed 100 events in 3.953269 s (wall clock)
	16:36:25	Sequencer	INF0	Average time per event: 39.055690 ms/event
Ē.				

1000 test propagations/event

full predictive navigation (material integration, hole search) propagation through 2 T field

Track Fitting



- ☑ Top level detector agnostic geometry description
- Calibration structure (PrepRawData -> RIO_OnTrack)
- <u>
 Measurement sorting</u>
- Many different interfaces





KalmanFitter Prototype status





Pattern recognition





Pattern recognition Strategy

Transcribe ATLAS pattern recognition code into ACTS code pattern

```
namespace Acts {
  /// doxygen documentation
 template <typename iterator_type>
  class PatternReco {
   /// @struct Config for To
    struct Cache {
       Store someCacheStore; ///< necessary cache for pattern reco</pre>
    };
    /// method to make the horse run
    /// @param pCache - cache for this pattern
    /// @param pConfig - configuration for this pattern
    /// @param coords - place where the horse should run to
    /// @return a result, horse may drop dead if max path is reached
      const Result seedFinder(
         const Config& pConfig,
         Cache& pCache,
         iterator_type start
         iterator_type end) const;
 };
```

Pattern recognition Strategy (1)



Overlapping regions ? Result merging ? Data pre-dividing ? Thread pools ?



Overlapping regions ? Result merging ? Data pre-dividing ? Thread pools ?

Seeding Status

Seeding was first module integrated from ATLAS pattern recognition

- all 'magic numbers' documented
- ATLAS specifics have been encapsulated

Tested and runs in AthenaMT

- gives comparable results to ATLAS seeding
- drop-in replacement not
 straight forward, as seeding
 in ATLAS is part of a higher level
 algoritm

```
template <typename SpacePoint>
float ATLASCuts<SpacePoint>::seedWeight(
   const InternalSpacePoint<SpacePoint>& bottom,
   const InternalSpacePoint<SpacePoint>&,
   const InternalSpacePoint<SpacePoint>& top) const {
  float weight = 0;
 if (bottom.radius() > 150) {
    weight = 400;
 if (top.radius() < 150) {</pre>
   weight = 200;
  return weight;
template <typename SpacePoint>
bool ATLASCuts<SpacePoint>::singleSeedCut(
    float weight, const InternalSpacePoint<SpacePoint>& b,
   const InternalSpacePoint<SpacePoint>&,
   const InternalSpacePoint<SpacePoint>&) const {
  return !(b.radius() > 150. && weight < 380.);</pre>
```

Define a seeding module in ATLAS

- Would also allow ML seeders to be deployed

CKF Prototype

Dedicated development meetings: https://indico.cem.ch/event/877617/

Efficiency/fake rate





Vertexing New to the family



Vertexing Status

acts > 🐽 acts-core > Merge Requests > 1535

Merged

Iterative vertex finder

Opened 2 months ago by 🚯 Bastian Schlag

Implements the Iterative Vertex Finder together with the ZScanVertexFinder us 4D vertexing, ready to be merged after 1576 (merged).	sed as the vertex seeding algorithm. Already adapted to
Edited 1 week ago by Bastian Schlag	
Request to merge iterative_vertex_finder to master	\$ -
acts > 🔞 acts-core > Merge Requests > !566	
Open Opened 1 month ago by 🚯 Bastian Schlag	Edit Close merge request -
WIP: Multi adaptive vertex fitter	
Implements the multi adaptive vertex fitter, depends on some features from	!535 (merged)
Edited 1 month ago by Bastian Schlag	
Request to merge MultiAdaptiveVertexF Co into master The source branch is 62 commits behind the target branch	Open in Web IDE Check out branch Q - Now iterating on the design and optimising

Edit

Report abuse

Full vertexing suite available now

z-Finder



Original module from ATLAS Atlantis

- fast (ms) z-finder on space points
- Code is fully implemented but not yet imported into acts



Chapter Three Configuration & integration

Status Binding to detector software & framework

Acts designed to have minimal overhead when being integrated in detector software

Algebra library is Eigen but dependencies are minimal

- may change to a template implementation (if beneficial)

No dependency on Identifier

- Detector calibration is resolved in detector geometry

Screen logging can be replaced by Id pre-loading

- needs a simple struct on the detector framework side that provides a logger() method.
- tested with different loggers:
 - Acts logger in acts-framework
 - Gaudi logger within FCCSW

Status Binding to framework configuration

ACTS tools have a nested configuration struct:

```
namespace Acts {
   /// doxygen documentation
   class WorkHorse {
        /// @struct Config for To
        struct Config {
           float coatColor; ///< configure the coat color
           float maxPath; ///< set the max path this horse can run
      };
   };
}</pre>
```

These structs are then configured by the detector framework,

e.g. through Gaudi/Athena

```
/// feed from Framework into ACTS configuration
declareProperty("CoatColor", m_cfg.coatColor);
declareProperty("MaxPath", m_cfg.maxPath);
```

tested with Gaudi for FCCSW & AthenaMT

Configuration Strategy

Nested configuration struct by convention

```
namespace Acts {
   /// doxygen documentation
   class SomeComponent {
      /// @struct Config for this Component
      struct Config {
         bool run_faster = false; ///< configuration flag
      };
      /// Constructor with config object
      SomeComponent(Config& cfg);
   };
}</pre>
```

Inside the framework Wrapper

```
#include "ACTS/Package/SomeComponent.hpp"
...
/// create the config sruct
Acts::SomeComponent::Config scConfig;
/// bind to your framework configuration
declareProperty("RunFastVersion", scConfig.run_faster);
Acts::SomeComponent sc(scConfig);
```

Chapter four multi-threading



const-correctness

Remove every use of "mutable" in ACTS !265 · opened 3 days ago by Hadrien Grasland



statelessness engines

- cache visitor pattern for calls that need to run concurrently

```
namespace Acts {
    /// doxygen documentation
    class WorkHorse {
        /// @struct Cache for the WorkHorse
        struct State {
            float accumulatedPath = 0.; ///< the passed path so far
        };
        /// method to make the horse run
        /// @param hState - cache tracker for this horse
        /// @param coords - place where the horse should run to
        /// @return a result, horse may drop dead if max path is reached
        const RunResult run(State& hState, const Vector3D& coords) const;
    };
}</pre>
```

Concurrency Tests

Acts test framework runs with TBB multithreaded mode

- running extrapolations through a test detector
- test programs are run using a single threaded setup vs. multi-threaded event processing
- this consistency check is part of
 - the acts-framework Cl



Intel Xeon e5-2698 v3, 2 sockets 32 Cores, 2 threads per core 64 Processors(cpu's)



ACTS with Context

Introduced context objects in **acts-core** & testes in **acts-framework**

```
- NOMEN est OMEN
/// Aggregated information to run one algorithm over one event.
struct AlgorithmContext
{
    size_t
    size_t
    whiteBoard&
    Acts::GeometryContext
    Acts::MagneticFieldContext
    Acts::CalibrationContext
};
```

While they are untouched in **acts-core** and simply defined as

#pragma once

```
/// Set the identifier PLUGIN
#ifdef ACTS_CORE_GEOMETRYCONTEXT_PLUGIN
#include ACTS_CORE_GEOMETRYCONTEXT_PLUGIN
#else
#include <any>
namespace Acts {
using GeometryContext = std::any;
using DefaultGeometryContext = GeometryContext;
} // namespace Acts
#endif
```

Contextual Detector The "clean" solution


Parallelism testbed

Test with different alignment every single event

<pre>salzburg\$./ACTFWAlignablePropagationExample -n10prop-ntests 1000bf-values 0 0 2output-root 1 12:49:10 Sequencer INF0 Added context decorator GeometryRotationDecorator 12:49:10 Sequencer INF0 Added service RandomNumbersSvc 12:49:10 Sequencer INF0 Appended algorithm PropagationAlgorithm 12:49:11 Sequencer INF0 Added writer RootPropagationStepsWriter 12:49:11 Sequencer INF0 Starting event loop for 12:49:11 Sequencer INF0 0 readers 12:49:11 Sequencer INF0 0 readers 12:49:11 Sequencer INF0 0 readers</pre>	
12:49:10SequencerINFOAdded context decorator GeometryRotationDecorator12:49:10SequencerINFOAdded service RandomNumbersSvc12:49:10SequencerINFOAppended algorithm PropagationAlgorithm12:49:11SequencerINFOAdded writer RootPropagationStepsWriter12:49:11SequencerINFOStarting event loop for12:49:11SequencerINFO1 services12:49:11SequencerINFO0 readers12:49:11SequencerINFO0 readers12:49:11SequencerINFO1 veriters	
12:49:10SequencerINFOAdded service RandomNumbersSvc12:49:10SequencerINFOAppended algorithm PropagationAlgorithm12:49:11SequencerINFOAdded writer RootPropagationStepsWriter12:49:11SequencerINFOStarting event loop for12:49:11SequencerINFO1 services12:49:11SequencerINFO0 readers12:49:11SequencerINFO0 readers12:40:11SequencerINFO1 veriters	
12:49:10SequencerINFOAppended algorithm PropagationAlgorithm12:49:11SequencerINFOAdded writer RootPropagationStepsWriter12:49:11SequencerINFOStarting event loop for12:49:11SequencerINFO1 services12:49:11SequencerINFO0 readers12:49:11SequencerINFO1 veritors	
12:49:11SequencerINFOAdded writer RootPropagationStepsWriter12:49:11SequencerINFOStarting event loop for12:49:11SequencerINFO1 services12:49:11SequencerINFO0 readers12:40:11SequencerINFO1 writers	
12:49:11SequencerINFOStarting event loop for12:49:11SequencerINFO1 services12:49:11SequencerINFO0 readers12:40:11SequencerINFO1 writers	
12:49:11SequencerINFO1 services12:49:11SequencerINFO0 readers12:40:11SequencerINFO1 writers	
12:49:11 Sequencer INFO 0 readers	
12.40.11 Converse TNEO 1. $vritere$	
IZ:49:II SEQUENCEN INFO I WITCEIS	
12:49:11 Sequencer INFO 1 algorithms	
12:49:11 Sequencer INFO Run the event loop	
12:49:11 Sequencer INFO start event 0 12:51:19 Sequencer INFO start event 0	
12:49:12 Sequencer INFO event 0 done 12:51:19 Sequencer INFO start event 5	
12:49:12 Sequencer INFO start event 1 12:51:19 Sequencer INFO start event 8	
12:49:13 Sequencer INFO event 1 done 12:51:19 Sequencer INFO start event 7	
12:49:13 Sequencer INFO start event 2 12:51:20 Sequencer INFO event 7 done	
12:49:14 Sequencer INFO event 2 done 12:51:20 Sequencer INFO start event 2	
12:49:14 Sequencer INFO start event 3 12:51:21 Sequencer INFO event 8 done	
12:49:15 Sequencer INFO event 3 done 12:51:21 Sequencer INFO start event 9	
12:49:15 Sequencer INFO start event 4 12:51:21 Sequencer INFO event 5 done	
12:49:16 Sequencer INFO event 4 done 12:51:21 Sequencer INFO start event 6	
12:49:16 Sequencer INFO start event 5 12:51:21 Sequencer INFO event 0 done	
12:49:17 Sequencer INFO event 5 done 12:51:21 Sequencer INFO start event 3	
12:49:17 Sequencer INFO start event 6 12:51:22 Sequencer INFO event 2 done	
12:49:19 Sequencer INFO event 6 done 12:51:22 Sequencer INFO start event 3	
12:49:19 Sequencer INFO start event 7 12:51:23 Sequencer INFO event 9 done	
12:49:19 Sequencer INFO event 7 done 12:51:23 Sequencer INFO start event 4	
12:49:19 Sequencer INFO start event 8 12:51:23 Sequencer INFO event 6 done	
12:49:20 Sequencer INFO event 8 done 12:51:23 Sequencer INFO event 1 done	
12:49:20 Sequencer INFO start event 9 12:51:23 Sequencer INFO event 3 done	
12:49:22 Sequencer INFO event 9 done 12:51:24 Sequencer INFO event 4 done	
12:49:22 Sequencer INFO Running end-of-run hooks of writers and services	

12 seconds

5 seconds

GeometryContext Comparing the two



salzburg\$ export ACTSFW_NUM_THREADS=1
salzburg\$ export ACTSFW_NUM_THREADS=4

GeometryContext In Action



g_y:g_x {event_nr == 0 && layer_id}



MagneticFieldContext In Action

g_y:g_x {event_nr==0}



Detectors & Framework Current support

FCC-hh detector

- via DD4Hep geometry description
- in Gaudi event processing framework

ACTS generic detector (TML)

- via python input
- in acts-framework

ATLAS Pixels and ITK detector

- currently via GDML input
- in acts-framework

Geometry Plugins available

- DD4Hep plugin
- TGeo plugin







TrackML Aftermath

Started to port first TrackML algorithms into acts-framework

Idea is to create a testbed for algorithm development and templating
provide several detectors to test on



A bunch of other detectors:



Examples Detectors in Acts

Contributions and developments

- Contributions mostly from ATLAS so far, but from outside too: Belle-2, FCC-hh
- TrackML challenge phase 2 completed²
- OpenDataDetector for open dataset under development



²Grand Finale in July 2019
³Talk by N. Braun
⁴CERN-THESIS-2019-136, J. Hrdinka
⁵A. Salzburger



OpenData Detector



A brain storming list for SPHE



Geometry & material description

- finish material maps (first fully layer based),
- will try to implement/demonstrate that on Wednesday
- Volume based material for TPC (faster)

Import z-finder into framework

- adapt to sPhenix parameters

Create truth tracking example - by-passing of pattern recognition

Running vertex reconstruction example