Flat(s)	Flat(%)	Sum(%)	Cum(s)	Cum(%)
0	0	0	303.71s	31.91
0	0	0	300.7	31.6
0	0	0	284.07	29.85
0	0	0	278.67	29.28
0	0	0	172.92	18.17
0	0	0	172.92	18.17
0.05	0.0053	0.0053	172.82	18.16
0	0	0.0053	157.16	16.51
0.07	0.0074	0.013	127.63	13.41
0.03	0.0032	0.034	66.76	7.01
26.15	2.75	2.79	54.03	5.68
0	0	2.79	39.76	4.18
0.01	0.0011	2.79	38.59	4.05
0	0	2.79	38.57	4.05
0	0	2.79	38.57	4.05
0	0	2.79	38.31	4.03
0	0	2.79	38.27	4.02
0.15	0.016	2.81	30.91	3.25
0.04	0.0042	2.81	28.42	2.99
0.06	0.0063	2.82	28.24	2.97
0.04	0.0042	2.82	26.59	2.79
0.02	0.0021	2.82	25.17	2.64
20.51	2.16	4.98	24.33	2.56
20.44	2.15	7.13	23.74	2.49
6.92	0.73	7.85	21.11	2.22
0.04	0.0042	7.86	20.55	2.16
0.28	0.029	7.89	20.31	2.13
0.08	0.0084	7.9	18.97	1.99
0.14	0.015	7.91	18.89	1.98
0.03	0.0032	7.91	18.2	1.91
0.03	0.0032	7.92	17.57	1.85
0.08	0.0084	7.92	16.86	1.77
0.09	0.0095	7.93	16.02	1.68
0	0	7.93	15.74	1.65
0	0	7.93	15.5	1.63
0.02	0.0021	7.94	15.18	1.6
0	0	7.94	11.23	1.18
0	0	7.94	11.23	1.18
0	0	7.94	11.21	1.18
0	0	7.94	11.19	1.18
0	0	7.94	11.19	1.18

Function
ActsExamples::TrackFindingAlgorithm::execute
(anonymous namespace)::TrackFinderFunctionImpl::operator()
Acts::CombinatorialKalmanFilter::findTracks
Acts::Propagator::propagate
Acts::ActionList::operator()
Acts::detail::action_list_impl::action
Acts::Propagator::propagate_impl
Acts::Propagator::propagate_impl
Acts::CombinatorialKalmanFilter::Actor::operator()
Acts::Delegate::operator()
Acts::Propagator::propagate_impl
Acts::CombinatorialKalmanFilter::Actor::operator()
Acts::CombinatorialKalmanFilter::Actor::finalize
Acts::Delegate::connect(Acts::GainMatrixSmoother const*)::{lambda(void const*, Acts::ContextType const&, A
Acts::Delegate::connect(Acts::GainMatrixSmoother const*)::{lambda(void const*, Acts::ContextType const&, A
Acts::GainMatrixSmoother::operator()
Acts::GainMatrixSmoother::operator()
Acts::CombinatorialKalmanFilter::Actor::filter
Acts::detail::transportCovarianceToBound
Acts::detail::boundState
Acts::Propagator::propagate_impl
Acts::MultiTrajectory::MultiTrajectory
Acts::GainMatrixSmoother::operator()(Acts::ContextType const&, Acts::MultiTrajectory&, unsigned long, Acts
Acts::(anonymous namespace)::boundToBoundJacobian
Acts::detail_lt::GrowableColumns::GrowableColumns
Acts::CombinatorialKalmanFilter::Actor::filter
Acts::Propagator::propagate_impl
Acts::CombinatorialKalmanFilter::Actor::filter
Acts::Navigator::resolveSurfaces
Acts::StepperExtensionList::finalize
Acts::Navigator::status
Acts::detail::GenericDefaultExtension::finalize
Acts::CombinatorialKalmanFilterResult::CombinatorialKalmanFilterResult
Acts::Propagator::propagate_impl
Acts::detail::transportCovarianceToBound
Acts::CombinatorialKalmanFilter::Actor::filter
template_switch
visit_measurement
template_switch
ActsExamples::TrackFindingAlgorithm::execute
ActsExamples::Trajectories::Trajectories

Flat = time spent at the location itself
Cum = total time spent in location and all of its callees
Sum = total time spent in program so far

File	Line No.
/acts/Examples/Algorithms/TrackFinding/src/TrackFindingAlgorithm.cpp	88
/acts/Examples/Algorithms/TrackFinding/src/TrackFindingAlgorithmFunction.cpp	40
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	1264
/acts/Core/include/Acts/Propagator/Propagator.ipp	139
/acts/Core/include/Acts/Propagator/ActionList.hpp	92 (inline)
/acts/Core/include/Acts/Propagator/detail/action_list_implementation.hpp	73 (inline)
/acts/Core/include/Acts/Propagator/detail/action_list_implementation.hpp	27 (inline)
/acts/Core/include/Acts/Propagator/Propagator.ipp	59
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	354
/acts/Core/include/Acts/Utilities/Delegate.hpp	120 (inline)
/acts/Core/include/Acts/Propagator/Propagator.ipp	44
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	448
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	1076
/acts/Core/include/Acts/Utilities/Delegate.hpp	107
/acts/Core/include/Acts/Utilities/Delegate.hpp	111 (inline)
/acts/Core/src/TrackFitting/GainMatrixSmoother.cpp	38
/acts/Core/include/Acts/EventData/MultiTrajectory.ipp	255 (inline)
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	695
/acts/Core/src/Propagator/detail/CovarianceEngine.cpp	340
/acts/Core/src/Propagator/detail/CovarianceEngine.cpp	275
/acts/Core/include/Acts/Propagator/Propagator.ipp	58
/acts/Core/include/Acts/EventData/MultiTrajectory.hpp	683 (partial-inline)
/acts/Core/src/TrackFitting/GainMatrixSmoother.cpp	60
/acts/Core/src/Propagator/detail/CovarianceEngine.cpp	117
/acts/Core/include/Acts/EventData/MultiTrajectory.hpp	58 (inline)
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	564
/acts/Core/include/Acts/Propagator/Propagator.ipp	64
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	610
/acts/Core/include/Acts/Propagator/Navigator.hpp	1147
/acts/Core/include/Acts/Propagator/StepperExtensionList.hpp	176
/acts/Core/include/Acts/Propagator/Navigator.hpp	341
/acts/Core/include/Acts/Propagator/detail/GenericDefaultExtension.hpp	103 (inline)
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	207 (partial-inline)
/acts/Core/include/Acts/Propagator/Propagator.ipp	25
/acts/Core/src/Propagator/detail/CovarianceEngine.cpp	353
/acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp	591
/acts/Core/include/Acts/Utilities/Helpers.hpp	379 (inline)
/acts/Core/include/Acts/EventData/MeasurementHelpers.hpp	56 (inline)
/acts/Core/include/Acts/Utilities/Helpers.hpp	382 (inline)
/acts/Examples/Algorithms/TrackFinding/src/TrackFindingAlgorithm.cpp	106
/acts/Examples/Framework/include/ActsExamples/EventData/Trajectories.hpp	48

Line
auto results = (*m_cfg.findTracks)(initialParameters, options);
return trackFinder.findTracks(initialParameters, options);
auto result = m_propagator.template propagate(sParameters, propOptions);
auto result = propagate_impl<ResultType>(state);
impl::action(tuple(), state, stepper, result);
action_caller<has_result>::action(this_action, state, stepper, result);
act(state, stepper, result.template get<detail::result_type_t<actor>>());
state.options.actionList(state, m_stepper, result);
auto res = filter(surface, state, stepper, result);
return std::invoke(m_function, m_payload, std::forward<Args>(args)...);
Result<double> res = m_stepper.step(state);
auto res = finalize(state, stepper, result);
auto smoothRes = m_extensions.smoother(state.geoContext, result.fittedStates, lastMeasurementIndex, getD
m_function = [](const void* payload, Args... args) -> return_type { assert(payload != nullptr && "Payload is req
return std::invoke(Callable, concretePayload, std::forward<Args>(args)...);};
trajectory.applyBackwards(prev_ts.previous(), [&prev_ts, &error, &logger](auto ts) {
bool proceed = callable(getTrackState(iendpoint));
auto res = stepper.boundState(state.stepping, *surface);
boundToBoundJacobian(geoContext, freeParameters, boundToFreeJacobian, freeTransportJacobian, freeToPa
transportCovarianceToBound(geoContext, covarianceMatrix, jacobian, transportJacobian, derivatives, boundT
m_navigator.status(state, m_stepper);
class MultiTrajectory {
BoundMatrix G = ts.filteredCovariance() * prev_ts.jacobian().transpose() * prev_ts.predictedCovariance().inve
fullTransportJacobian = freeToBoundJacobian * (FreeMatrix::Identity() + freeToPathDerivatives * freeToPath)
struct GrowableColumns {
stepper.transportCovarianceToBound(state.stepping, *surface);
m_navigator.target(state, m_stepper);
auto procRes = processSelectedTrackStates(state.geoContext, selectedTrackStateRange.first, selectedTrackSta
state.navigation.navSurfaces = navLayer->compatibleSurfaces(state.geoContext, stepper.position(state.steppi
return impl(std::integral_constant<int, nExtensions>{}, impl);
if (resolveSurfaces(state, stepper)) {state.navigation.navigationStage = Stage::surfaceTarget; return;}
return transportMatrix(state, stepper, h, D);
struct CombinatorialKalmanFilterResult {
state.options.actionList(state, m_stepper, result);
reinitializeJacobians(geoContext, freeTransportJacobian, freeToPathDerivatives, boundToFreeJacobian, freePa
createSourceLinkTrackStates(state.geoContext, result, boundState, prevTip, slBegin, slEnd);
return Callable<N>::invoke(std::forward<Args>(args)...);
return template_switch<detail::visit_measurement_callable, 1, eBoundSize>(dim, param, cov, lambda);
return template_switch<Callable, N + 1, NMAX>(v, std::forward<Args>(args)...);
std::move(trackFindingOutput.fittedParameters));
Trajectories(const MultiTrajectory& multiTraj, const std::vector<size_t>& tTips, const IndexedParameters& pa

)ummyLogger());
uired, but not set"); const auto* concretePayload = static_cast<const Type*>(payload); return std::invok

ıthDerivatives, fullTransportJacobian, surface);
oFreeJacobian, parameters, surface);

rse();
* freeTransportJacobian * boundToFreeJacobian;

ıteRange.second, result, isOutlier, prevTipState, nBranchesOnSurface, logger);
ng), stepper.direction(state.stepping), navOpts);

ırameters, surface);

ırameters) : m_multiTrajectory(multiTraj), m_trackTips(tTips), m_trackParameters(parameters) {}

e(Callable, concretePayload, std::forward<Args>(args)...);};