# Track Fitting for the LCD

Norman A. Graf

SLAC

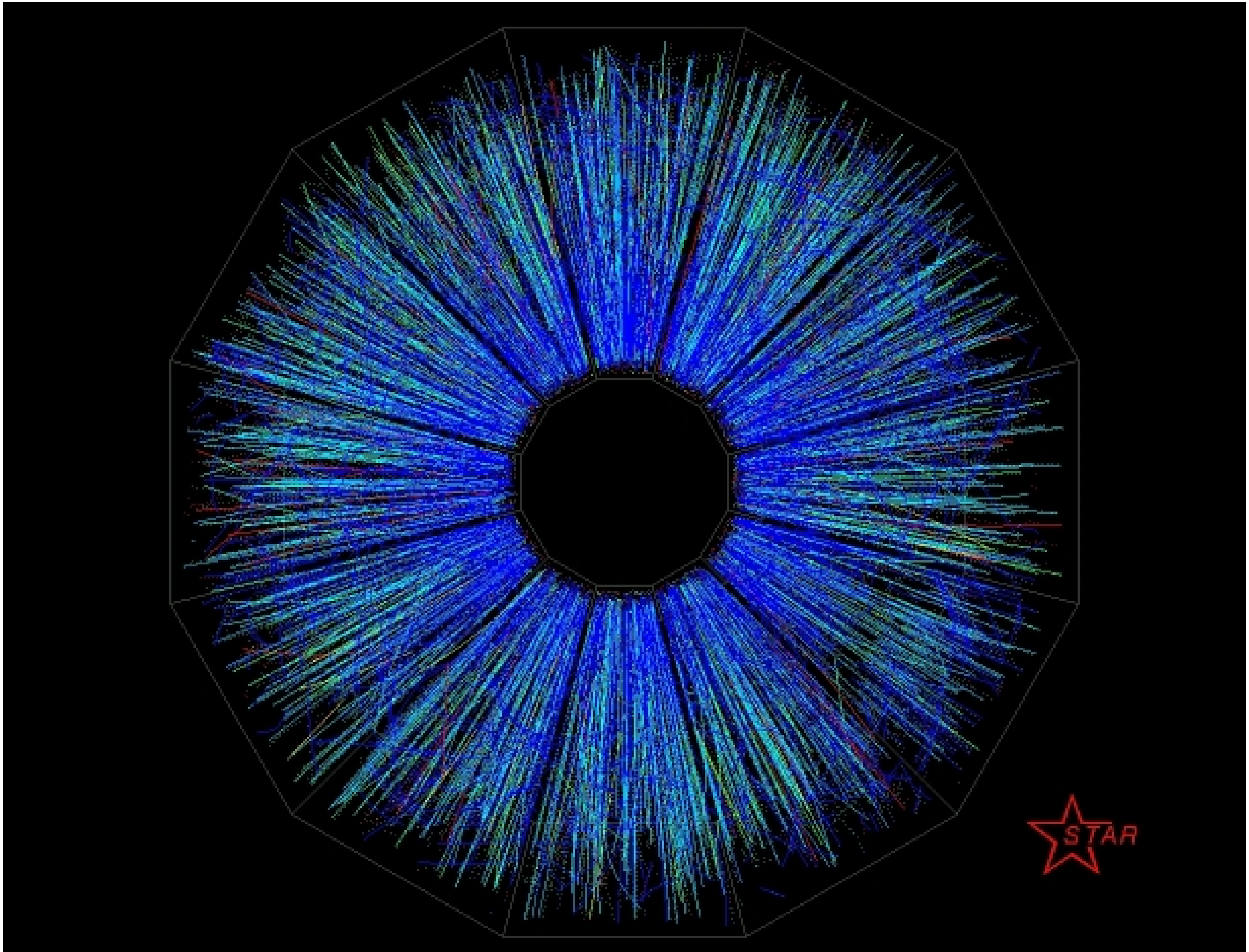
LCWS2000

# Problem Statement

- The tracking system is tasked with finding and reconstructing the trajectories of charged particles with high efficiency and precision.
  - Essential for the stringent vertexing capabilities demanded by flavor tagging.
  - Required if jet energy resolutions are to be improved using the “Energy Flow” algorithm.
  - Z invariant mass reconstruction for Higgs mass via recoil.

# Pattern Recognition

- **Easier at $e^+e^-$ than at hadron colliders:**
  - **intrinsically cleaner event structure,**
  - **reduced multiple interactions,**
  - **low occupancies in tracking detectors,**
  - **tightly constrained interaction point,**
  - **many measurements in most tracker designs,**
  - **3D measurements in most of the proposed detectors.**


STAR

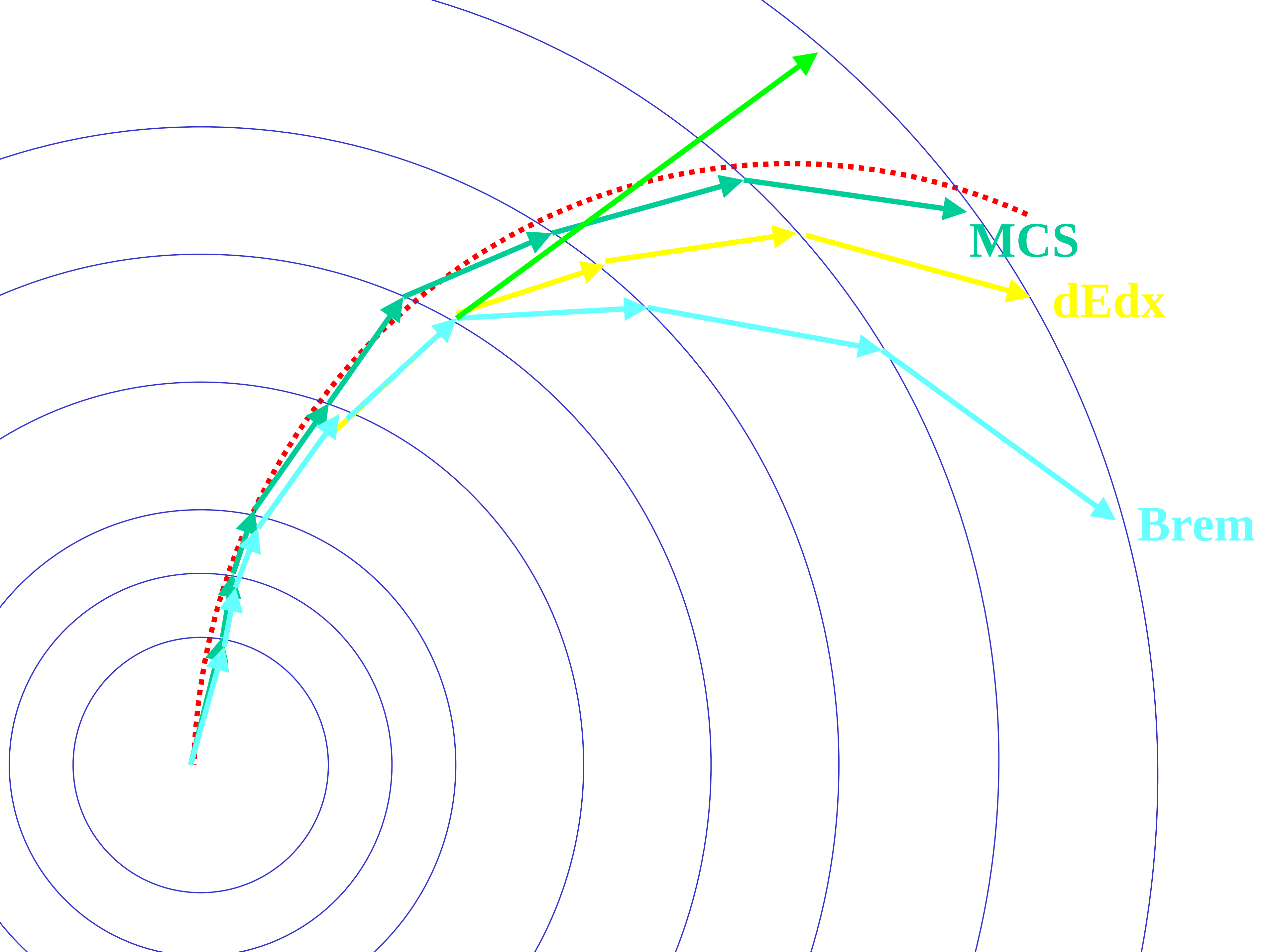
Omega 235.2
Theta 127.3
Phi 208.0

# Track Fitting

- ❖ Need to provide best possible fit to the track parameters and also their uncertainties.
  - Vertex pattern recognition and Least-Squares fitting are very sensitive to track uncertainties.
  - Tight matching of track momentum and direction with calorimeter cluster energy and position necessary for optimal jet energy resolutions.
- ❖Requiring best fit at both ends of track logically leads to a local track fitting strategy.

# Track Definition

- **Six parameters are required to determine a charged particle's path in a magnetic field.**
- **However, knowing these parameters at a single point (*e.g.* the distance of closest approach to the beam,** dca**) is insufficient for precision fits due to material effects (dE/dx, MCS, bremsstrahlung) and field inhomogeneities.**
  - **No global functional form for the fit.**


MCS
dEdx
Brem

# Track Definition

- **We define a track as an ordered list of hits (or misses) at measurement surfaces along with the best fit at that surface** (TrackStates)**.**
- **The track fit consists of five parameters appropriate to the surface plus one parameter which is provided by the constraint that the track lie on the surface.**

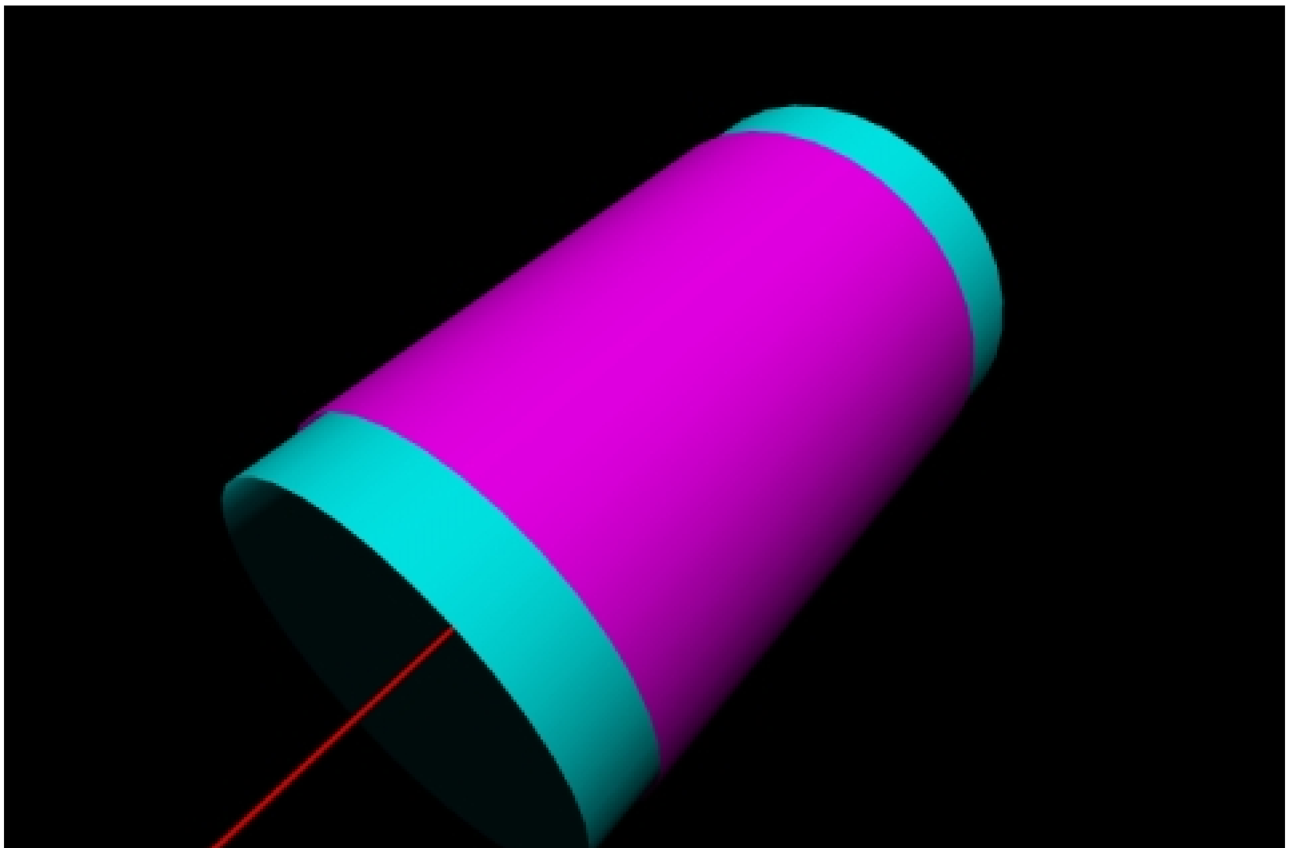
# Surfaces

- Surfaces generally correspond to geometric shapes representing detector devices.
- They provide a basis for tracks, and constrain one of the track parameters.
- The track vector at a surface is expressed in parameters which are “natural” for that surface.

# Cylinder

- Surface defined coaxial with z, therefore specified by a single parameter r.
- Track Parameters: $(\phi, z, \alpha, \tan\lambda, q/p_T)$
- Bounded surface adds $z_{min}$ and $z_{max}$.
- Supports 1D and 2D hits:
  - 1D Axial: $\phi$
  - 1D Stereo: $\phi+\kappa z$
  - 2D Combined: $(\phi, z)$

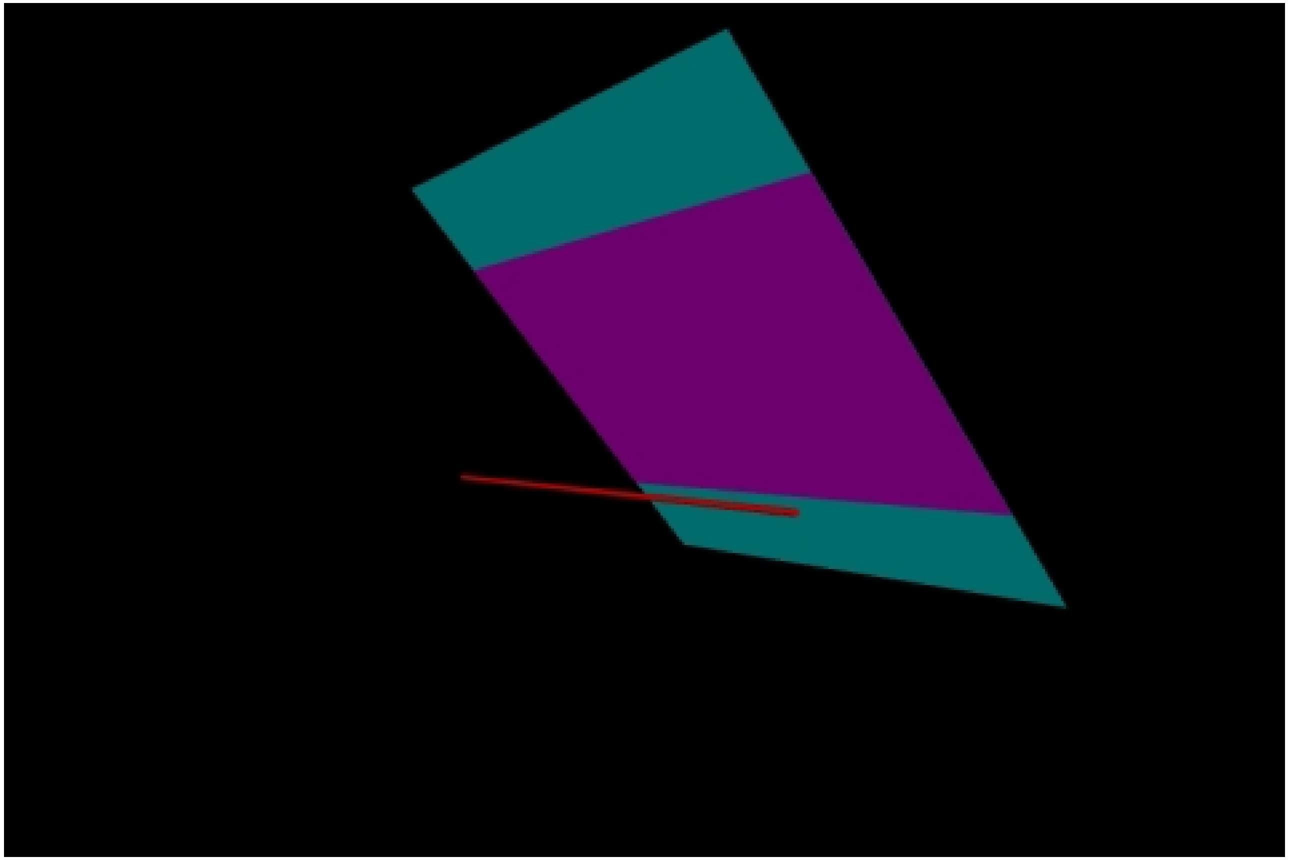
# Cylinder



# XY Plane

- **Surface defined parallel with z, therefore specified by distance u and angle $\phi$.**
- **Track Parameters: (v, z, dv/du, dz/du, q/p)**
- **Bounded surface adds rectangular boundaries.**
- **Supports 1D and 2D hits:**
  - **1D Stereo: $w_v*v + w_z*z$**
  - **2D Combined: (v, z)**

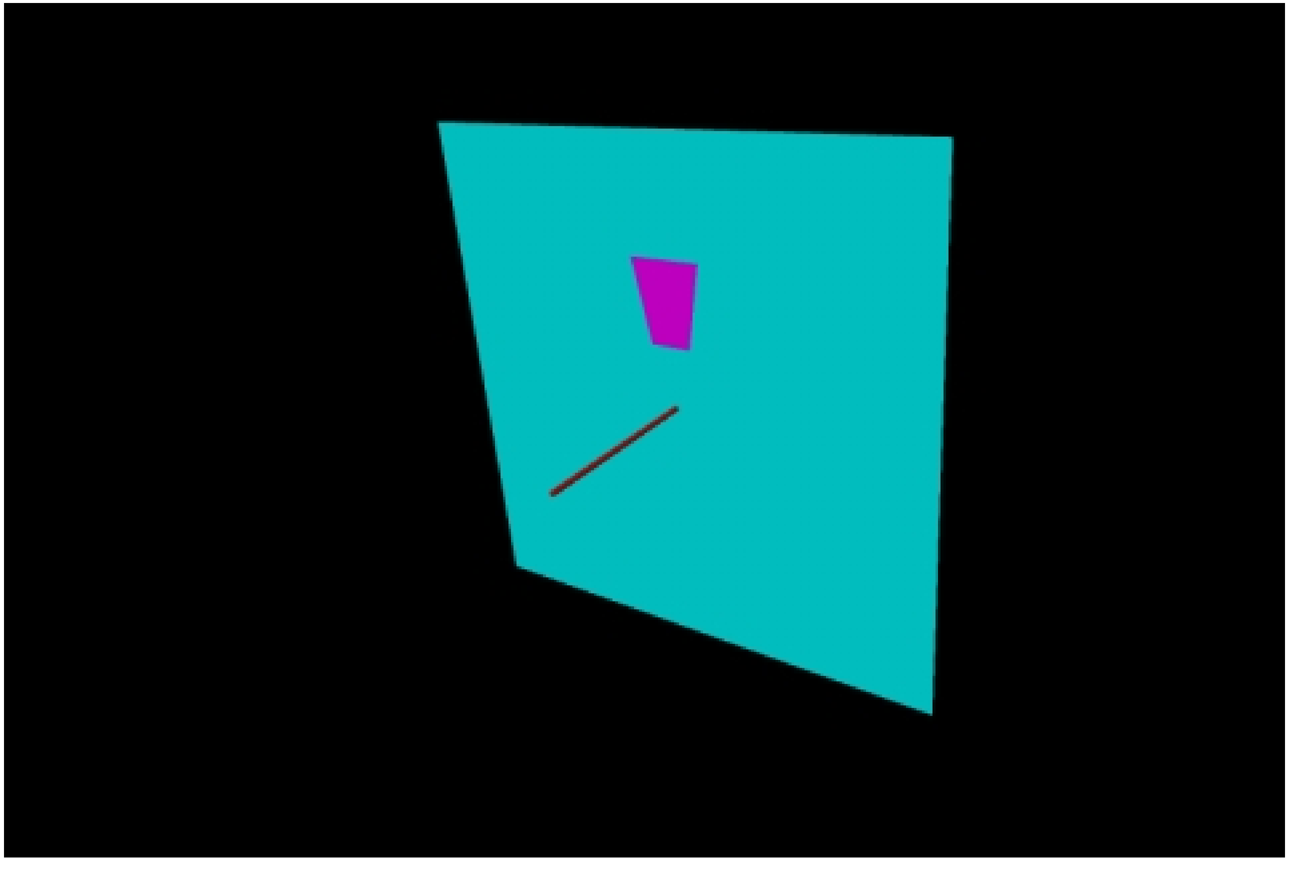
# XY Plane



# Z Plane

- Surface defined perpendicular to z, therefore specified by single parameter z.
- Track Parameters: (x, y, dx/dz, dy/dz, q/p)
- Bounded surface adds polygonal boundaries.
- Supports 1D and 2D hits:
  - 1D Stereo: $w_x*x + w_y*y$
  - 2D Combined: (x,y)

# Z Plane



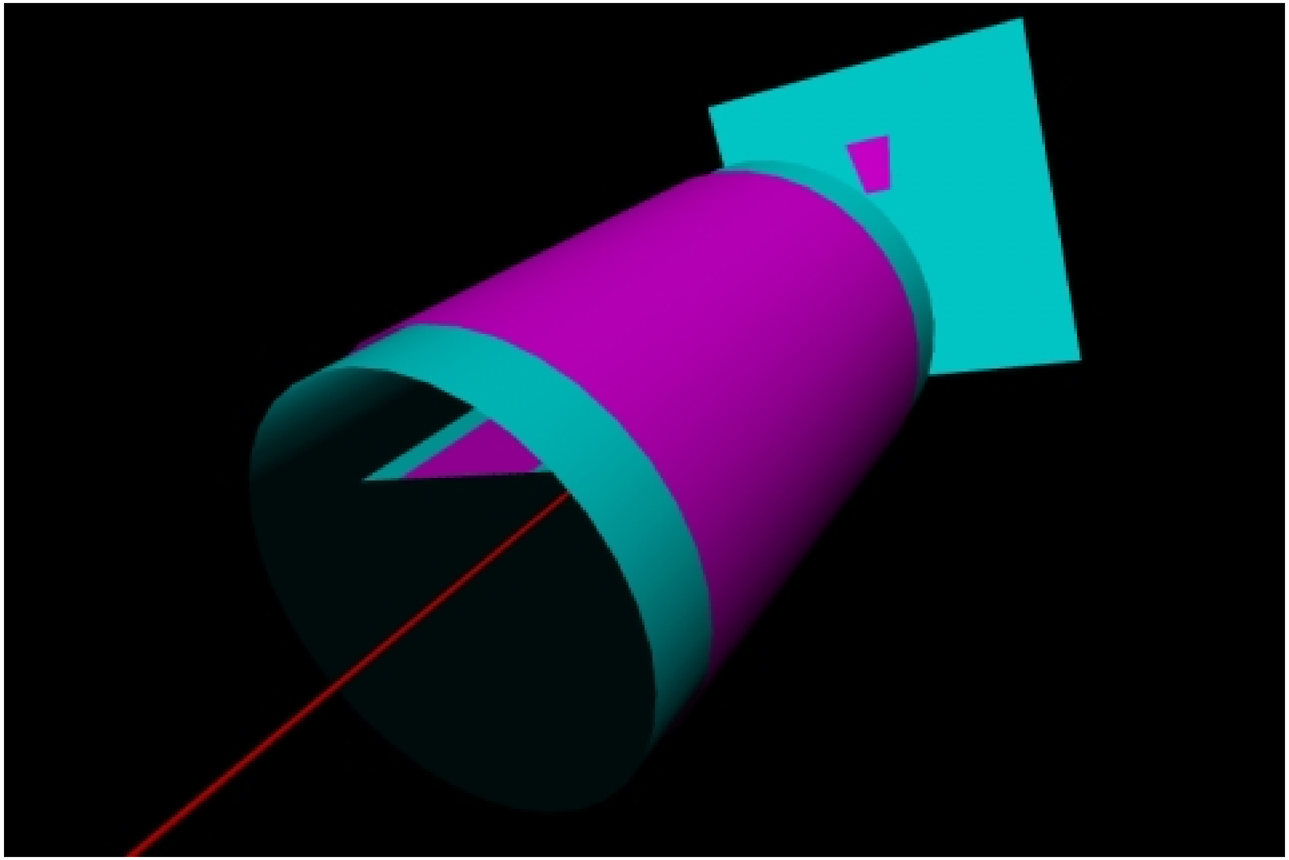
# Distance of Closest Approach

- **DCA is also a 5D** Surface **in the 6 parameter space of points along a track.**
- **It is** not **a 2D surface in 3D space.**
- **Characterized by the track direction and position in the (x,y) plane being normal;** $\alpha = \pi/2$.
- **Track Parameters:** $(r, z, \phi_{dir}, \tan\lambda, q/p_T)$

# Detector

- **A** Detector **describes a collection of** Layers **which are organized in a hierarchy of detectors.**
- Layers **describe the geometry of the detector by holding** Surfaces**, either directly or through sub** Layers

# Detector



# Propagators

- Propagators **propagate a track (with or without covariance matrix) to a new surface.**
  - Propagators **to and from all the surfaces are defined, *e.g.***
    - **PropCylCyl**
    - **PropDcaCyl**
    - **PropXYZ**
- **Currently defined for homogeneous fields.**

# Interactions with Material

- **Interactions with material affect the track state by perturbing the track covariance matrix (*e.g.* stochastic processes such as MCS) or the track vector itself (dE/dx).**
- **This behavior is encapsulated in an abstract** Interactor
  - **Specific instances inherit from this, such as** ThinCylMs.
  - **Energy loss is handled by abstract** DeDX
    - DeDxBethe or DeDxFixed

# Track Fitting

- **Can be combined with track finding to accomplish both tasks at once.**
- **Can also fit hits which have been identified as constituents of a track by a separate pattern recognition package.**

# Track Fitting

- Pattern recognition program delivers a list of hits and an estimate of the global track parameters.
- Track Fit uses the Kalman Filter algorithm to reconcile the track hypothesis with the hit measurements in an iterative manner.
- After fitting each hit, the track covariance matrix is updated to account for the effects of MCS, and the track vector is modified to account for dE/dx.
- The track is then propagated to the next surface.

# Track Fitting

- $\chi^2$ at each surface can be used to reject outliers or search for kinks caused by decays in flight or bremsstrahlung.
- Misses are added with a probability which reflects the efficiency of the detector
  - Cut on combined probability, not number of misses.
- End up with the best fit at the extrema of the track, project to vertex or calorimeter.
  - Smoothing gives the best fit at all points.

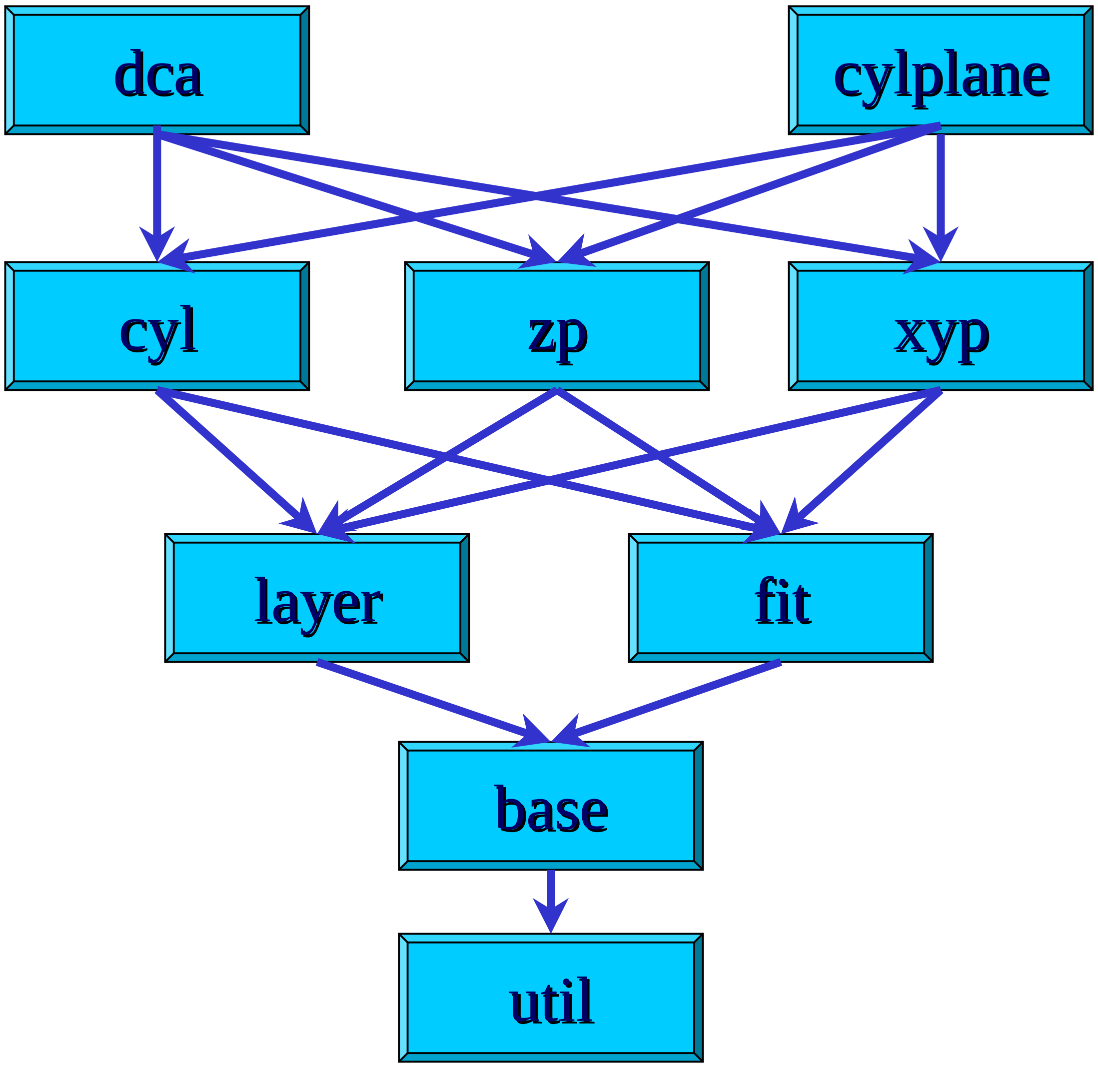
# Simulations

- Simulators are provided to generate hits and account for MCS and energy loss.
- Can be used for fast simulation:
  - Particles from MC event are propagated to each detector element.
  - The appropriate hit is generated from the intersection of the track with the surface.
  - Track vector is smeared for MCS and modified for energy loss, then propagated to next element.

# Status

- The trf++ tracking toolkit packages are available in both C++ and Java.
- Supports both finding+fitting or fitting of tracks provided by external pattern recognition.
- Currently being used at D0.
- Integration into LCD framework to be initiated after LCWS2000.

# Package Dependencies



# Credits

- David Adams (UTA) is the principal architect of the trf++ package.
- Developed in C++ and intended to be a general purpose track finding and fitting toolkit.
- Contributions from others, especially Slava Kulik.
- Experiment-neutral implementation.