dRICH Optics

– Status, Issues & Plans –

Christopher Dilks
dRICH Meeting
25 January 2023

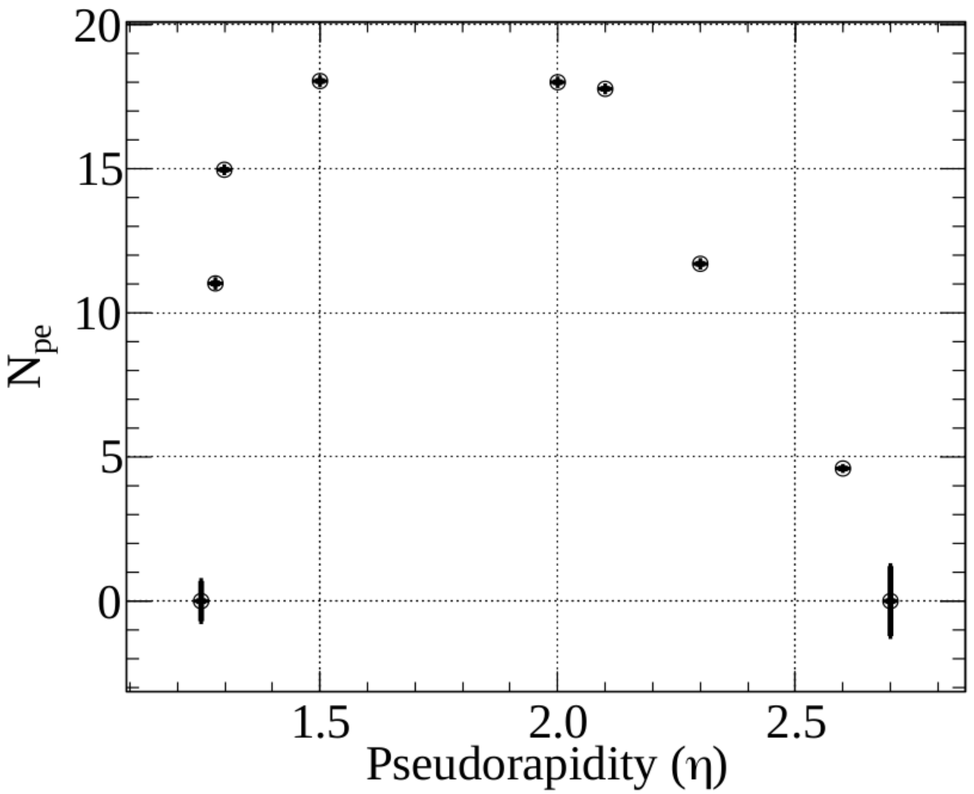
Acceptance shown at Collaboration Meeting

- 50 GeV pions
- Number of Photoelectrons (NPE) from gas radiator
- Acceptance limits:

  $1.3 < \eta < 2.3$

  $11.5° < \theta < 30°$

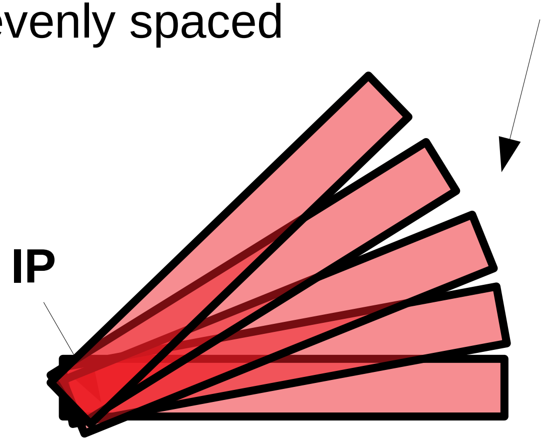
  Integrated over ϕ
- Optics could be improved…



Study from C. Chatterjee

Parallel-to-point Focusing

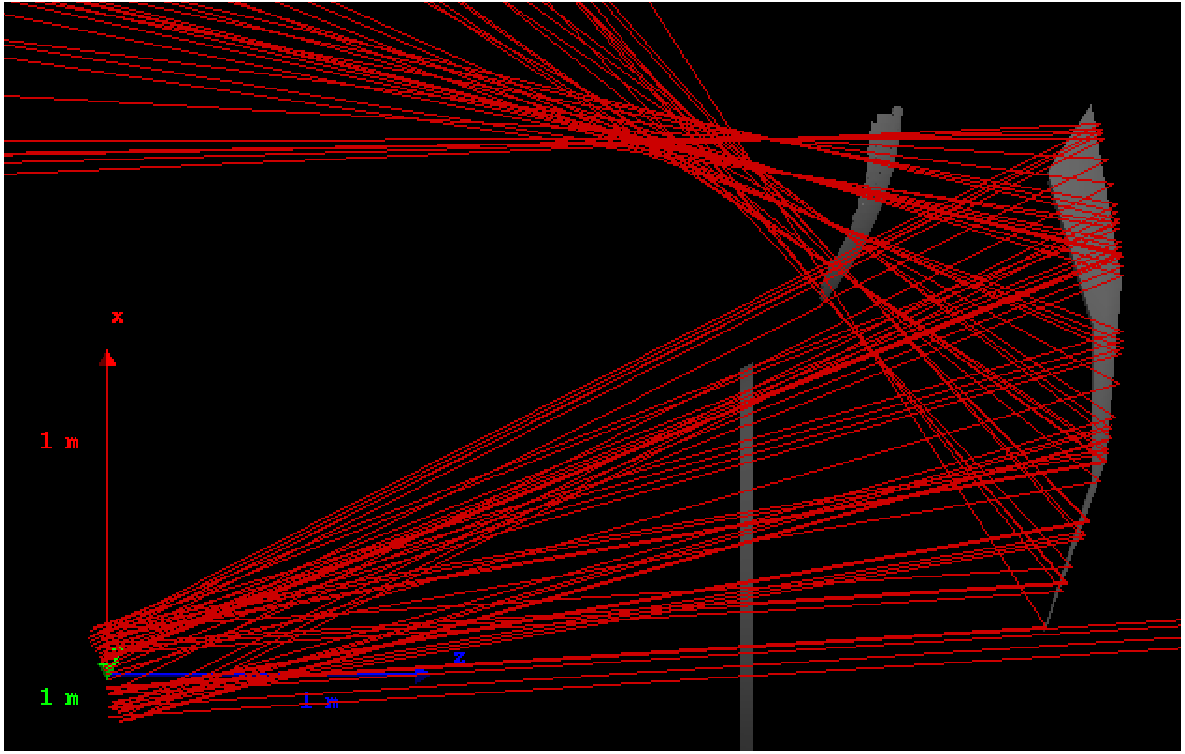
- 5, wide collimated photon beams
  - Emitted from IP
  - Within full dRICH polar acceptance, evenly spaced



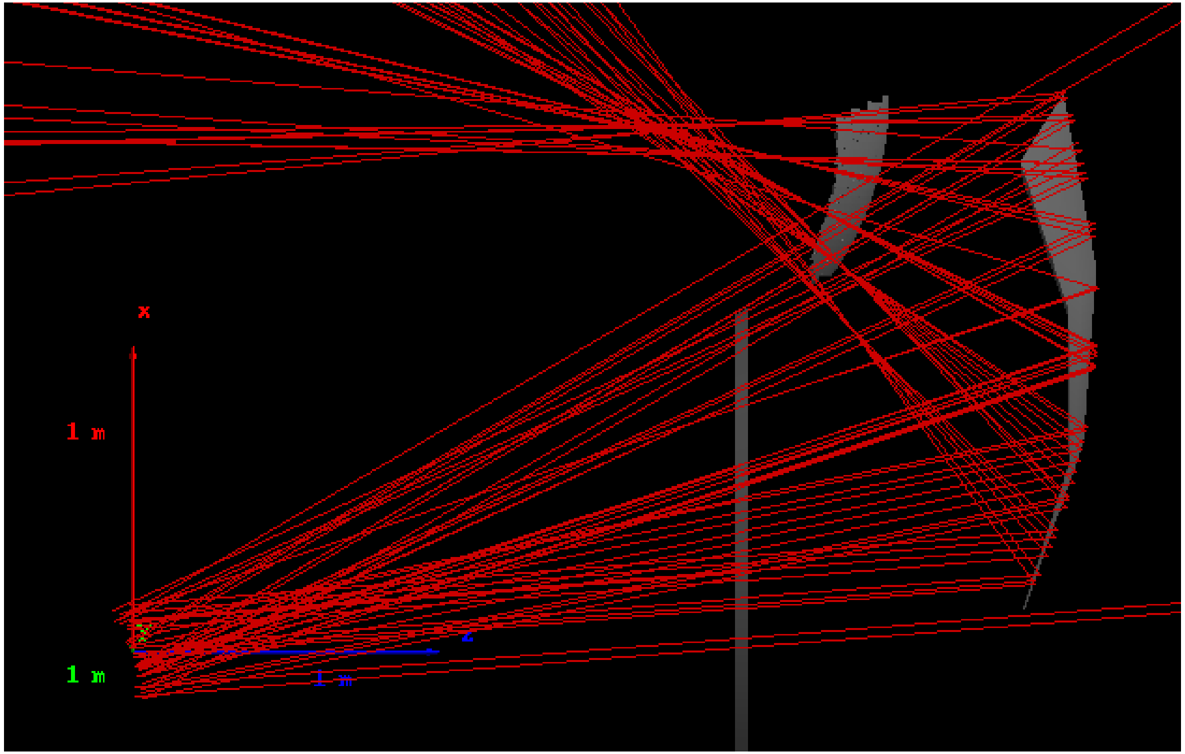
- Each beam will focus in a region → this is parallel-to-point focusing
- The closer the sensors are to this region, the smaller our Cherenkov ring resolutions are… **approximately...**
- A more realistic test involves checking the focus for *all* possible Cherenkov cones originating from *all* possible charged particle track points in *both* radiators
- Our studies so far indicate this parallel-to-point focal region is a decent approximation to the "real" focal region

The optics we had in August 2022 …

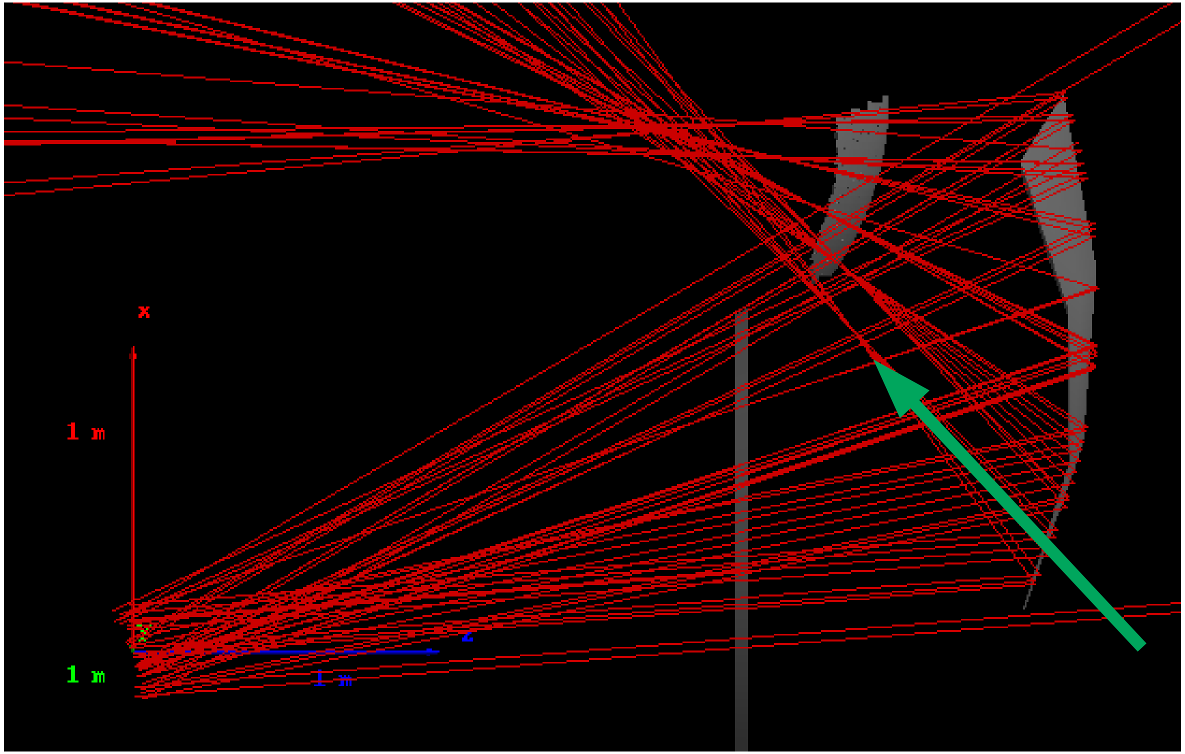
Tuned in https://github.com/eic/epic/pull/24



… vs. what we have now



… vs. what we have now

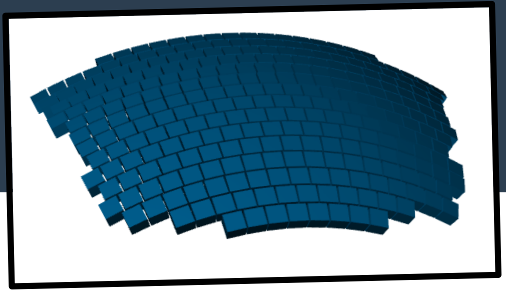


High η misses the sensors!
This is easy to fix!

How could this happen?

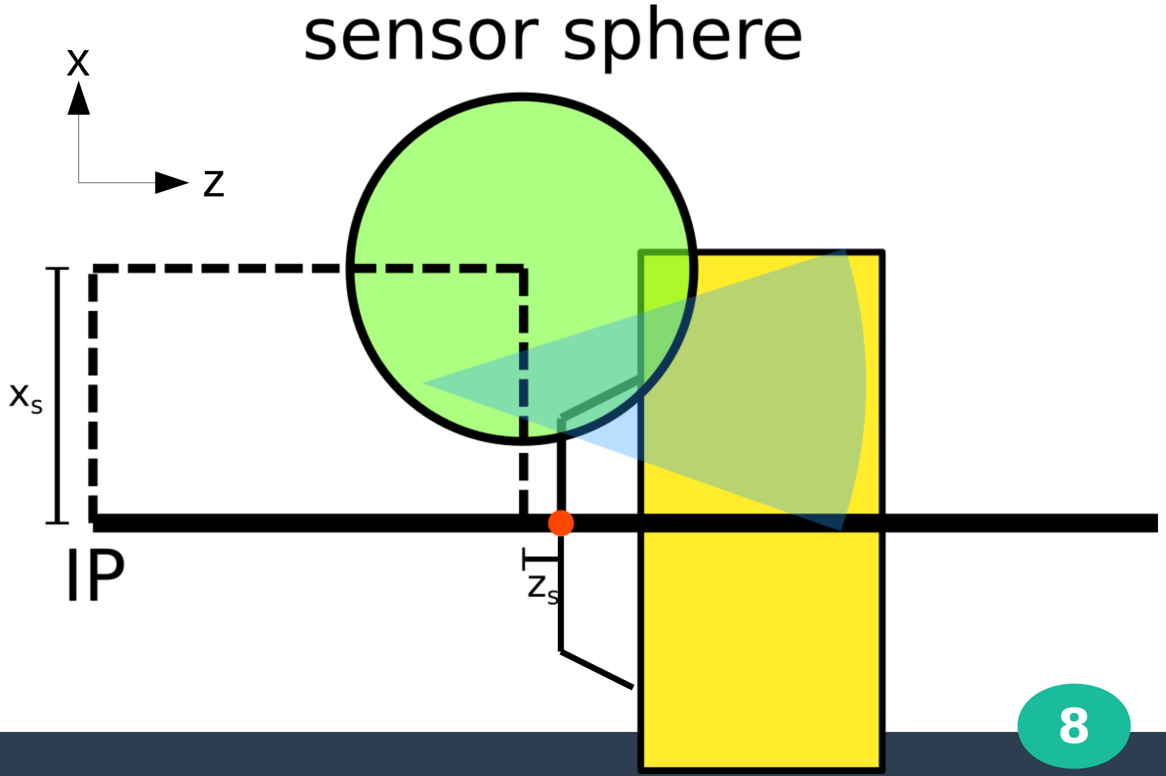
- **Optics optimized in August 2022 to be “good enough” to proceed with PID implementation, but never fully optimized → there were higher priorities**
- ***Many* small geometry changes between August 2022 and 1st campaign**
  - Optics looked okay, so we did not do any more tuning
  - Overlooked that the high-η photons were suddenly *missing* the sensors
  - Importance of continuously testing everything!
- **We must improve:**
  - Restoring full η acceptance: easy to do
  - Getting good focus across all η: difficult, and *impossible* with a single spherical mirror (per sector)
  - Need to also improve sensor placement – it’s not a sphere

Sensor Parameterization



- **Sensor sphere**: sensors are tiled on a sphere with specified radius and center coordinates (z_s,x_s), defined with signs specified with respect to vessel snout front (red point)
- “spherical patch” cuts are used to take a subset of the sphere within the vessel

```
<sphere
  centerz="-70.0*cm"
  centerx="220.0*cm"
  radius="140.0*cm"
  />
<sphericalpatch
  phiw="18*degree"
  rmin="DRICH_rmax1 + 1.0*cm"
  rmax="DRICH_rmax2 - 4.0*cm"
  zmin="DRICH_snout_length + 3.0*cm"
  />
```



Mirror Parameterization

Spherical Mirror: Need 3 numbers: center position (2 numbers) and a radius

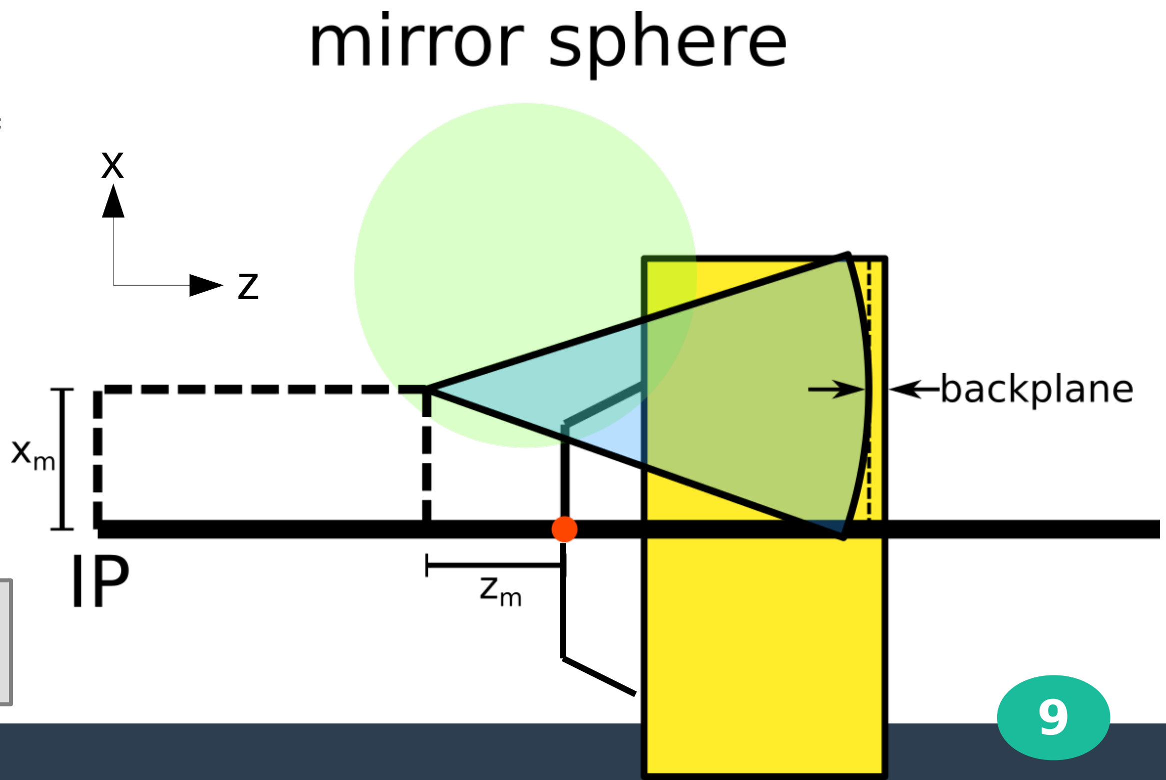
In practice: reparameterize in terms of 3 other numbers

Mirror position (z_m,x_m) determined with the help of “tune” parameters (see next slide)

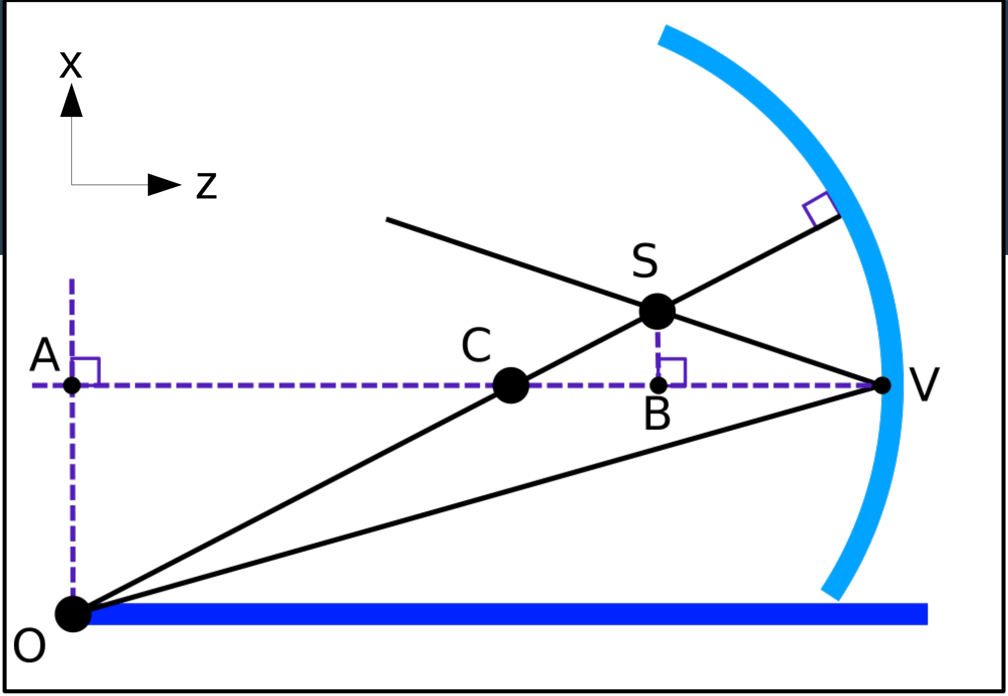
Radius r_m determined from z_m, given a fixed “backplane” distance: the minimum distance between the mirror and the vessel backplane

```
<mirror
  backplane="DRICH_window_thickness + 2.0*cm"
  rmin="DRICH_rmin1 + DRICH_wall_thickness - 1.0*cm"
  rmax="DRICH_rmax2 - DRICH_wall_thickness - 3.0*cm"
  phiw="59.5*degree"
  thickness="0.2*cm"
  focus_tune_x="-5.0*cm"
  focus_tune_z="0.0*cm"
  />
```

```
radius   = 218.5 cm
center_z = -100.6 cm
center_x = 113.9
```



Spherical Mirror Optics



Goal: Focus an object at O (e.g., the IP, or photon vertex) at the position S

C = mirror center (radius = [CV])
A,C,B,V are along the mirror's optical axis; at V the mirror tangent plane is parallel to the xy-plane

Given positions O and S, along with the length d=[AV], one can solve the following similar triangle relations for mirror center and radius

$$d = [AV] \qquad S = (z_S, x_S)$$

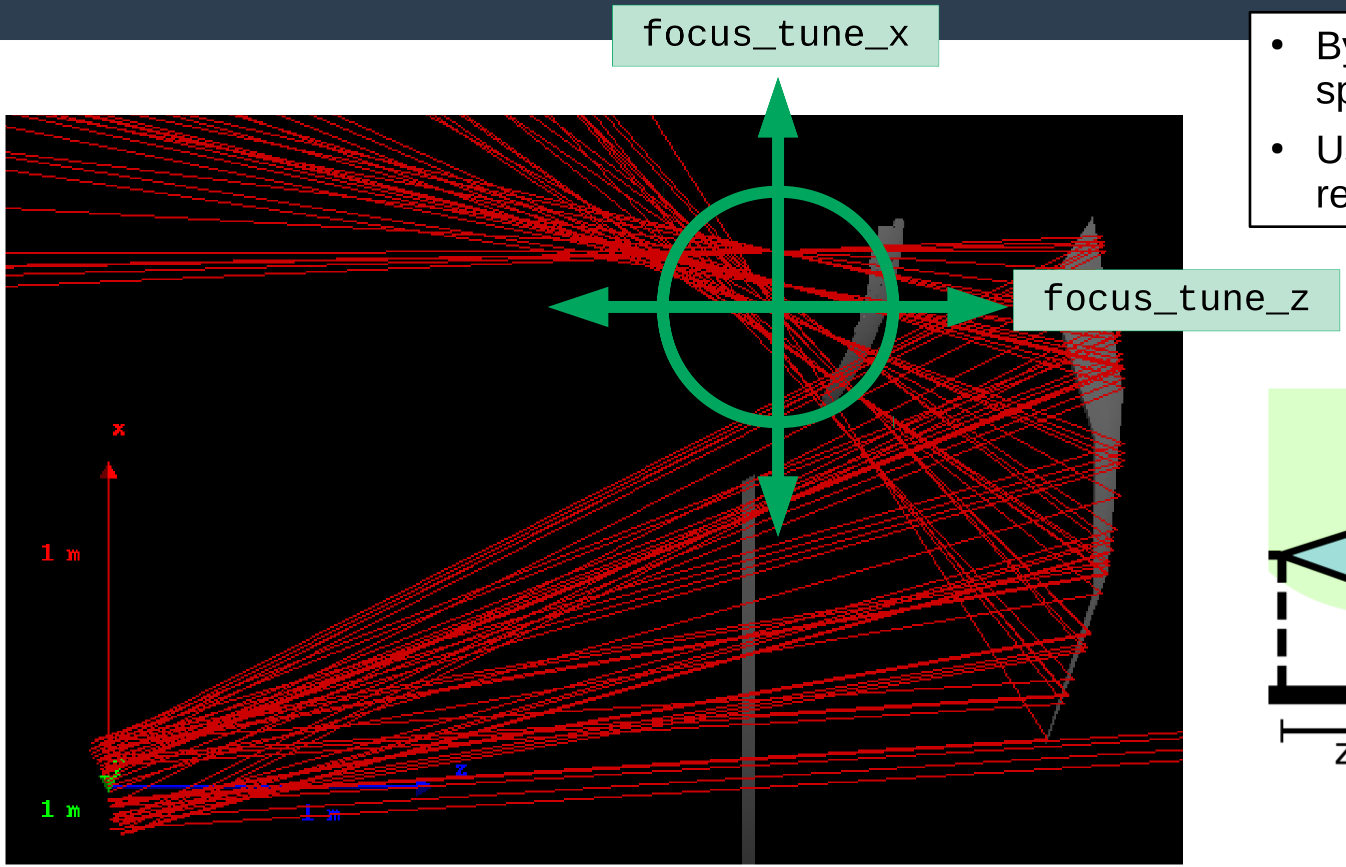
$$\begin{cases} \Delta ACO \sim \Delta BCS \\ \Delta AVO \sim \Delta BVS \end{cases}$$

$$C = (z_M, x_M) = \left(\frac{d z_S}{2d - z_S}, \frac{d x_S}{2d - z_S} \right)$$

$$r_M = d - z_M$$

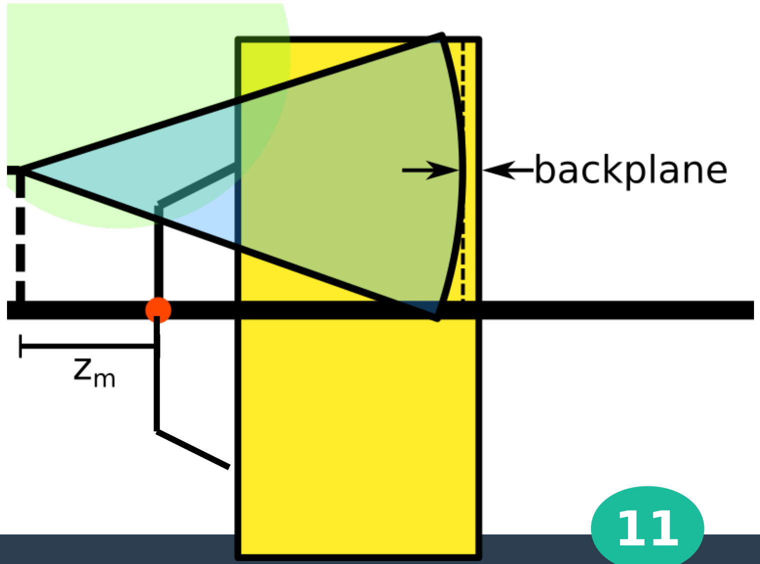
Caveat: if O is far from the optical axis, that is if distance [AO] is large, then spherical aberrations will cause the image to not appear exactly at S

Steering the Focal Region



- By default, S is set to the sensor sphere center
- Use focus tunes to *steer* the focal region w.r.t. the sensor sphere center

backplane distance stays fixed



Brute Force Optimizer

- **Scan the parameter space, hoping to find a reasonable region**
- **Start with a coarse, 5-dimensional lattice**
  - Sensor sphere center (z,x) and radius
  - Mirror focus tunes (z,x)
- **Choose the "best" option(s)**
  - Still lacking a quantitative performance metric, so the option is chosen "by eye"; looking for:
    - Small ring resolution
    - Covers full acceptance, from low to high η
    - Sensors are not blocking the Cherenkov cones
    - Photons are close to normal incidence on the sensors
    - Check for "holes" in the acceptance, along sector boundaries
- **Repeat with a finer lattice near this "best" region, until we converge with good optics for PID**
  - May need to constrain some variables, or run on a lower dimensional lattice
  - Use intuition built from several difficult days' experience tuning ATHENA dRICH optics by hand

Example Lattice for Brute Force Optimizer

- **The <u>most recent</u> (finest) lattice that was run to finalize the August 2022 optics tune:**
  - Mirror:
    - focus_tune_x: 5 points, from –20 to 0 cm
    - focus_tune_z: 5 points, from 0 to 30 cm
  - Sensor sphere:
    - radius: 5 points, from 80 to 120 cm
    - center_z: constrained to be “50 – sensor sphere radius”

Note: it is possible to constrain variables as functions of other (varying) variables, allowing for much more flexibility

Did not run a finer lattice, since at the time our goal was “good enough” optics

Automated Parameter Variation

scripts/vary_params.rb

- Input user configuration:
  - Which parameters to vary, and how to vary them
  - Fixed parameter values (which differ from the default)
  - Derived parameters, which depend on varied parameter values
  - Simulation pipeline – the code you want to run for each variant (shell commands)

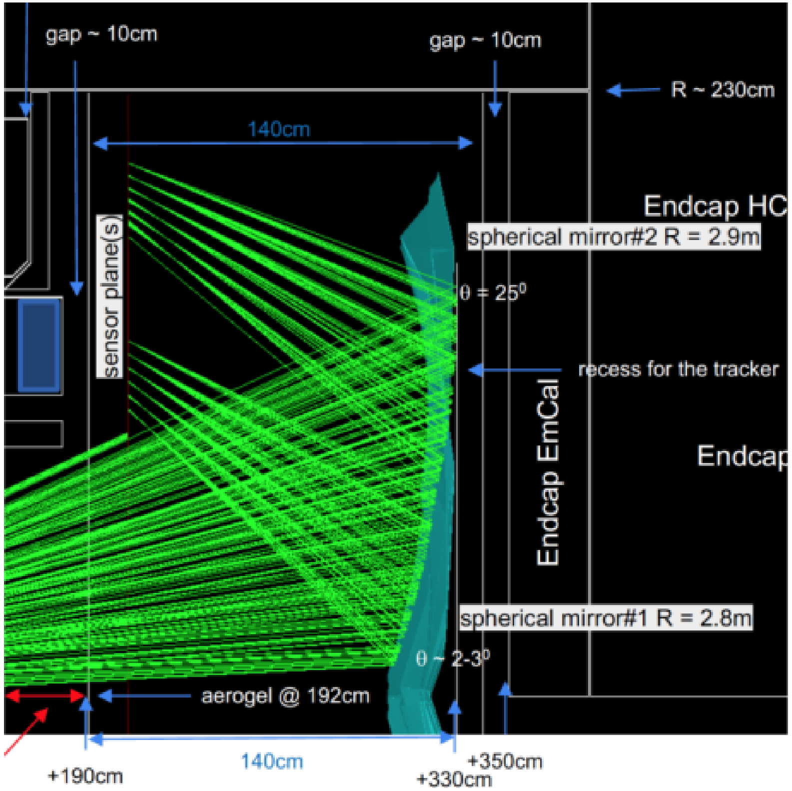
- Execution
  - Takes the "product" of all possible variants
  - Calculates derived parameters for each variant
  - Generates dRICH compact (XML) files for each variant
  - Runs simulation pipelines, multi-threaded, one thread per variant

- Outputs for each variant:
  - Simulation pipeline output, as well as logs for stdout and stderr
  - Info files, listing the variant's parameters
  - Compact files, config files, etc.
  - TODO: performance metric

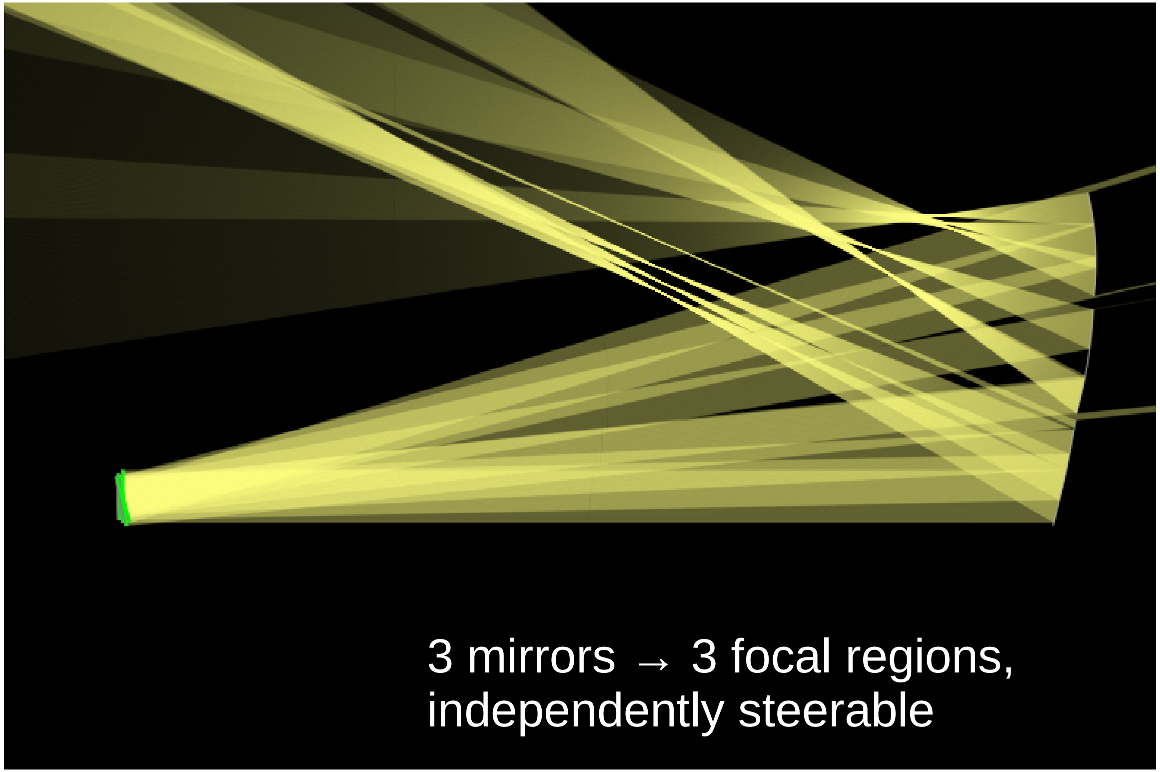
- **Runs the Brute Force optimizer**
- **Designed to be an entry point for smarter optimizers**
- **Entry point for Machine Learning**
- **Desperately need quantitative performance metric**

Multiple Mirrors → Sensor Placement Flexibility

Alexander's Dual Mirror approach:



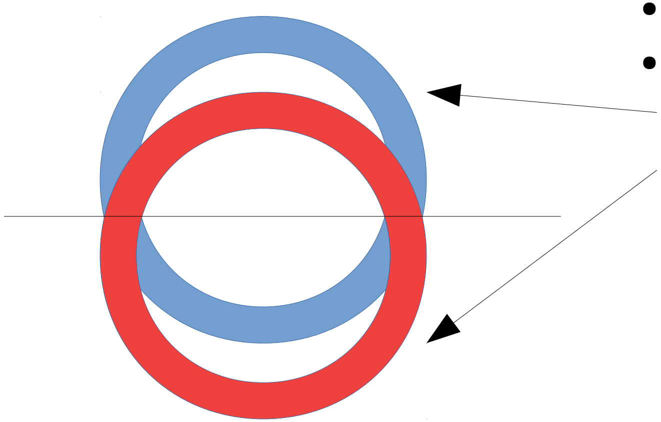
Multiple Mirrors → Sensor Placement Flexibility



- Multi-mirror configuration allows for independent focal region steering for each mirror
- Use this approach to “tighten” the focal region
- Allows for easier sensor placement, given hardware constraints
- Possible issue: expense of the mirror molds

toy ray optics simulator: https://ricktu288.github.io/ray-optics/simulator/

Multiple Mirrors → Sensor Placement Flexibility



- Intersection of spheres is a plane
- Take “divergent” combination:
  - Blue mirror above the plane
  - Red mirror below

- With the divergent combination, each mirror will therefore correspond one-to-one to a set of sensors
- But… given a hit on a sensor, will we know which mirror it came from?
  - Maybe…
  - Alexander has been adding features to the IRT code to handle a similar situation for the pfRICH
- Rings that span the boundary of 2 mirrors should be straightforward to handle with ray tracing

Implementation Strategy for Multi-Mirrors

- We already have multi-mirror geometry code from ATHENA
  - Recover and update it for ePIC
  - Improve it and make it work
  - Then figure out where to put the sensors
- **Need a person who enjoys geometry and code to dedicate time and effort to do this**

Unrealistic Ideal

- Parabolic mirror has less spherical aberration than a spherical mirror
- Hyperbolic mirror is even better
- Both are significantly more expensive to manufacture, and much harder to achieve the same level of precision as a spherical mirror

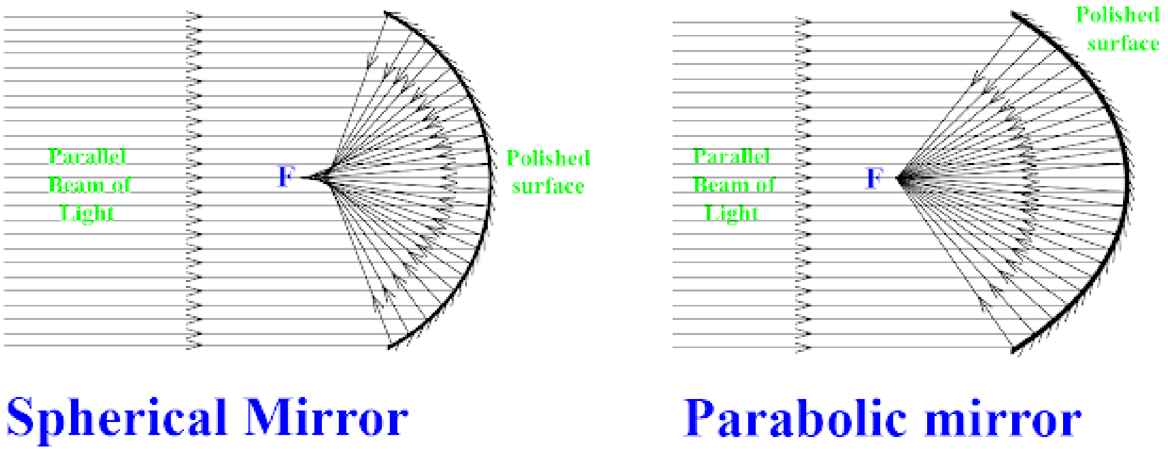


image from
https://qsoluti0n.blogspot.com/2020/07/Curved-mirror.html

Sensor Placement Guidance

Connor Pecar's Focus Finder

- Automatically finds the parallel-to-point focal region, in 3D
- Can draw boxes in DD4hep at this region → could be extended to guide sensor placement
- Should be straightforwardly compatible with multi-mirror configurations

code:

https://github.com/eic/epic/pull/351

https://github.com/eic/drich-dev/pull/55

2x2 SiPM Modules

Readout modular unit and services

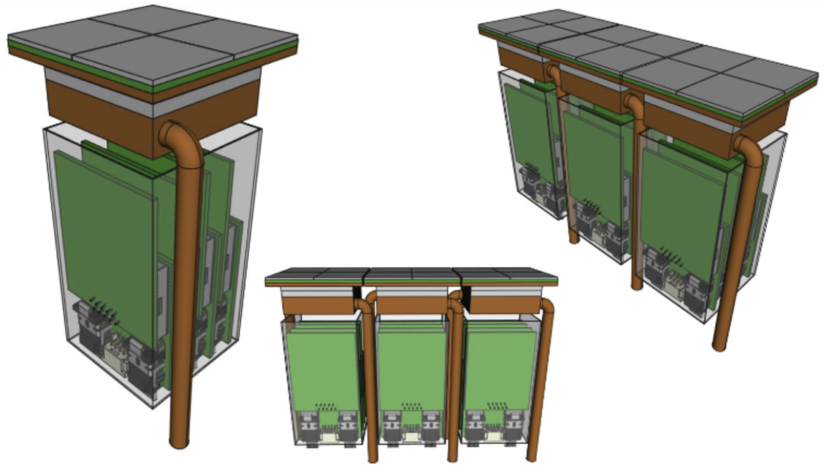


Figure from Marco Contalbrigo

- 20 cm behind the sensors
- Combined this reality with Connor’s focus finder → spherical placement of sensors is not ideal
- Need to take this into consideration along with the multi-mirror plan

Summary and Outlook

- **Optics issues**
  - Limited η acceptance → easy to resolve, issue slipped under the radar
  - Good focusing across all η → much more difficult
    - Need multi-mirror configuration
    - Need to improve placement of sensors
    - Need to study the “actual” focal region (not just the parallel-to-point focal region “approximation”)
    - Need people power!
      - The tools are here, we need someone willing to use them, improve them as needed, and do the work