3rd IAEA Technical Meeting on Fusion Data Processing, Validation and Analysis

Forward modelling for the design and analysis of polarisation imaging diagnostics.

Oliver. P. Ford[1]

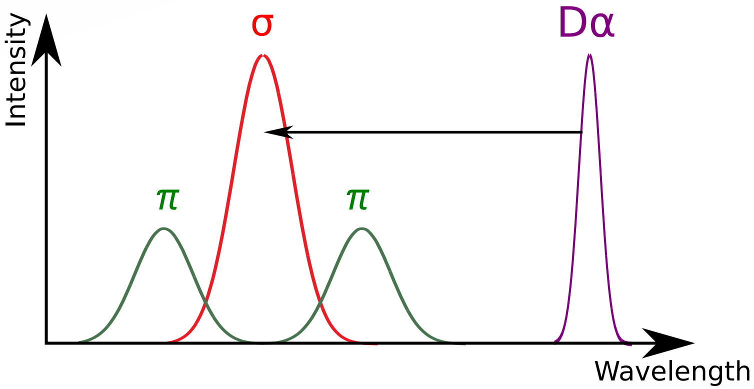
A. Burckhart[1], J.Howard[2], A. Meakins[3], M. Reich[1], J. Svensson[1], R. Wolf[1], ASDEX-Upgrade Team[1]

1: Max-Planck Institut für Plasmaphysik, Greifswald/Garching, Germany
2: Australian National University, Canberra, Australia
3: UKAEA Fusion Association, Culham Science Centre, OX14 3DB, UK

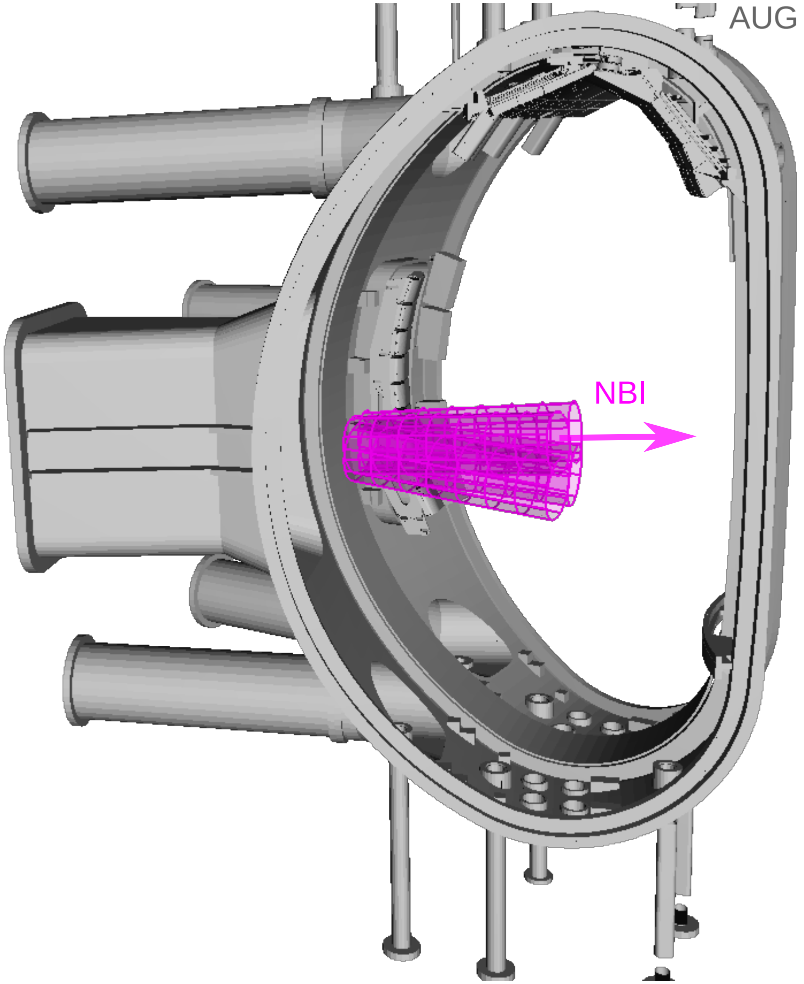
(Imaging) Motional Stark Effect at AUG

ASDEX Upgrade has an existing 10-channel MSE system.

- Hα/Dα beam emission is Doppler shifted and split by the Motional Stark Effect into π and σ components.



Conventional MSE

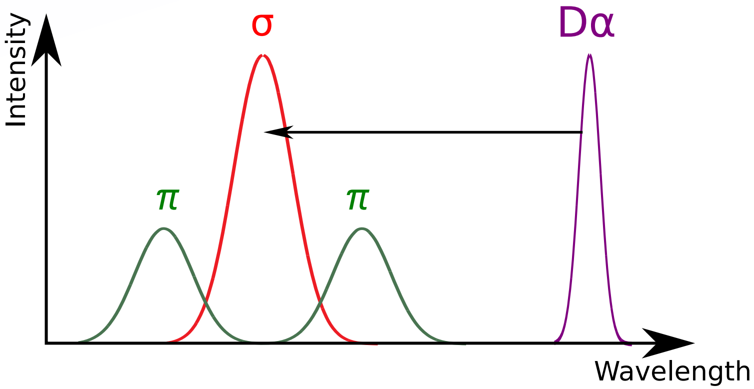


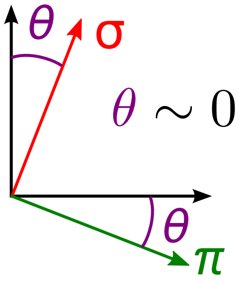
(Imaging) Motional Stark Effect at AUG

ASDEX Upgrade has an existing 10-channel MSE system.

- Hα/Dα beam emission is Doppler shifted and split by the Motional Stark Effect into π and σ components.

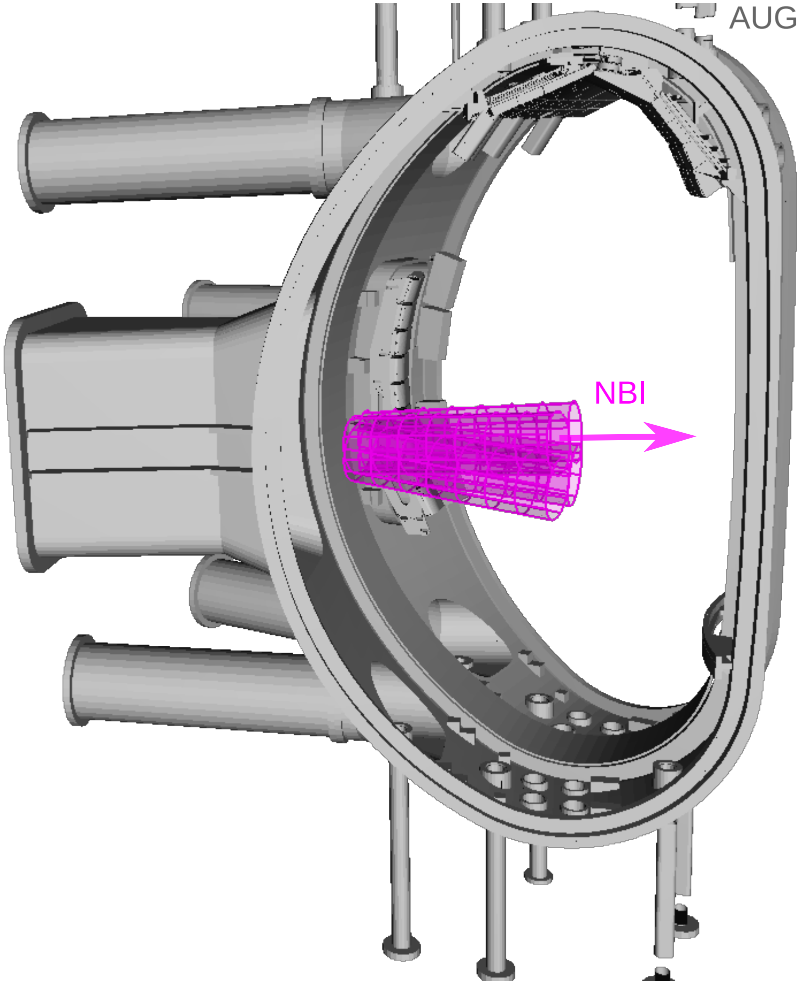
- Components are polarised perpendicular and parallel to projected v x B direction:





$$\theta \sim 0.6\frac{B_z}{B_\phi}$$

Conventional MSE

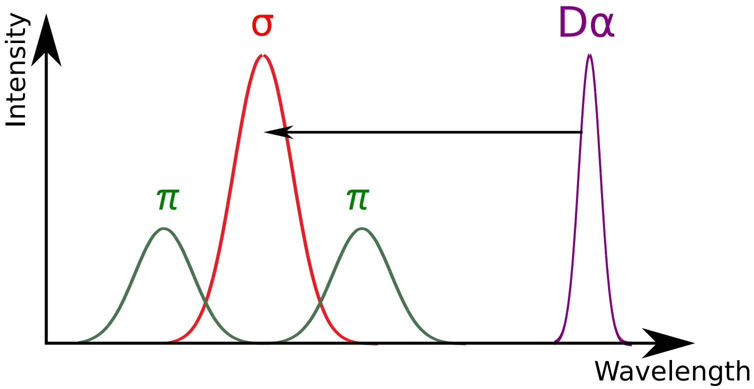


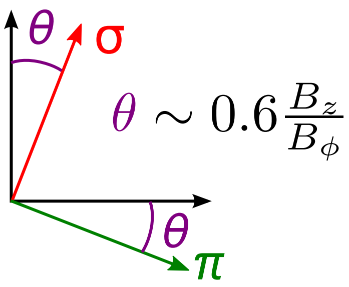
(Imaging) Motional Stark Effect at AUG

ASDEX Upgrade has an existing 10-channel MSE system.

- Hα/Dα beam emission is Doppler shifted and split by the Motional Stark Effect into π and σ components.

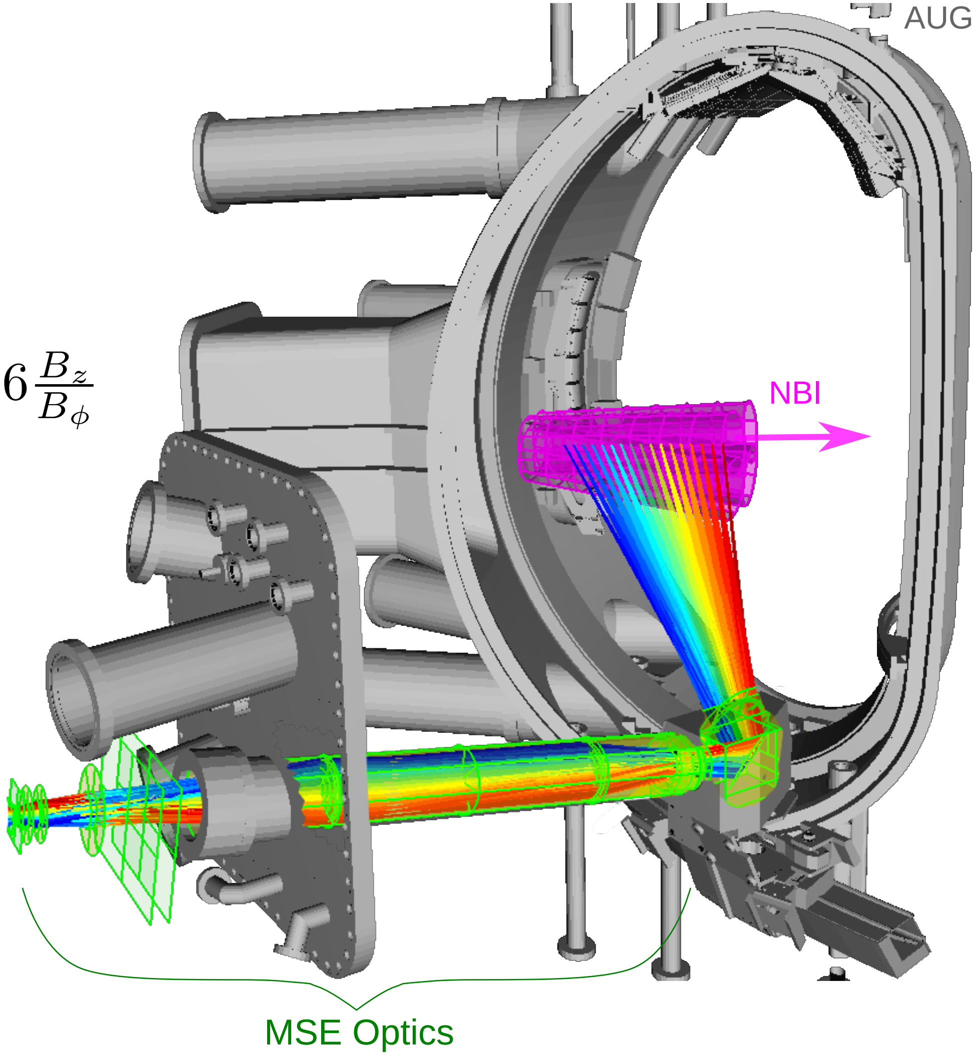
- Components are polarised perpendicular and parallel to projected v x B direction:





$$\theta \sim 0.6\frac{B_z}{B_\phi}$$

Conventional MSE

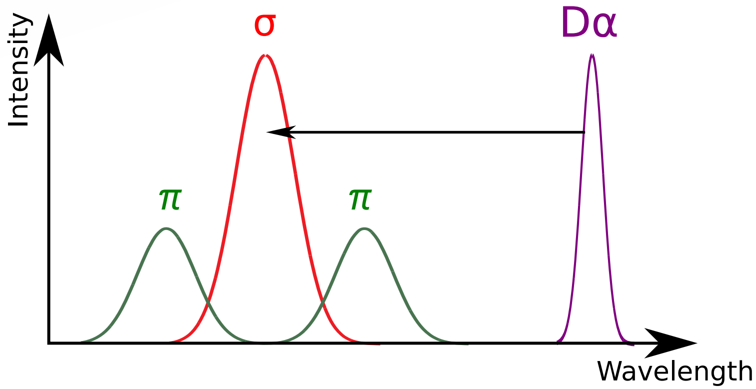


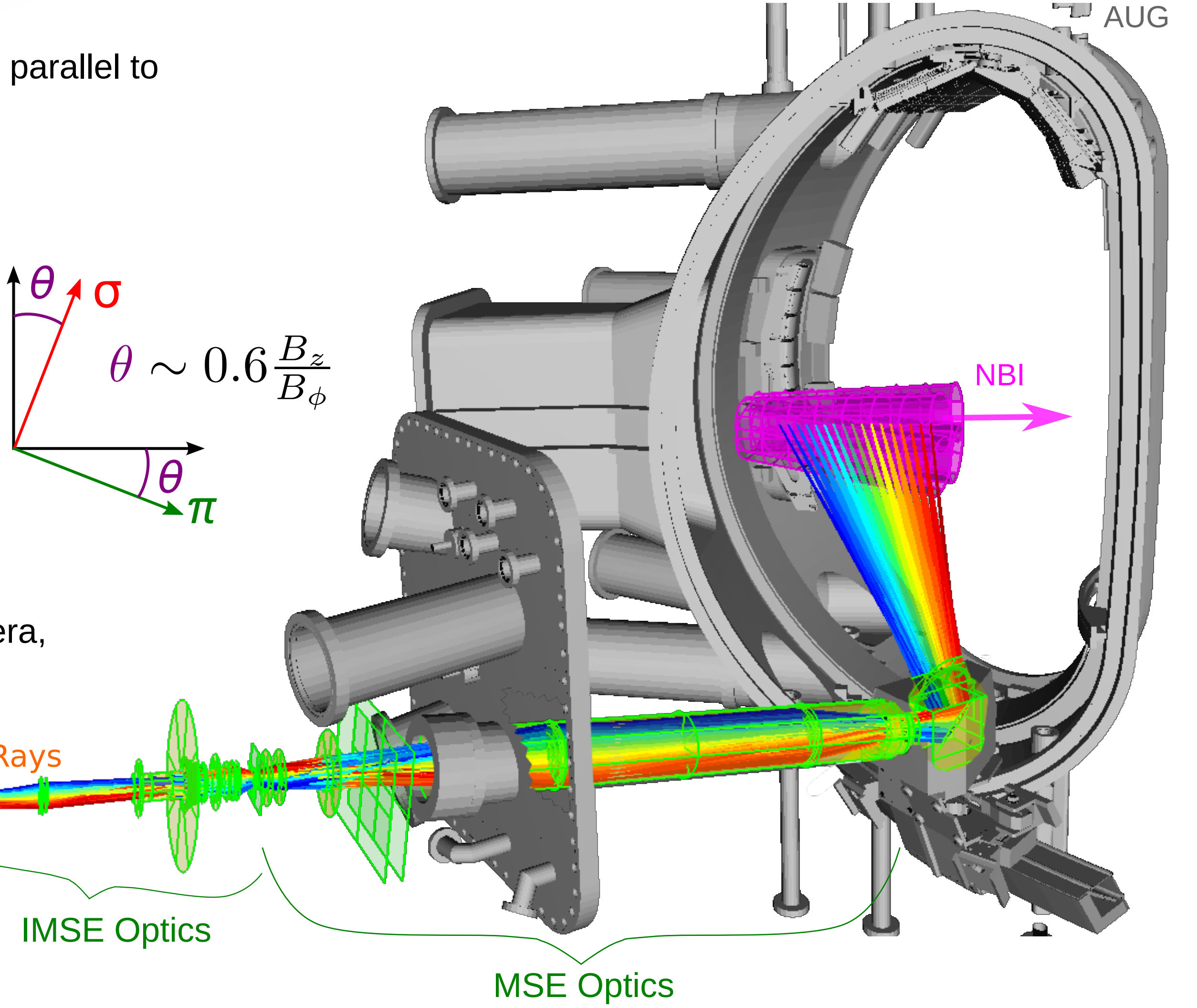
(Imaging) Motional Stark Effect at AUG

ASDEX Upgrade has an existing 10-channel MSE system.

- Hα/Dα beam emission is Doppler shifted and split by the Motional Stark Effect into π and σ components.

- Components are polarised perpendicular and parallel to projected v x B direction:





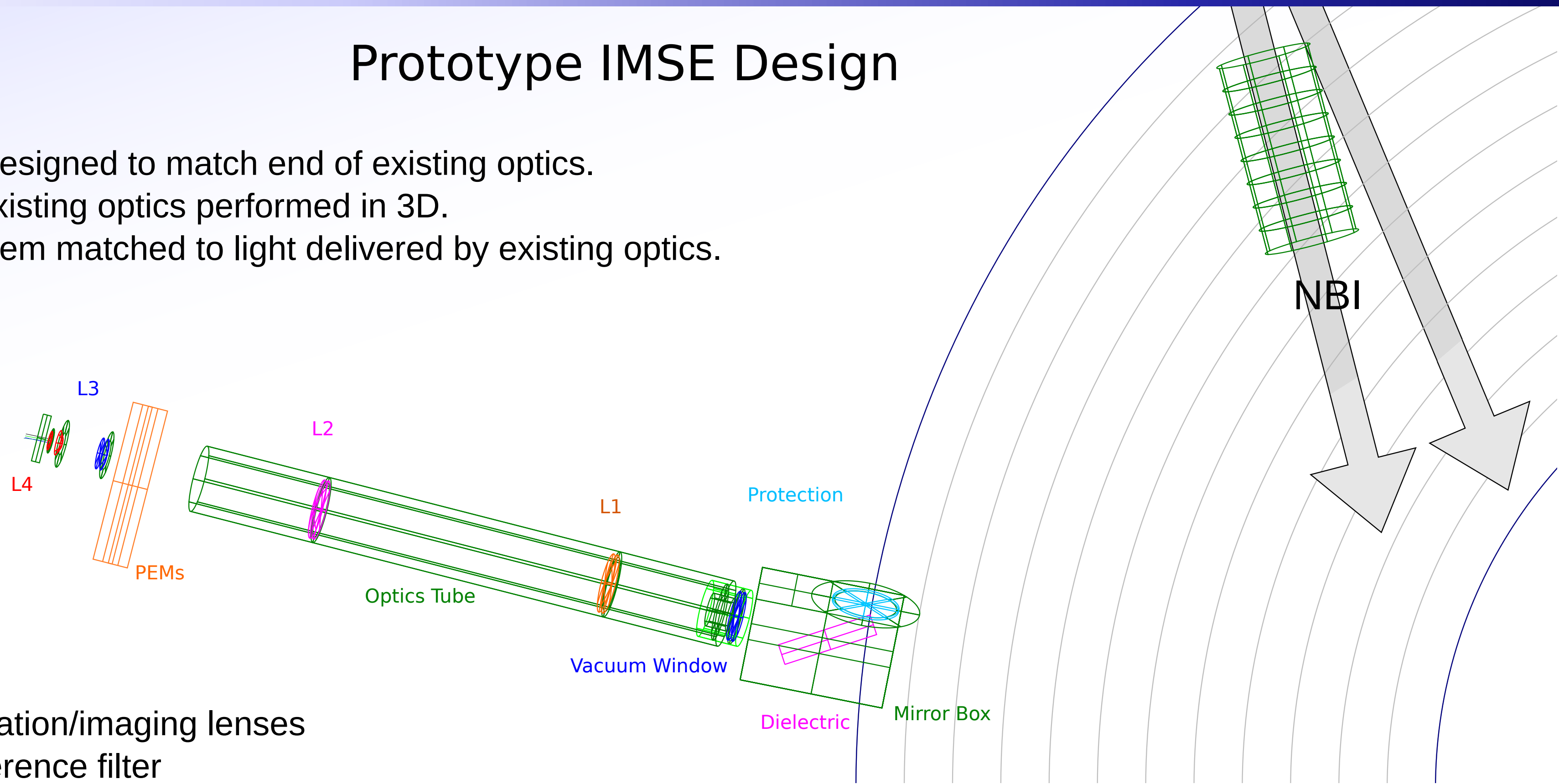
$$\theta \sim 0.6\frac{B_z}{B_\phi}$$

Image of the beam emission using a CCD camera,
> 60x60 θ measurements.

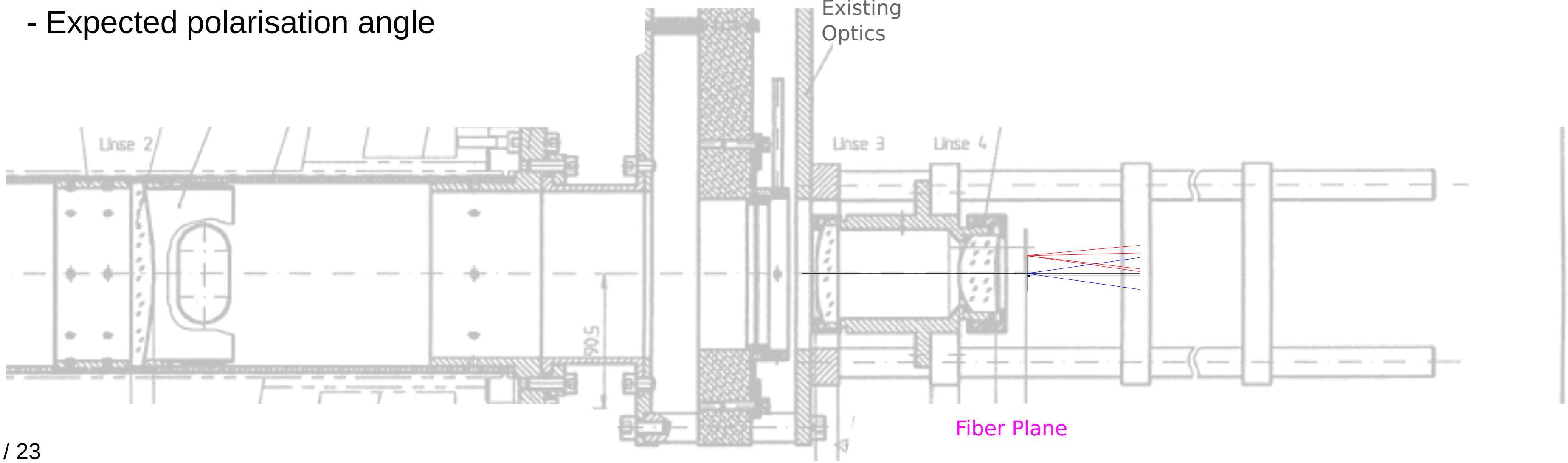
Replaced the MSE for two short periods of plasma operation to test the basic principle.

Prototype IMSE Design

Prototype IMSE designed to match end of existing optics.
- Ray tracing of existing optics performed in 3D.
- New optical system matched to light delivered by existing optics.

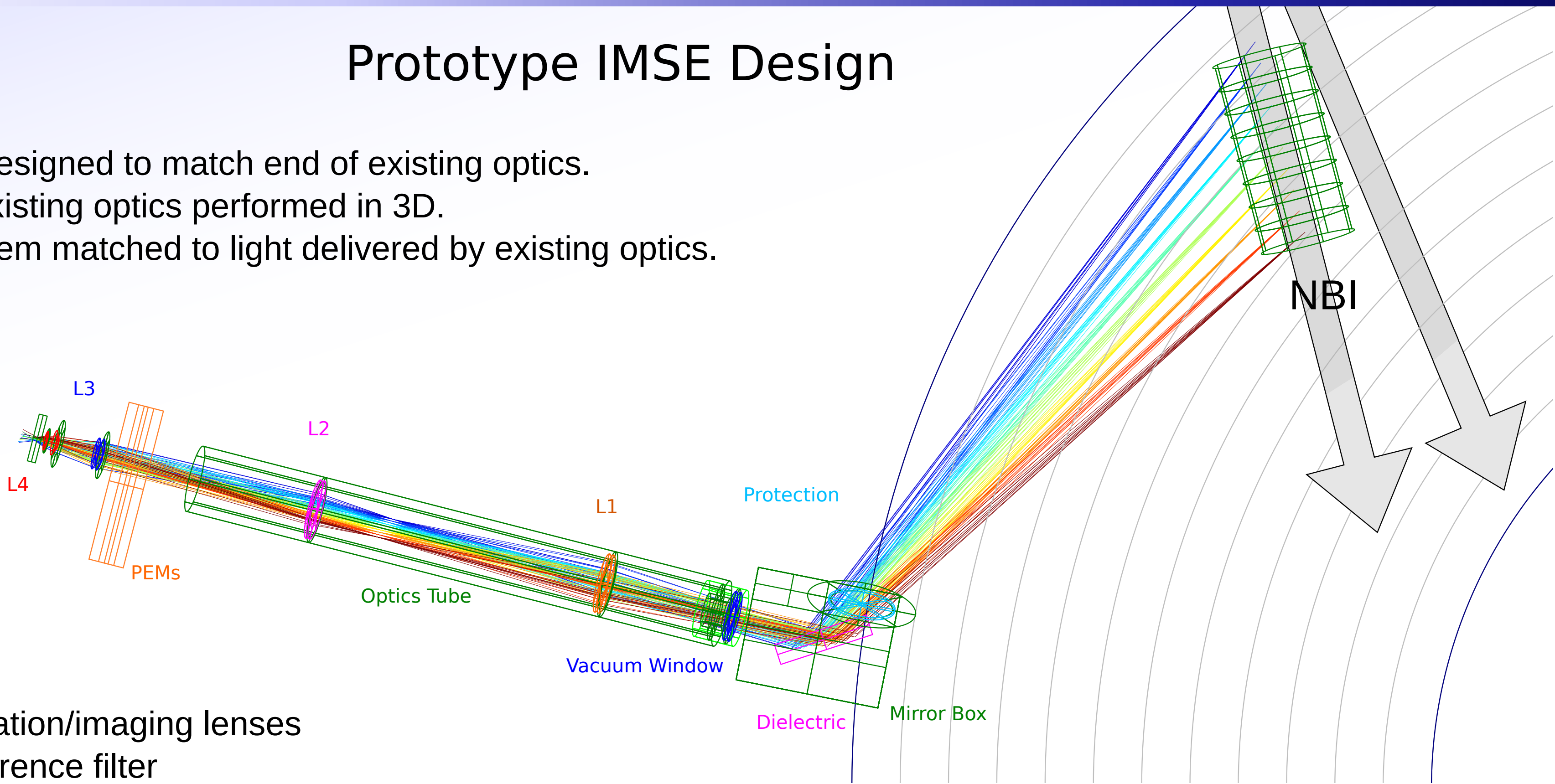


Determination of:
- Optimal collimation/imaging lenses
- Optimal interference filter
- Expected polarisation angle

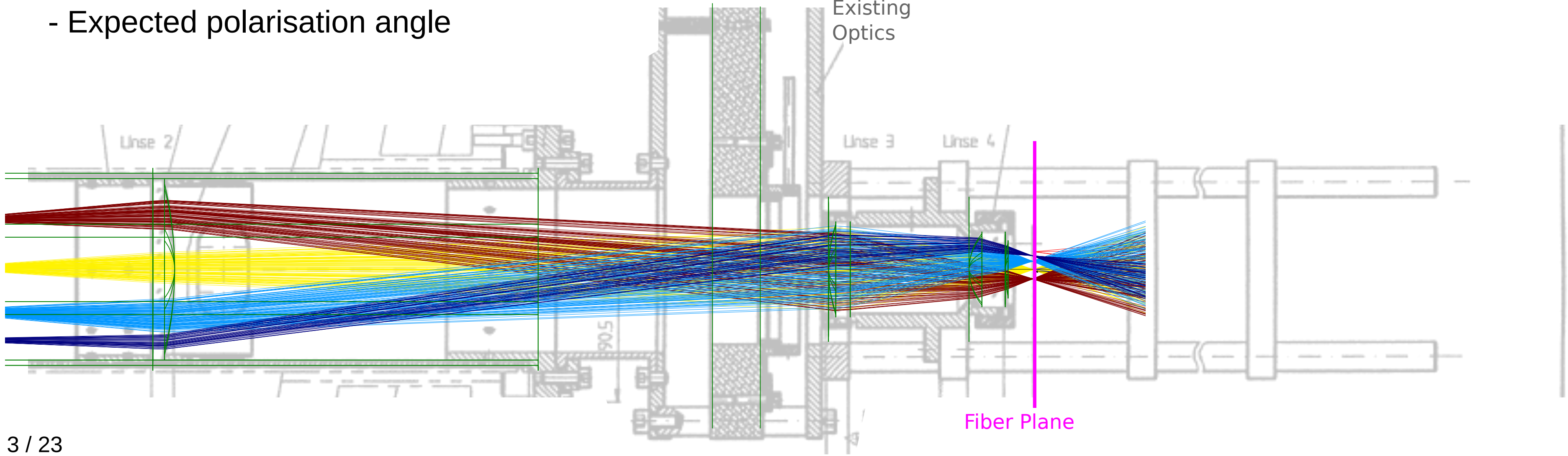


Prototype IMSE Design

Prototype IMSE designed to match end of existing optics.
- Ray tracing of existing optics performed in 3D.
- New optical system matched to light delivered by existing optics.



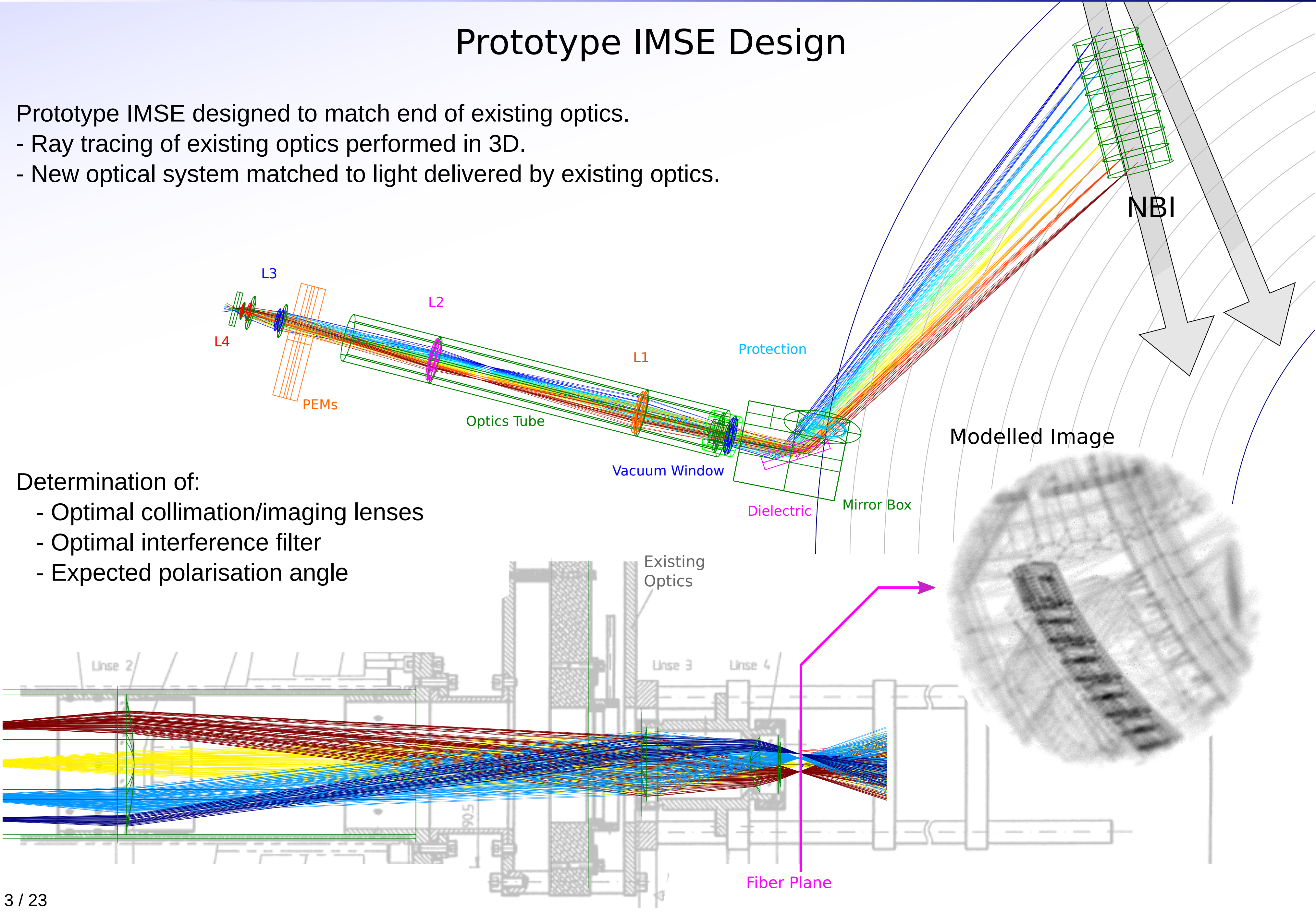
Determination of:
- Optimal collimation/imaging lenses
- Optimal interference filter
- Expected polarisation angle



Prototype IMSE Design

Prototype IMSE designed to match end of existing optics.
- Ray tracing of existing optics performed in 3D.
- New optical system matched to light delivered by existing optics.

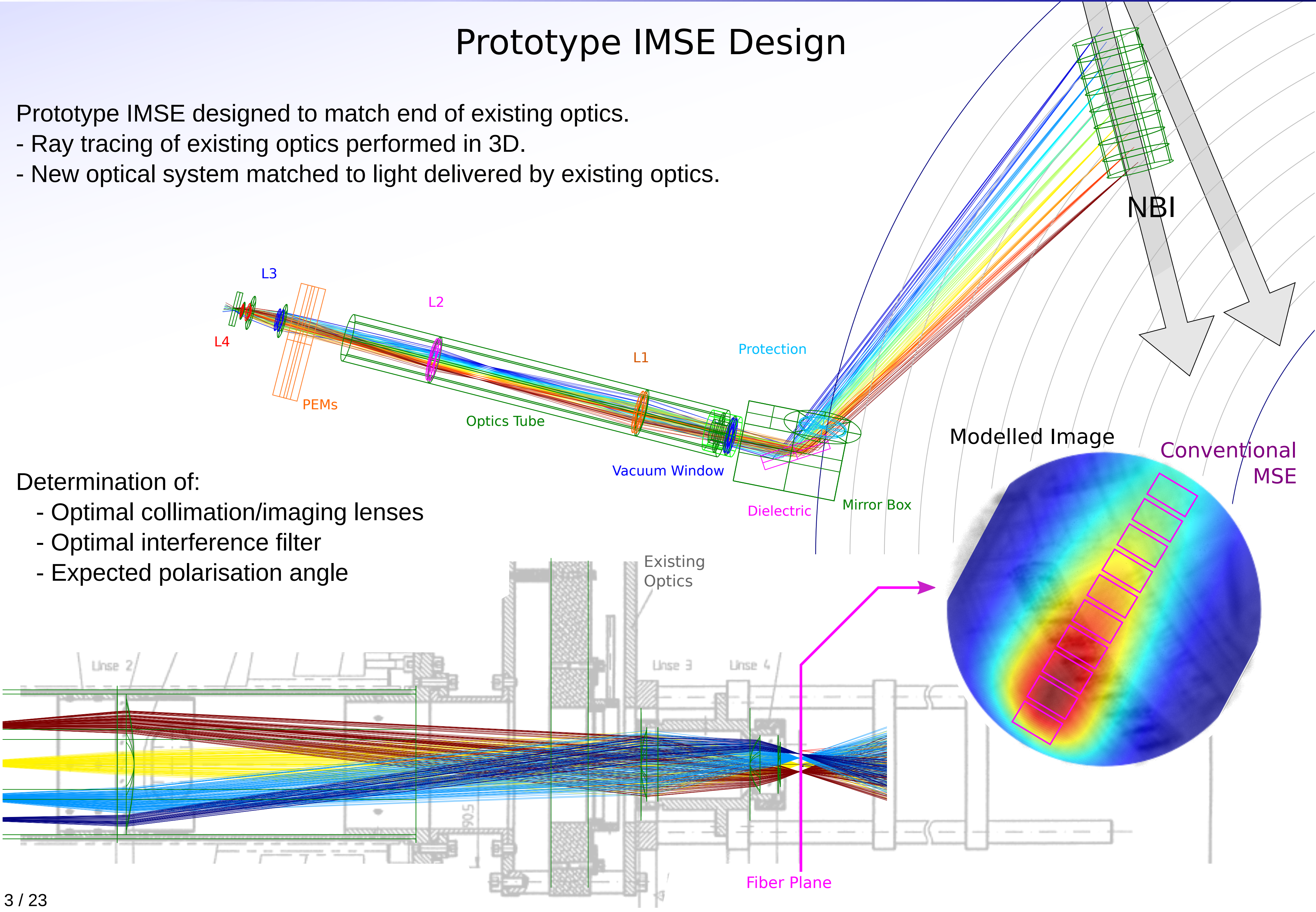
Determination of:
- Optimal collimation/imaging lenses
- Optimal interference filter
- Expected polarisation angle



Prototype IMSE Design

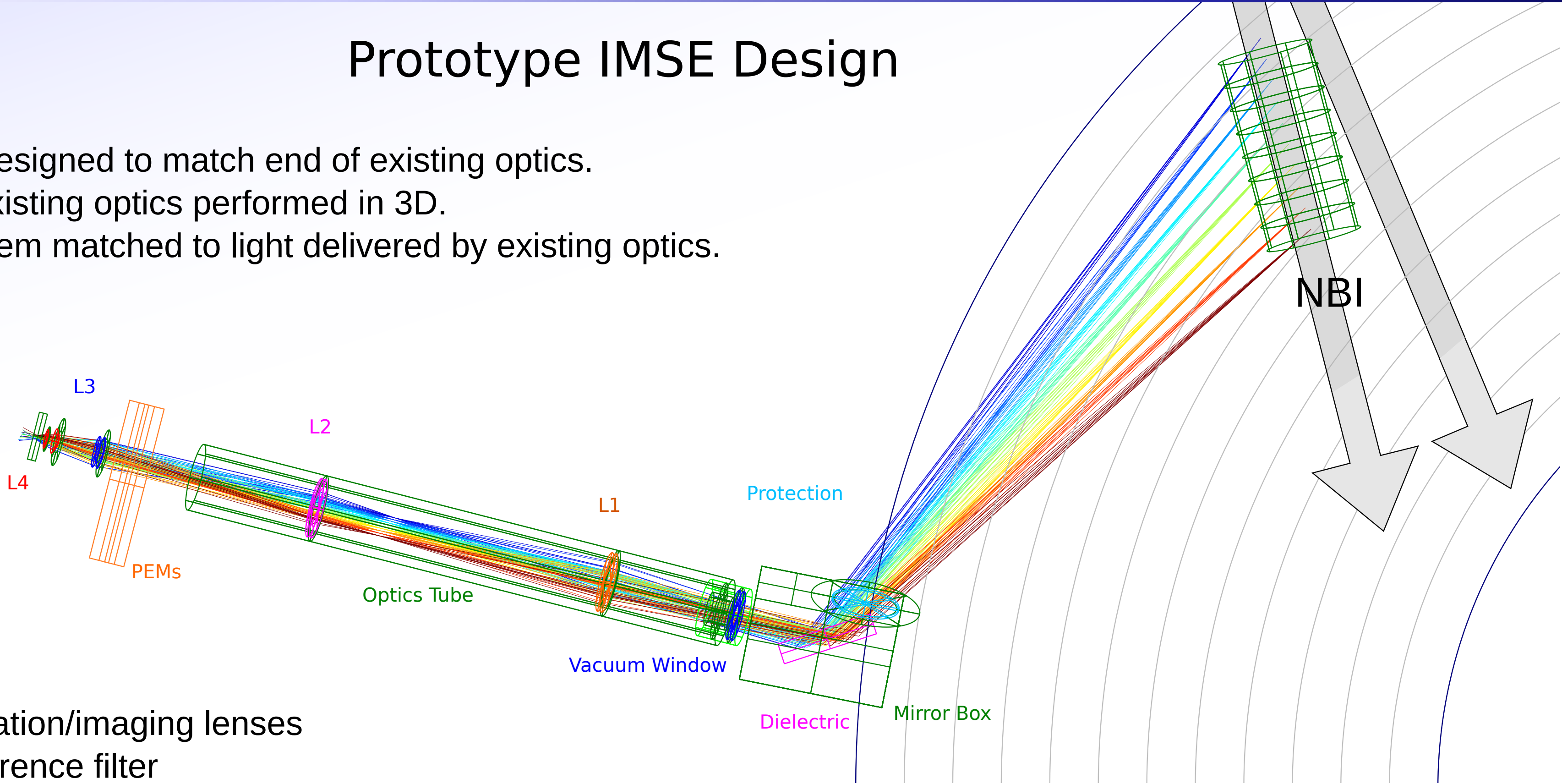
Prototype IMSE designed to match end of existing optics.
- Ray tracing of existing optics performed in 3D.
- New optical system matched to light delivered by existing optics.

Determination of:
- Optimal collimation/imaging lenses
- Optimal interference filter
- Expected polarisation angle

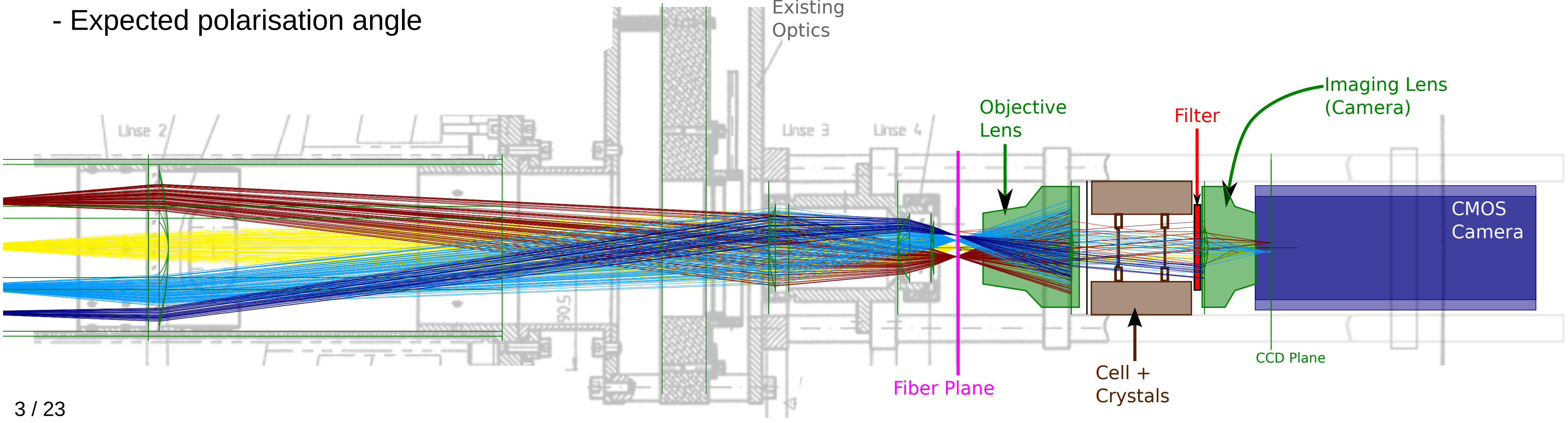


Prototype IMSE Design

Prototype IMSE designed to match end of existing optics.
- Ray tracing of existing optics performed in 3D.
- New optical system matched to light delivered by existing optics.



Determination of:
- Optimal collimation/imaging lenses
- Optimal interference filter
- Expected polarisation angle

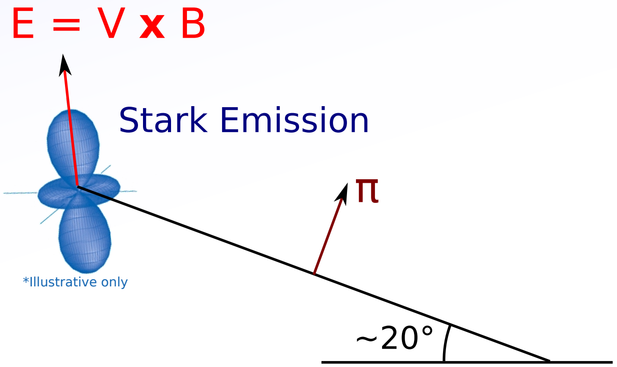


Ray-traced forward model

To fully understand effects of optics, everything put into ray-tracing model:

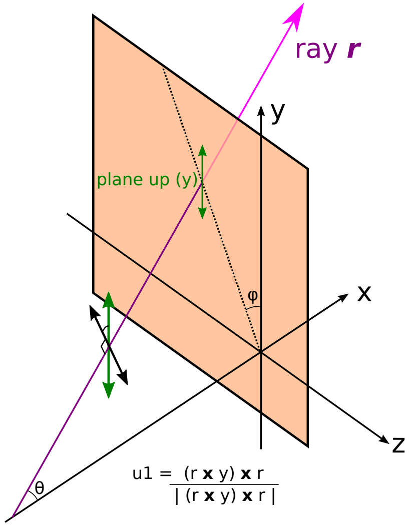
1) Field of view effects:

- Subtlety of how polarisation is created, defined and measured.



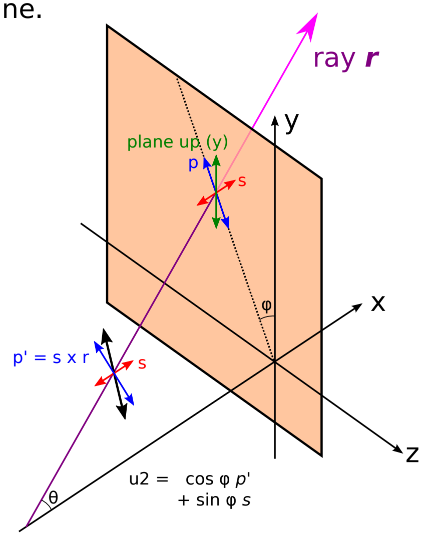
POLAR consistent

u1 = Nearest vector to 'up' ≠



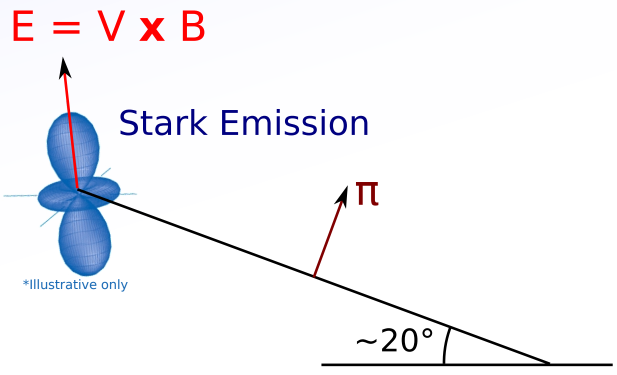
PLANAR consistent

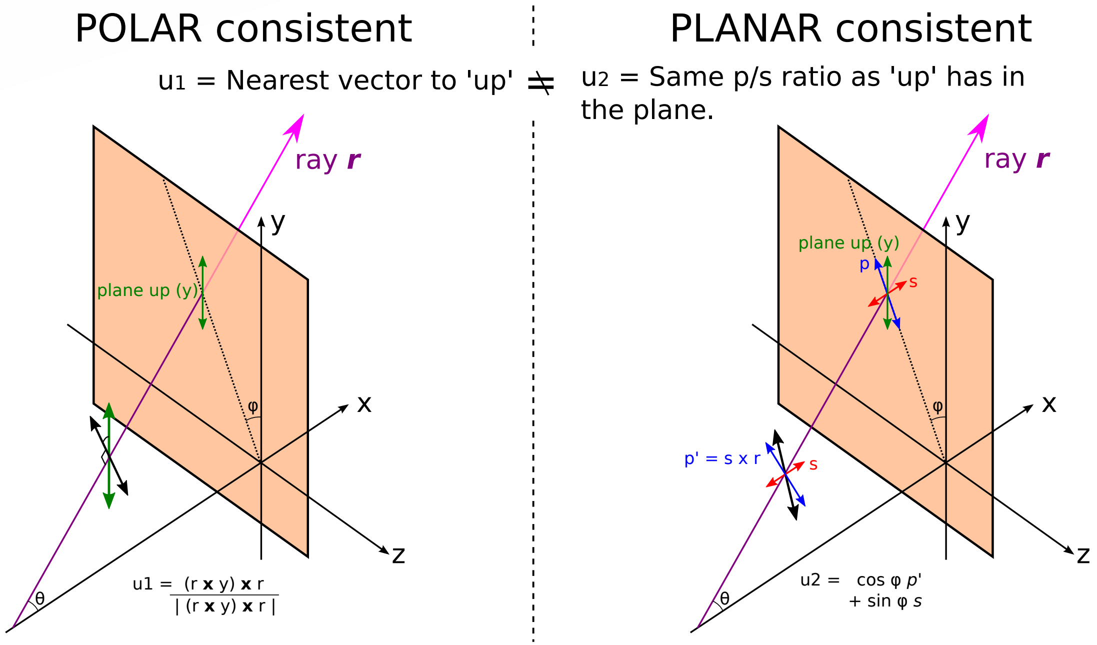
u2 = Same p/s ratio as 'up' has in the plane.

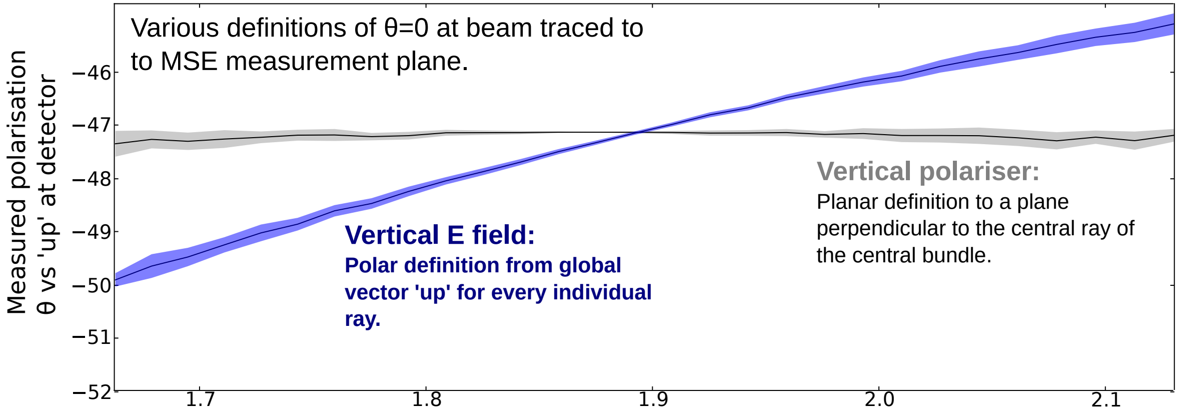


Ray-traced forward model

To fully understand effects of optics, everything put into ray-tracing model:
1) Field of view effects:
- Subtlety of how polarisation is created, defined and measured.




POLAR consistent
u1 = Nearest vector to 'up' ≠
PLANAR consistent
u2 = Same p/s ratio as 'up' has in the plane.
ray ***r***
y
x
z
plane up (y)
φ
θ
u1 = (r **x** y) **x** r / | (r **x** y) **x** r |
p
s
p' = s x r
u2 = cos φ *p'* + sin φ *s*



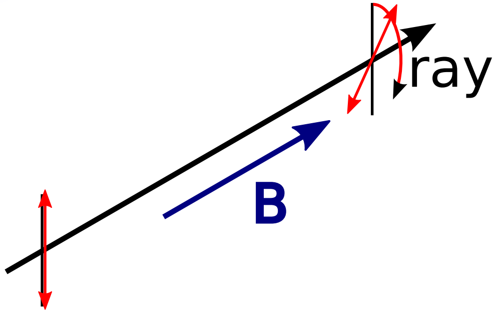
Various definitions of θ=0 at beam traced to to MSE measurement plane.

Vertical E field:
Polar definition from global vector 'up' for every individual ray.

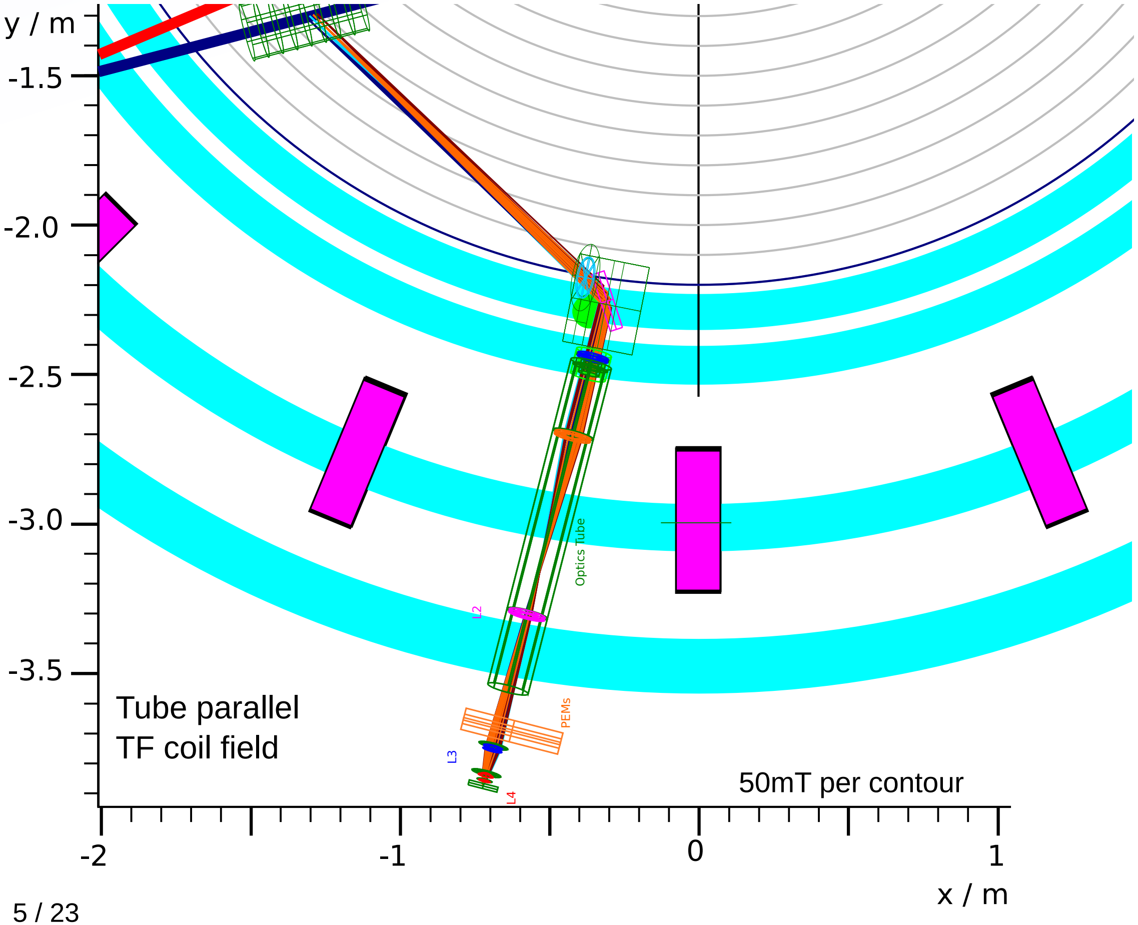
Vertical polariser:
Planar definition to a plane perpendicular to the central ray of the central bundle.

Polarisation effects

To fully understand effects of optics, everything put into ray-tracing model:
2) Faraday rotation
- Fields due to both:
  - TF coils: Strong, but static
  - PF coils. Weak, but time-varying.

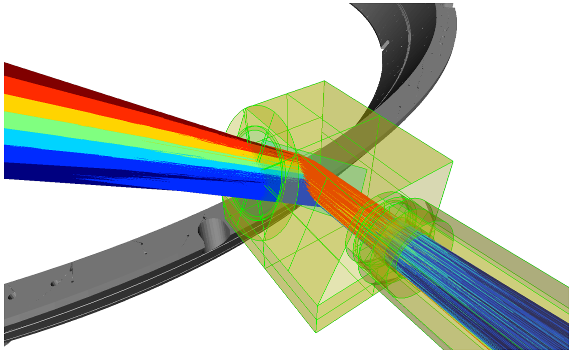


Biot-savart B.l (Minerva)



Tube parallel
TF coil field

50mT per contour

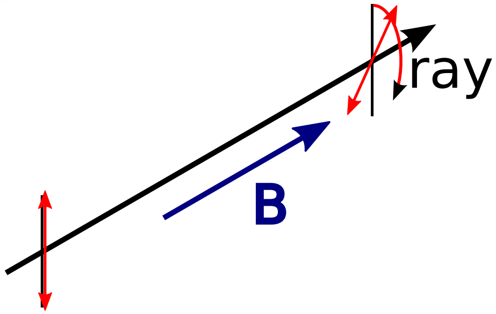


Polarisation effects

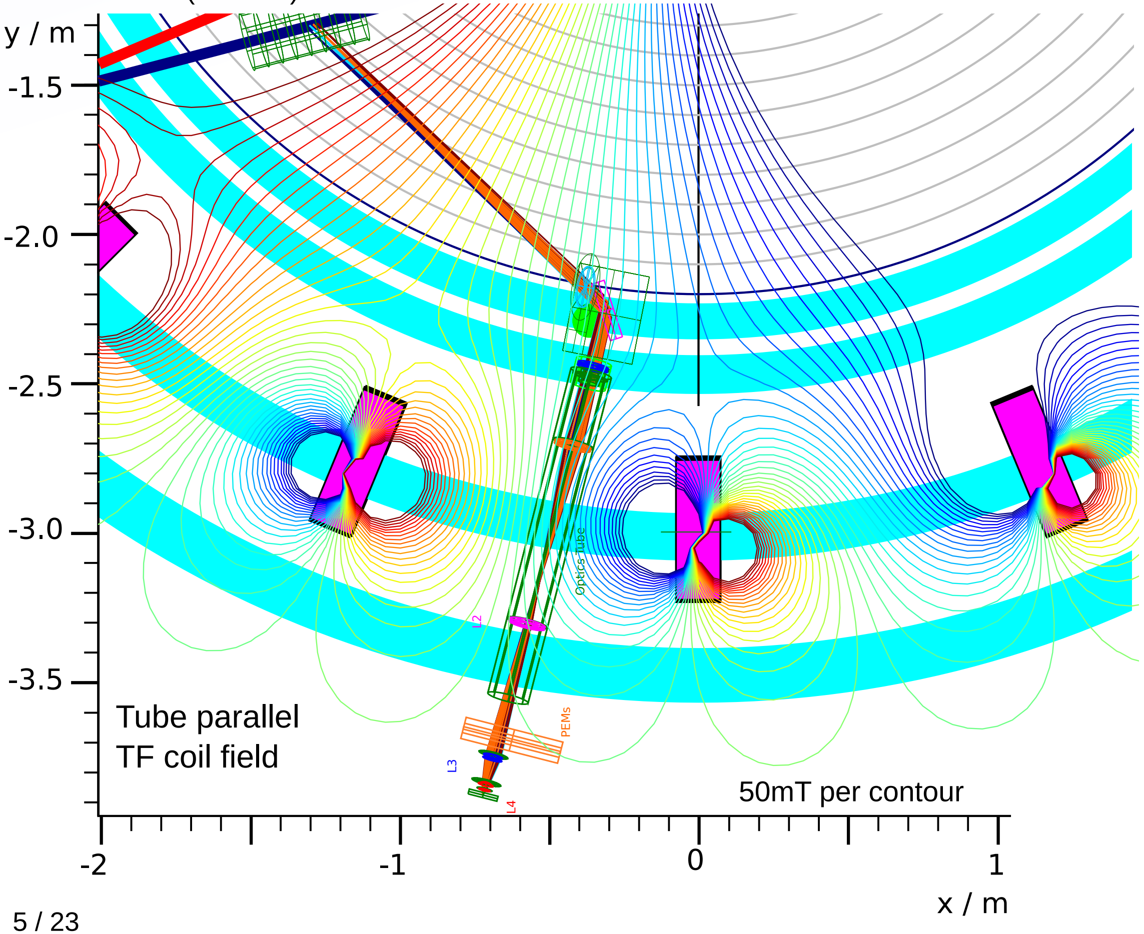
To fully understand effects of optics, everything put into ray-tracing model:

2) Faraday rotation

- Fields due to both:
    - TF coils: Strong, but static
    - PF coils. Weak, but time-varying.

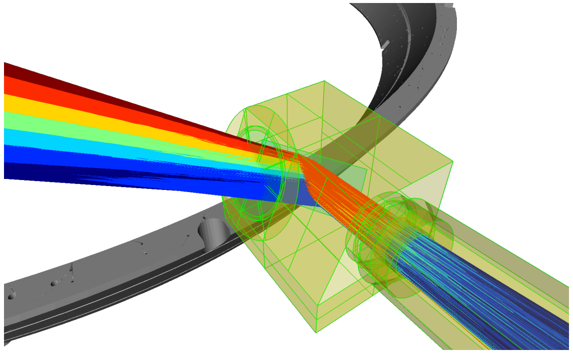


Biot-savart B.l (Minerva)



Tube parallel
TF coil field

50mT per contour

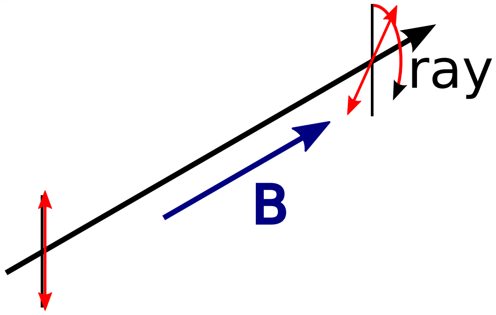


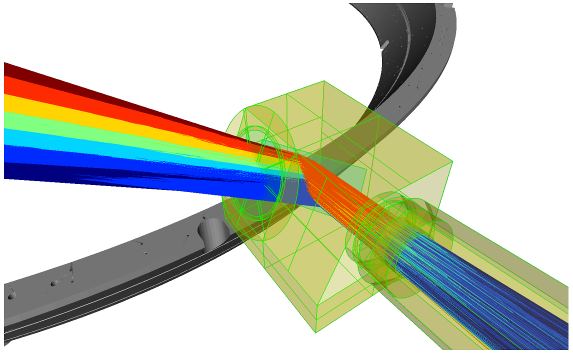
Polarisation effects

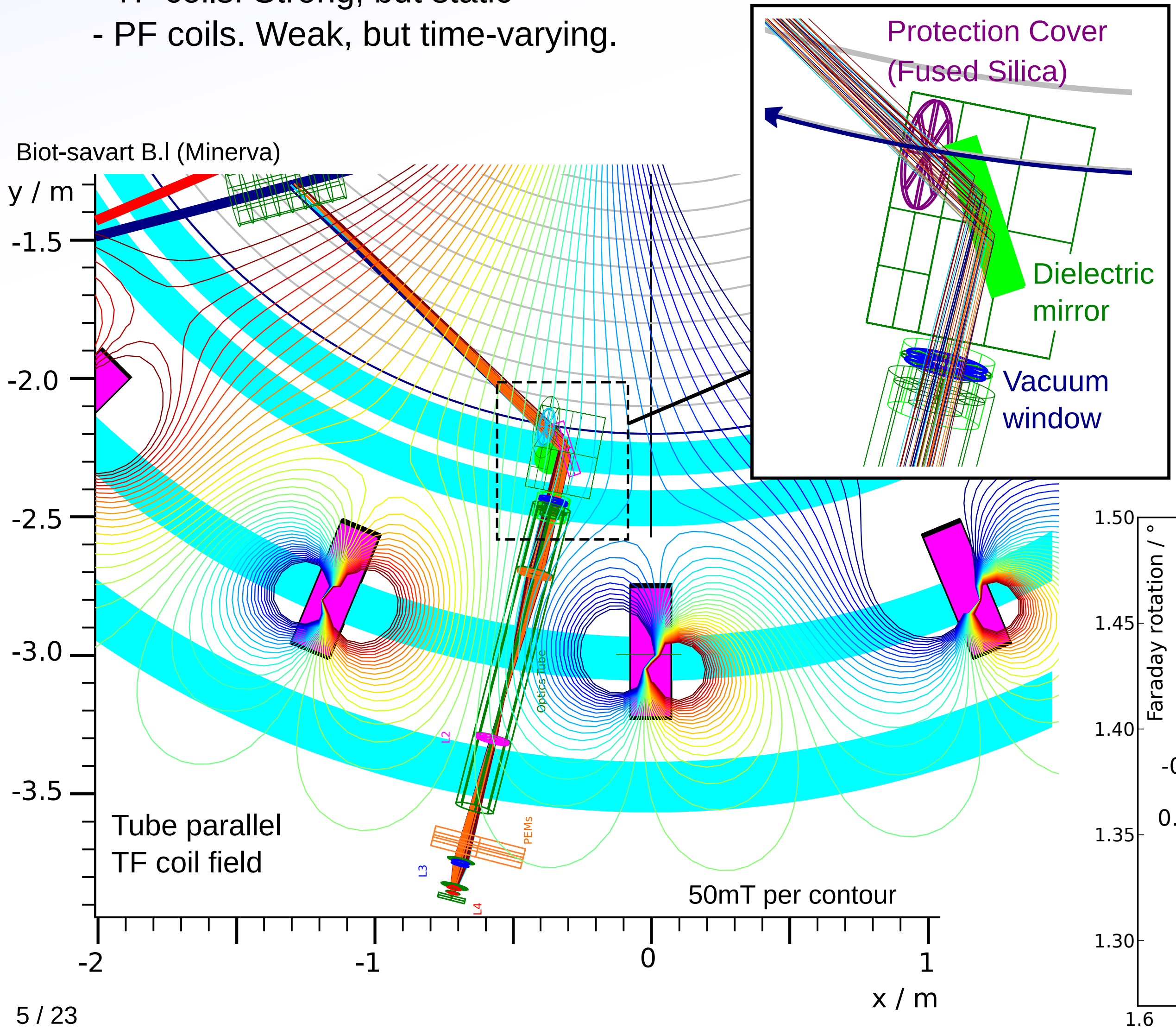
To fully understand effects of optics, everything put into ray-tracing model:

2) Faraday rotation

- Fields due to both:
    - TF coils: Strong, but static
    - PF coils. Weak, but time-varying.

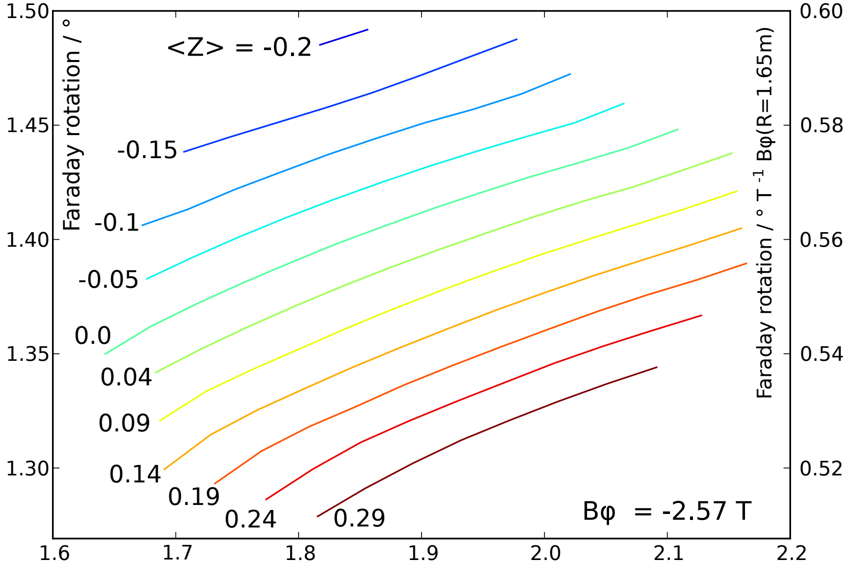






Tube parallel
TF coil field

50mT per contour



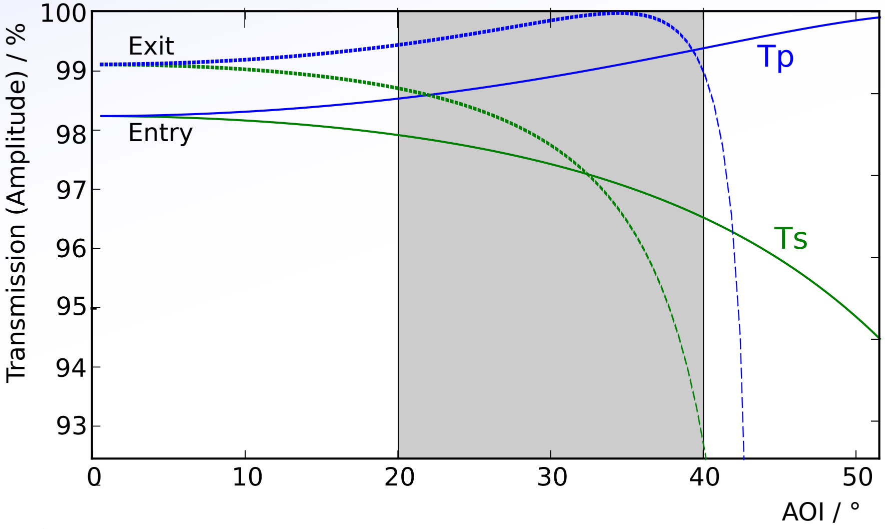
Oliver Ford
IPP Greifswald

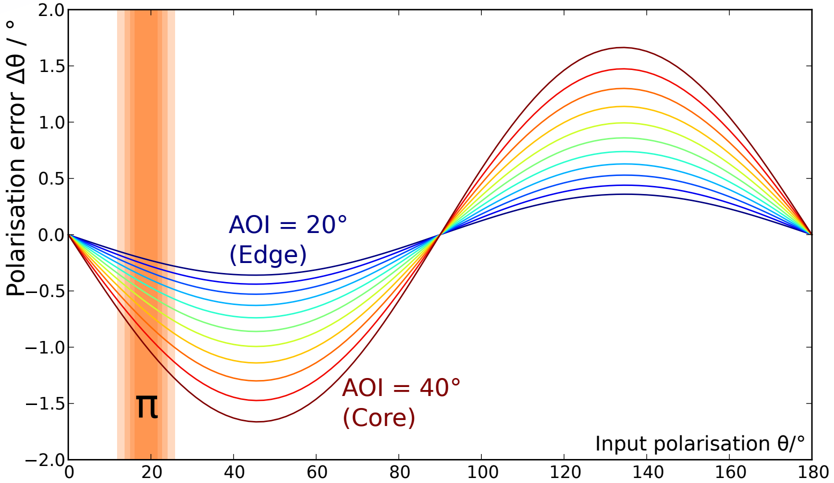
gmds/AUG/29317

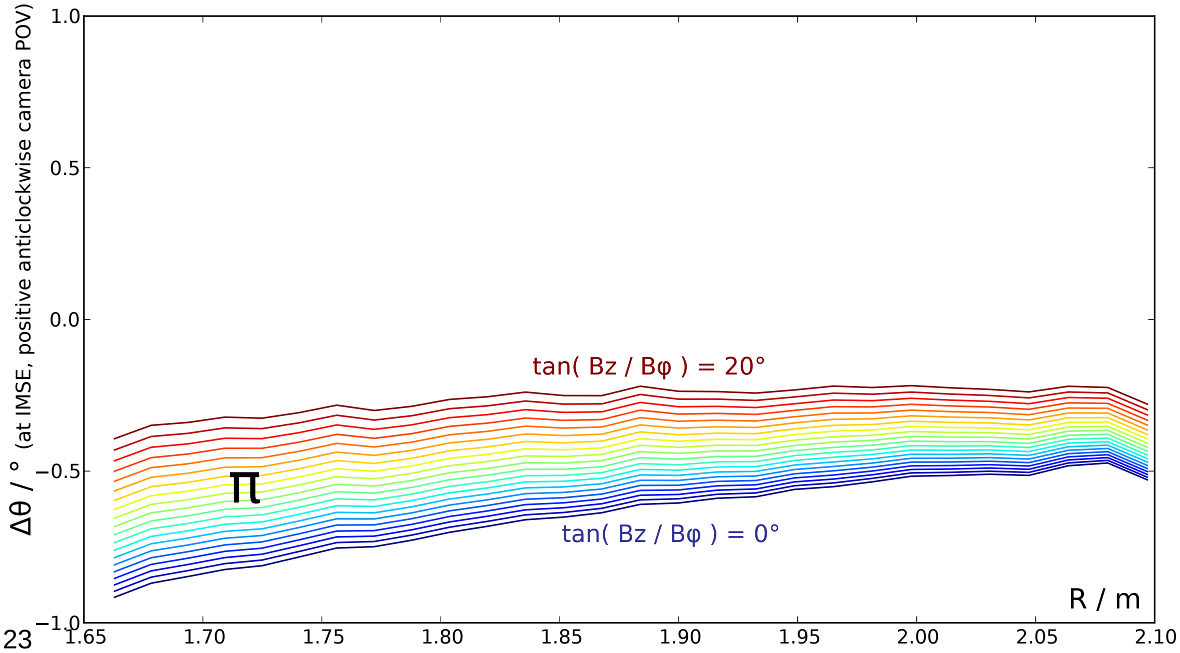
Polarisation effects

3) Fresnel coefficient effects (e.g. uncoated protection cover)

- Variation of transmission/reflectance with AOI --> Non-linear rotation of polarisation







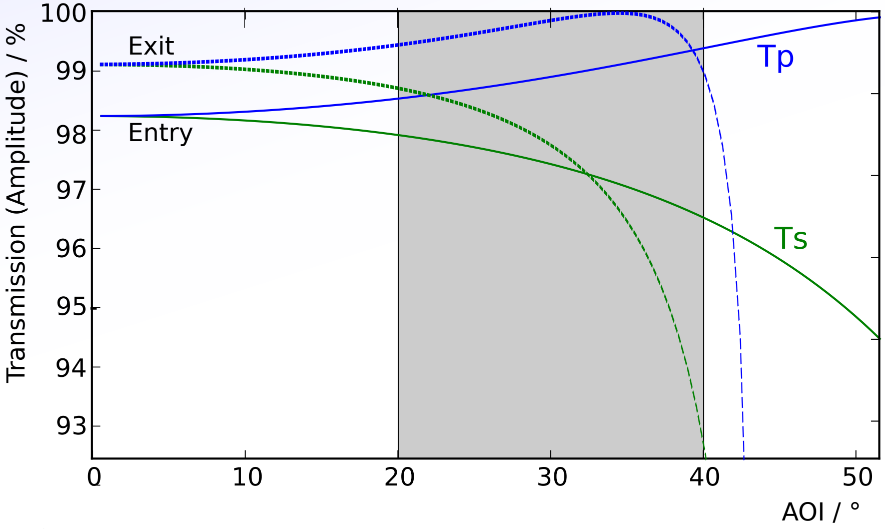
Mostly cancels for σ+π (IMSE), but is important for MSE

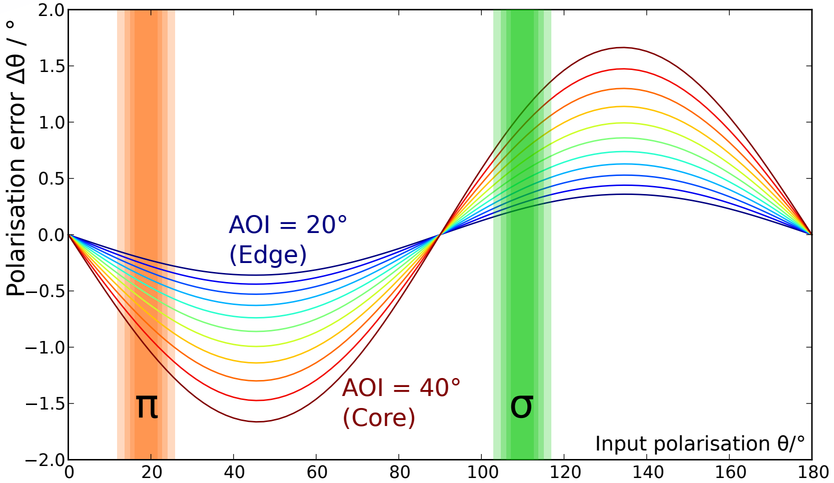
Oliver Ford
IPP Greifswald
gmds/AUG/29317

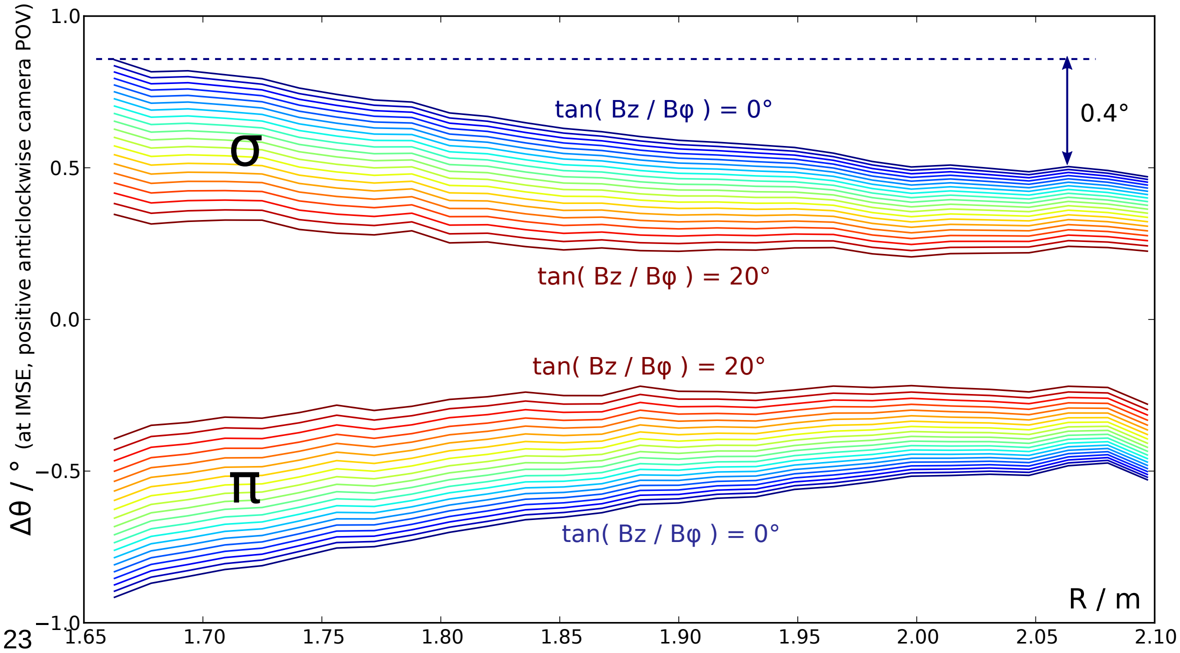
Polarisation effects

3) Fresnel coefficient effects (e.g. uncoated protection cover)

- Variation of transmission/reflectance with AOI --> Non-linear rotation of polarisation



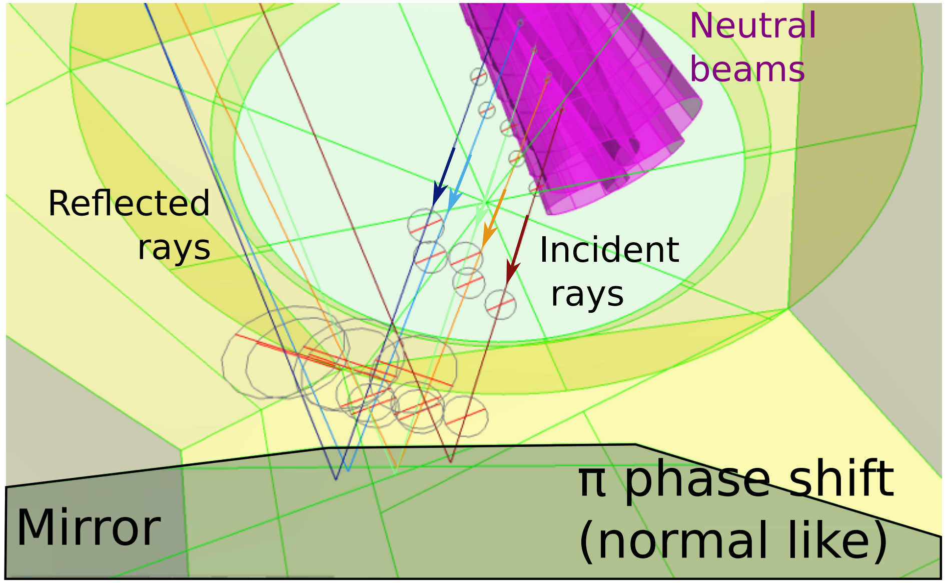




Mostly cancels for σ+π (IMSE), but is important for MSE

Polarisation effects

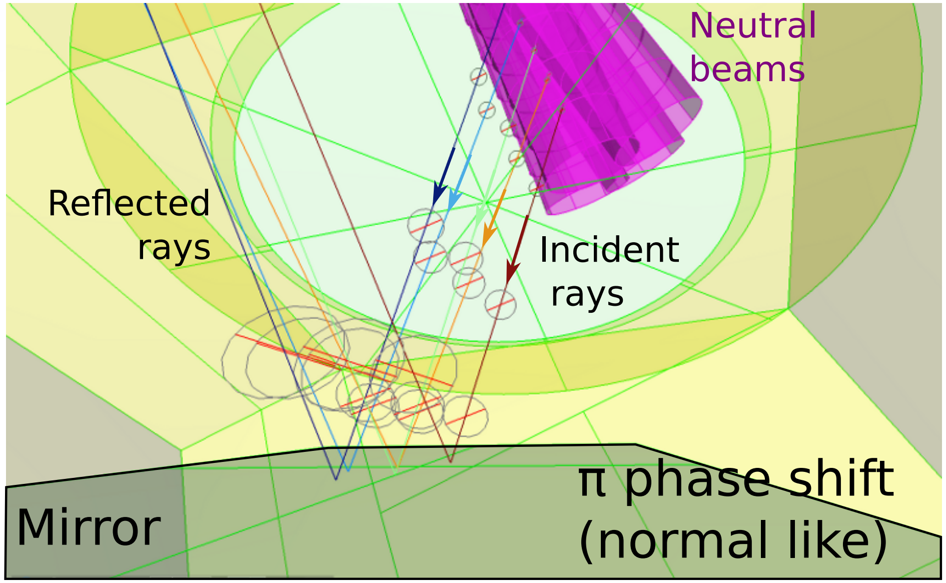
4) Mirrors

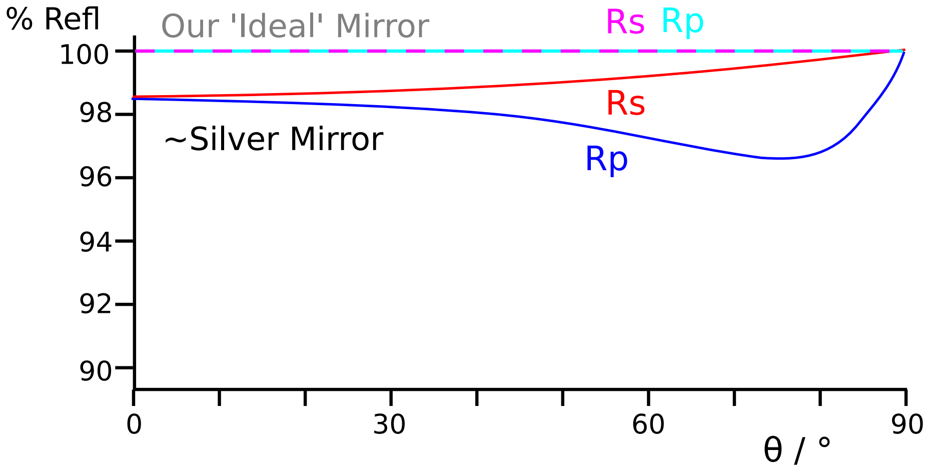


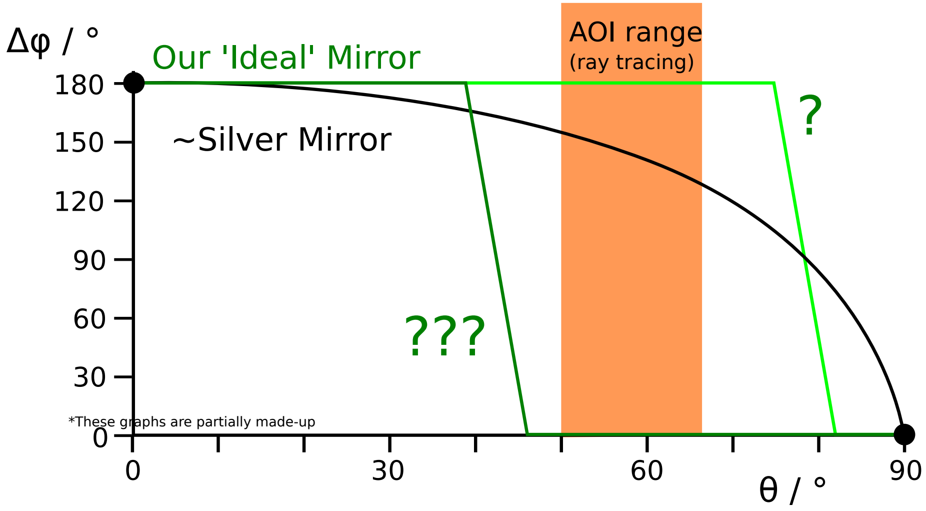
Polarisation effects

4) Mirrors

- Simple 'ideal π phase shift' for dielectric mirror.
- Later with measured spectro-polarisation transfer properties of mirror.



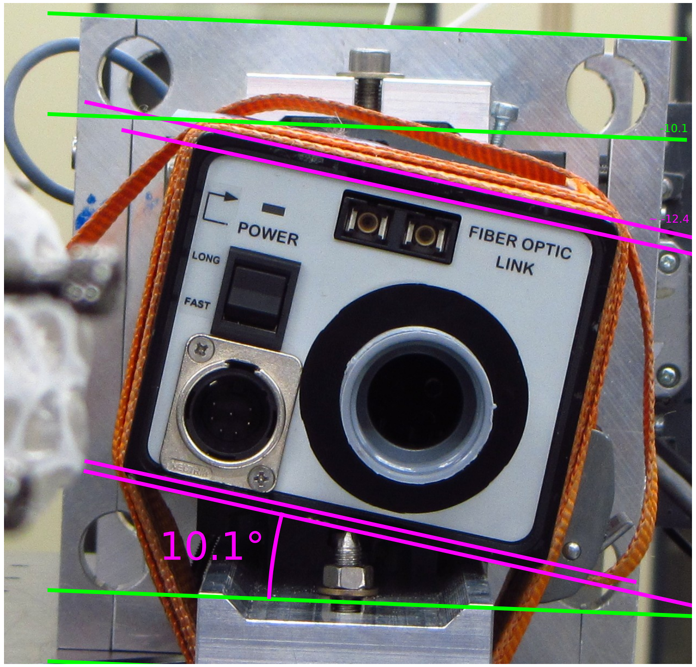


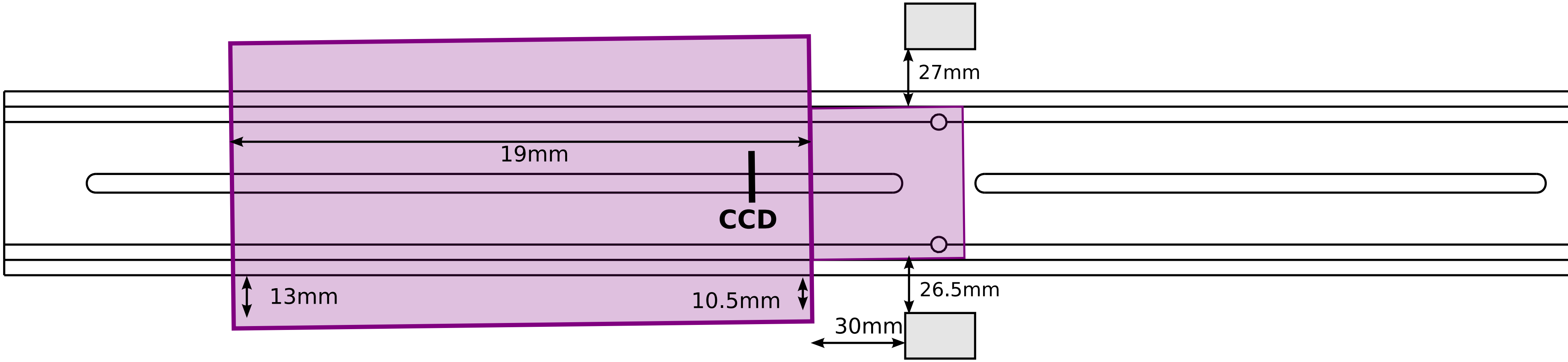


*These graphs are partially made-up

Polarisation effects

5) Mechanical angles/positions
- Angle/position of camera - Measured with uncertainties
- Small inaccuracies in lens/mirror positions - fitted...

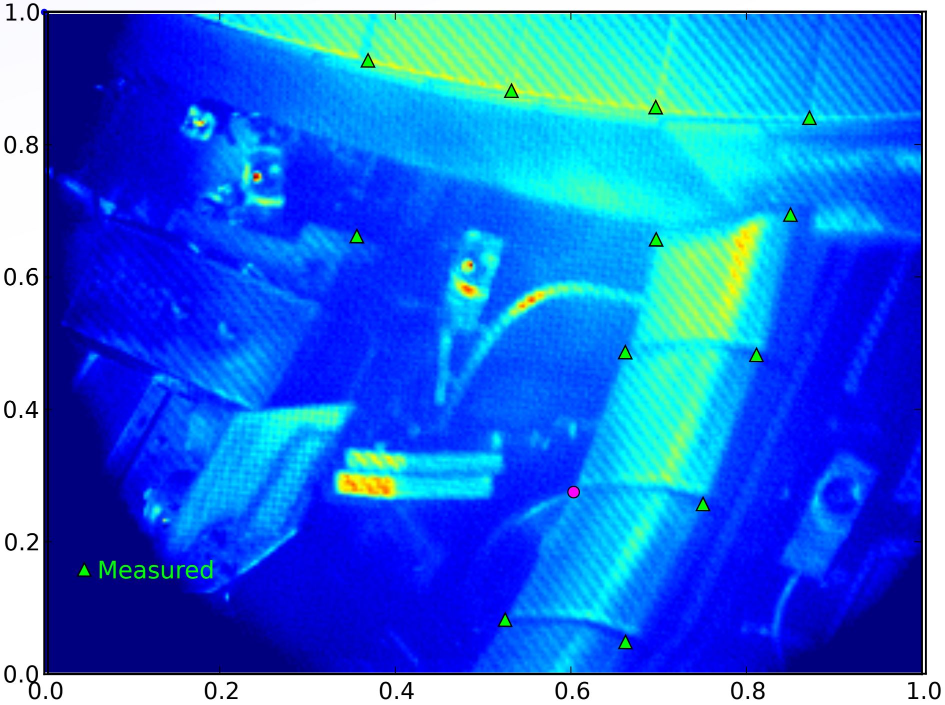


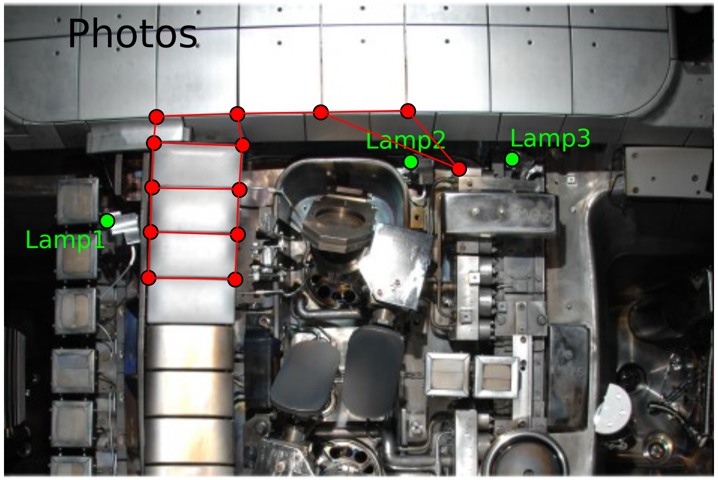


Ray-tracing match

- Fit mechanical model parameters to match ray-traced CAD 3D points.
- Match vignetting curves from different stages in system

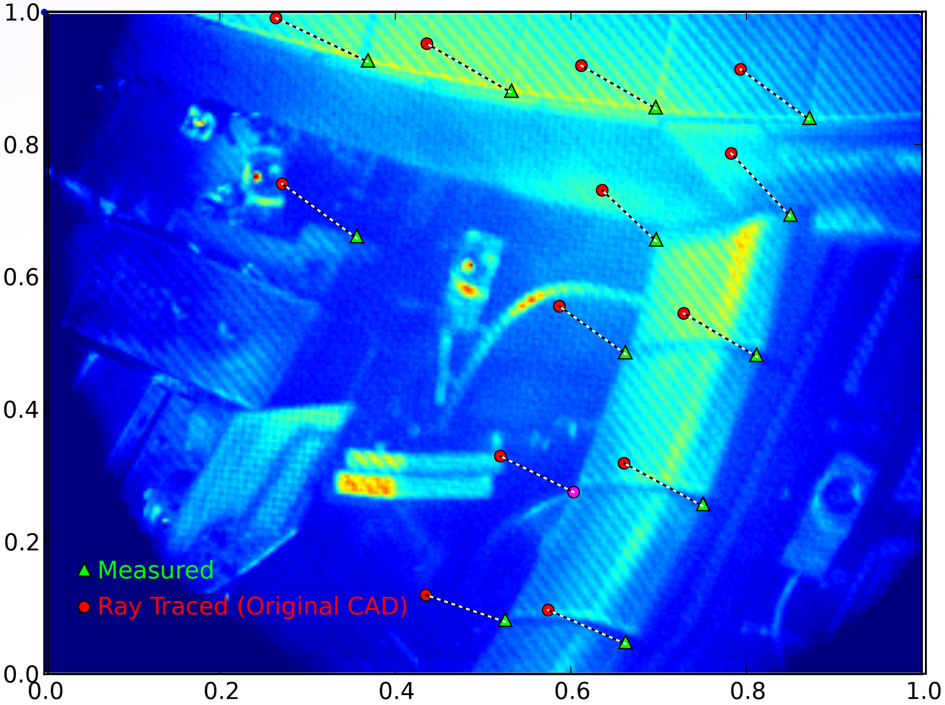
    --> Determines most mechanical parameters.

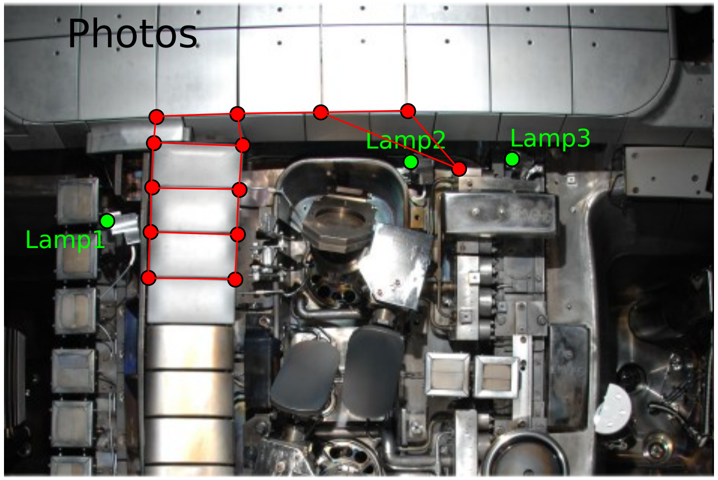




Ray-tracing match

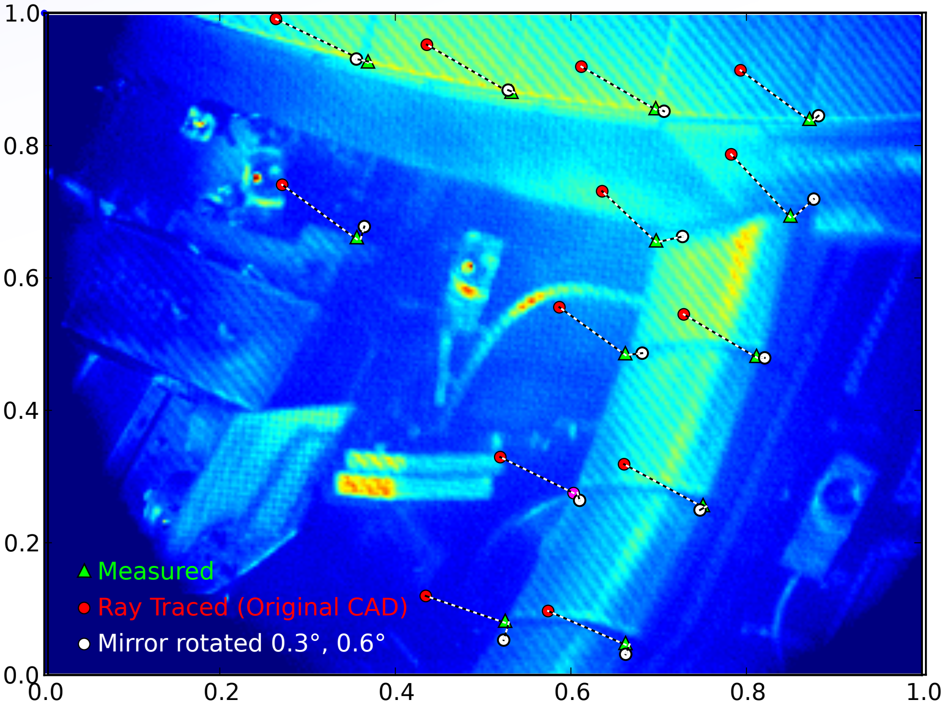
- Fit mechanical model parameters to match ray-traced CAD 3D points.
- Match vignetting curves from different stages in system
  --> Determines most mechanical parameters.

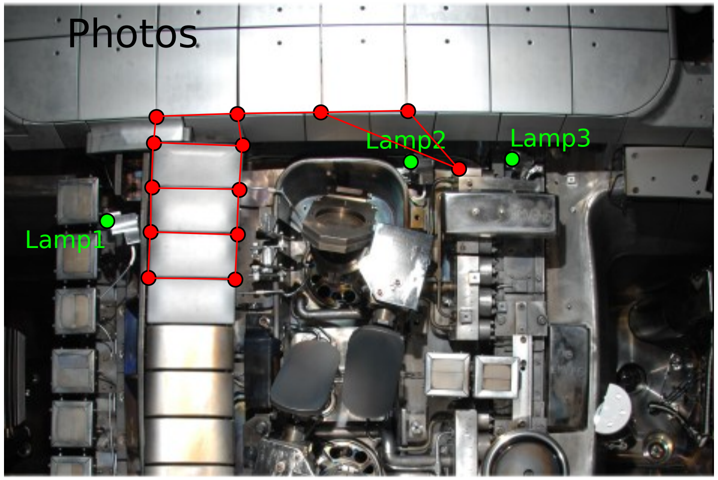




Ray-tracing match

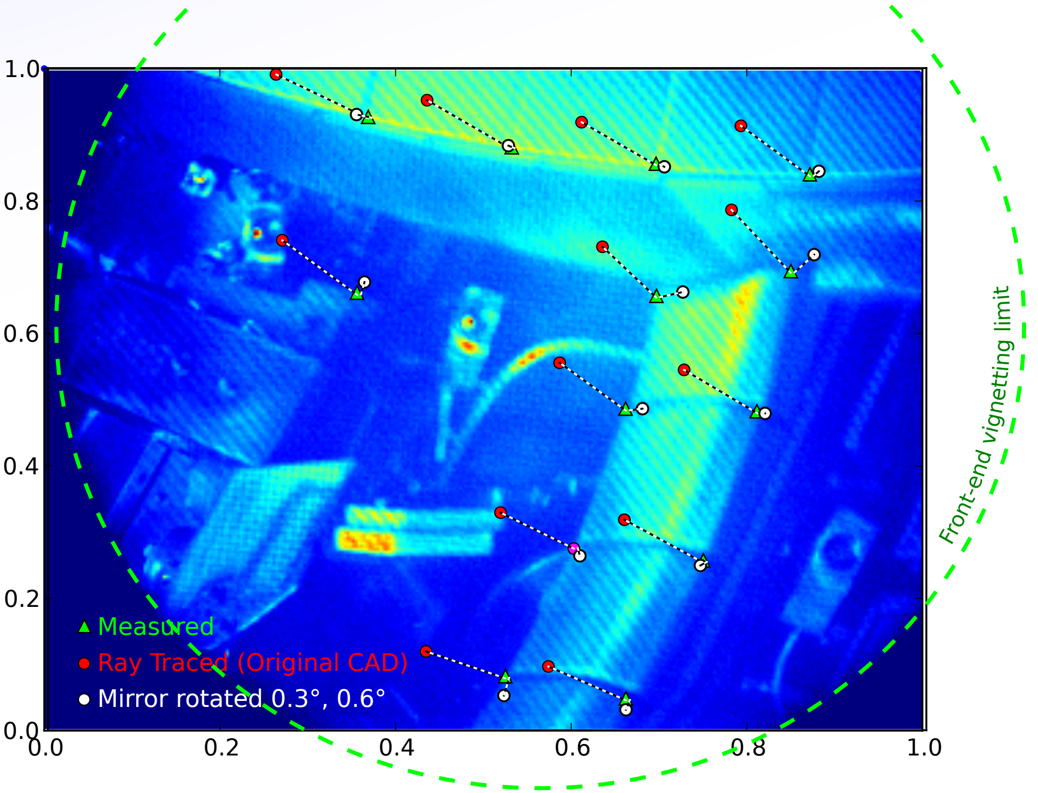
- Fit mechanical model parameters to match ray-traced CAD 3D points.
- Match vignetting curves from different stages in system
    --> Determines most mechanical parameters.

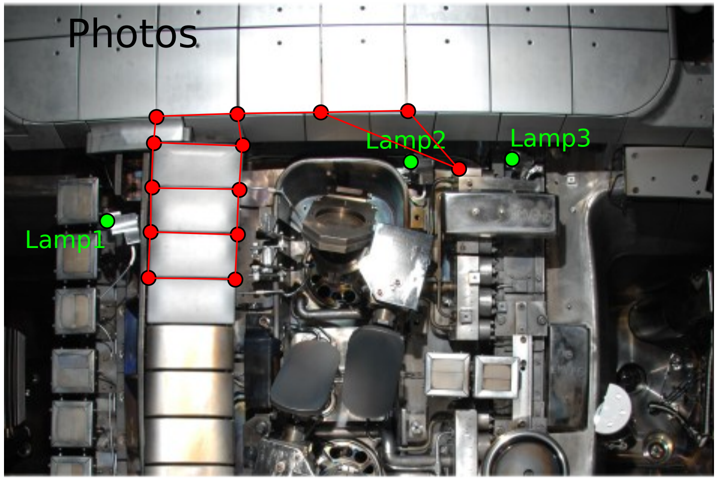




Ray-tracing match

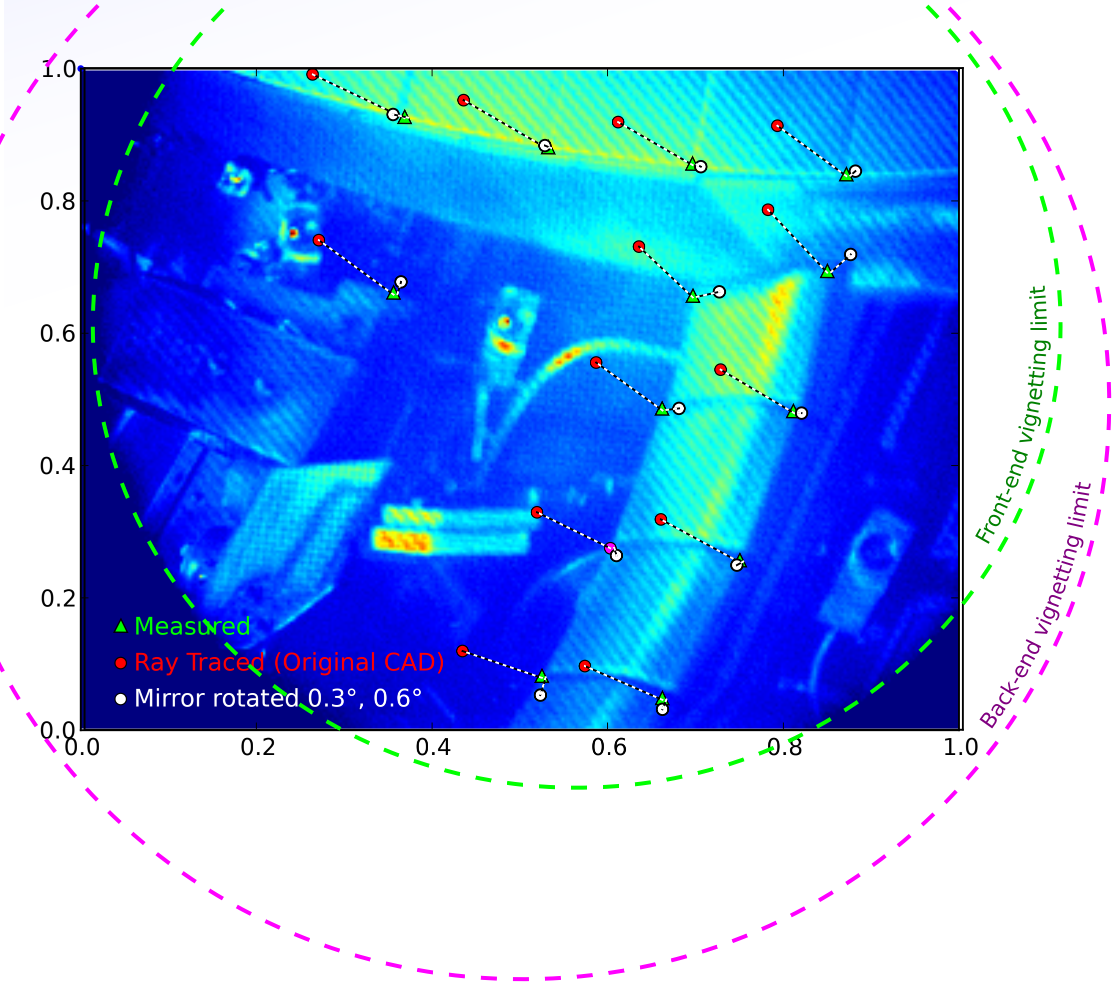
- Fit mechanical model parameters to match ray-traced CAD 3D points.
- Match vignetting curves from different stages in system
  --> Determines most mechanical parameters.

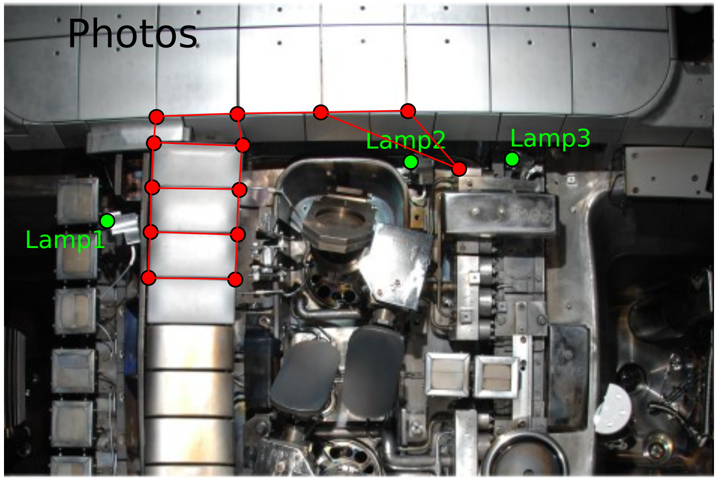




Ray-tracing match

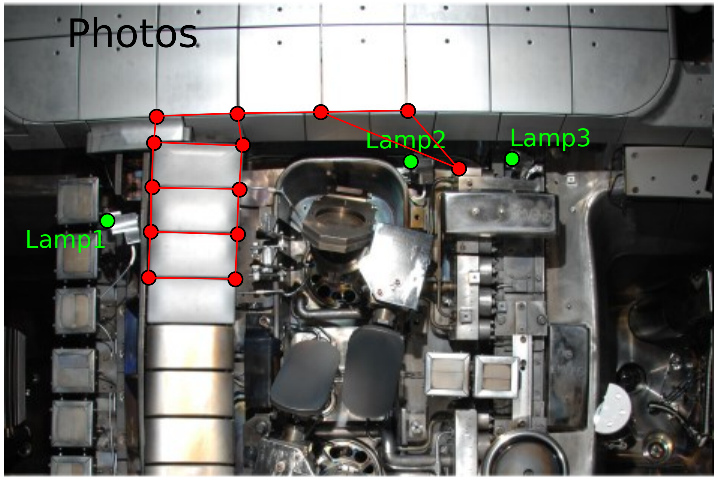
- Fit mechanical model parameters to match ray-traced CAD 3D points.
- Match vignetting curves from different stages in system
  --> Determines most mechanical parameters.

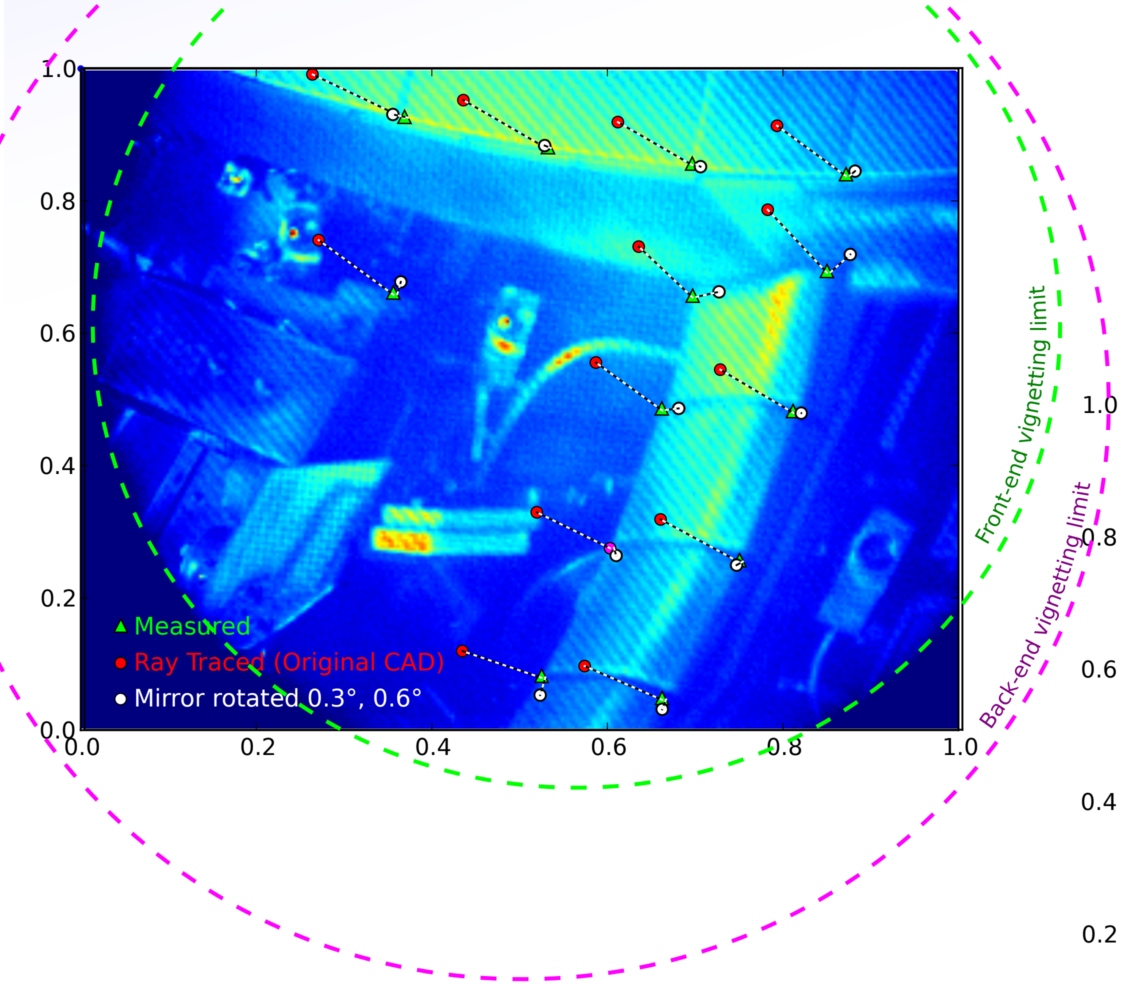




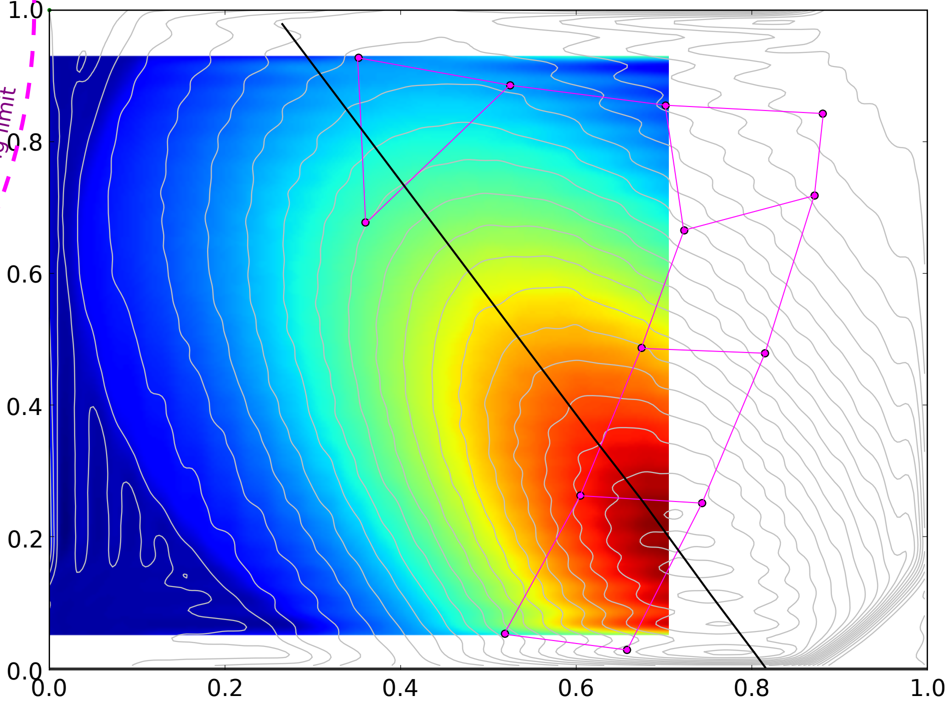
Ray-tracing match

- Fit mechanical model parameters to match ray-traced CAD 3D points.
- Match vignetting curves from different stages in system
  --> Determines most mechanical parameters.





Correctly predicts beam emission intensity axis:

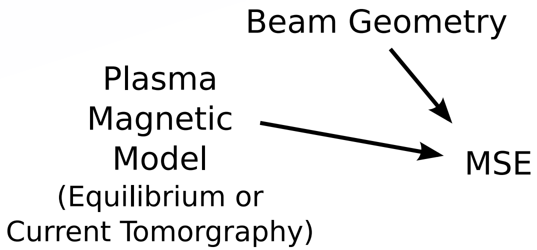


Model agreement

- Final prediction entirely without calibration

Model agreement

- Final prediction entirely without calibration

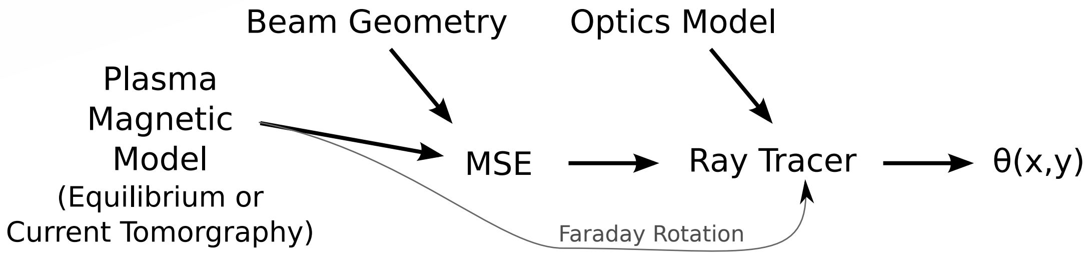

Beam Geometry

Plasma
Magnetic
Model
(Equilibrium or
Current Tomorgraphy)

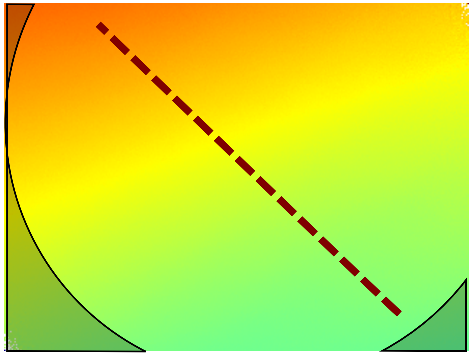
MSE

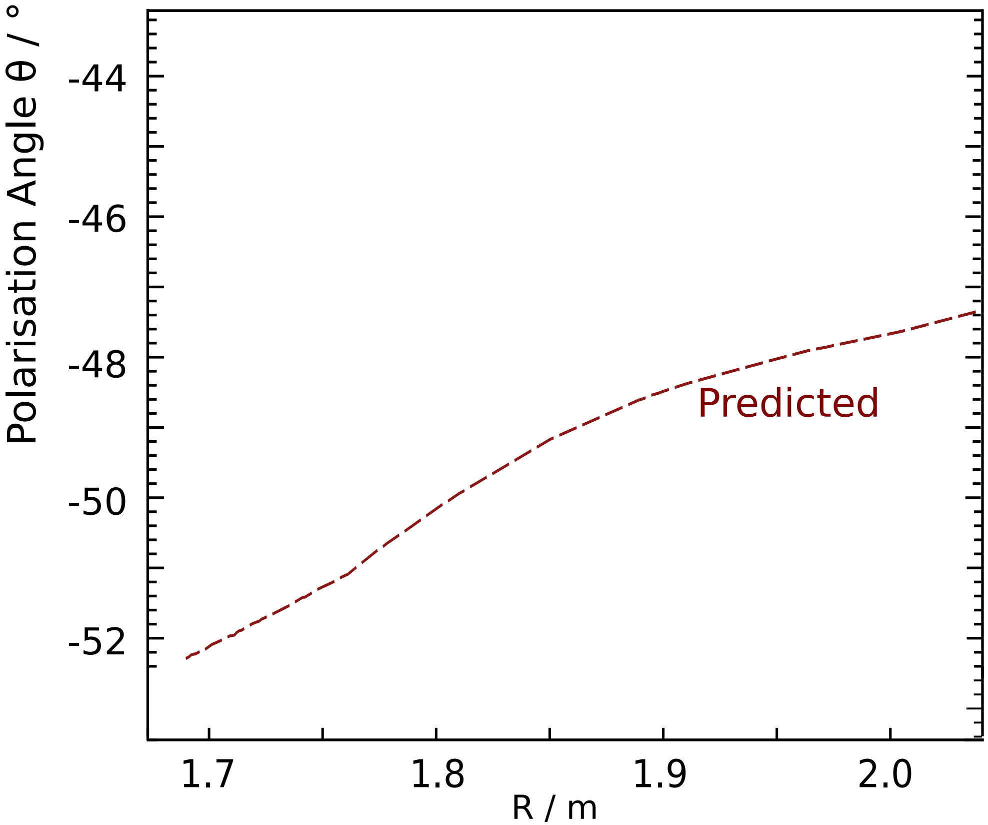
Model agreement

- Final prediction entirely without calibration



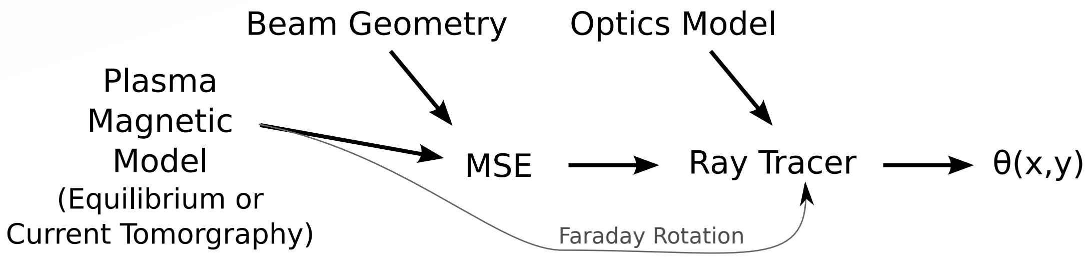
Forward Model





Model agreement

- Final prediction entirely without calibration



Beam Geometry

Optics Model

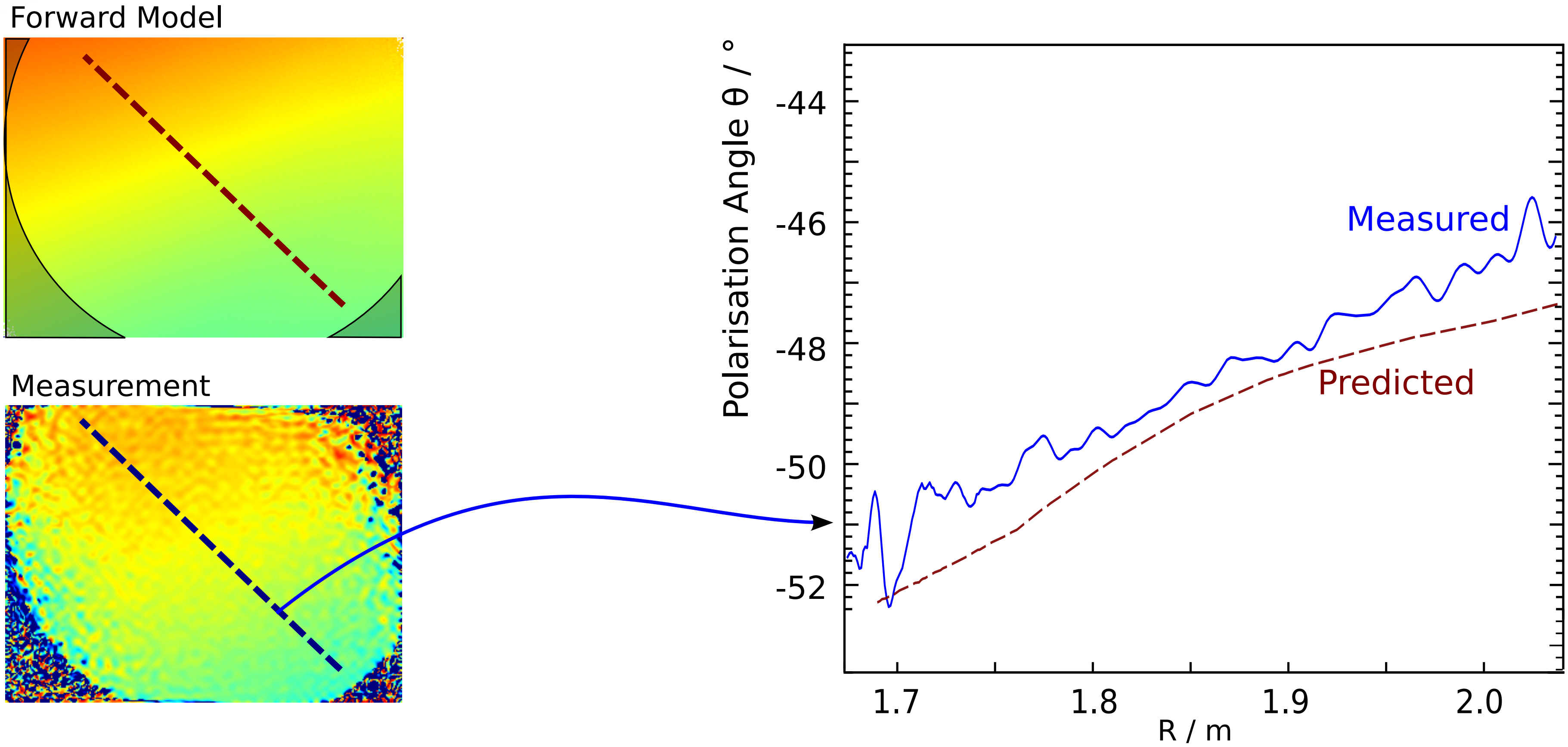
Plasma
Magnetic
Model
(Equilibrium or
Current Tomorgraphy)

MSE

Ray Tracer

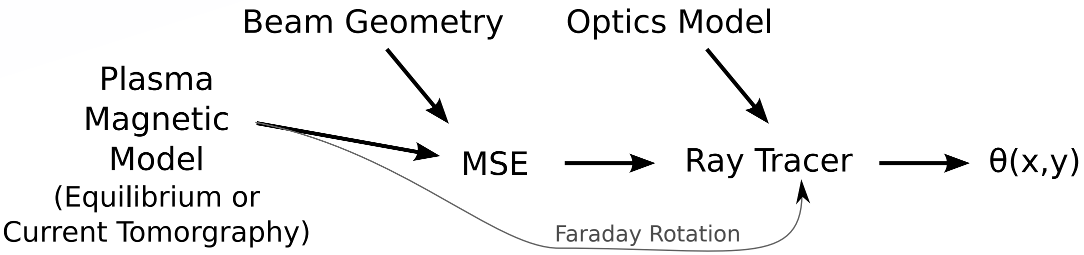
θ(x,y)

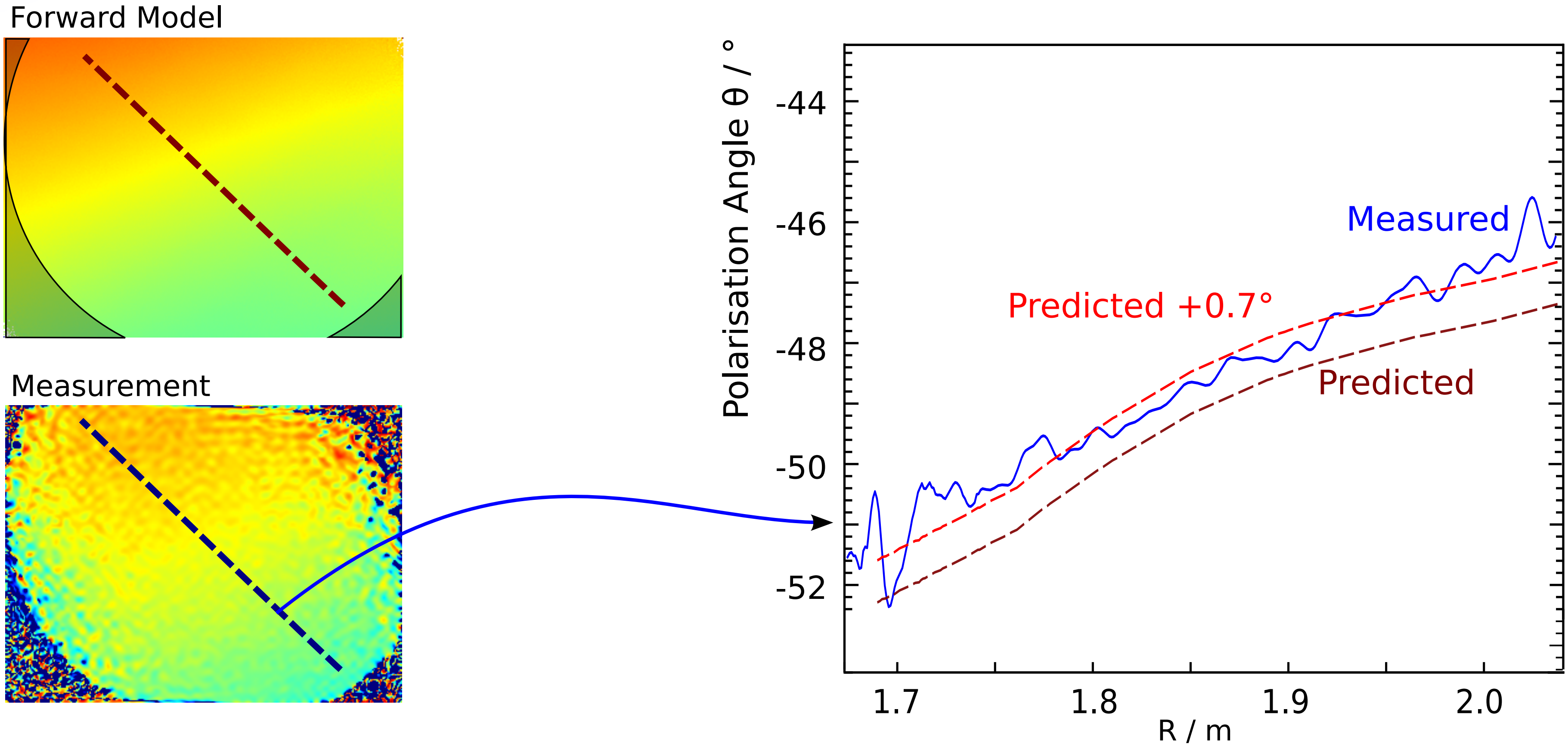
Faraday Rotation



Model agreement

- Final prediction entirely without calibration within ~0.7° of measurement.
- Remaining uncertainties only affect offset.
- *dθ/dr* most important - well modelled.

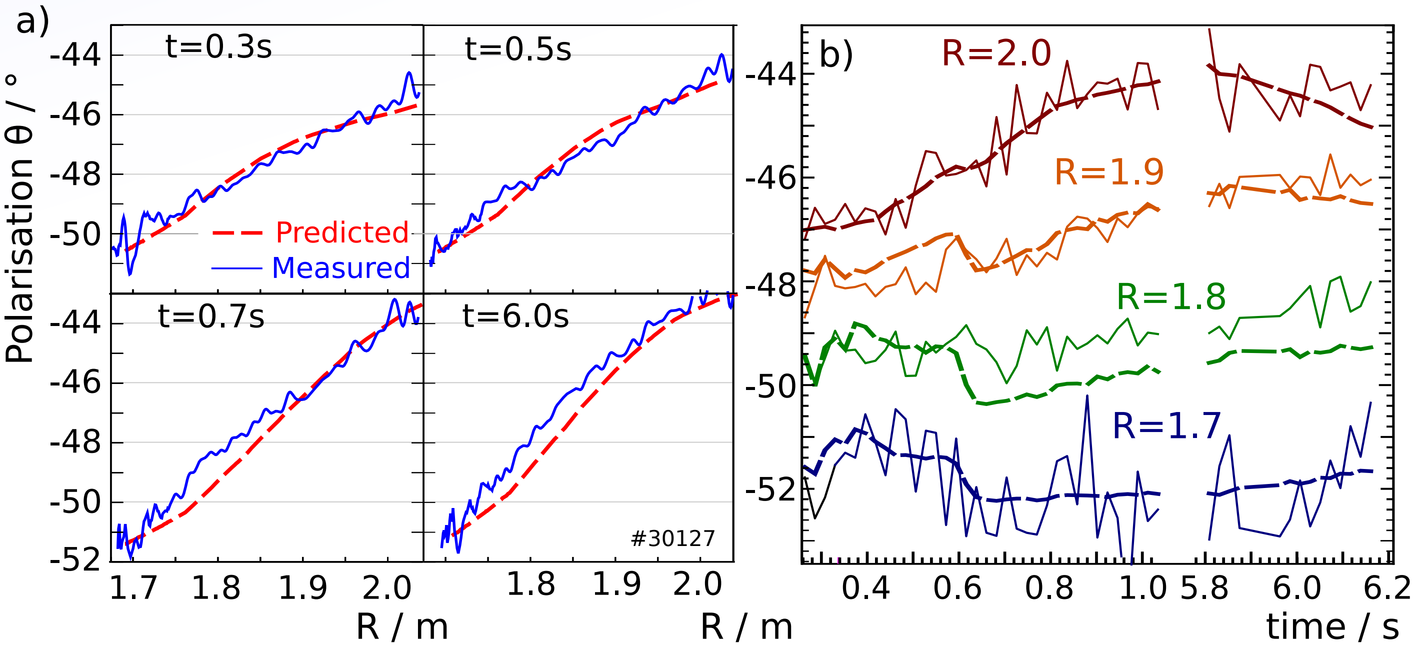



Forward Model

Measurement

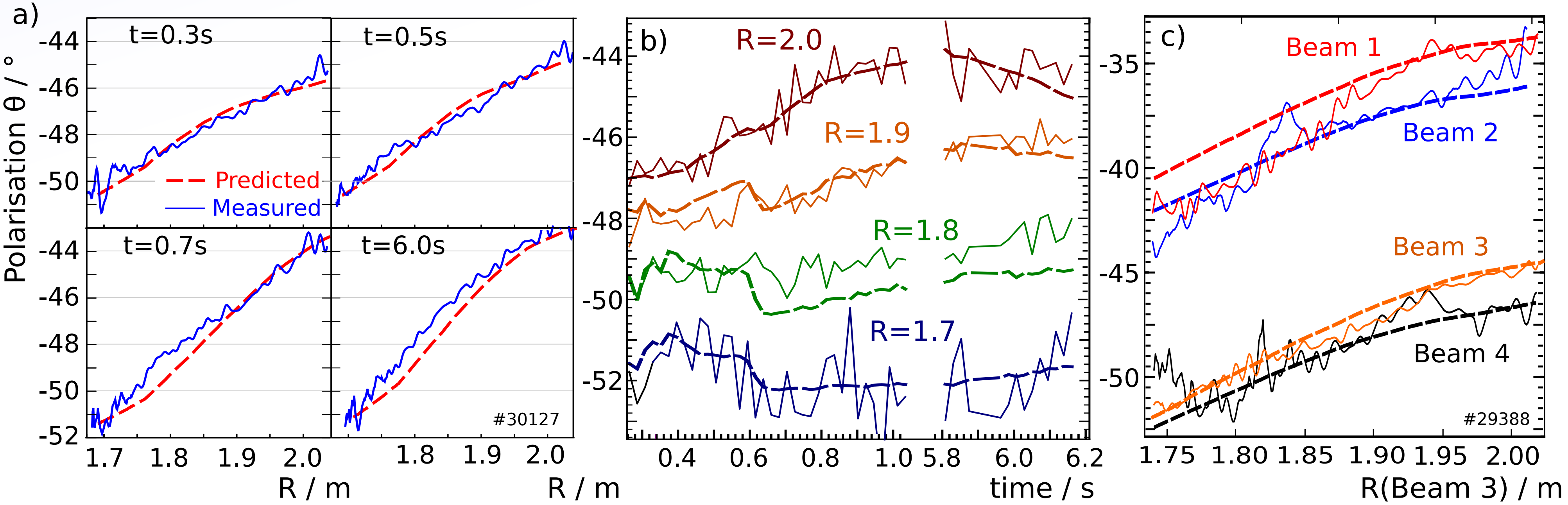
Model agreement

- Remaining disagreement is largely what is unknown in equilibrium code.



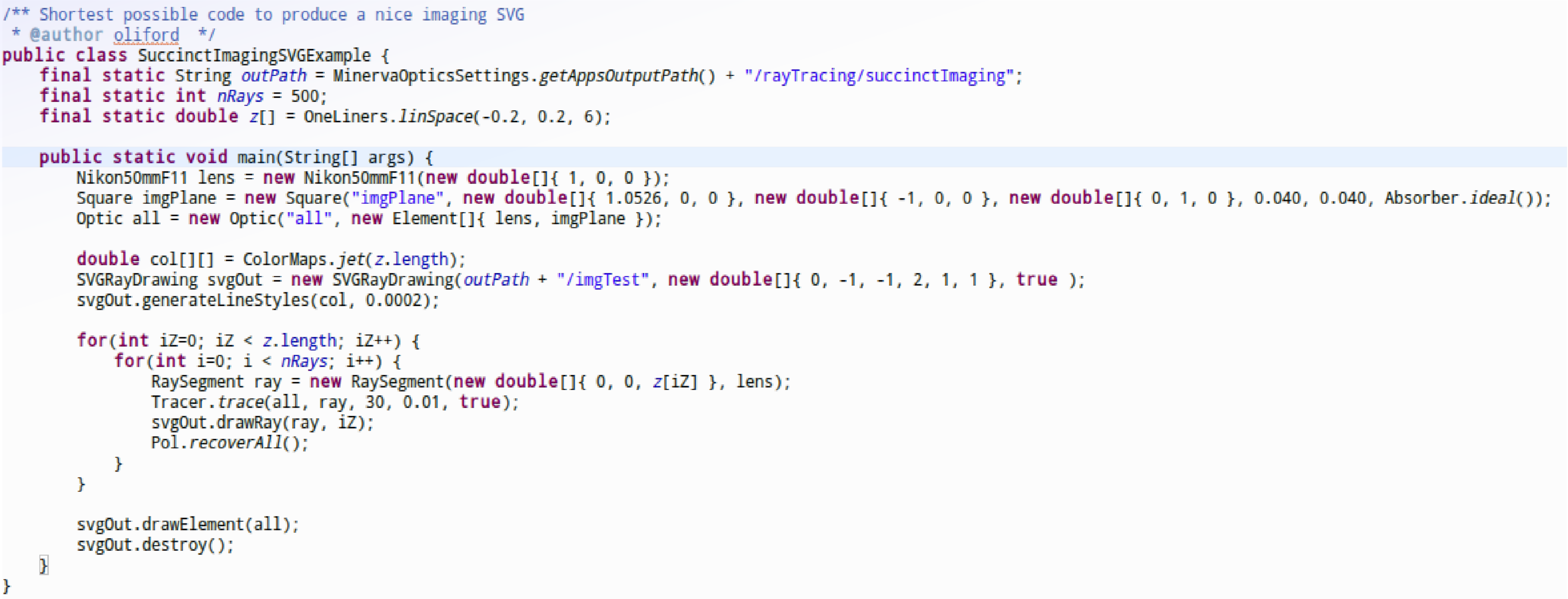
Model agreement

- Remaining disagreement is largely what is unknown in equilibrium code.
- Different beam geometries also approximately predicted
  --> Confirmation of geometric effects, variation over image etc.



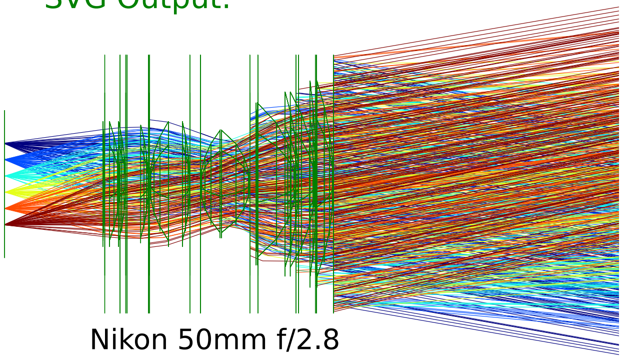
FusionOptics Ray Tracer

- Ray tracing core is a relatively simple to use Java library.



```
/** Shortest possible code to produce a nice imaging SVG
 * @author oliford  */
public class SuccinctImagingSVGExample {
    final static String outPath = MinervaOpticsSettings.getAppsOutputPath() + "/rayTracing/succinctImaging";
    final static int nRays = 500;
    final static double z[] = OneLiners.linSpace(-0.2, 0.2, 6);

    public static void main(String[] args) {
        Nikon50mmF11 lens = new Nikon50mmF11(new double[]{ 1, 0, 0 });
        Square imgPlane = new Square("imgPlane", new double[]{ 1.0526, 0, 0 }, new double[]{ -1, 0, 0 }, new double[]{ 0, 1, 0 }, 0.040, 0.040, Absorber.ideal());
        Optic all = new Optic("all", new Element[]{ lens, imgPlane });

        double col[][] = ColorMaps.jet(z.length);
        SVGRayDrawing svgOut = new SVGRayDrawing(outPath + "/imgTest", new double[]{ 0, -1, -1, 2, 1, 1 }, true );
        svgOut.generateLineStyles(col, 0.0002);

        for(int iZ=0; iZ < z.length; iZ++) {
            for(int i=0; i < nRays; i++) {
                RaySegment ray = new RaySegment(new double[]{ 0, 0, z[iZ] }, lens);
                Tracer.trace(all, ray, 30, 0.01, true);
                svgOut.drawRay(ray, iZ);
                Pol.recoverAll();
            }
        }

        svgOut.drawElement(all);
        svgOut.destroy();
    }
}
```

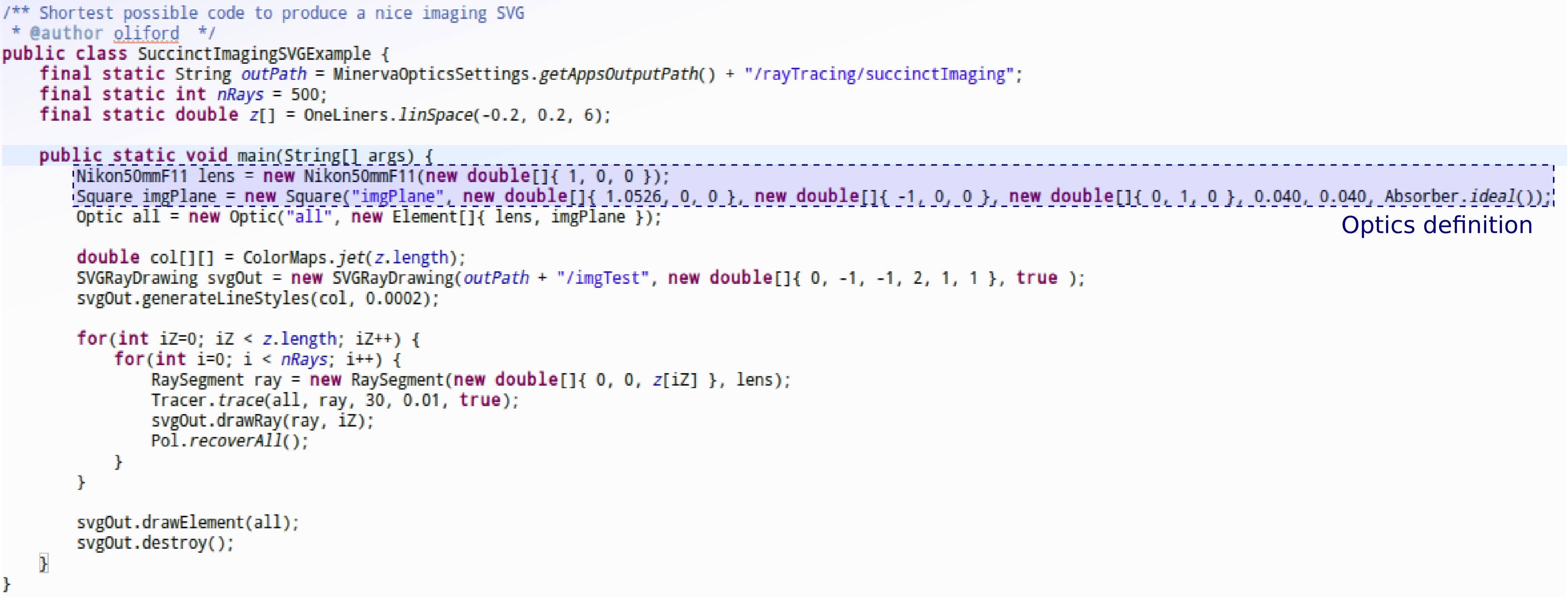
SVG Output:



Nikon 50mm f/2.8

FusionOptics Ray Tracer

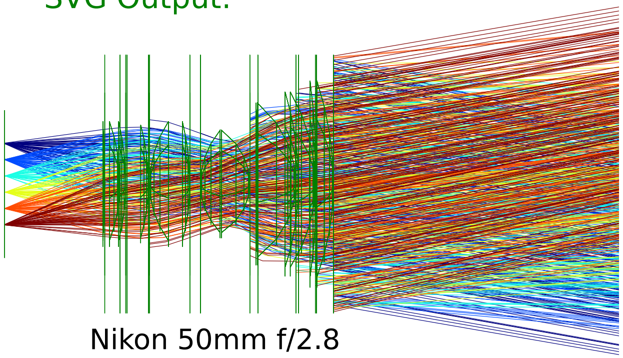
- Ray tracing core is a relatively simple to use Java library.



```
/** Shortest possible code to produce a nice imaging SVG
 * @author oliford  */
public class SuccinctImagingSVGExample {
    final static String outPath = MinervaOpticsSettings.getAppsOutputPath() + "/rayTracing/succinctImaging";
    final static int nRays = 500;
    final static double z[] = OneLiners.linSpace(-0.2, 0.2, 6);

    public static void main(String[] args) {
        Nikon50mmF11 lens = new Nikon50mmF11(new double[]{ 1, 0, 0 });
        Square imgPlane = new Square("imgPlane", new double[]{ 1.0526, 0, 0 }, new double[]{ -1, 0, 0 }, new double[]{ 0, 1, 0 }, 0.040, 0.040, Absorber.ideal());
        Optic all = new Optic("all", new Element[]{ lens, imgPlane });

        double col[][] = ColorMaps.jet(z.length);
        SVGRayDrawing svgOut = new SVGRayDrawing(outPath + "/imgTest", new double[]{ 0, -1, -1, 2, 1, 1 }, true );
        svgOut.generateLineStyles(col, 0.0002);

        for(int iZ=0; iZ < z.length; iZ++) {
            for(int i=0; i < nRays; i++) {
                RaySegment ray = new RaySegment(new double[]{ 0, 0, z[iZ] }, lens);
                Tracer.trace(all, ray, 30, 0.01, true);
                svgOut.drawRay(ray, iZ);
                Pol.recoverAll();
            }
        }

        svgOut.drawElement(all);
        svgOut.destroy();
    }
}
```

Optics definition

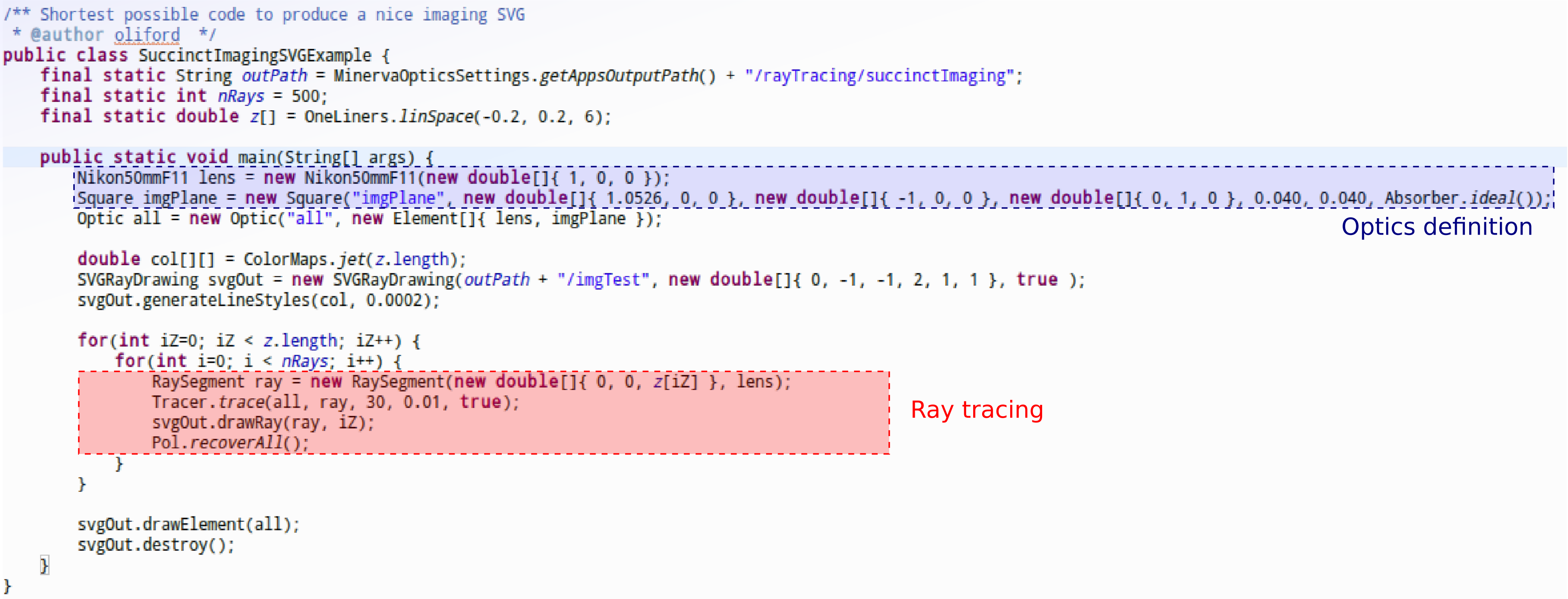
SVG Output:



Nikon 50mm f/2.8

FusionOptics Ray Tracer

- Ray tracing core is a relatively simple to use Java library.

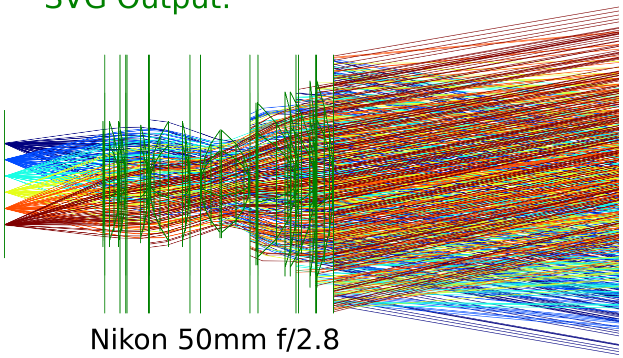


```
/** Shortest possible code to produce a nice imaging SVG
 * @author oliford  */
public class SuccinctImagingSVGExample {
    final static String outPath = MinervaOpticsSettings.getAppsOutputPath() + "/rayTracing/succinctImaging";
    final static int nRays = 500;
    final static double z[] = OneLiners.linSpace(-0.2, 0.2, 6);

    public static void main(String[] args) {
        Nikon50mmF11 lens = new Nikon50mmF11(new double[]{ 1, 0, 0 });
        Square imgPlane = new Square("imgPlane", new double[]{ 1.0526, 0, 0 }, new double[]{ -1, 0, 0 }, new double[]{ 0, 1, 0 }, 0.040, 0.040, Absorber.ideal());
        Optic all = new Optic("all", new Element[]{ lens, imgPlane });

        double col[][] = ColorMaps.jet(z.length);
        SVGRayDrawing svgOut = new SVGRayDrawing(outPath + "/imgTest", new double[]{ 0, -1, -1, 2, 1, 1 }, true );
        svgOut.generateLineStyles(col, 0.0002);

        for(int iZ=0; iZ < z.length; iZ++) {
            for(int i=0; i < nRays; i++) {
                RaySegment ray = new RaySegment(new double[]{ 0, 0, z[iZ] }, lens);
                Tracer.trace(all, ray, 30, 0.01, true);
                svgOut.drawRay(ray, iZ);
                Pol.recoverAll();
            }
        }

        svgOut.drawElement(all);
        svgOut.destroy();
    }
}
```

Optics definition

Ray tracing

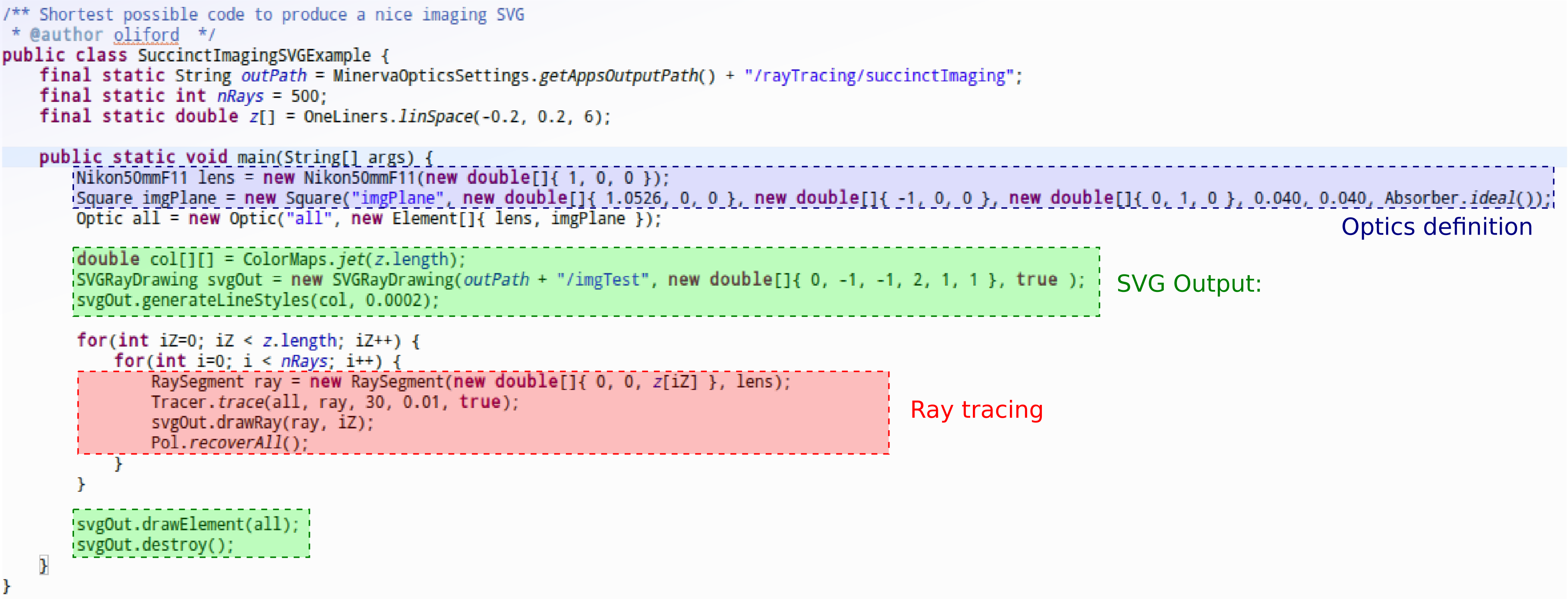
SVG Output:



Nikon 50mm f/2.8

FusionOptics Ray Tracer

- Ray tracing core is a relatively simple to use Java library.



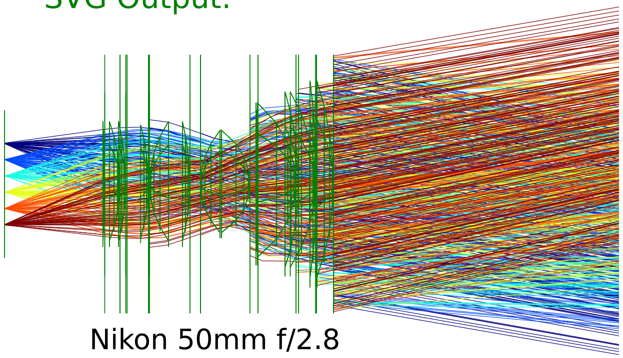
```
/** Shortest possible code to produce a nice imaging SVG
 * @author oliford  */
public class SuccinctImagingSVGExample {
    final static String outPath = MinervaOpticsSettings.getAppsOutputPath() + "/rayTracing/succinctImaging";
    final static int nRays = 500;
    final static double z[] = OneLiners.linSpace(-0.2, 0.2, 6);

    public static void main(String[] args) {
        Nikon50mmF11 lens = new Nikon50mmF11(new double[]{ 1, 0, 0 });
        Square imgPlane = new Square("imgPlane", new double[]{ 1.0526, 0, 0 }, new double[]{ -1, 0, 0 }, new double[]{ 0, 1, 0 }, 0.040, 0.040, Absorber.ideal());
        Optic all = new Optic("all", new Element[]{ lens, imgPlane });

        double col[][] = ColorMaps.jet(z.length);
        SVGRayDrawing svgOut = new SVGRayDrawing(outPath + "/imgTest", new double[]{ 0, -1, -1, 2, 1, 1 }, true );
        svgOut.generateLineStyles(col, 0.0002);

        for(int iZ=0; iZ < z.length; iZ++) {
            for(int i=0; i < nRays; i++) {
                RaySegment ray = new RaySegment(new double[]{ 0, 0, z[iZ] }, lens);
                Tracer.trace(all, ray, 30, 0.01, true);
                svgOut.drawRay(ray, iZ);
                Pol.recoverAll();
            }
        }

        svgOut.drawElement(all);
        svgOut.destroy();
    }
}
```

Optics definition

SVG Output:

Ray tracing

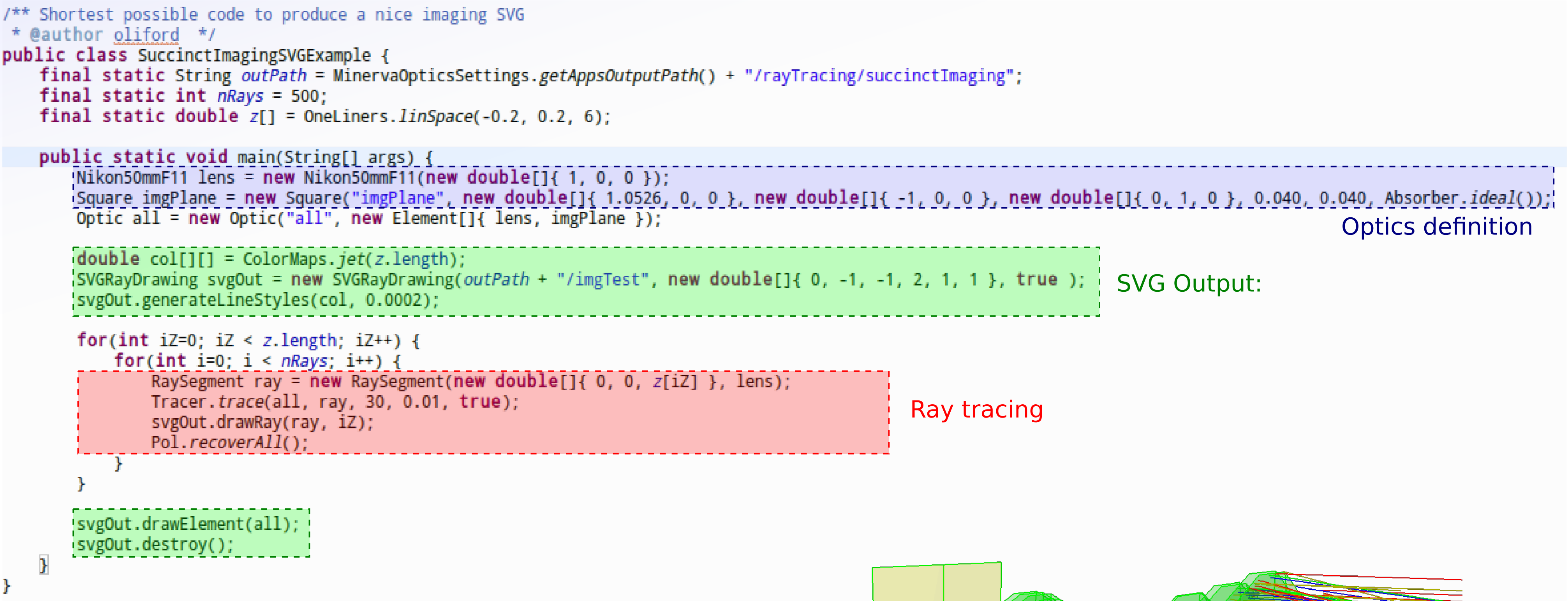
SVG Output:



Nikon 50mm f/2.8

FusionOptics Ray Tracer

- Ray tracing core is a relatively simple to use Java library.



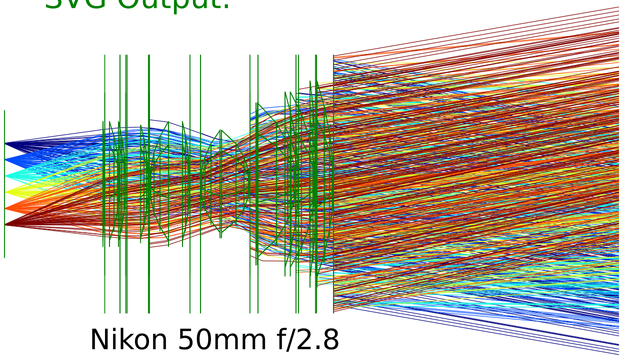
```
/** Shortest possible code to produce a nice imaging SVG
 * @author oliford  */
public class SuccinctImagingSVGExample {
    final static String outPath = MinervaOpticsSettings.getAppsOutputPath() + "/rayTracing/succinctImaging";
    final static int nRays = 500;
    final static double z[] = OneLiners.linSpace(-0.2, 0.2, 6);

    public static void main(String[] args) {
        Nikon50mmF11 lens = new Nikon50mmF11(new double[]{ 1, 0, 0 });
        Square imgPlane = new Square("imgPlane", new double[]{ 1.0526, 0, 0 }, new double[]{ -1, 0, 0 }, new double[]{ 0, 1, 0 }, 0.040, 0.040, Absorber.ideal());
        Optic all = new Optic("all", new Element[]{ lens, imgPlane });

        double col[][] = ColorMaps.jet(z.length);
        SVGRayDrawing svgOut = new SVGRayDrawing(outPath + "/imgTest", new double[]{ 0, -1, -1, 2, 1, 1 }, true );
        svgOut.generateLineStyles(col, 0.0002);

        for(int iZ=0; iZ < z.length; iZ++) {
            for(int i=0; i < nRays; i++) {
                RaySegment ray = new RaySegment(new double[]{ 0, 0, z[iZ] }, lens);
                Tracer.trace(all, ray, 30, 0.01, true);
                svgOut.drawRay(ray, iZ);
                Pol.recoverAll();
            }
        }

        svgOut.drawElement(all);
        svgOut.destroy();
    }
}
```

Optics definition

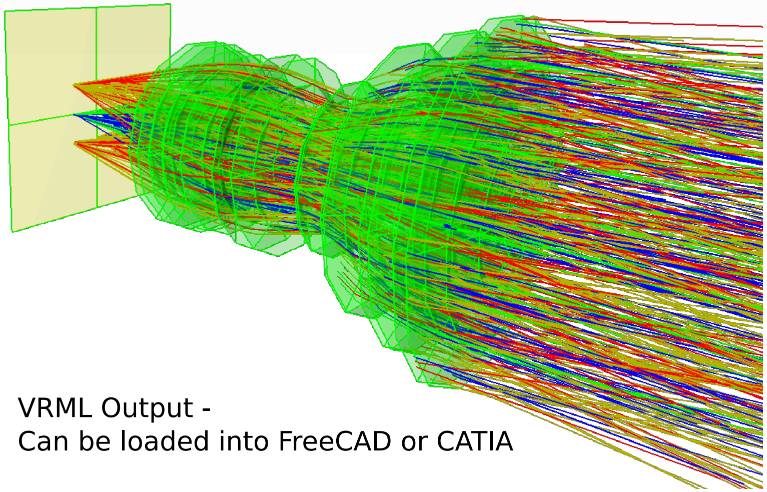
SVG Output:

Ray tracing

SVG Output:



Nikon 50mm f/2.8



VRML Output -
Can be loaded into FreeCAD or CATIA

FusionOptics Ray Tracer - Aspherics

Many optimisation algorithms available (Hooke & Jeeves, Genetic Algorithms, ...), so easy to optimise any parameters to any cost function. e.g:

- Auto-focus (moving elements).
- Determining unknown lens properties (e.g. refractive indices) by fitting measured image.
- Aspheric surface optimisation for aberration control.

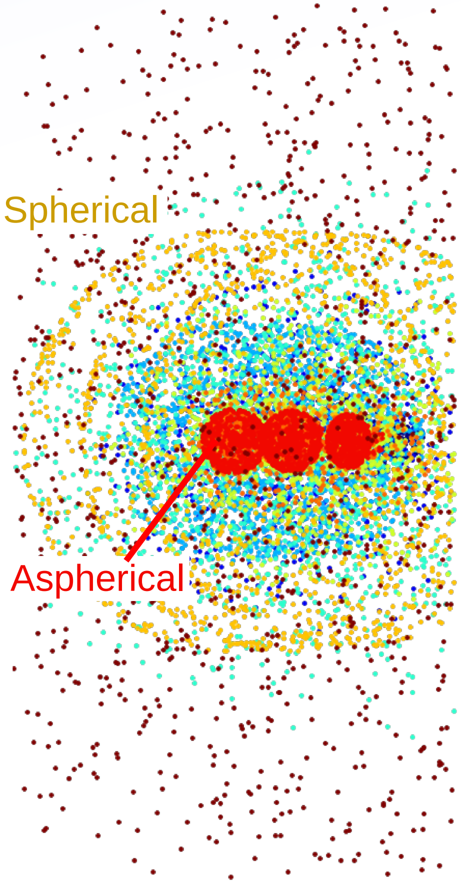
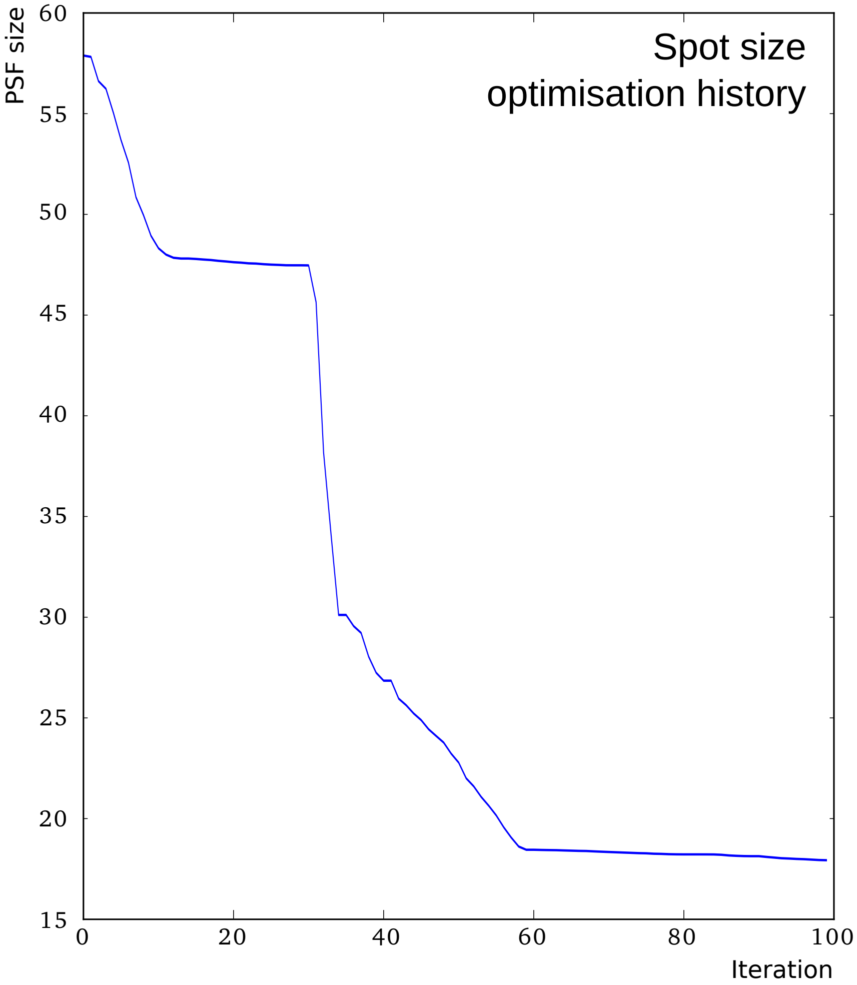
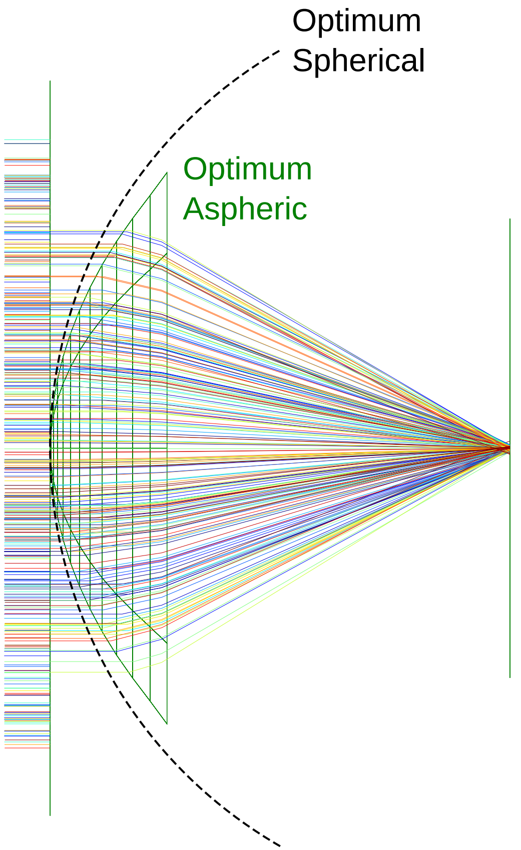


Image plane (Monte-carlo)

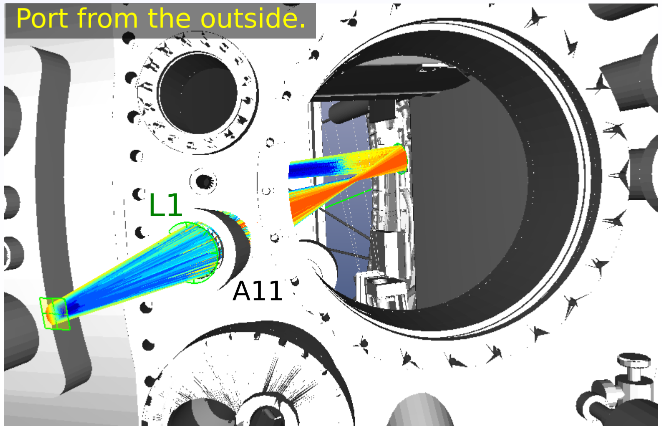


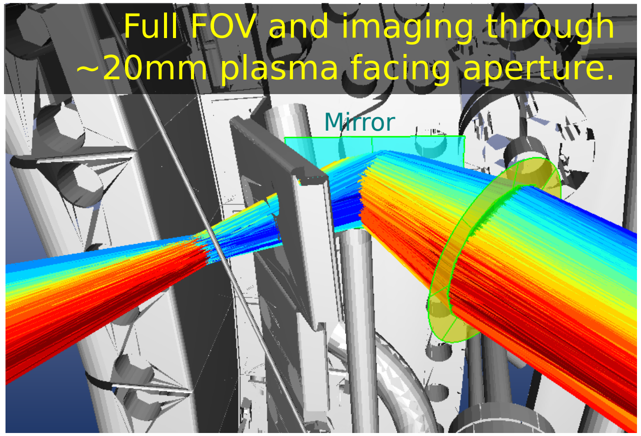


FusionOptics Ray Tracer - In place design

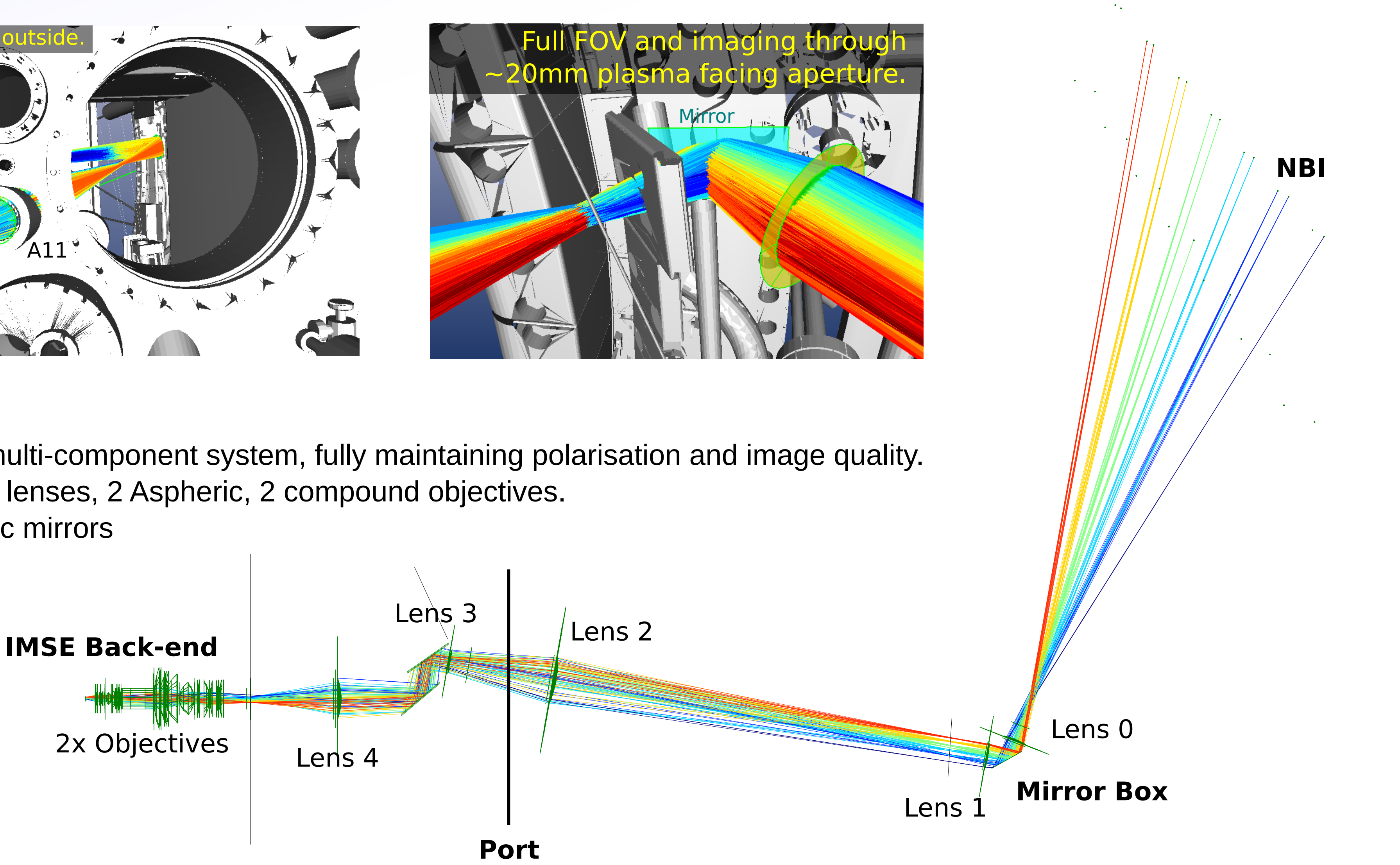
Has now been used for optical design/analysis of various systems at IPP:

- Permanent IMSE at ASDEX Upgrade:





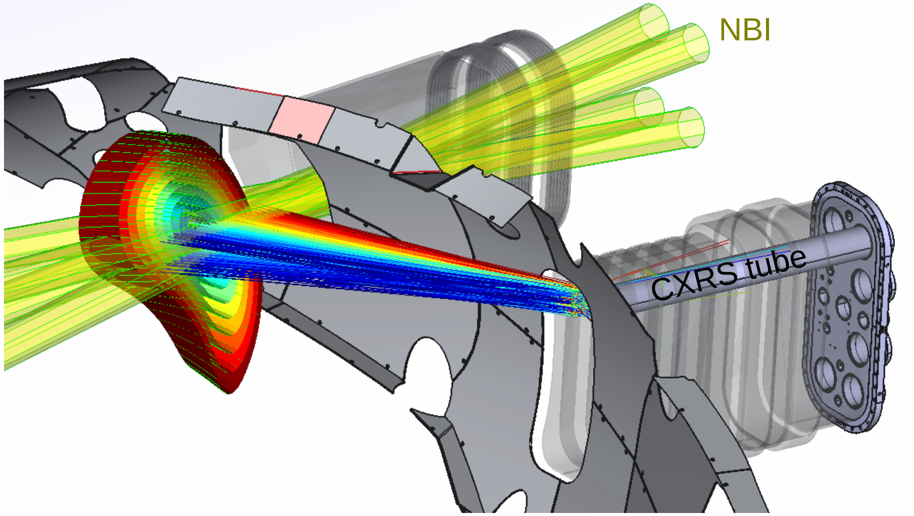
- Complex multi-component system, fully maintaining polarisation and image quality.
  - 3 Spheric lenses, 2 Aspheric, 2 compound objectives.
  - 3 dielectric mirrors



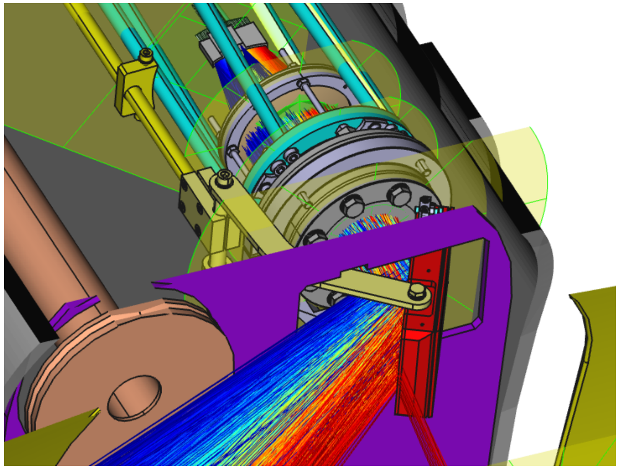
FusionOptics Ray Tracer - In place design

Has now been used for optical design/analysis of various systems at IPP:

- CXRS at W7-X: Full optical system design.



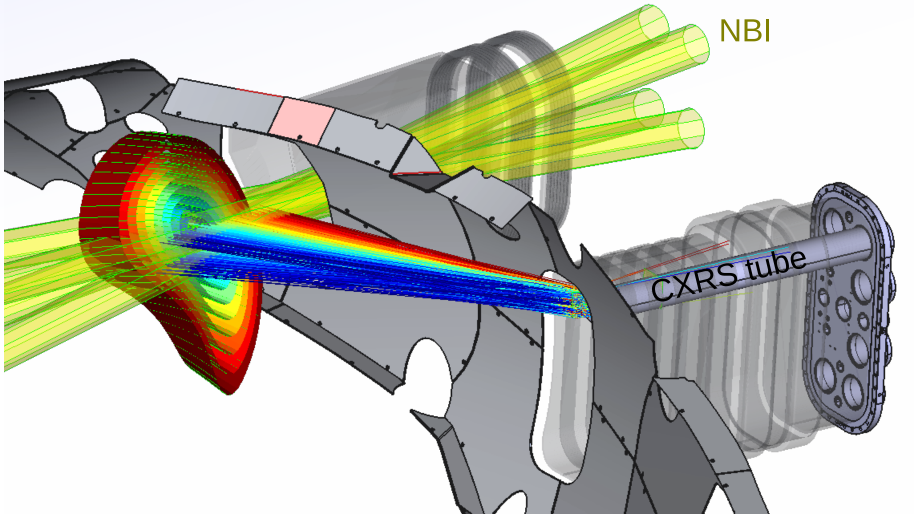
Minimal window exposure:



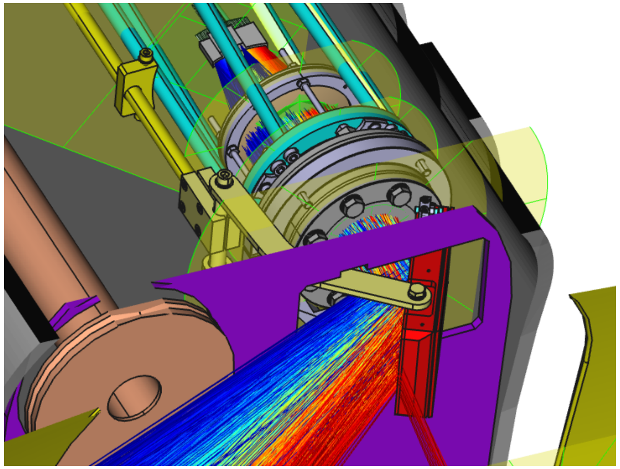
FusionOptics Ray Tracer - In place design

Has now been used for optical design/analysis of various systems at IPP:

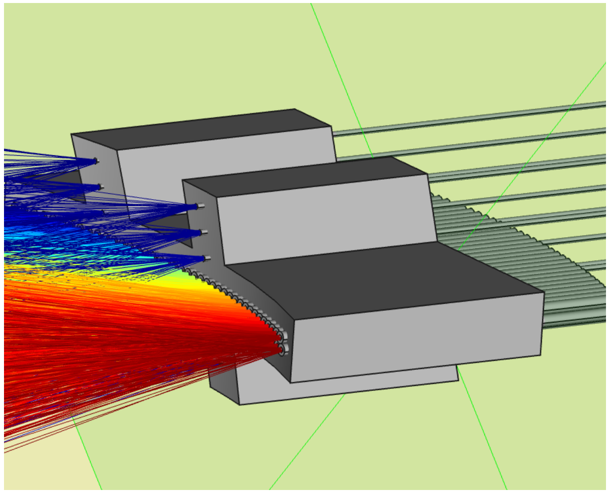
- CXRS at W7-X: Full optical system design.



Minimal window exposure:



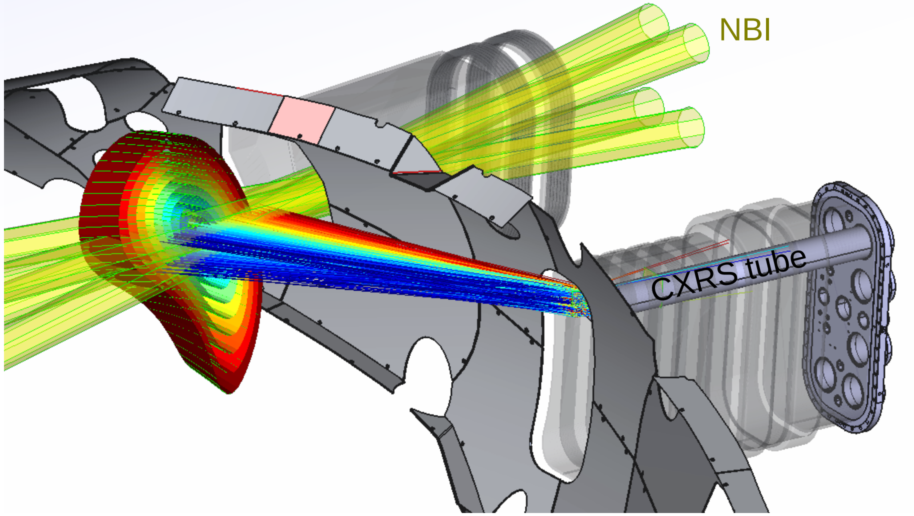
Fibre head design for optimal focus:



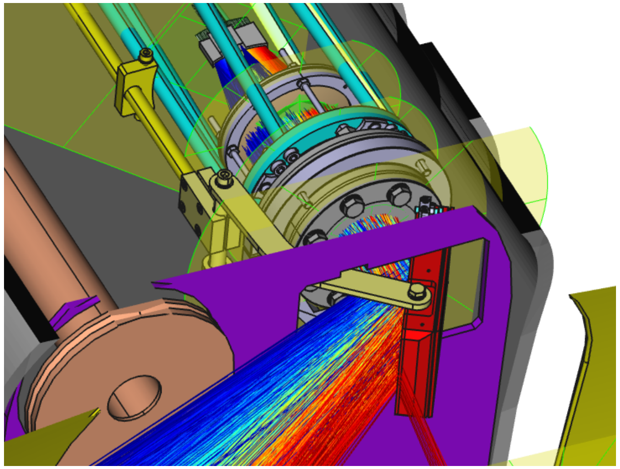
FusionOptics Ray Tracer - In place design

Has now been used for optical design/analysis of various systems at IPP:

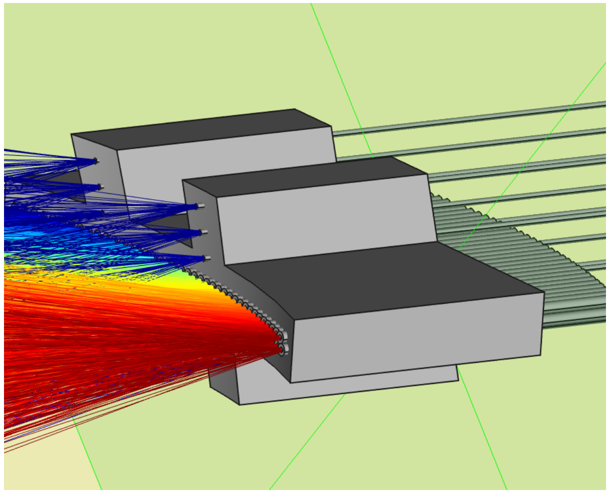
- CXRS at W7-X: Full optical system design.

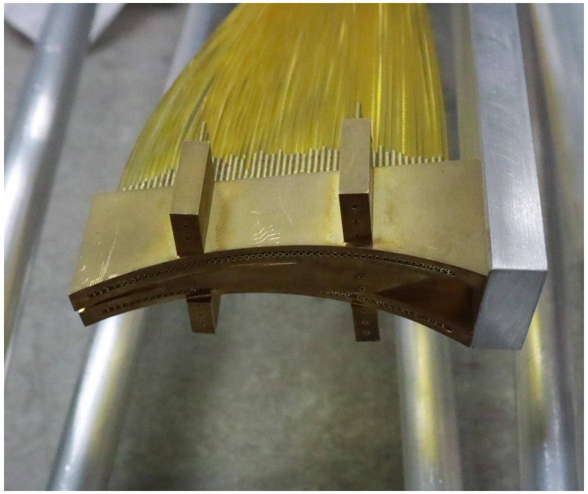


Minimal window exposure:



Fibre head design for optimal focus:

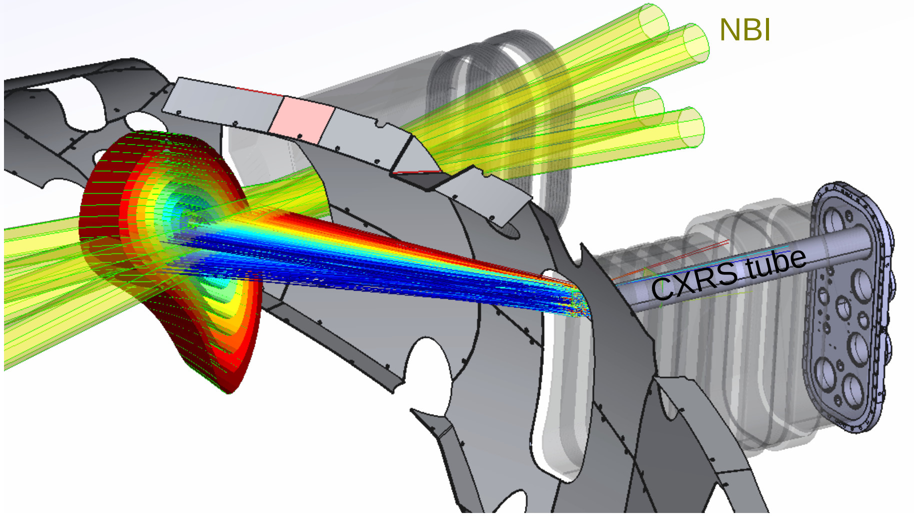




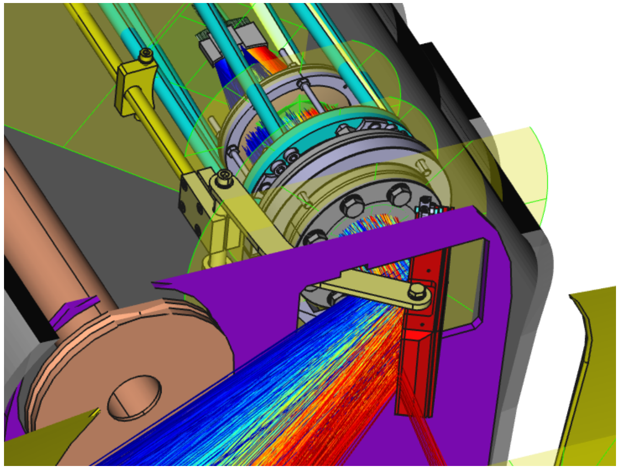
FusionOptics Ray Tracer - In place design

Has now been used for optical design/analysis of various systems at IPP:

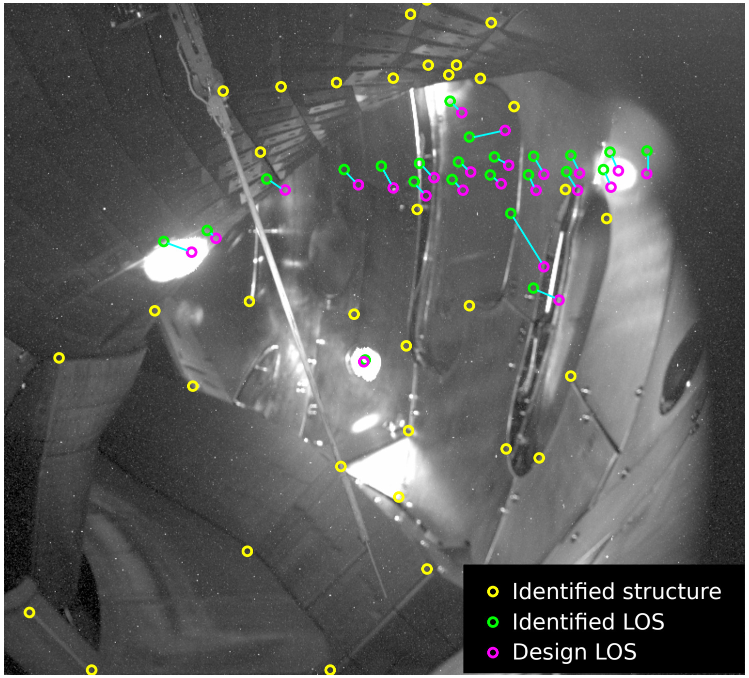
- CXRS at W7-X: Full optical system design.



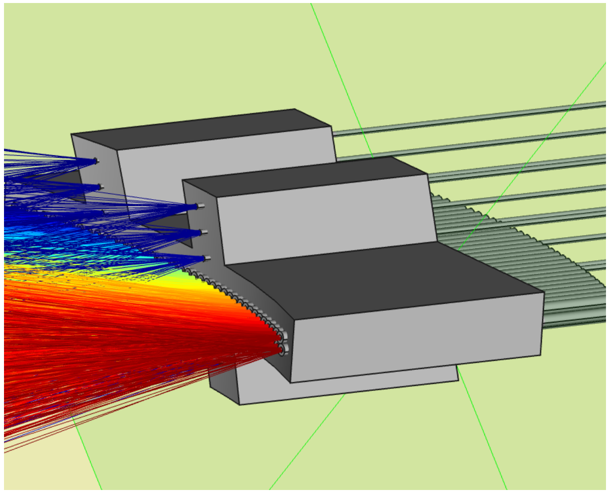
Minimal window exposure:

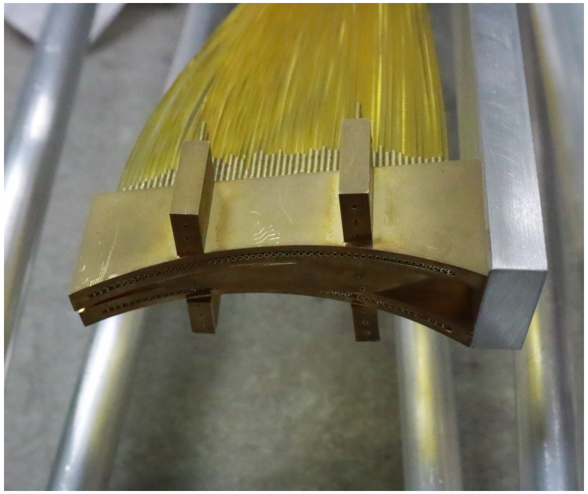


Fit model alignment from backlighting:



Fibre head design for optimal focus:

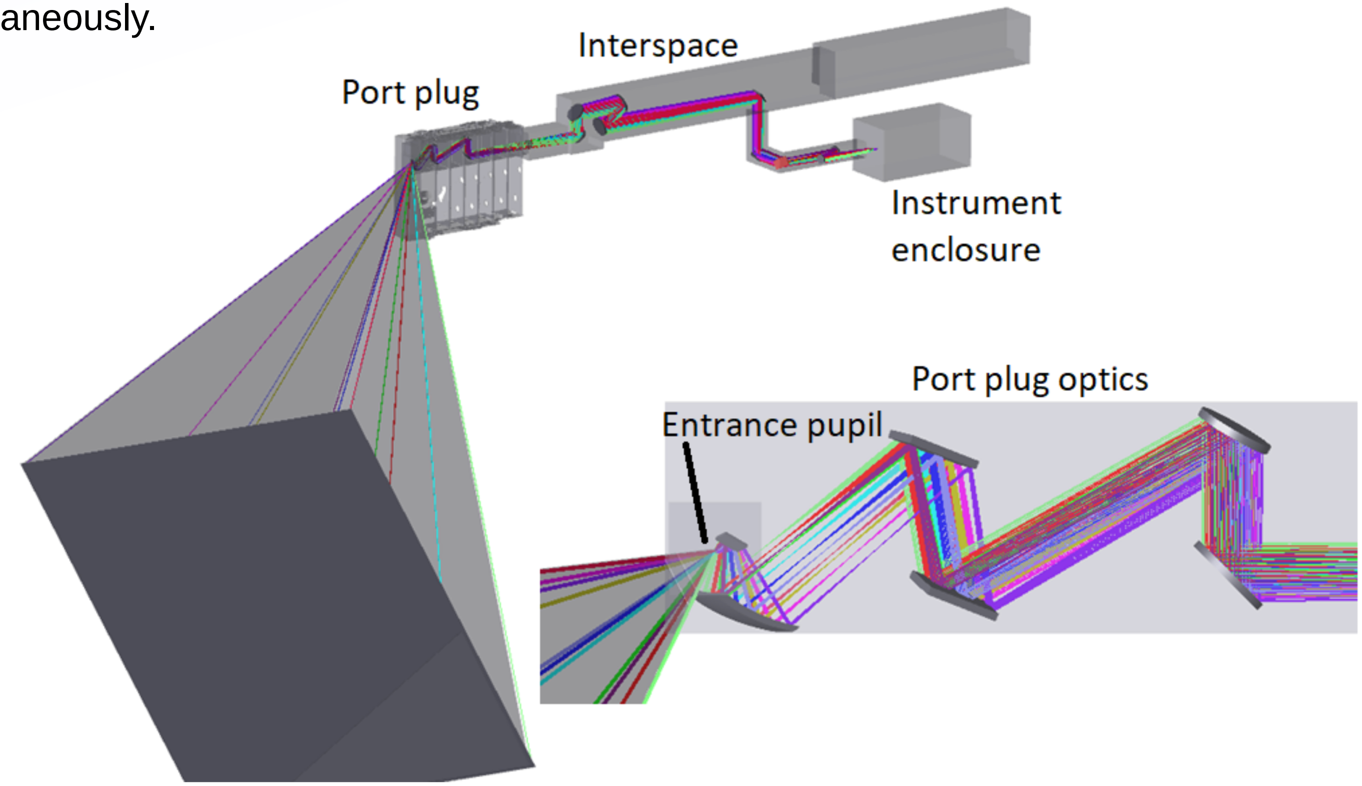




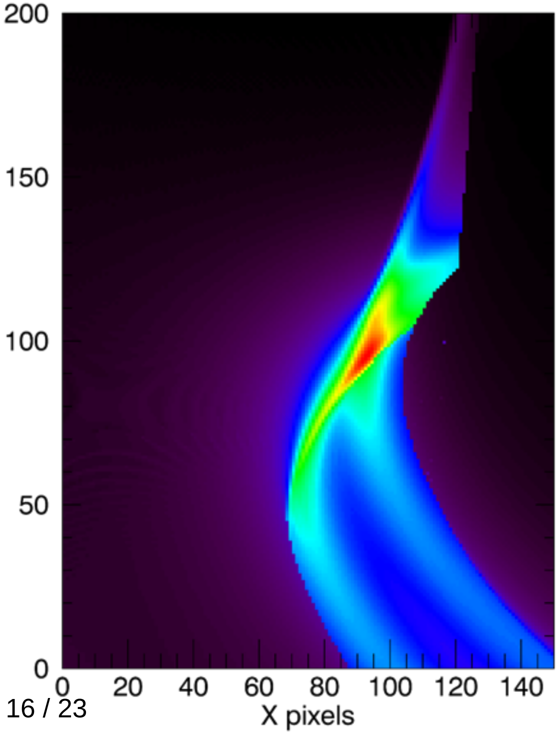
ITER Flow Monitor

ITER 'Flow Monitor'
- Coherence imaging system for SOL impurity/bulk flow.
- Use polarisation to discriminate reflections.
- Several other measurements simultaneously.

Optical relay



Predicted view of divertor



Summary

FusionOptics:

- 3D general ray-tracer developed for design/analysis of optical diagnostics.

- Intended for coupling into diagnostics forward models.
- Simple modular object-oriented structure.
- Detailed treatment of many realistic components (mirror, lenses, glasses, filters etc)

- Good coupling to CAD programs.
- Full 3D treatment of polarisation states, easy to understand and visualise.
- Now used for several diagnostics at AUG and W7-X.

Summary

FusionOptics:

- 3D general ray-tracer developed for design/analysis of optical diagnostics.

- Intended for coupling into diagnostics forward models.
- Simple modular object-oriented structure.
- Detailed treatment of many realistic components (mirror, lenses, glasses, filters etc)

- Good coupling to CAD programs.
- Full 3D treatment of polarisation states, easy to understand and visualise.
- Now used for several diagnostics at AUG and W7-X.

General:

- Fitting of known image points and vignetting/limit circles constrains many unknowns such that polarisation state is well predicted.

- Coupling of 3D ray-tracing and CAD allows easy simultaneous convergence of optic and mechanical design.

- Very complex optical chains can be handled without too much difficulty - (e.g. most optical ITER diangnostics)

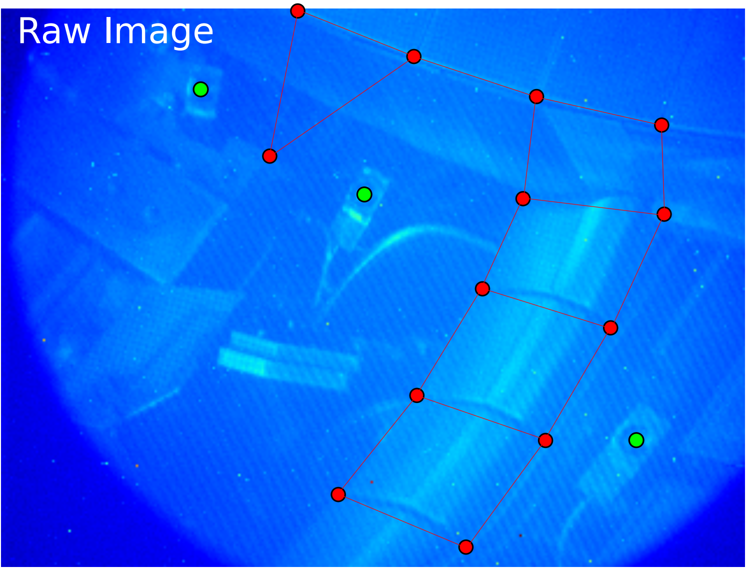
- Small polarisation effects (~0.1°) are numerous, but can be modelled.

- Particular relevance for ITER Flow Monitor (coherence imaging / divertor viewing).

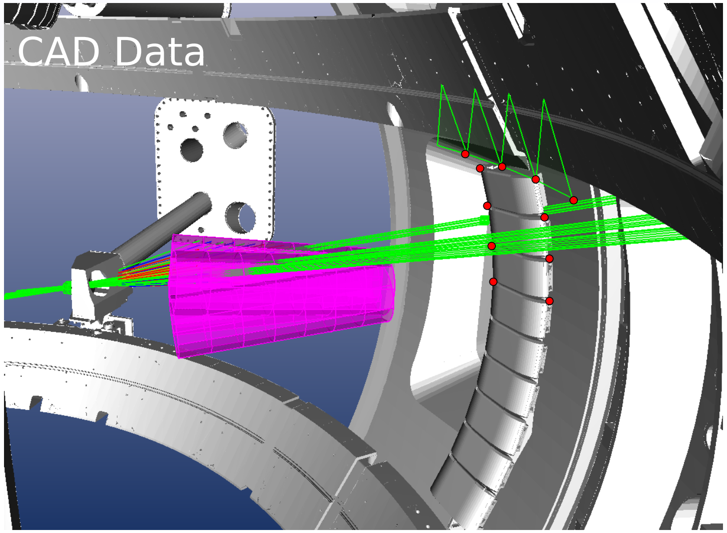
gmds/AUG/29317

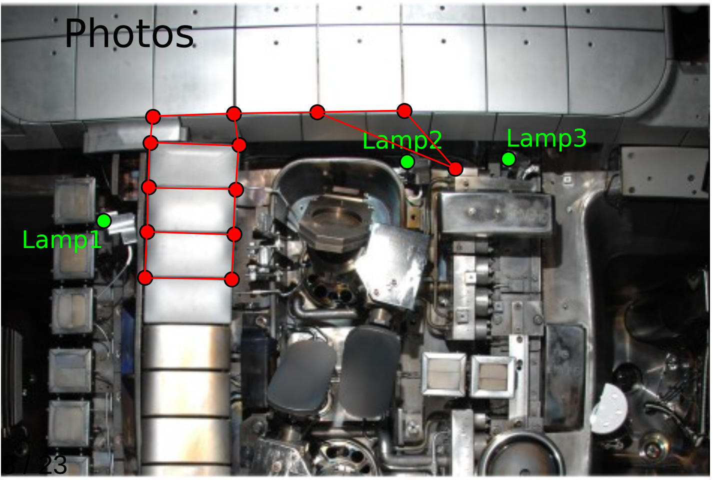
Image Transform

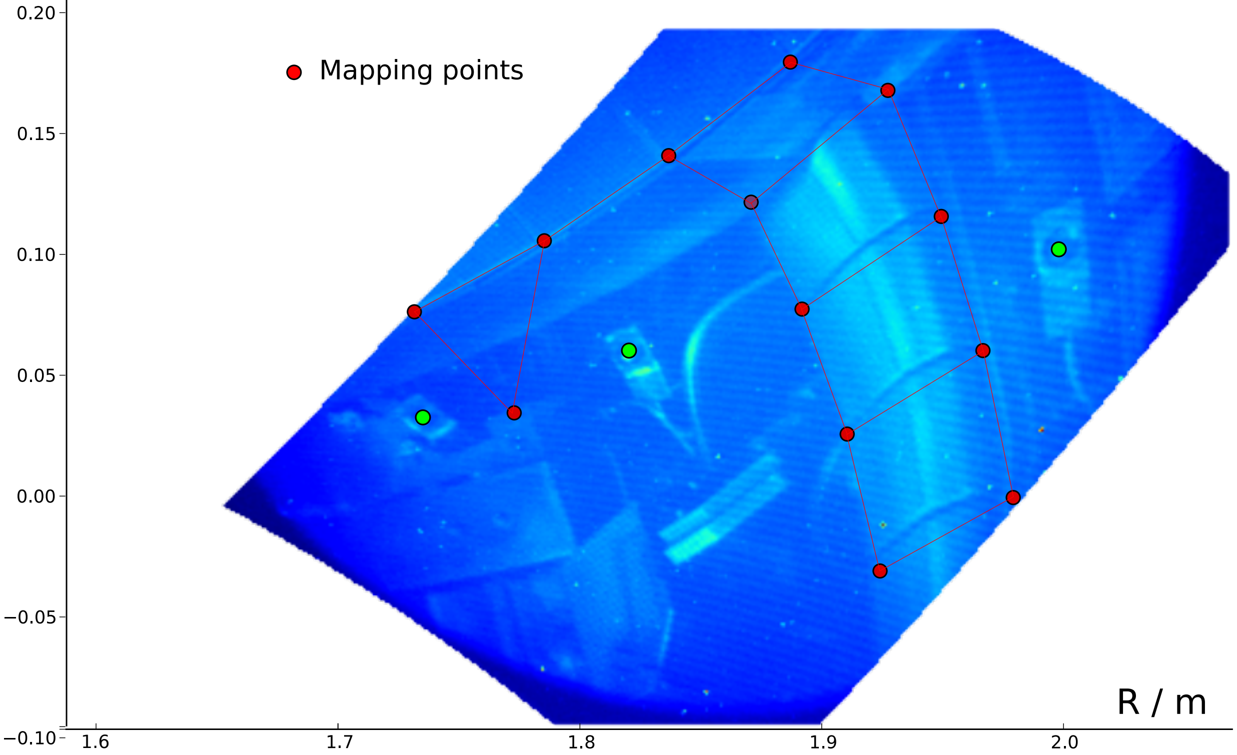
- Points with known 3D positions (CAD)
- Define affine/cubic transform directly (x,y) --> (φ, θ) --> (R, Z)
- Fit unknown optics model parameters so ray-tracing matches:
  - a) Camera position ±6µm
  - b) Mirror angles ±0.1°



Project in (R,Z)



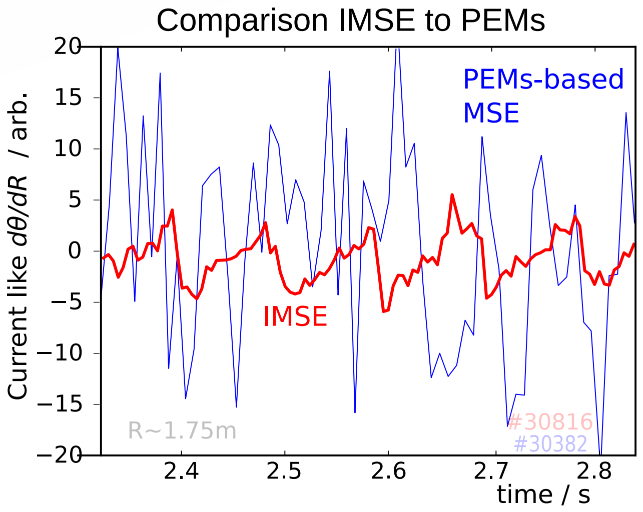


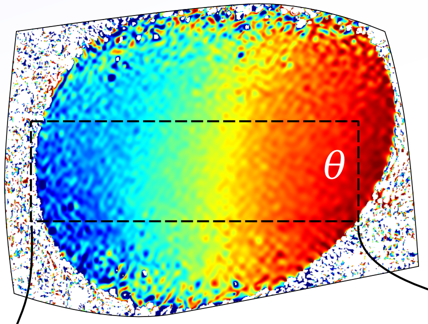


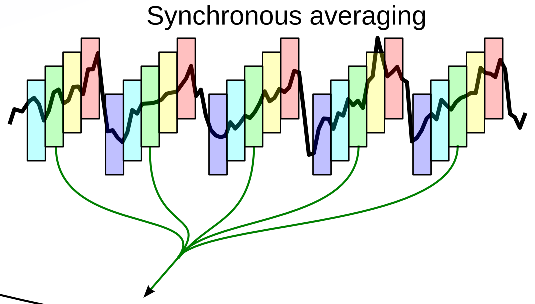
Sawteeth - Magnetic Reconnection

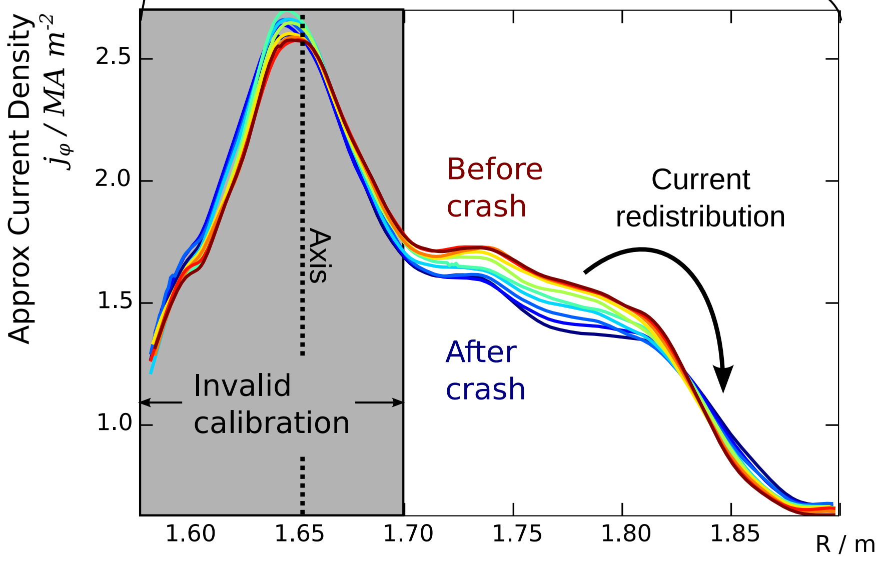
What do we see in the IMSE data?

- Sawtooth changes are ***very*** small - need good statistics.
- Average over Z near axis
- Synchronous average over many sawteeth in time.

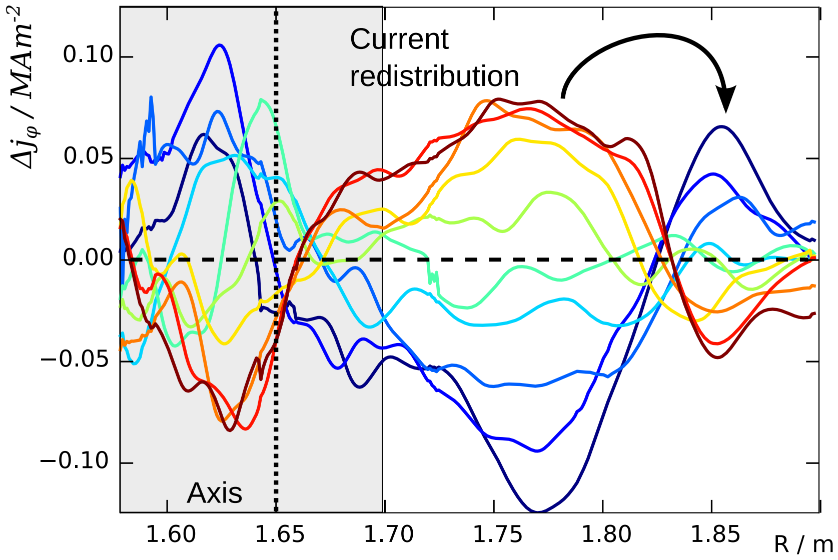








Difference from average profile:



Current redistribution: $\Delta j \sim 0.050\ MA\ m^{-2}$

Measurements every ~3cm (resolution): $\Delta(d\theta/dR) \sim 0.7° m^{-1}$ --> $\boldsymbol{\Delta\theta \pm 0.02°}$ ***required for*** $\boldsymbol{\Delta R = 3cm}$

[R. Fischer et. al.]

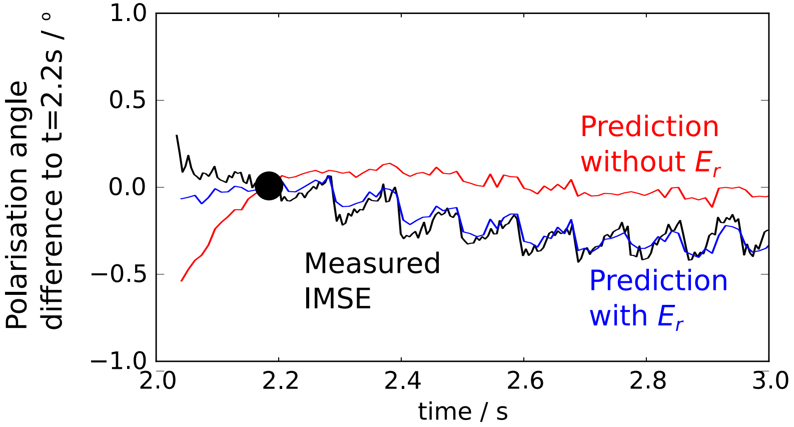
Integrated Equilibrium vs IMSE - Sawteeth

Required precision is so high, many other factors become important:

Plasma radial electric field:

$$E = v \times B + E_r$$

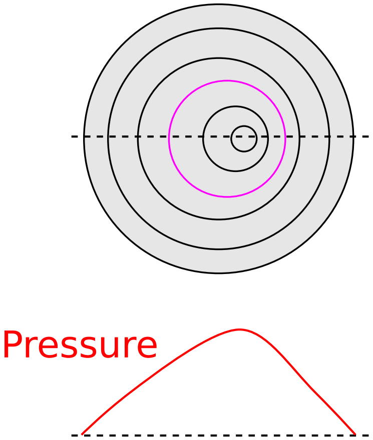
At some locations, ΔE_r during sawtooth dominates measurement:

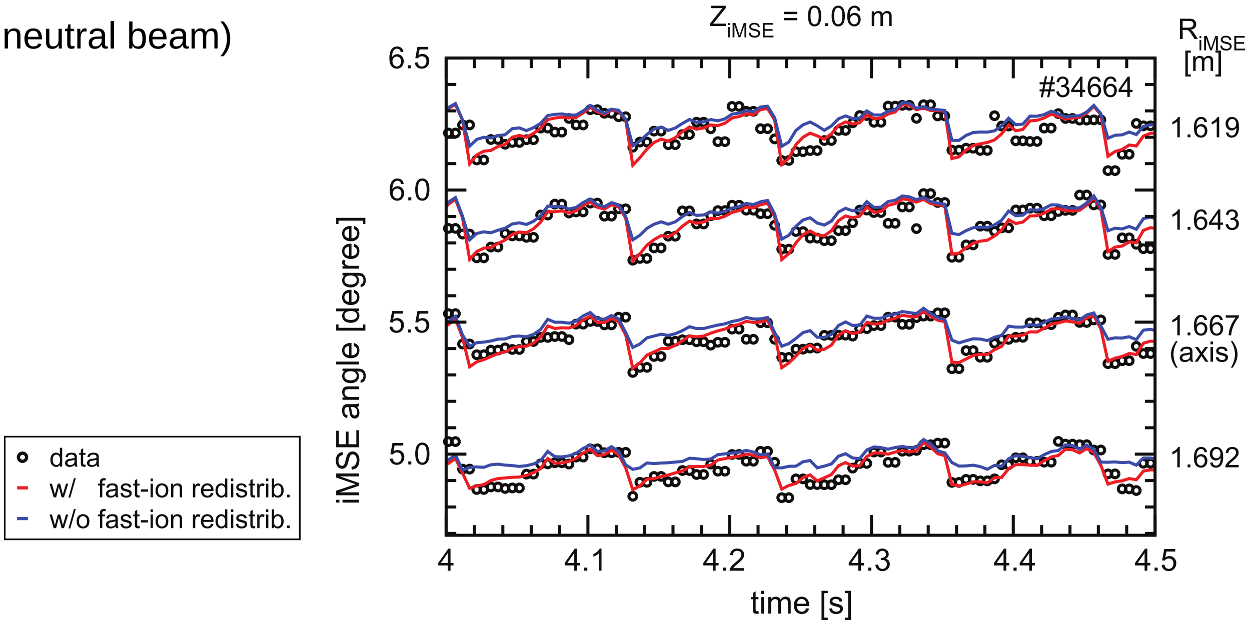


Shafranov shift:

Movement of plasma axis with pressure.

(including redistribution of fast-ions from neutral beam)





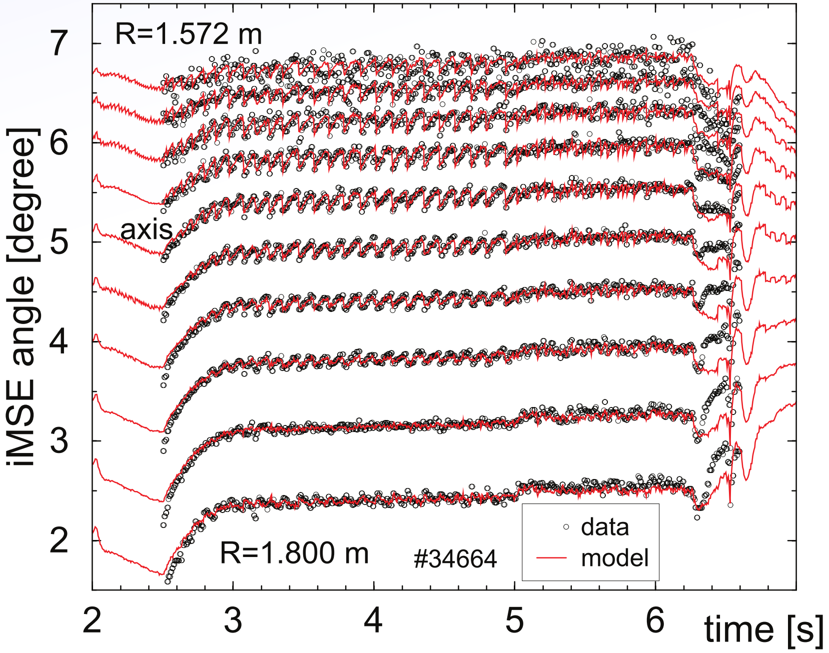
[R. Fischer et. al.]

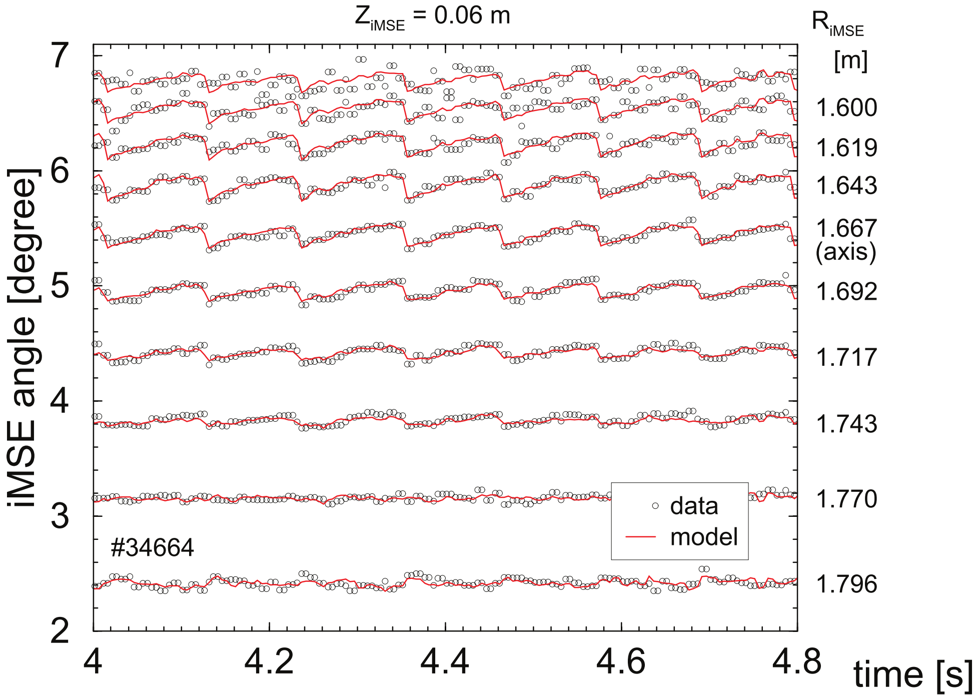
Integrated Equilibrium vs IMSE - Sawteeth

Required precision is so high, many other factors become important:

but...

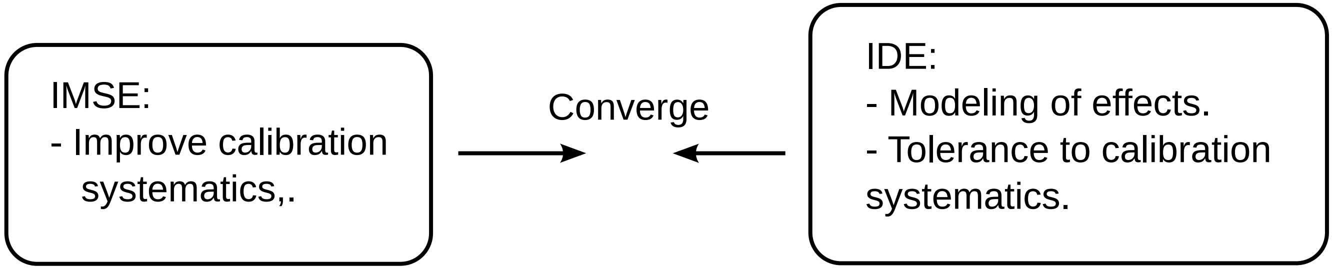
we now have good agreement between full integrated model and IMSE measurements for sawtooth evolution in θ.





- This is where we are - 'the state of the ~~art~~ ... science'

What next?



IMSE:
- Improve calibration systematics,.

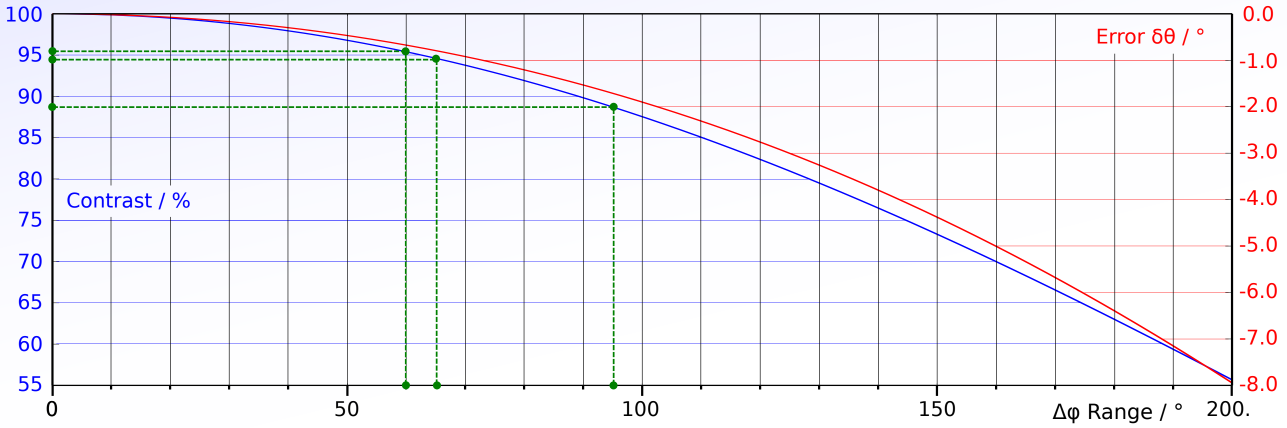
Converge

IDE:
- Modeling of effects.
- Tolerance to calibration systematics.

The return of the Magic Number

Oliver Ford
IPP Greifswald
gmds/AUG/75-97

The crystal parallelism isn't enough to explain all of the magic number. E.g. United Crystals plate A has < 60° and is very flat in the middle (< 5° variation). That should give a constrast of > 98%, but the measured constrast is always below 90%.



So, there is more to the story....
Using a big sphere to light all of the CCD/lens and looking at the full 16mm CCD shows a consistent pattern:

United Crystals A

United Crystals B

CLaser Displacer