AIRY OPTICS

THE POLARIZATION EXPERTS

Interactive Analysis of Complex Optical Phenomena

Jonathan Heinz, Lead Developer
Airy Optics Inc.

Overview

Airy Optics Inc. develops Polaris-M, a polarization ray-tracing software written completely in Mathematica. This talk has three parts:

- Visualizations of Polarization and Polarized Rays
- Dealing with Polarization Ray-Tracing Data
- Bringing it All Together

These tools have helped us provide analysis to our customers, including contributions to the development of the next generation of AR/VR headsets.



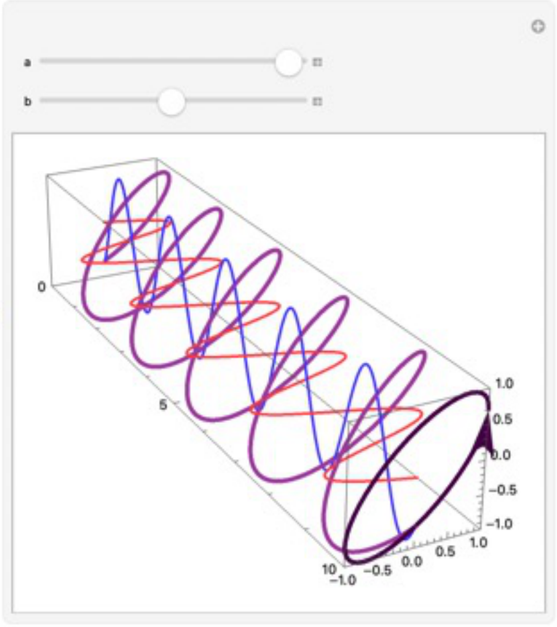
Dynamic Learning Tools

Polarization Ellipses

Though one can often think of a single axis of linear polarization, in general polarization can be described using ellipses.

In[139]:=
```
Manipulate[Show[ParametricPlot3D[{{Sin[s Pi], 0, s}, {0, a Sin[(s - b) Pi], s}, {Sin[s Pi], a Sin[(s - b) Pi], s}}, {s, 0, 10},
   PlotStyle → {Directive[Lighter[Red]], Lighter[Blue], Directive[Thickness[0.01], Lighter[Purple]]}, PlotRange → {{-1, 1}, {-1, 1}, {0, 10}}],
  Graphics3D[Ellipse3D[{1, a Exp[I b Pi], 0}, {0, 0, 10}, Color → Darker[Purple]]], ViewVertical → {0, 1, 0}, ViewPoint → {-1, 1, 2}], {a, 0, 1},
 {b, 0, 0.5}]
```

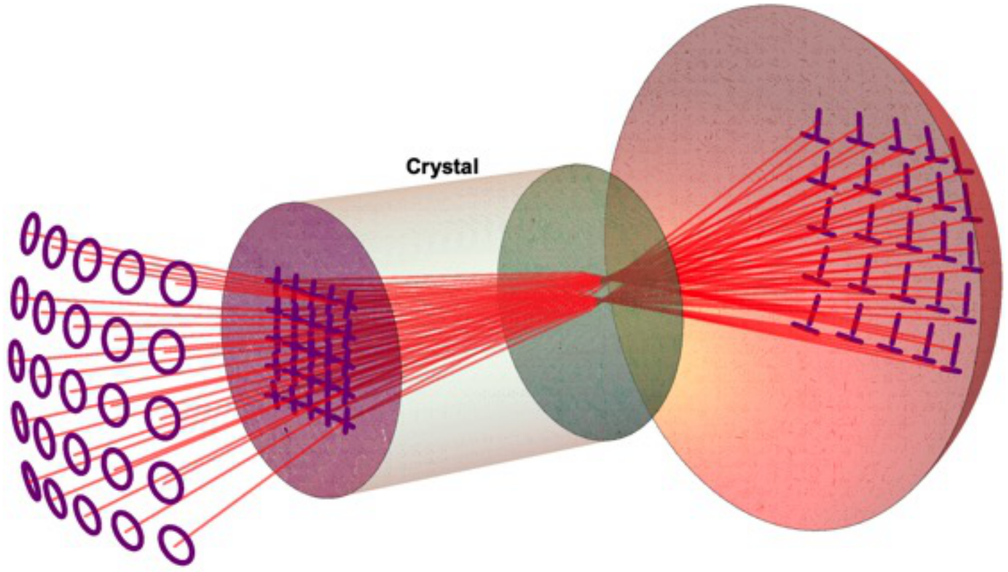
Out[139]=



In[74]:=

Polarization Raytracing

Out[ ]=



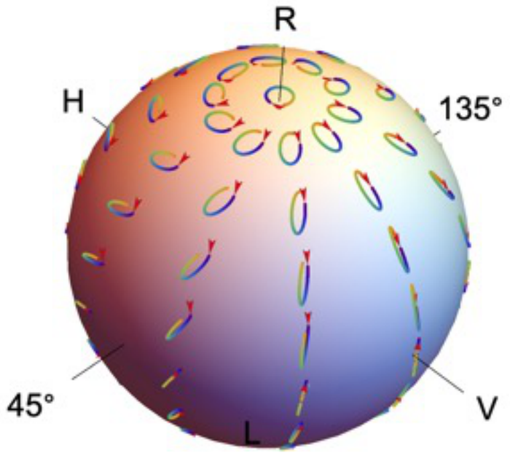
In[74]:=

The Poincare Sphere

The Poincare Sphere is a useful way to visually represent different polarization states of light.

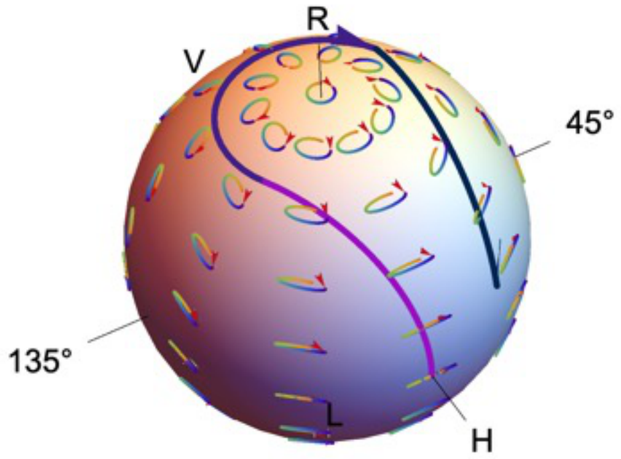
In[68]:=
```
Graphics3D[{Sphere[], PoincareEllipses[10, 10, ScaleEllipse → 1.3, RadiusFunction → 0.005],
   PoincareAxes}, Boxed → False]
```

Out[ ]=



In[69]:=
```
Retlist = {{π / 2, π / 3, 0}, {π, 0, π / 2}, {π / 2, -π / 4, 0}};
Graphics3D[{RetarderTrajectory[Retlist, {1, 1, 0, 0}, ShowSphere → False], Sphere[],
  PoincareEllipses[10, 10, ScaleEllipse → 1.3, RadiusFunction → 0.005], PoincareAxes}, Boxed → False]
```

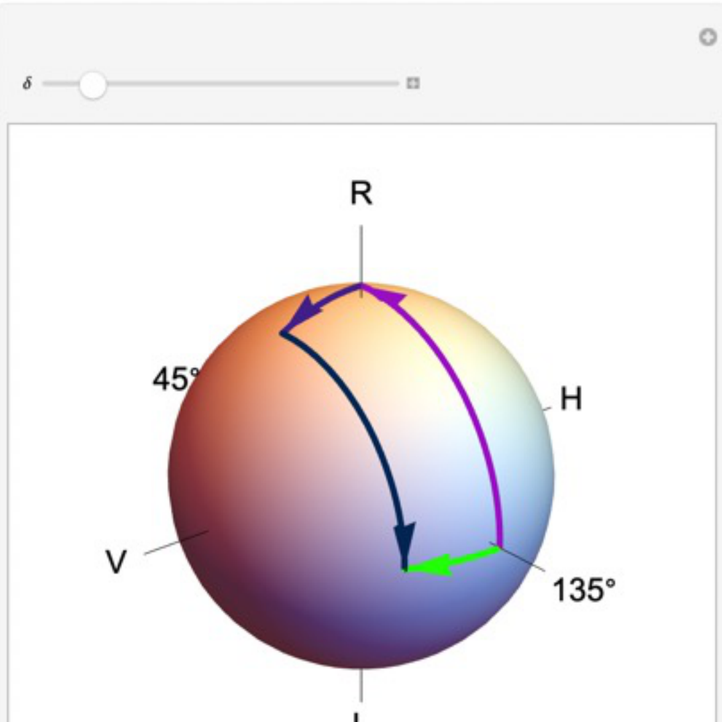
Out[ ]=



In[71]:=
```
horizontalret = {π / 2, 0, 0};
verticalret = {π / 2, π / 2, 0};
Manipulate[
```

In[71]:=
```
horizontalret = {π / 2, 0, 0};
verticalret = {π / 2, π / 2, 0};
Manipulate[
 Graphics3D[{RetarderTrajectory[{horizontalret, {δ, π / 4, 0}, verticalret}, {1, 0, -1, 0},
    Opts`ShowSphere → False], RetarderTrajectory[{δ, 0, π / 2}, {1, 0, -1, 0}, Opts`Color → Green,
    Opts`ShowSphere → False], Sphere[{0, 0, 0}, 0.95], PoincareAxes}, Boxed → False,
  ViewPoint → {-2.0, -2.4, 1.5}], {{δ, π / 4}, π / 16, π}]
```

Out[ ]=

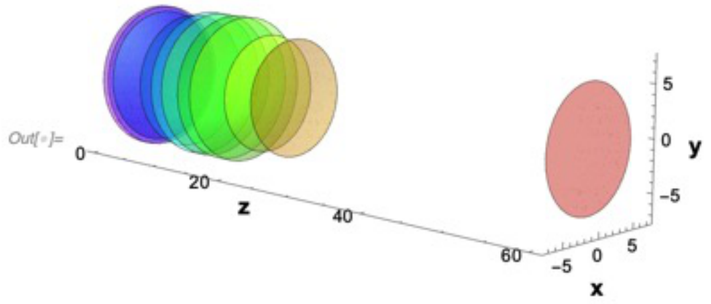


Flexible Styling of Plots with Many Surfaces

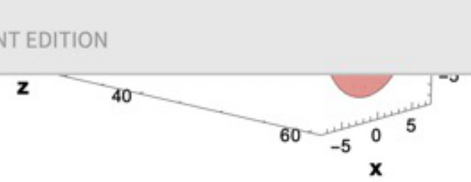
Plot System

In[97]:=
```
os = Tutorial OS ;
PlotSystem[os]
```

Out[•]=

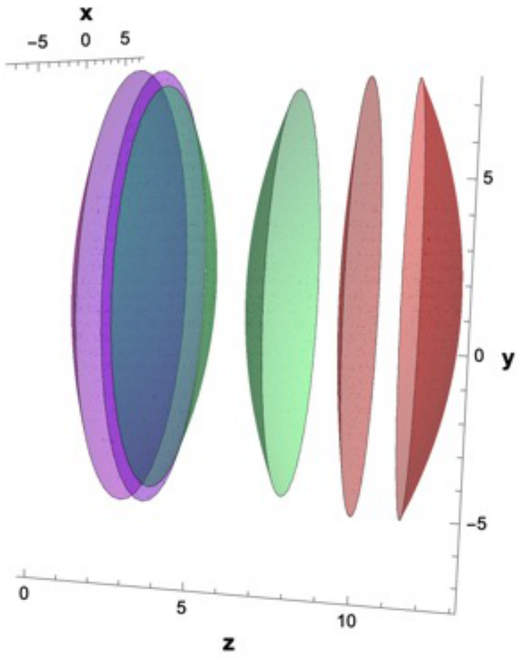


In[99]:=
```
PlotSystem[os[[1 ;; 6]], SurfaceGroups → {{1, 2}, {3, 4}, {5, 6}}]
```



In[99]:=
```
PlotSystem[os[[1 ;; 6]], SurfaceGroups → {{1, 2}, {3, 4}, {5, 6}}]
```

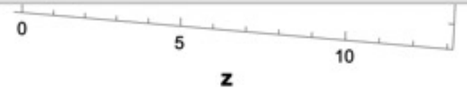
Out[ ]=



In[100]:=

In[101]:=

In[ ]:=
```
Grid[RandomReal[1, {2, 2}], Dividers → All]
```



In[100]:=

In[101]:=

In[102]:=
```
Grid[RandomReal[1, {2, 2}], Dividers → All]
```

Out[ ]=

0.130405	0.773566
0.968951	0.471584

In[103]:=
```
Grid[RandomReal[1, {2, 2}], Dividers → {{All, 2 → Red}, {All, 1 → Blue}}]
```

Out[ ]=

0.639211	0.989318
0.707121	0.362454

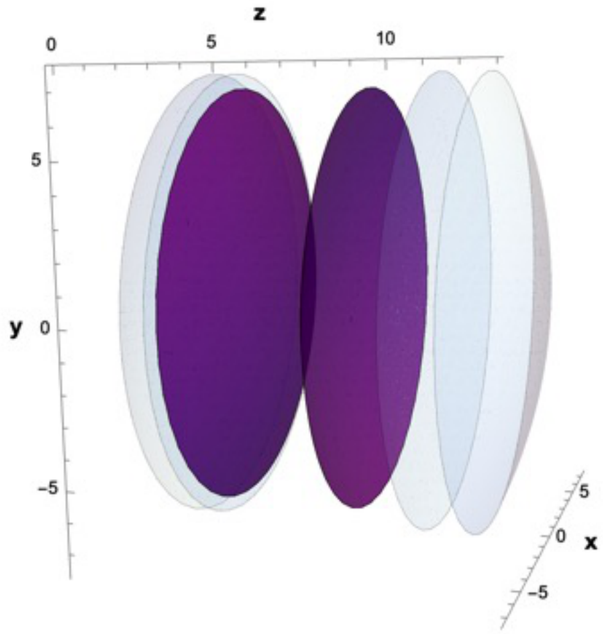
Why not use this input form for multi-surface plots?

In[104]:=
```
PlotSystem[os[[1 ;; 6]], SurfaceGroups → {{1, 2}, {3, 4}, {5, 6}},
 SurfaceStyle → {Directive[Purple, Opacity[0.8]], 1 → Opacity[0.2], 3 → Opacity[0.2]}]
```

z

Why not use this input form for multi-surface plots?

In[104]:=
```
PlotSystem[os[[1 ;; 6]], SurfaceGroups → {{1, 2}, {3, 4}, {5, 6}},
 SurfaceStyle → {Directive[Purple, Opacity[0.8]], 1 → Opacity[0.2], 3 → Opacity[0.2]}]
```



In[105]:=

Style Lists

The style list is a list of instructions for the style of each part of the plot.

In[106]:=
```
{groups, styles} = PlotSystemParams[os[[1 ;; 4]], SurfaceGroups → {{1, 2}, {3, 4}}];
FormattedTable[Keys[styles[[1]]], Values /@ styles]
```

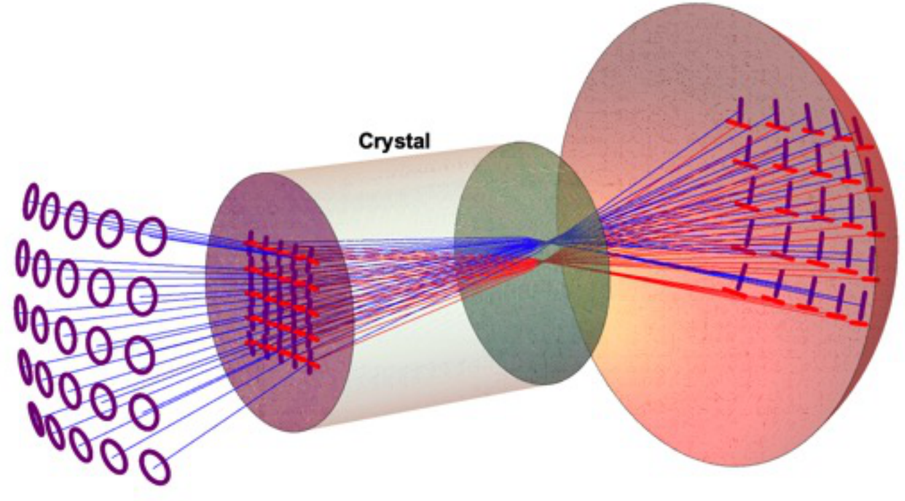
Out[ ]=

SurfaceStyle	SurfaceStyleInherit	LineStyle	LineSt
Directive[Glow[■], Opacity[0.4], Specularity[■, 10], ■]	None	■	
{}	1	{}	
Directive[Glow[■], Opacity[0.4], Specularity[■, 10], ■]	None	■	
{}	3	{}	

We can use the style list to do even more specific styling tasks.

```
(* Plot *)
Show[label,
 PlotSystem[groupedRays, crystal, RayStyleList → styles]]
```

Out[215]=



Now, how do we deal with all of this data?

Interactive Data Tables

FormattedTable

In[85]:=
```
heads = {"String", "Number", "Matrix", "List", "Label"};
data =
  Transpose@{{"apple", "banana", "orange", "pear", "tangerine", "persimmon", "pineapple",
     "prickly pear"}, RandomReal[1, 8], RandomReal[1, {8, 2, 2}], RandomReal[1, {8, 10}],
    {"Keeps the Doctor away", "Part of a healthy breakfast", "The color and the fruit",
     "A great shape to be", "Isn't that just another orange?",
     "Okay, you're just making things up now", "This one is not actually like an apple",
     "Desert pokey bois with good juice"}};
FormattedTable[heads, data]
```

String	Number	Matrix	List	Label
			0.0294918	

```
FormattedTable[heads, data]
```

Out[ ]=

String	Number	Matrix	List	Label
apple	0.560857	$\begin{pmatrix} 0.969355 & 0.410601 \\ 0.902506 & 0.40227 \end{pmatrix}$	$\begin{pmatrix} 0.0294918 \\ 0.0898914 \\ 0.4497 \\ 0.452227 \\ 0.494272 \\ 0.503242 \\ 0.246522 \\ 0.177347 \\ 0.00142325 \\ 0.11999 \end{pmatrix}$	Keeps the Doctor away
banana	0.195707	$\begin{pmatrix} 0.473527 & 0.220787 \\ 0.245463 & 0.552219 \end{pmatrix}$	$\begin{pmatrix} 0.915132 \\ 0.908592 \\ 0.694552 \\ 0.312436 \\ 0.812013 \\ 0.0256433 \\ 0.185176 \\ 0.266113 \\ 0.181368 \\ 0.358157 \end{pmatrix}$	Part of a healthy breakfa

In[89]:=

In[286]:=
```
Manipulate[FormattedTable[heads, data,
  Opts`RowOrientation → a0,
  Opts`HeightLimit → b0,
  Opts`ShowFullTable → c0,
  Opts`MaxRowNumber → d0,
  Opts`TrimDigits → e0,
  Opts`RoundingPrecision → f0,
  Opts`ClickToCopy → g0,
  Opts`UseMatrixForm → h0,
  Opts`UseMulticolumn → i0], {{a0, Horizontal, "RowOrientation"}, {Horizontal, Vertical}},
 {{b0, 0.7, "HeightLimit"}, {0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}},
 {{c0, True, "ShowFullTable"}, {False, True}}, {{d0, 5, "MaxRowNumber"}, {3, 4, 5, 6}},
 {{e0, False, "TrimDigits"}, {False, True}}, {{f0, 4, "RoundingPrecision"}, {1, 2, 3, 4, 5, 6}},
 {{g0, False, "ClickToCopy"}, {False, True}}, {{h0, False, "UseMatrixForm"}, {False, True}},
 {{i0, False, "UseMulticolumn"}, {False, True}}]
```

RowOrientation: Horizontal | Vertical
HeightLimit: 0.7
ShowFullTable
MaxRowNumber: 3 | 4 | 5 | 6
TrimDigits
RoundingPrecision: 2
ClickToCopy
UseMatrixForm
UseMulticolumn

Out[286]=

String	Number	Matrix	List	Label
apple	0.31	$\begin{pmatrix} 0.45 & 0.35 \\ 0.5 & 1. \end{pmatrix}$	$\left(\begin{array}{c\|cc} 0.31 & 0.22 & 0.38 \\ 0.87 & 0.2 & 0.03 \\ 0.34 & 0.62 & \\ 0.44 & 0.34 & \end{array}\right)$	Keeps the Doctor a
banana	0.6	$\begin{pmatrix} 0.54 & 0.82 \\ 0.42 & 0.57 \end{pmatrix}$	$\left(\begin{array}{c\|cc} 0.93 & 0.41 & 0.95 \\ 0.26 & 0.95 & 0.69 \\ 0.59 & 0.08 & \\ 0.32 & 0.08 & \end{array}\right)$	Part of a healthy bre
orange	0.55	$\begin{pmatrix} 0.79 & 0.21 \\ 0.66 & 0.88 \end{pmatrix}$	$\left(\begin{array}{c\|cc} 0.18 & 0.23 & 0.26 \\ 0.06 & 0.08 & 0.04 \\ 0.42 & 0.65 & \\ 0.4 & 0.06 & \end{array}\right)$	The color and the

Left click an element to copy the full data to the clipboard.

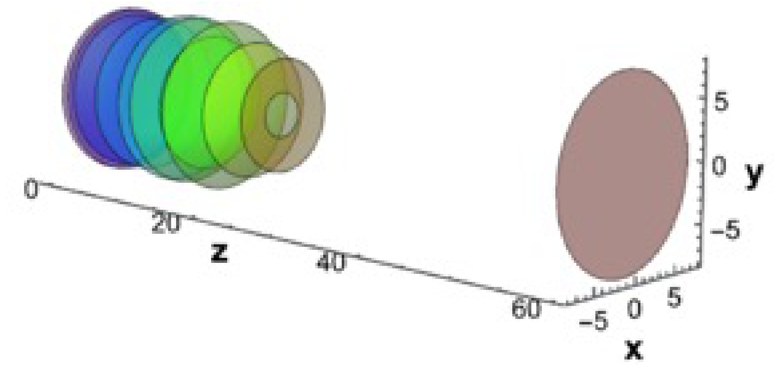
In[91]:=

```
{{0.45111, 0.354491}, {0.498887, 0.998764}}
```

PrintSystem

In[92]:=
```
opticalSystem = Example Optical System ;
PlotSystem[opticalSystem]
```

Out[ ]=



In[96]:=
```
PrintSystem[opticalSystem, CollapseNamedLists → True, UseInfoButtons → True, UseMatrixForm → False]
```

Display Options

Geometric:	☐	sys`v sys`a sys`shape sys`aperture
Identification:	☑	sys`surfID sys`surfaceLabel
Material Information:	☐	sys`material1 sys`material2 sys`coating
Interaction Mode:	☑	sys`mode sys`surfaceScatter

	surfID	mode	surfaceScatter	surfaceLabel

In[96]:= `PrintSystem[opticalSystem, CollapseNamedLists → True, UseInfoButtons → True, UseMatrixForm → False]`

Display Options

Geometric:	☑	sys`v sys`a sys`shape sys`aperture
Identification:	☑	sys`surfID sys`surfaceLabel
Material Information:	☑	sys`material1 sys`material2 sys`coating
Interaction Mode:	☑	sys`mode sys`surfaceScatter

Out[•]=

	surfID	shape	material1	material2
Surface1	1	Sphere	Air	const_{isotropic,{1.460
Surface2	2	Sphere	const_{isotropic,{1.46096}}	air
Surface3	3	Sphere	Air	const_{isotropic,{1.498
Surface4	4	Asphere	const_{isotropic,{1.49800}}	air
Surface5	5	Sphere	{Asphere, $\frac{1}{100}$, 3.5, {0., 2.5×10^{-6}, -1.25×10^{-6}}}	ic,{1.460
Surface6	6	Sphere	const_{isotropic,{1.46096}}	air
Surface7	7	Sphere	Air	const_{isotropic,{1.460
Surface8	8	Sphere	Crystal	air

Stokes Vectors

In[287]:= `StokesEllipticalSelect[]`

Out[287]=

S_0 = 1

ψ = $\frac{3\pi}{10}$

ϵ = −0.5

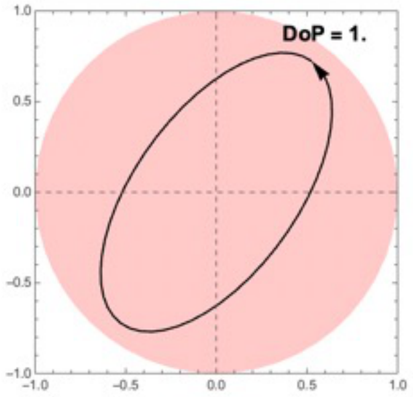
DoP = 1

Stokes Elliptical Select

$$\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} 1. \\ -0.19 \\ 0.57 \\ -0.8 \end{pmatrix}$$

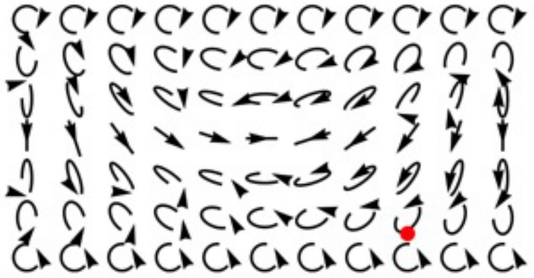
Copy result

Amplitude Polarization Ellipse



Show ψ Show ϵ

Polarization State on Poincare Sphere



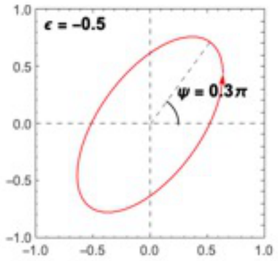
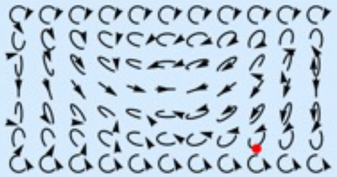
In[76]:=

In[288]:=
```
StokesProperties[{1.`, -0.18541019662496827`, 0.5706339097770922`, -0.7999999999999997`}]
```

Out[288]=

Stokes Properties

Click Cell to Copy Value To Clipboard

Property	Value
Stokes Parameters	$\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} 1. \\ -0.19 \\ 0.57 \\ -0.8 \end{pmatrix}$
Amplitude Polarization Ellipse	
Polarization State on 2D Poincare Map	
Flux, S_0	1.
Major Axis Orientation ψ	radians 0.3π degrees 54.°

Out[288]=

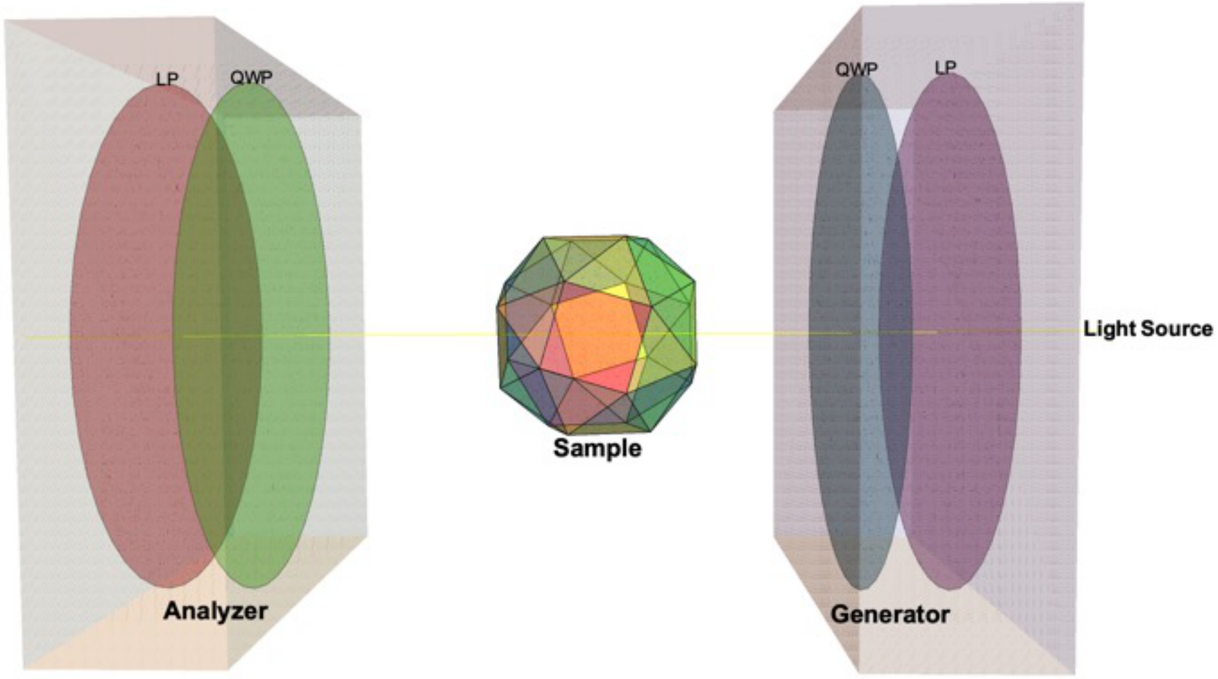
Polarization State on 2D Poincare Map	
Flux, S_0	1.
Major Axis Orientation ψ	radians: 0.3π degrees: 54.°
Ellipticity	-0.5
Degree of Polarization	1.
Degree of Linear Polarization	0.6
Degree of Circular Polarization	-0.8
Polarized Flux Vector	$\begin{pmatrix} I_H \\ I_V \\ I_{45} \\ I_{135} \\ I_L \\ I_R \end{pmatrix} = \begin{pmatrix} 0.41 \\ 0.59 \\ 0.79 \\ 0.21 \\ 0.1 \\ 0.9 \end{pmatrix}$
Jones Vector	Cartesian: $\begin{pmatrix} 0.64 \\ 0.45 + 0.63\,i \end{pmatrix}$ Polar: $\begin{pmatrix} 0.64e^{i0.\pi} \\ 0.77e^{i0.3\pi} \end{pmatrix}$
semi-major and semi-minor axis length	$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0.69 \\ 0.45 \end{pmatrix}$

Copy To Clipboard

Analyzing Real Polarization Data

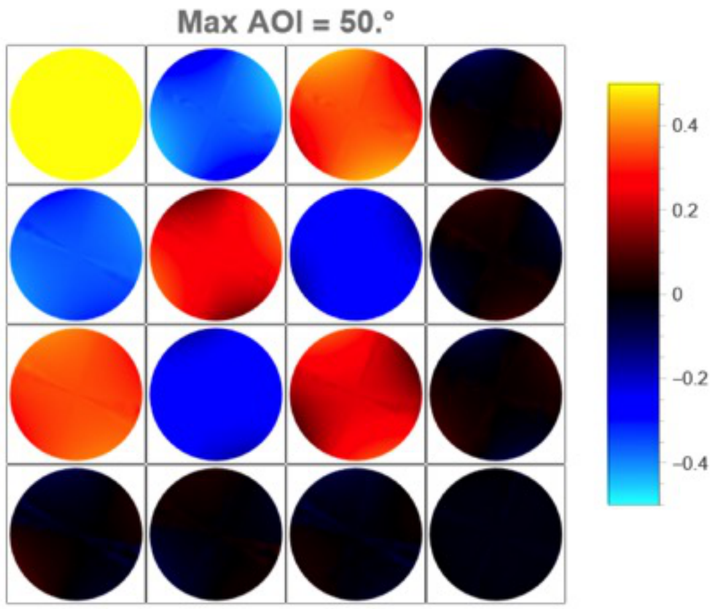
Spreadsheets full of measured polarization data can be visualized and analyzed in a single view.

Out[ ]=



Out[ ]=

Rotation Angle: 22.5 ° **Plot Type:** Mueller Matrix



Conclusion

Mathematica's capabilities for interactivity have helped us provide faster analysis to our customers in visually engaging ways, contributing substantially to the development of the next generation of AR/VR headsets as well as many other exciting applications.

Questions?