Supplementary note 1: Calculation of the polarization properties of the retroreflector assemblies

Calculation of analytic formulas for the symmetry parameters

```
In[1]:= MLinearRetarderId[δ_, θ_] :=
  ( 1  0                              0                              0
    0  Cos[2 θ]^2 + Sin[2 θ]^2 * Cos[δ]  Sin[2 θ] Cos[2 θ] (1 - Cos[δ])  -Sin[2 θ] Sin[δ]
    0  Sin[2 θ] Cos[2 θ] (1 - Cos[δ])  Sin[2 θ]^2 + Cos[2 θ]^2 Cos[δ]  Cos[2 θ] Sin[δ]
    0  Sin[2 θ] Sin[δ]                 -Cos[2 θ] Sin[δ]                Cos[δ] )
  (* Eq. S1 *)

In[2]:= MLinearDiattenuator[θ_] :=
  1 / 2 ( 1         Cos[2 θ]           Sin[2 θ]           0
          Cos[2 θ]  Cos[2 θ]^2         Sin[2 θ] Cos[2 θ]  0
          Sin[2 θ]  Sin[2 θ] Cos[2 θ]  Sin[2 θ]^2         0
          0         0                  0                  0 ) (* Eq. S2 *)

In[3]:= MM = ( 1  0  0   0
               0  1  0   0
               0  0  -1  0
               0  0  0   -1 ) ; (* Eq. S3 *)

In[4]:= MCCR[θ_, α_, type_] :=
   If[type == 0, (MLinearRetarderId[(90 °) , - (θ)].MLinearRetarderId[(90 °) , - (θ + α)].
      MM.MLinearRetarderId[(90 °) , θ + α].MLinearRetarderId[(90 °) , θ]),
    (MLinearRetarderId[(90 °) , - (θ)].MLinearDiattenuator[- (θ + α)].MM.
      MLinearDiattenuator[θ + α].MLinearRetarderId[(90 °) , θ])];
  (* Eq. S4 with type ==0 for assemblies with 2nd waveplate and type ==
   1 fo assemblies with polarizer *)

In[5]:= I1[θ_, α_, type_] := (MLinearDiattenuator[0 °].
      MLinearRetarderId[90 °, 45 °].MCCR[θ, α, type].{1, 0, 0, 1}) [[1]]; (*Eq. S5*)
  I2[θ_, α_, type_] := (MLinearDiattenuator[0 °].MLinearRetarderId[90 °, -45 °].
      MCCR[θ, α, type].{1, 0, 0, 1}) [[1]];
  I3[θ_, α_, type_] := (MLinearDiattenuator[0 °].MLinearRetarderId[90 °, 45 °].
      MCCR[θ, α, type].{1, 0, 0, -1}) [[1]];
  I4[θ_, α_, type_] := (MLinearDiattenuator[0 °].MLinearRetarderId[90 °, -45 °].
      MCCR[θ, α, type].{1, 0, 0, -1}) [[1]];

In[9]:= Ia[θ_, α_, type_] := I1[θ, α, type] + I2[θ, α, type];
  Ib[θ_, α_, type_] := I3[θ, α, type] + I4[θ, α, type];

In[11]:= SymPar1[θ_, α_, type_] :=
   Simplify[(Ia[θ, α, type] - Ib[θ, α, type]) / (Ia[θ, α, type] + Ib[θ, α, type])];
  SymPar2[θ_, α_, type_] := Simplify[
    (I1[θ, α, type] - I2[θ, α, type]) / (I1[θ, α, type] + I2[θ, α, type])];
  SymPar3[θ_, α_, type_] := Simplify[(I3[θ, α, type] - I4[θ, α, type]) /
     (I3[θ, α, type] + I4[θ, α, type])];
```

In[14]:=

```
Simplify[SymPar1[θ, α, 0]] (*Eq. S6*)
Simplify[SymPar2[θ, α, 0]]
Simplify[SymPar3[θ, α, 0]]
```

Out[14]= 0

Out[15]= Cos[4 α]

Out[16]= -Cos[4 α]

In[17]:=

```
Simplify[SymPar1[θ, α, 1]] (*Eq. S7*)
Simplify[SymPar2[θ, α, 1]]
Simplify[SymPar3[θ, α, 1]]
```

Out[17]= Sin[2 α]

Out[18]= -Sin[2 α]

Out[19]= -Sin[2 α]

Generation of Supplementary Figure 1

In[20]:=

```
xlabelPos = {-45, -15, 0, 15, 45};
xlabel = {"CCR3", "CCR4", "CCR5", "CCR6", "CCR7"};
xlabelPos2 = {0, 45};
xlabel2 = {"CCR1", "CCR2"};
```

In[24]:=

```
SupFig1P1 = Plot[{0, Cos[4 * x * Degree], -Cos[4 * x * Degree]}, {x, 0, 45 },
   Frame → True, FrameLabel → {{"Symmetry Parameter", ""}, {"α / °", ""}},
   PlotLegends → {"P1", "P2", "P3"}, PlotStyle → {Automatic, Automatic,
     Automatic, {Dashed, Red}, {Dashed, Purple}, {Dashed, Pink}}, FrameTicks →
    {{Automatic, Automatic}, {Automatic, Transpose[{xlabelPos2, xlabel2}]}},
   GridLines → {xlabelPos2, {-2, 3}}, GridLinesStyle → Directive[Blue, Dashed],
   Axes → False, ImageSize → 300, PlotLabel → "CCRs with two quarter waveplates:"];
```

In[25]:=

```
SupFig1P2 = Plot[{Sin[2 * x * Degree], -Sin[2 * x * Degree]}, {x, -45, 45 },
   Frame → True, FrameLabel → {{"Symmetry Parameter", ""}, {"α / °", ""}},
   PlotLegends → {"P1", "P2=P3"}, PlotStyle →
    {Automatic, Automatic, Automatic, {Dashed, Red}, {Dashed, Purple}, {Dashed, Pink}},
   FrameTicks → {{Automatic, Automatic}, {Automatic, Transpose[{xlabelPos, xlabel}]}},
   GridLines → {xlabelPos, {-2, 2}},
   GridLinesStyle → Directive[Blue, Dashed], Axes → False, ImageSize → 300,
   PlotLabel → "CCRs with quarter waveplate and polarizer:"];
```

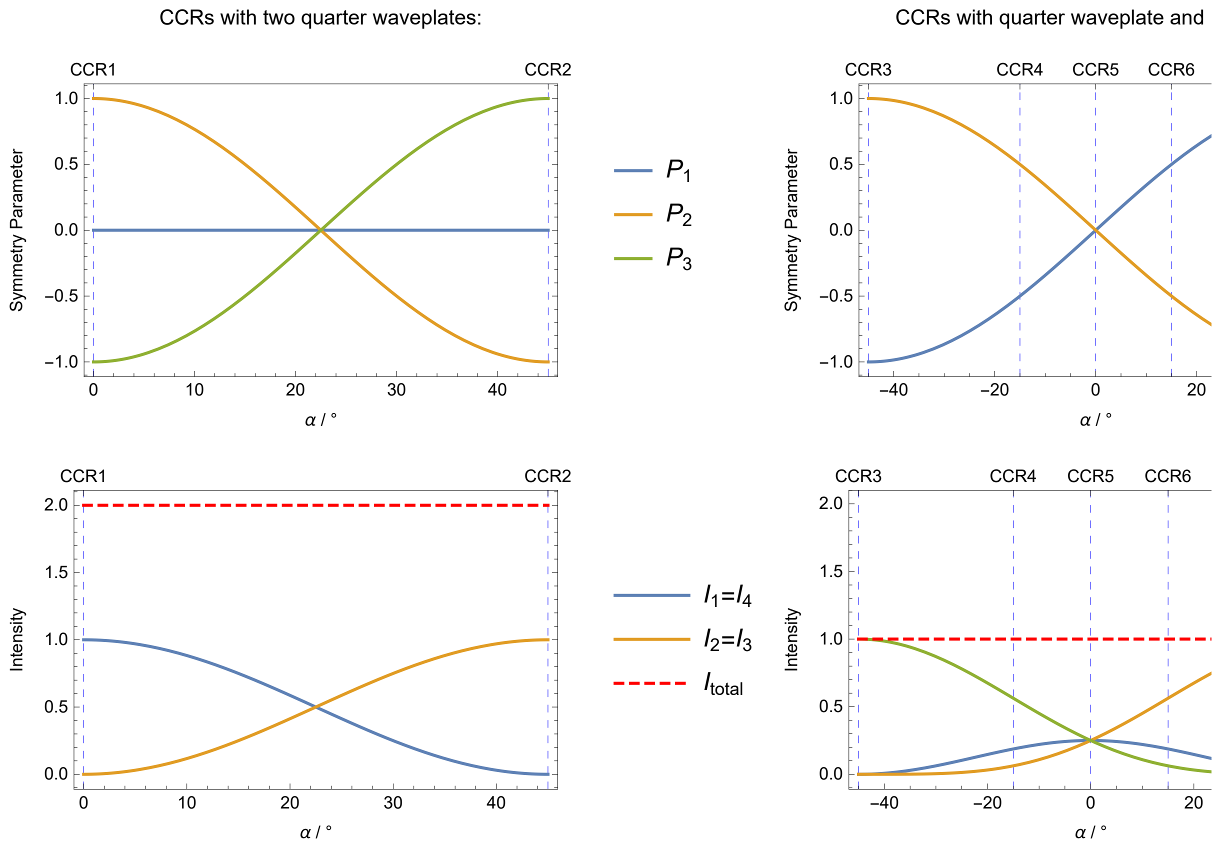
In[26]:=

```
SupFig1P3 = Plot[{I1[0, α, 0] /. α → (x Degree),
    I2[0, α, 0] /. α → (x Degree), (Ia[0, α, 0] + Ib[0, α, 0]) /. α → (x Degree)},
   {x, 0, 45 }, Frame → True, FrameLabel → {{"Intensity", ""}, {"α / °", ""}},
   PlotLegends → {"I1=I4", "I2=I3", "Itotal"},
   PlotStyle → {Automatic, Automatic, {Dashed, Red}, {Dashed, Purple}, {Dotted, Pink}},
   GridLines → {xlabelPos2, {-2, 3}}, GridLinesStyle → Directive[Blue, Dashed],
   Axes → False, ImageSize → 300, FrameTicks →
    {{Automatic, Automatic}, {Automatic, Transpose[{xlabelPos2, xlabel2}]}}];
```

In[27]:=
```
SupFig1P4 = Plot[{I1[0, α, 1] /. α → (x Degree), I2[0, α, 1] /. α → (x Degree),
    I3[0, α, 1] /. α → (x Degree), (Ia[0, α, 1] + Ib[0, α, 1]) /. α → (x Degree)},
   {x, -45, 45}, Frame → True, FrameLabel → {{"Intensity", ""}, {"α / °", ""}},
   PlotLegends → {"I1=I4", "I2", "I3", "Itotal"}, PlotStyle →
    {Automatic, Automatic, Automatic, {Dashed, Red}, {Dashed, Purple}, {Dotted, Pink}},
   GridLines → {xlabelPos, {-2, 3}}, GridLinesStyle → Directive[Blue, Dashed],
   Axes → False, ImageSize → 300, PlotRange → {All, {All, 2.1}},
   FrameTicks → {{Automatic, Automatic}, {Automatic, Transpose[{xlabelPos, xlabel}]}}];
```

In[28]:=
```
PlInt = Grid[{{SupFig1P1, SupFig1P2}, {SupFig1P3, SupFig1P4}},
  Alignment → {{Left, Left}, {Left, Left}}]
```

Out[28]=



In[29]:=
```
SetDirectory[NotebookDirectory[]];
Export["PlInt.jpg", PlInt, ImageResolution → 500]
```

Out[30]= PlInt.jpg

Supplementary note 4: Radar link equation

In[31]:=
```
λ = 1064 * 10^-9; (* Wavelength of SLR station in m *)
Et = 150 * 10^-6; (* Laser pulse energy in J *)
ηt = 0.5; (* Efficiency of transmitter *)
h = 6.626 * 10^-34; (*Planck constant in J/Hz*)
c = 3 * 10^8; (*Speed of light in m/s *)
ρ = 0.9; (*Reflectance of retroreflector *)
Acc[rCC_] := Pi * rCC^2; (* Area of retroreflector front face*)
σ0[rCC_] := ρ * ((4 Pi) / λ^2) Acc[rCC]^2; (* Retroreflector peak,
on-axis cross section, see eq. 6.1.1 of Degnan,
"Millimeter Accuracy Satellite Laser Ranging: A Review
" *)
σkorr = 0.05; (* Correction factor accounting for the CCR
cross section as function of incidence angle and
velocity aberration *)
Ar = Pi / 4 * ((0.23^2 - 0.1^2)^2); (* Area of the receiving telescope *)
ηr = 0.5; (*efficiency of receiving optics *)
ηq = 0.3 ; (* efficiency of detector *)
Ta = 0.8; (* atmospheric transmission *)
Tc = 0.8; (* atmospheric transmission *)
θt = 50 * 10^-6; (* Laser beam divergence *)
Gt = (8 / θt^2) * Exp[-2 * 0.5^2]; (* Transmitter gain  *)
ηpol = 0.25; (* polarimetric transmission factor *)
ηpc = 1; (* Transmission factor in case of a pulse collision avoidance *)
ηLCVR = 1 - (2 * 50. + 2 * 1)/(4 * 50); (* correction factor for the switching time of the LCVR *)
reprate = 75000; (* Laser pulse rep. rate in Hz *)
```

In[51]:=
```
etap[d_, rCC_] := (Et * λ)/(h * c) ηt Gt/(4 * Pi * d^2) σ0[rCC]/(4 * Pi * d^2) * σkorr * Ar * ηr * ηq * Ta^2 * Tc^2 (*Eq. S8*)
```

In[52]:=
```
RE = 6387; (* Erdradius / km *)
hs = 600; (* Satellitenhöhe /km *)
```

In[54]:=
```
dis[ϕ_] := Sqrt[RE^2 * Cos[ϕ]^2 + 2 * RE * hs + hs^2] - RE * Cos[ϕ] (* Eq. S9 *)
```

In[55]:=
```
Angle[dis2_] :=
  (ϕ /. Solve[dis2 == Sqrt[RE^2 * Cos[ϕ]^2 + 2 * RE * hs + hs^2] - RE * Cos[ϕ], ϕ][[2, 1]]) / Degree
```

In[56]:=
```
MajorTicksV = 10;
MajorWerteTabelle = Table[i * 1., {i, 0, 90, MajorTicksV}];
MajorWertePos =
  Table[dis[MajorWerteTabelle[[i]] * Degree], {i, 1, Length[MajorWerteTabelle]}];
MajorTicks = Transpose[{MajorWertePos, MajorWerteTabelle}];
```

In[60]:=
```
MajorWerteTabelle = {600.01, 700., 800., 1000., 1200.};
MajorWerteTabelle2 = {600, 700, 800, 1000, 1200};
MajorWertePos =
   Table[Angle[MajorWerteTabelle[[i]]], {i, 1, Length[MajorWerteTabelle]}]; // Quiet
MajorTicks = Transpose[{MajorWertePos, MajorWerteTabelle2}];
```

In[64]:=
```
CountPerSecondTable = {10, 20, 50, 100, 500, 1000, 5000, 10000, 50000, 100000};
CountPerSecondTicks = Transpose[{CountPerSecondTable / reprate, CountPerSecondTable}];
```

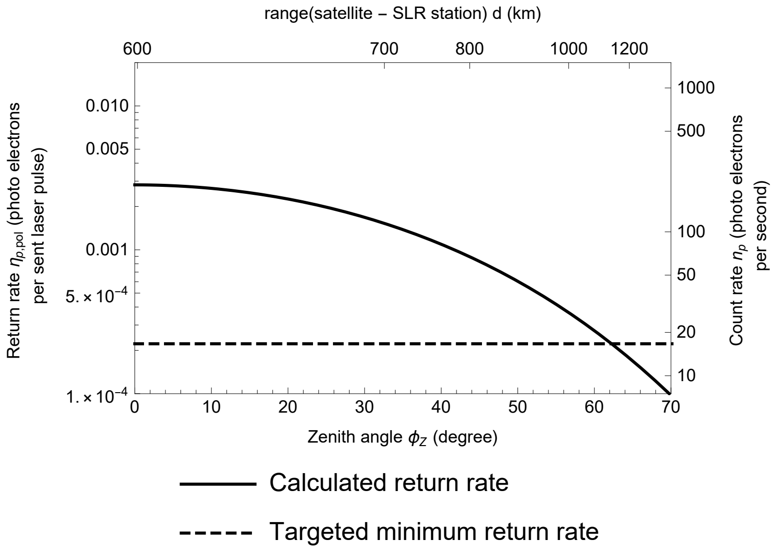
In[66]:=
$$\sigma korrCalc[rCC\_, \phi\_] := \left(\frac{2 * BesselJ\left[1, \frac{2 Pi}{\lambda} * rCC * Sin[\phi]\right]}{\frac{2 Pi}{\lambda} Sin[\phi]}\right)$$

Corrections due to polarimetric effects

In[67]:=
```
etapkorr[d_, rCC_] := etap[d, rCC] * ηpol * ηpc * ηLCVR
```

In[68]:=
```
RadarLink =
  LogPlot[{etapkorr[dis[ϕ * Degree] * 1000, 6.1 * 10^-3], 1000 / reprate / 60, 5000},
   {ϕ, 0, 90}, PlotRange → {{0, 70}, {1 * 10^-4, 2 * 10^-2}}, Frame → True,
   FrameTicks → {{Automatic, CountPerSecondTicks}, {Automatic, MajorTicks}},
   PlotStyle → {Black, {Black, Dashed}},
   FrameLabel → {{"Return rate η_p,pol (photo electrons \n per sent laser pulse)",
      "Count rate n_p (photo electrons \n per second)"}, {"Zenith angle ϕ_Z (degree)",
      "range(satellite - SLR station) d (km)"}}, ImageSize → 400, PlotLegends →
    Placed[{"Calculated return rate", "Targeted minimum return rate"}, Below]]
```

Out[68]=



In[69]:=
```
SetDirectory[NotebookDirectory[]];
Export["ReturnRate.jpg", RadarLink, ImageResolution → 500]
```

Out[70]= ReturnRate.jpg

Supplementary note 4 : Calculation of the retroreflector cross section

Definitions

In[71]:=
```
ArcSecond = 4.84814 * 10^-6; (*conversion from rad to μrad  *)
```

```
In[72]:= microrad = 1 * 10^-6;

In[73]:= CalcDFμ[θi_ , r_, n_, l_, λ_, x_] :=
  Module[{θip, Fd, F, μ, μD, ηd(*Local variables*)},
   θip = ArcSin[Sin[θi]/n];
   Fd[ηd_] := Cos[θi] * (Sqrt[r^2 - ηd^2] - l * Tan[θip]);
   μD = r * Sqrt[1 - (l / r)^2 * Tan[θip^2]];
   F[ηd_] := Fd[ηd * r]/r;
   μ = μD/r;
   {F[x], μ}(*Output to be returned*)]

In[74]:= CalcNormCrossSection[θi_ , r_, n_, l_, λ_] :=
  Module[{F, μ, ξm, ηm, αm , SinCC, Intm, W, h, Num, x, y, NormCrossSection,
    NormCrossSection2, AnzahlPunkte, XMax(*Local variables*)},
   F[x_] := CalcDFμ[θi, r, n, l, λ, x][[1]];
   μ = CalcDFμ[θi , r, n, l, λ, 0][[2]];
   Num = 100; (* Number of Integration Intervals *)
   h = μ/Num;
   ξm = Table[0.5 * (F[m * h] + F[(m - 1) * h]), {m, 1, Num}];
   ηm = Table[(m - 0.5) * h, {m, 1, Num}];
   αm = Table[0.5 * (F[m * h] - F[(m - 1) * h]), {m, 1, Num}];
   SinCC[ω_] := If[ω == 0, 1, Sin[ω / 2]/(ω / 2)];
   Intm[x_, y_, m_] :=
    (2 * h)/x * (Sin[ξm[[m]] * x + ηm[[m]] * y] * SinCC[h * (y + αm[[m]] * x)] +
       Sin[ξm[[m]] * x - ηm[[m]] * y] * SinCC[h * (y - αm[[m]] * x)]);
   W[x_, y_] := Sum[Intm[x, y, m], {m, 1, Num}];
   AnzahlPunkte = 100;
   XMax = 6;
   NormCrossSection =
    Table[{ArcTan[(y*λ)/(2*Pi*r)]/microrad, W[y, 0.001]^2/9.86172620865356}, {y, 0.0001, XMax, XMax/AnzahlPunkte}];
   NormCrossSection2 = Table[{ArcTan[(y*λ)/(2*Pi*r)]/microrad, W[0.001, y]^2/9.86172620865356},
     {y, 0.0001, XMax, XMax/AnzahlPunkte}];
   {NormCrossSection, NormCrossSection2}(*Output to be returned*)]

In[75]:=
```

Calculation of the cross-section

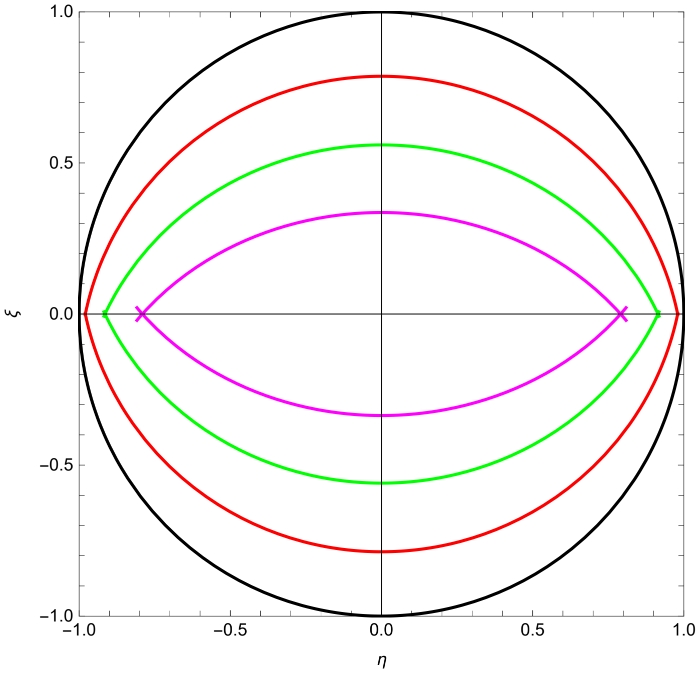
In[76]:=
```
rSet = 6.1 * 10^-3;
nSet = 1.4496; (* fused silica at 1064 nm *)
lSet = 10.16 * 10^-3;
λSet = 1064 * 10^-9;
```

In[80]:=
```
θiTable = {0.01, 10, 20, 30} * Degree;
ColorTable = {Black, Red, Green, Magenta};
```

In[82]:=
```
F1[ηd_, θi_] := CalcDFμ[θi, rSet, nSet, lSet, λSet, ηd][[1]]
μ1[θi_] := CalcDFμ[θi , rSet, nSet, lSet, λSet, 0][[2]]
```

In[84]:=
```
FigA = Show[Table[Plot[{F1[ηd, θiTable[[i]]], -F1[ηd, θiTable[[i]]]},
    {ηd, -μ1[θiTable[[i]]], μ1[θiTable[[i]]]}, Frame → True,
    PlotStyle → {ColorTable[[i]], ColorTable[[i]]}, AspectRatio → Automatic,
    FrameLabel → {"η", "ξ"}, PlotRange → {{-1, 1}, {-1, 1}}], {i, 1, Length[θiTable]}]]
```

Out[84]=



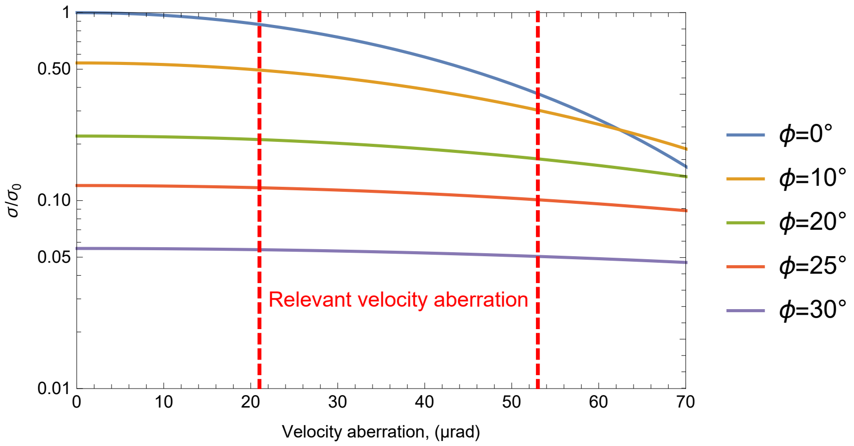
In[85]:=
```
Cl1[θ_] := CalcNormCrossSection[θ , rSet, nSet, lSet, λSet][[1]]
```

In[86]:=
```
line1 = Line[{{21, Log[0.01]}, {21, Log[1.1]}}];
line2 = Line[{{53, Log[0.01]}, {53, Log[1.1]}}];
```

In[88]:=
```
lineStyle = {Thick, Red, Dashed};
```

In[89]:=
```
Pla = ListLogPlot[
  {Cl1[0.01 Degree], Cl1[10 Degree], Cl1[20 Degree], Cl1[25 Degree], Cl1[30 Degree]},
  PlotRange → {{0, 70}, {0.01, 1}}, Joined → True, Axes → False,
  Frame → {{True, True}, {True, True}}, FrameStyle -> Directive[Black] ,
  FrameLabel → {"Velocity aberration, (μrad)", "σ/σ0"},
  PlotLegends → {"ϕ=0°", "ϕ=10°", "ϕ=20°", "ϕ=25°", "ϕ=30°"},
  Epilog → {Directive[lineStyle], line1, line2,
    Style[Text["Relevant velocity aberration", {37, Log[0.03]}], 11]}]
```

Out[89]=



In[90]:=
```
SetDirectory[NotebookDirectory[]];
Export["RetroreflectorCrossSection.jpg", Pla, ImageResolution → 500]
```

Out[91]= RetroreflectorCrossSection.jpg

Supplementary note 4 : Statistical error in the determination of the symmetry parameters

We assume that there is a single polarimetric retroreflector with the symmetry parameters P1 = P2 = P3 = 0, meaning that we should measure equal intensities (I1=I2=I3=I4) for a large number of detected photons. To simulte the effect of a limited number of the detected photons, we generate a random dataset of integers in the range between 1 to 4.

In[92]:=
```
IntegerList[no_] := RandomInteger[{1, 4}, no]
```

In[93]:=
```
IntegerList[50]
```

Out[93]=
```
{2, 2, 2, 4, 4, 1, 1, 3, 3, 3, 1, 2, 3, 4, 1, 2, 2, 4, 2, 4, 3, 3, 2, 2,
 4, 4, 4, 1, 4, 2, 2, 4, 1, 1, 4, 3, 4, 1, 1, 4, 2, 4, 4, 4, 2, 2, 3, 1, 3, 4}
```

The count of the element “1” would be I1, the count of element “2” is I2, the count of element “3” is I3 and the count of element “4” is I4. From these intensities, we can calculate the symmetry parameters P1 to P3.

```
In[94]:= CalculateSymmetries[DataSet_] := Module[{I1, I2, I3, I4, Ia, Ib, P1, P2, P3},
    I1 = Count[DataSet, 1];
    I2 = Count[DataSet, 2];
    I3 = Count[DataSet, 3];
    I4 = Count[DataSet, 4];
    Ia = I1 + I2;
    Ib = I3 + I4;
    P1 = (Ia - Ib)/(Ia + Ib);
    P2 = (I1 - I2)/(I1 + I2);
    P3 = (I3 - I4)/(I3 + I4);
    {P1, P2, P3}];
```

The calculation of the symmetry paramters is repeated for 100 times and we build an array of values in P1, P2, P3. The “error” is defined as the difference between the maximum and minimum value in any of the symmetry parameters divided by 2.

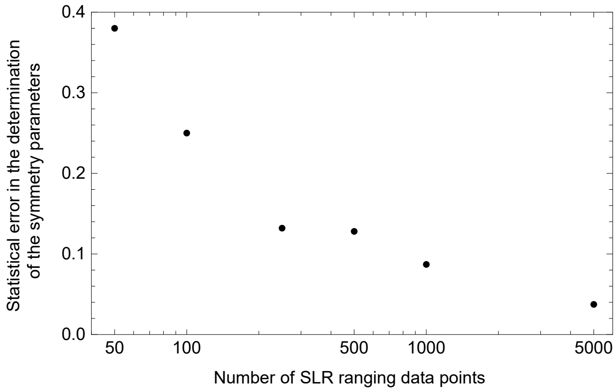
```
In[95]:= CalculateError[noPhotons_] :=
  Module[{P1Array, P2Array, P3Array, P1, P2, P3, TotalPArray, Error},
   P1Array = {};
   P2Array = {};
   P3Array = {};
   For[i = 0, i < 100, i++,
    {P1, P2, P3} = CalculateSymmetries[IntegerList[noPhotons]];
    AppendTo[P1Array, P1];
    AppendTo[P2Array, P1];
    AppendTo[P3Array, P1]
   ];
   TotalPArray = Join[P1Array, P2Array, P3Array];
   Error = (Max[TotalPArray] - Min[TotalPArray]) / 2
  ]
```

Now we generate Supplementary Figure 12. Note that the data is slightly different, everytime the calculation is run. This is because it uses data from random number generation.

```
In[96]:= noPhotons = {50, 100, 250, 500, 1000, 5000};
ErrorList = Table[CalculateError[i], {i, noPhotons}];
```

In[98]:=
```
ErrorPlot = ListLogLinearPlot[Transpose[{noPhotons, ErrorList}],
  PlotRange → {{40, 6000}, {0, 0.4}}, Frame → True, FrameLabel →
   {{"Statistical error in the determination \n of the symmetry parameters", ""},
    {"Number of SLR ranging data points ", ""}}, PlotStyle → Black, ImageSize → 330]
```

Out[98]=



In[99]:=

In[100]:=
```
SetDirectory[NotebookDirectory[]];
Export["Accuracy.jpg", ErrorPlot, ImageResolution → 500]
```

Out[101]= Accuracy.jpg