

# Polarization Skyrmions

## Function definitions and other admin

---

### Initial admin

#### Sundry admin

```
In[ ]:= Get[FileNameJoin[{NotebookDirectory[], "Color-functions.m"}]]
```

```
In[ ]:= SetDirectory[FileNameJoin[{NotebookDirectory[], "..", "figures"}]];
```

#### ComplexFocus package

```
In[ ]:= Needs["ComplexFocus`",  
  FileNameJoin[{NotebookDirectory[], "ComplexFocus", "ComplexFocus.m"}]]
```

```
In[ ]:= $ComplexFocusVersion
```

```
Out[ ]:= ComplexFocus v1.0, Tue 1 Dec 2020 22:39:54
```

#### MaTeX

```
In[ ]:= << MaTeX`
```

```
In[ ]:= SetOptions[MaTeX, FontSize → 12, Magnification → 1, "Preamble" → {}];
```

If MaTeX is not installed, make sure the MaTeX-1.7.7.paclet file is in the notebook's directory (obtainable from <https://github.com/szhorvat/MaTeX>) and run the cell below.

```
In[ ]:= Needs["PacletManager`"]  
PacletInstall[FileNameJoin[{NotebookDirectory[], "MaTeX-1.7.7.paclet"}]];
```

```
In[ ]:= MaTeX["x^2"]
```

```
Out[ ]:=  $x^2$ 
```

# Plotting admin

## Plotting functions

### sphereANE

*In[#:]=*

```
sphereANE[axeslbl_, ts_ : 26, font_ : "Times New Roman"] := Block[{sphere, ecuators},
  sphere = Graphics3D[{
    {
      Opacity[0.2], Specularity[Gray, 5], Lighter[Gray, 2/5],
      Sphere[{0, 0, 0}]
    },
    Black, Thick, PointSize[0.015],
    Point[{0, 0, 0}],
    Thickness[0.004],
    Line[{{-1, 0, 0}, {1, 0, 0}}],
    Line[{{0, -1, 0}, {0, 1., 0}}],
    Line[{{0, 0, -1}, {0, 0, 1.}}],
    Text[Style[axeslbl[[1]], ts, FontFamily → font], {1.2, 0, 0}],
    Text[Style[axeslbl[[2]], ts, FontFamily → font], {0, 1.2, 0}],
    Text[Style[axeslbl[[3]], ts, FontFamily → font], {0, 0, 1.2}]
  ]];
  ecuators = ParametricPlot3D[
    {{Cos[t], Sin[t], 0}, {0, Cos[t], Sin[t]}, {Cos[t], 0, Sin[t]}}
    , {t, 0, 2  $\pi$ }
    , PlotStyle → {{Black, Dashed}, {Black, Dashed}}
    , Axes → False
  ];
  Show[sphere, ecuators]
]
```

## PoincaranaSpinX

In[ ]:=

```
PoincaranaSpinX[vecL_] := Block[{sphere, equators, ts = 26},
  Show[
    sphereANE[{"s_x", "s_y", "s_z"}],
    Graphics3D[{
      If[
        Norm[#[[1]] + #[[2]]] == 0,
        Black,
        Darker[
          ColorData["Rainbow"][ $\frac{1}{2} - \frac{1}{2} \left( \frac{\#[[1]] + \#[[2]]}{\text{Norm}[\#[[1]] + \#[[2]]]} \right) [[3]]$ ], 1 - Norm[ $\frac{\#[[1]] + \#[[2]]}{2}$ ]]
        ],
      PointSize[0.02],
      Line[#, Point[#],
      If[
        Norm[#[[1]] + #[[2]]] ≠ 0,
        Arrow[{ {0, 0, 0},  $\frac{\#[[1]] + \#[[2]]}{\text{Norm}[\#[[1]] + \#[[2]]]}$  }],
        Null
      ]
    } & /@ (Poincarana /@ vecL)
  ]
, Boxed → False
, ViewPoint → {7, 6, 3}
, ImageSize → 400
, Method → {"ShrinkWrap" → True}
]
```

## PoincaranaMajorX

In[ ]:=

```
PoincaranaMajorX[vecL_] := Block[{sphere, equators, ts = 26},
  Show[
    sphereANE[{"a_x", "a_y", "a_z"}],
    Graphics3D[{
      ColorData["BlueGreenYellow"][ $-(\#) / \text{Norm}[\#] [[3]] / 2 + 1 / 2$ ],
      PointSize[0.02],
      Point[# / Norm[#]], Arrow[{ {0, 0, 0}, (#) }]
    } & /@ (JonesMajorAxisA[#, Norm] & /@ vecL)
  ]
, Boxed → False
, ViewPoint → {7, 6, 3}
, ImageSize → 400
, Method → {"ShrinkWrap" → True}
]
```

## PlotSkyrmionFigure components and assembly

### plotSkyrmionIntensity

```
In[ ]:= plotSkyrmionIntensity[fieldFunction_, rm_, scAmp_, maxIntensity_] := Plot3D[
  scAmp Norm[fieldFunction[x, y] / maxIntensity]
, {x, -rm, rm}, {y, -rm, rm}
, PlotTheme → "Web"
, ColorFunction → Function[{x, y, z}, Viridis[z]]
, PlotPoints → 25
, Boxed → False
, LabelStyle → Black
, Axes → False
, BoxRatios → Automatic
, PlotRange → All
]
```

### skyrmionPlotColorScale

```
In[ ]:= skyrmionPlotColorScale[scAmp_, rm_, lowerLimit_] := Graphics3D[{
  Inset[MaTeX["| | \mathbf{E} | |", FontSize → 10, Magnification → 0.9],
    {rm, rm, 5 scAmp / 4}, Scaled[{0.4, 0.5}]],
  Inset[MaTeX[NumberForm[lowerLimit, {1, 2}], FontSize → 10, Magnification → 0.7],
    {rm, rm, 3 scAmp / 4}],
  Inset[MaTeX["1.00", FontSize → 10, Magnification → 0.7], {rm, rm, 7 scAmp / 4}],
  Lighting → {"Ambient", White},
  ReplaceAll[
    First[
      DensityPlot[y, {x, -.5 rm / 7, .5 rm / 7},
        {y, 0, scAmp}, ColorFunction → Viridis, Frame → False]
    ],
    {x_?AtomQ, y_?AtomQ} :>
      {6 rm / 7, 6 rm / 7, 3 scAmp / 4} + {x Cos[ $\pi$  / 4], x Sin[ $\pi$  / 4], y}
    (*lifting from 2D to 3D*)
  ]
}]
```

## skyrmionPlotAxes

In[ ]:=

```
skyrmionPlotAxes[rm_, axS_] := Graphics3D[{
  Line[{{-rm, -rm, 0}, {rm, -rm, 0}}],
  Line[{{rm, -rm, 0}, {rm, rm, 0}}],
  Table[{
    Inset[MaTeX[ToString[x], FontSize → 9], {x, -axS rm, 0}],
    Inset[MaTeX[ToString[x], FontSize → 9], {axS rm, x, 0}]
  }, {x, -2, -rm, -2}],
  Table[{
    Line[{{-x, -rm, 0}, {-x, -.95 rm, 0}}],
    Line[{{x, -rm, 0}, {x, -.95 rm, 0}}],
    Line[{{rm, -x, 0}, {.95 rm, -x, 0}}],
    Line[{{rm, x, 0}, {.95 rm, x, 0}}],
    Inset[MaTeX[ToString[x], FontSize → 9], {x, -axS rm, 0}],
    Inset[MaTeX[ToString[x], FontSize → 9], {axS rm, x, 0}]
  }, {x, 0, rm, 2}],
  Inset[MaTeX["k\\,x", FontSize → 9], {0, -1.32 rm, 0}],
  Inset[MaTeX["k\\,y", FontSize → 9], {1.32 rm, 0, 0}]
}]
```

## skyrmionPlotCircleLift

In[ ]:=

```
skyrmionPlotCircleLift[translation_,
  lowerCircleHeight_, sm_, zoom_, lineAngle_ : (- $\pi$  / 4)] := Graphics3D[{
  (*FaceForm[],
  EdgeForm[{Thickness[0.003], Red, Dashing[{0.015, 0.015}]}], *)
  (*EdgeForm[{Thickness[0.003], Black, Dashing[{0.015, 0.9}]}], *)
  (*Cylinder[{{0, 0, lowerCircleHeight}, {0, 0, lowerCircleHeight + 0.01}}, sm],
  Cylinder[{translation, translation + {0, 0, 0.01}}, zoom sm], *)

  Thickness[0.003], Black, Dashing[{0.015, 0.015}],
  Line[
    Join[#, #[[1]]] &[Map[Join[#, {lowerCircleHeight}] &, CirclePoints[sm, 50]]]],

  Line[Join[#, #[[1]]] &[Map[Join[#, {translation[[3]]}] &,
    CirclePoints[translation[[1, 2]], zoom × sm, 50]]]],

  With[{ $\theta$ r = lineAngle}, {
    Thickness[0.003],
    Dashing[{0.015, 0.015}],
    Line[{{sm Cos[ $\theta$ r], sm Sin[ $\theta$ r], lowerCircleHeight},
      {zoom sm Cos[ $\theta$ r], zoom sm Sin[ $\theta$ r], 0} + translation}],
    Line[{{sm Cos[ $\pi$  +  $\theta$ r], sm Sin[ $\pi$  +  $\theta$ r], lowerCircleHeight},
      {zoom sm Cos[ $\pi$  +  $\theta$ r], zoom sm Sin[ $\pi$  +  $\theta$ r], 0} + translation}]
  }]
}]
```

plotSkyrmionVectorComponents

In[8]:=

```

plotSkyrmionVectorComponents[kind_, field_, rm_, showLegend_ : False] :=
Block[{colorList, legendTags, legend, vectorFunction, vectPlot, style},
  colorList = Map[RGBColor,  $\frac{1}{255}$  {{0, 119, 187}, {0, 153, 136}, {238, 119, 51},
    {204, 51, 17}, {238, 51, 119}, {51, 187, 238}, {187, 187, 187}}];
  legendTags = MaTeX[{"||\\mathbf{" <#>"}||", # <"_\\rho", # <"_\\phi", # <
    "_z"}, FontSize → 10, Magnification → 0.8] &@If[kind == "major", "A", "S"];
  legend = Row[Flatten[Table[{
    Graphics[{colorList[[j]], AbsoluteThickness[1.35], Line[{0, 0}, {1, 0}]}],
    PlotRange → {{0, 1}, {-0.5, 0.1}}, PlotRangePadding → None, ImageSize → 7],
    legendTags[[j]],
    Spacer[2]
  }, {j, 1, 4}]]];
  style = Table[{AbsoluteThickness[1.35], colorList[[j]], {j, 1, 4}}];

  vectorFunction[x_] :=
    If[kind == "major", JonesMajorAxisA[field[x, 0], Norm], SpinE[field[x, 0]]];

  Plot[
    {Norm[vectorFunction[x]],
      vectorFunction[x][[1]], vectorFunction[x][[2]], vectorFunction[x][[3]]
    , {x, 0, rm}
    , PlotStyle → style
    , PlotRange → {{0, rm}, {-1.075, 1.075}}
    , PlotRangePadding → None
    , Frame → True
    , Axes → False
    , PlotLegends → Placed[legend, Scaled[{0.5, 1.}]]
    , Epilog → {
      Inset[MaTeX["k\\,\\rho", FontSize → 10, Magnification → 0.85], {rm/2, -1.75}],
      Inset[Rotate[MaTeX[If[kind == "major", "\\mathrm{Major\\ axis}",
        "\\mathrm{Spin}"], FontSize → 8.5, 90°], {-1.9 rm / 7, 0}]
    ]
    , Method → {"AxesInFront" → True}
    , FrameStyle → Directive[GrayLevel[0.3], AbsoluteThickness[0.01]]
    , FrameTicksStyle → Directive[GrayLevel[0.3], AbsoluteThickness[0.01]]
    , FrameTicks → {{
      Join[{#, MaTeX[ToString[NumberForm[#, {1, 1}]]],
        FontSize → 10, Magnification → 0.6], {0.02, 0}} & /@
        Range[-1, 1, 0.5] (*, {#, "", {0.01, 0}} & /@ Range[-1, 1, 0.1] *)],
      Join[{#, "", {0.02, 0}} & /@ Range[-1, 1, 1] (*,
        {#, "", {0.01, 0}} & /@ Range[-1, 1, 0.2] *)]
    }, {
      Join[{#, MaTeX[ToString[#, FontSize → 10, Magnification → 0.6], {0.02, 0}} & /@
        Range[0, 7, 1], {#, "", {0.0005, 0}} & /@ Range[0, 7, 0.5]],
      Join[{#, "", {0.02, 0}} & /@ Range[0, 7, 1],
        {#, "", {0.0005, 0}} & /@ Range[0, 7, 0.5]]
    ]}
    , ImageSize → 108
    , PlotRangeClipping → False
    , ImagePadding → {{27, 2}, {20, 15}}
  ]
]

```

## PlotSkyrmionFigure

Here:

- **rm** is the maximal coordinate of the square plot range of the intensity plot
- **sm** is the radius of the circle in which the arrows plot is plotted, with the corresponding lift from the intensity plot

*In[ ]:=*

```
PlotSkyrmionFigure[fieldFunction_, kind_, rm_, sm_] :=
Block[{
  axes, intensityPlot, colorScale, circleLift, arrowsPlot, componentsPlot,
  intensityRange, lowerCircleHeight, circleTranslation = {-1.3 rm/2, -1.3 rm/2, 11},
  zoom = 5 × 1.1 / sm, scAmp = 4, axS = 1.15,
  shrinkWrapSafeLine
},
intensityRange = Last[
  PlotRange[Plot[Norm[fieldFunction[x, 0]], {x, 0, √2 rm}, PlotRange → Full]]];
lowerCircleHeight = scAmp Norm[fieldFunction[sm, 0]] / intensityRange[[2]];

axes = skyrmionPlotAxes[rm, axS];
circleLift =
  skyrmionPlotCircleLift[circleTranslation, lowerCircleHeight, sm, zoom];
arrowsPlot = Map[
  Translate[#, circleTranslation] &,
  PlotFieldArrows[fieldFunction, kind, sm, zoom]
];
intensityPlot =
  plotSkyrmionIntensity[fieldFunction, rm, scAmp, intensityRange[[2]]];
colorScale = skyrmionPlotColorScale[scAmp, rm,
  intensityRange[[1]] / intensityRange[[2]]];
componentsPlot = plotSkyrmionVectorComponents[kind, fieldFunction, rm, True];

shrinkWrapSafeLine =
  Graphics3D[{GrayLevel[0.99], Line[{rm, -rm + 1, -1.2}, {rm - 1, -rm, -1.2}]}];

Show[{
  intensityPlot, arrowsPlot, circleLift, axes, colorScale, shrinkWrapSafeLine
},
, BaseStyle → {FontFamily → "Times", FontSize → 28}
, Method → {"ShrinkWrap" → True}
, ViewPoint → {8, -8, 10}
, ViewVertical → {0, 0, 1}
, ImageSize → 247
, Epilog → {
  Inset[MaTeX["\\rm (a)"], {0.15, 0.78}],
  Inset[MaTeX["\\rm (b)"], {0.55, 0.78}],
  Inset[MaTeX["\\rm (c)"], {0.18, 0.42}],
  Inset[componentsPlot, {0.95, 0.85}, ImageScaled[{1, 1}]]
}
, Boxed → False
]
]
```



# C-skyrmions

## Bloch (Spiral) skyrmion

### Field definition

```
In[ ]:= BlochCSkyrmionQuasiCircular::usage =
  "BlochSkyrmionQuasiCircular[ξg,ξv,γ,{x,y,z}] gives the Bloch
  C-skyrmion field by superposing a gaussian-like CF beam with q=qg
  with a vortex CF beam of l=m=2 at q=qv, with a mixing angle γ.";

BlochCSkyrmionQuasiCircular[ξg_, ξv_, γ_, {x_, y_, z_}] = Simplify[
  Cos[γ] αQC[0, 0, ξg] × ComplexFocusHelicityE[0, 0, 1, {x, y, z}, ξg]
  - Sin[γ] αQC[2, 2, ξv] × ComplexFocusHelicityE[2, 2, -1, {x, y, z}, ξv]
];
```

### Initial search

The Manipulate below explores the Bloch spiral Skyrmion on the spin Poincarana sphere, with the goal of finding a set of parameters that does a reasonably good job at the required task, i.e., passing from right-circular polarization at the center to left-circular at the edge, via a complete flip of the polarization plane without losing circularity.

The results from the initial manual search are encoded as the defaults below.

```
In[ ]:= Manipulate[
  Row[{
    PoincaranaSpinX[
      Map[
        BlochCSkyrmionQuasiCircular[ξg, ξv, γ, {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ]
  }],
  {
    {ξg, 9.7}, {0.01, 10}
    , {{ξv, 4.2}, {0.01, 10}
    , {{γ, 1.3}, {0, π}
    , {xMax, {5, 10, 15}}
    , {{pts, 20}, {5, 10, 20}}
    , SaveDefinitions → True
  ]
```

### Optimization

After the manual search above we perform a machine optimization, by minimizing  $\int \left| \frac{\mathbf{E} \cdot \mathbf{E}}{|\mathbf{E}|^2} \right| dx$  over a radial cut (with the ideal optimum being zero, as  $\mathbf{E} \cdot \mathbf{E} = 0$  for CP light) while requiring that the field remain left-circular at the edge.

```

In[ ]:= OptimizeSkyrmionBlochCQC[initialParameters_, xMax_, pointNumber_] :=
  Block[{sampledField, costFunction, spinAtEnd, constraint, field},
    field[ξg_, ξv_, γ_, x_?NumericQ] =
      BlochCSkyrmionQuasiCircular[ξg, ξv, γ, {x, 0., 0.}];
    sampledField = Table[field[ξg, ξv, γ, x],
      {x, xMax / pointNumber, xMax, xMax / pointNumber}];
    costFunction = Total[Abs[(Dot[#, #] / Norm[#]^2)] & /@ sampledField];
    spinAtEnd = SpinE[sampledField[[-1]]];
    constraint =  $\frac{\text{spinAtEnd}[[3]]}{\text{Norm}[\text{spinAtEnd}]} + 1$ ;

    FindMinimum[{costFunction, constraint < 0.01, ξg > 0., ξv > 0.}, initialParameters]
  ]

```

```

In[ ]:= AbsoluteTiming[
  optimizedSkyrmionBlochCQC =
    OptimizeSkyrmionBlochCQC[{ξg, 9.1}, {ξv, 4.1}, {γ, 1.3}], 5., 5]
]

```

FindMinimum: The algorithm does not converge to the tolerance of  $10^{-6}$  in 500 iterations. The best estimated solution, with feasibility residual, KKT residual, or complementary residual of {0.0178415, 0.0631366, 0.00647327}, is returned.

```

Out[ ]:= {81.4054, {0.098796, {ξg → 9.10029, ξv → 4.10013, γ → 1.30221}}}

```

Retaking the precomputed outcome:

```

In[ ]:= optimizedSkyrmionBlochCQC = {0.09879600246648725`,
  {ξg → 9.100286211783686`, ξv → 4.1001348610849595`, γ → 1.3022133876510562`}};

```

Visualizing the machine-optimized solution:

```

In[ ]:= {
  PoincaranaSpinX[
    Map[
      BlochCSkyrmionQuasiCircular[
        Apply[Sequence, {ξg, ξv, γ} /. optimizedSkyrmionBlochCQC[[2]], {#, 0, 0.001}] &,
        Subdivide[0, 5, 20]
      ]
    ],
    Show[{
      PlotFieldArrows[BlochCSkyrmionQuasiCircular[Apply[Sequence,
        {ξg, ξv, γ} /. optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] &, "spin", 5, 1, 1.1],
      PlotFieldEllipses[BlochCSkyrmionQuasiCircular[Apply[Sequence,
        {ξg, ξv, γ} /. optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] &, 0.4, 5, Norm, 1.1]
    ]
  ], ImageSize → 650
]
}

```

## Figure Export

```

In[ ]:=DateString[]
blochCSkyrmionPlot = PlotSkyrmionFigure[
  BlochCSkyrmionQuasiCircular[
    Apply[Sequence, {ξg, ξv, γ} /. optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] &
    , "spin", 7., 5.];
DateString[]
Out[ ]:= Tue 1 Dec 2020 21:12:07

Out[ ]:= Tue 1 Dec 2020 21:13:10

In[ ]:= blochCSkyrmionPlot


In[ ]:= AbsoluteTiming[FileSize[Export["Bloch-C-Skyrmion.pdf",
  Show[blochCSkyrmionPlot, ImageSize → 247], ImageResolution → 1200]]]
Out[ ]:= {8.29034, 1.35564 MB }

```

## Skyrmion-ellipse figure

### Full version

```

In[ ]:= initialSkyrmionPlot = Show[{
  PlotFieldArrows[BlochCSkyrmionQuasiCircular[Apply[Sequence,
    {ξg, ξv, γ} /. optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] & "spin", 5, 1.1],
  PlotFieldEllipses[BlochCSkyrmionQuasiCircular[Apply[Sequence, {ξg, ξv, γ} /.
    optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] & 0.3, 5, Norm, 1.1, {0, 0, 0}
    (*, PlotStyle → Directive[Opacity[0.9]] *)
    , PlotStyle → Directive[EdgeForm[{Thickness[0.001], Black}],
      FaceForm[{

```

## Thumbnail version

```
initialSkyrmionPlotThumbnail = Show[{
  PlotFieldArrows[BlochCSkyrmionQuasiCircular[
    Apply[Sequence, {ξg, ξv, γ} /. optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] &,
    "spin", 3.5, 1.1, {0, 0, 0}, RadialPoints → 6],
  PlotFieldEllipses[BlochCSkyrmionQuasiCircular[Apply[Sequence, {ξg, ξv, γ} /.
    optimizedSkyrmionBlochCQC[[2]], {#1, #2, 0}] &, 0.3, 3.5, Norm, 1.1, {0, 0, 0}
    , PlotStyle → Directive[EdgeForm[{Thickness[0.001], Black}],
    FaceForm[{

```

## Néel (hedgehog) skyrmion

### Field definition

```
In[ ]:= NeelCSkyrmionQuasiCircular::usage =
  "NeelCSkyrmionQuasiCircular[γ,δ,{x,y,z}] gives the Néel
  C-skyrmion field by superposing two standing-wave monopolar
  fields of opposite polarization and different symmetry.";

NeelCSkyrmionQuasiCircular[γ_, δ_, {x_, y_, z_}] = Simplify[
  Cos[δ] (
    Cos[γ] ComplexFocusMagneticE[1, 0, UnitE[1], {x, y, z}, 0]
    - Sin[γ] ComplexFocusMagneticE[3, 2, UnitE[-1], {x, y, z}, 0]
  )
  - Sin[δ] ComplexFocusElectricE[1, 1, UnitE[0], {x, y, z}, 0]
];
```

## Initial search

```

In[ ]:= Manipulate[
  Row[{
    PoincaranaSpinX[
      Map[
        NeelCSkyrmionQuasiCircular[ $\gamma$ ,  $\delta$ , {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ],
    Show[{
      PlotFieldArrows[NeelCSkyrmionQuasiCircular[ $\gamma$ ,  $\delta$ , {#1, #2, 0}] &, "spin", 5, 1.1],
      PlotFieldEllipses[
        NeelCSkyrmionQuasiCircular[ $\gamma$ ,  $\delta$ , {#1, #2, 0}] &, 0.4, 5, Norm, 1.1]
    ]
    , ImageSize → 650
  ]
  ],
  {{ $\gamma$ , 1.3}, 0,  $\pi$ }
  , {{ $\delta$ , 0.13}, 0,  $\pi$ }
  , {xMax, {5, 10, 15}}
  , {{pts, 10}, {5, 10, 20}}
  , SaveDefinitions → True
]

```

## Optimization

Same as above

```

In[ ]:= OptimizeSkyrmionNeelCQC[initialParameters_, xMax_, pointNumber_] :=
  Block[{sampledField, costFunction, spinAtEnd, constraint, field},
    field[ $\gamma$ _,  $\delta$ _, x_?NumericQ] = NeelCSkyrmionQuasiCircular[ $\gamma$ ,  $\delta$ , {x, 0., 0.}];

    sampledField =
      Table[field[ $\gamma$ ,  $\delta$ , x], {x, xMax / pointNumber, xMax, xMax / pointNumber}];
    costFunction = Total[Abs[(Dot[#, #] / Norm[#]^2)] & /@ sampledField];
    spinAtEnd = SpinE[sampledField[[-1]]];
    constraint =  $\frac{\text{spinAtEnd}[[3]]}{\text{Norm}[\text{spinAtEnd}]} + 1$ ;

    FindMinimum[{costFunction, constraint < 0.01,  $\delta$  > 0}, initialParameters]
  ]

```

```

In[ ]:= AbsoluteTiming[
  optimizedSkyrmionNeelCQC = OptimizeSkyrmionNeelCQC[{{ $\gamma$ , 1.3}, { $\delta$ , .75}}, 4.5, 5]
]

```

**FindMinimum:** The algorithm does not converge to the tolerance of  $10^{-6}$  in 500 iterations. The best estimated solution, with feasibility residual, KKT residual, or complementary residual of  $\{3.62015 \times 10^{-10}, 2.86032, 4.78041 \times 10^{-21}\}$ , is returned.

```
Out[ ]:= {3.38823, {0.119554, { $\gamma \rightarrow 1.53038$ ,  $\delta \rightarrow 0.125239$ }}}
```

Retaking the precomputed outcome:

```
In[ ]:= optimizedSkyrmionNeelCQC =
  {0.11955364369018709`, { $\gamma \rightarrow 1.5303842173550015`$ ,  $\delta \rightarrow 0.12523875939626397`$ }};
```

Visualizing the machine-optimized solution:

```
{
  PoincaranaSpinX[
    Map[
      NeelCSkyrmionQuasiCircular[
        Apply[Sequence, { $\gamma$ ,  $\delta$ } /. optimizedSkyrmionNeelCQC[[2]], {#, 0, 0.001}] &,
        Subdivide[0, 5, 20]
      ]
    ],
    Show[ {
      PlotFieldArrows[NeelCSkyrmionQuasiCircular[Apply[Sequence,
        { $\gamma$ ,  $\delta$ } /. optimizedSkyrmionNeelCQC[[2]], {#1, #2, 0}] &, "spin", 5, 1, 1.1],
      PlotFieldEllipses[NeelCSkyrmionQuasiCircular[Apply[Sequence,
        { $\gamma$ ,  $\delta$ } /. optimizedSkyrmionNeelCQC[[2]], {#1, #2, 0}] &, 0.4, 5, Norm, 1.1]
    ]
  ], ImageSize -> 650
]
```

The intensity as a function of the radial coordinate:

```
In[ ]:= Plot[
  Norm[
    NeelCSkyrmionQuasiCircular[
      Apply[Sequence, { $\gamma$ ,  $\delta$ } /. optimizedSkyrmionNeelCQC[[2]], {x, 0, 0}]
    ]
  ], {x, 0, 7}
, Frame -> True
, GridLines -> All
, PlotStyle -> Darker[Blue, 0.3]
, PlotRange -> {0, Automatic}
, PlotRangePadding -> None
]

In[ ]:= Block[{Fnorm},
  Fnorm[x_] := Norm[NeelCSkyrmionQuasiCircular[
    Apply[Sequence, { $\gamma$ ,  $\delta$ } /. optimizedSkyrmionNeelCQC[[2]], {x, 0, 0}]]];
  {FindMaximum[Fnorm[x], {x, 4}][[2]],  $\frac{\text{Fnorm}[0]}{\text{FindMaximum[Fnorm[x], \{x, 4\}][[1]]}}$ 
]
```

... **FindMaximum**: The line search decreased the step size to within the tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient increase in the function. You may need more than MachinePrecision digits of working precision to meet these tolerances.

... **FindMaximum**: The line search decreased the step size to within the tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient increase in the function. You may need more than MachinePrecision digits of working precision to meet these tolerances.

```
Out[ ]:= {{x -> 3.83506}, 0.111896}
```

## Figure Export

```

In[ ]:=DateString[]
NeelCSkyrmionPlot = PlotSkyrmionFigure[
  NeelCSkyrmionQuasiCircular[
    Apply[Sequence, { $\gamma$ ,  $\delta$ } /. optimizedSkyrmionNeelCQC[[2]], {#1, #2, 0}] &
    , "spin", 7., 5.];
DateString[]
Out[ ]:= Tue 1 Dec 2020 21:13:22

Out[ ]:= Tue 1 Dec 2020 21:13:43

NeelCSkyrmionPlot

In[ ]:= AbsoluteTiming[FileSize[Export["Neel-C-Skyrmion.pdf",
  Show[ NeelCSkyrmionPlot, ImageSize -> 247], ImageResolution -> 1200]]]
Out[ ]:= {7.61924, 1.64203 MB }

```

# L-skyrmions

## Forward Bloch

### Field definition

```

In[ ]:= BlochLSkyrmion::usage =
  "BlochLSkyrmion[ $\xi r, \xi a, \gamma, \{x, y, z\}$ ] gives the Bloch L-skyrmion field.
  BlochLSkyrmion[ $\gamma, \{x, y, z\}$ ] gives the Bloch L-skyrmion field for  $\xi r = \xi a = 0$ .";

BlochLSkyrmion[ $\xi r_$ ,  $\xi a_$ ,  $\gamma_$ , { $x_$ ,  $y_$ ,  $z_$ }] = Simplify[
  -Cos[ $\gamma$ ]  $\alpha$ TEM[0, 0,  $\xi r$ ]  $\times$  ComplexFocusElectricE[0, 0, UnitE[0], {x, y, z},  $\xi r$ ]
  +  $\mathfrak{i}$  Sin[ $\gamma$ ]  $\alpha$ TEM[0, 0,  $\xi a$ ]  $\times$  ComplexFocusMagneticE[0, 0, UnitE[0], {x, y, z},  $\xi a$ ]
];
BlochLSkyrmion[ $\gamma_$ , { $x_$ ,  $y_$ ,  $z_$ }] = Simplify[
  -Cos[ $\gamma$ ]  $\alpha$ TEM[0, 0, 0]  $\times$  ComplexFocusElectricE[0, 0, UnitE[0], {x, y, z}, 0]
  +  $\mathfrak{i}$  Sin[ $\gamma$ ]  $\alpha$ TEM[0, 0, 0]  $\times$  ComplexFocusMagneticE[0, 0, UnitE[0], {x, y, z}, 0]
];

```

## Initial search

```
In[ ]:= Manipulate[
  Row[{
    PoincaranaMajorX[
      Map[
        BlochLSkyrmion[ $\xi r$ ,  $\xi a$ ,  $\gamma$ , {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ],
    PoincaranaSpinX[
      Map[
        BlochLSkyrmion[ $\xi r$ ,  $\xi a$ ,  $\gamma$ , {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ]
  ]],
  {{ $\xi r$ , 0.1}, 0.01, 10},
  {{ $\xi a$ , 1.92}, 0.01, 10},
  {{ $\gamma$ , 1.02}, 0,  $\pi$ },
  {xMax, {5, 10, 15}},
  {{pts, 10}, {5, 10, 20}},
  SaveDefinitions -> True
]
```

## Optimization

Same as above

```
In[ ]:= OptimizeSkyrmionBlochL[initialParameters_, xMax_, pointNumber_] :=
  Block[{sampledField, costFunction, majorAxisAtEnd, constraint, field},
    field[ $\xi r$ _,  $\xi a$ _,  $\gamma$ _, x_?NumericQ] = BlochLSkyrmion[ $\xi r$ ,  $\xi a$ ,  $\gamma$ , {x, 0., 0.}];

    sampledField =
      Table[field[ $\xi r$ ,  $\xi a$ ,  $\gamma$ , x], {x, xMax/pointNumber, xMax, xMax/pointNumber}];
    costFunction = Total[Norm[JonesMajorAxisA[#, Norm]] & /@ sampledField];
    majorAxisAtEnd = JonesMajorAxisA[sampledField[[-1]], Norm];
    constraint =  $\frac{\text{majorAxisAtEnd}[[3]]}{\text{Norm}[\text{majorAxisAtEnd}]} + 1$ ;

    FindMaximum[{costFunction, constraint < 0.01,
      10 >  $\xi r$  > 0.1, 10 >  $\xi a$  > 0.1, 0 <  $\gamma$  < 0.65  $\pi$  / 2}, initialParameters]
  ]
```

```
In[ ]:= AbsoluteTiming[
  optimizedSkyrmionBlochL =
    OptimizeSkyrmionBlochL[{{ $\xi r$ , 2.12}, { $\xi a$ , 2.16}, { $\gamma$ , .88}}, 5., 5]
]
```

```
Out[ ]:= {0.573343, {4.99927, { $\xi r \rightarrow 0.100007$ ,  $\xi a \rightarrow 1.92149$ ,  $\gamma \rightarrow 1.02096$ }}}
```

Retaking the precomputed outcome:



```
In[ ]:= optimizedSkyrmionBlochL = {4.999269850042667`,
  {ξr → 0.10000691291778394`, ξa → 1.9214853124291764`, γ → 1.0209638616941368`}};
```

Visualizing the machine-optimized solution:

```
In[ ]:= {
  PoincaranaMajorX[
    Map[
      BlochLSkyrmion[
        Apply[Sequence, {ξr, ξa, γ} /. optimizedSkyrmionBlochL[[2]], {#, 0, 0.001}] &,
        Subdivide[0, 5, 20]
      ]
    ],
  PoincaranaSpinX[
    Map[
      BlochLSkyrmion[
        Apply[Sequence, {ξr, ξa, γ} /. optimizedSkyrmionBlochL[[2]], {#, 0, 0.001}] &,
        Subdivide[0, 5, 20]
      ]
    ],
  Show[{
    PlotFieldArrows[
      BlochLSkyrmion[Apply[Sequence, {ξr, ξa, γ} /. optimizedSkyrmionBlochL[[2]],
        {#1, #2, 0}] &, "major", 5, 1, 1.1],
    PlotFieldEllipses[BlochLSkyrmion[Apply[Sequence,
      {ξr, ξa, γ} /. optimizedSkyrmionBlochL[[2]], {#1, #2, 0}] &, 0.4, 5, Norm, 1.1]
    ]
  ], ImageSize → 500
  ]
}
```

## Figure Export

```
In[ ]:= DateString[]
BlochLSkyrmionplot = PlotSkyrmionFigure[
  BlochLSkyrmion[
    Apply[Sequence, {ξr, ξa, γ} /. optimizedSkyrmionBlochL[[2]], {#1, #2, 0}] &
    , "major", 7., 5.];
DateString[]
```

```
Out[ ]:= Tue 1 Dec 2020 21:13:52
```

```
Out[ ]:= Tue 1 Dec 2020 21:14:10
```

**BlochLSkyrmionplot**

```
In[ ]:= AbsoluteTiming[FileSize[Export["Bloch-L-Skyrmion.pdf",
  Show[BlochLSkyrmionplot, ImageSize → 247], ImageResolution → 1200]]]
```

```
Out[ ]:= {8.27962, 1.28329 MB }
```

# Forward Néel

## Field definition

```
In[ ]:= NeelLSkymion::usage =
  "NeelLSkymion[ξ1,ξ2,γ,{x,y,z}] gives the Néel L-skyrmion field.
  NeelLSkymion[γ,{x,y,z}] gives the Néel L-skyrmion field for qr1=qr2=0.";

NeelLSkymion[ξ1_, ξ2_, γ_, {x_, y_, z_}] = -Simplify[
  Cos[γ] αTEM[0, 0, ξ1] × ComplexFocusElectric[0, 0, UnitE[0], {x, y, z}, ξ1]
  +  $\frac{i}{2}$  Sin[γ] αTEM[1, 0, ξ2] × ComplexFocusElectric[1, 0, UnitE[0], {x, y, z}, ξ2]
];

NeelLSkymion[γ_, {x_, y_, z_}] = -Simplify[
  Cos[γ] αTEM[0, 0, 0] × ComplexFocusElectric[0, 0, UnitE[0], {x, y, z}, 0]
  +  $\frac{i}{2}$  Sin[γ] αTEM[1, 0, 0] × ComplexFocusElectric[1, 0, UnitE[0], {x, y, z}, 0]
];
```

## Initial search

```
In[ ]:= Manipulate[
  Row[{
    PoincaranaMajorX[
      Map[
        NeelLSkymion[ξ1, ξ2, γ, {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ],
    PoincaranaSpinX[
      Map[
        NeelLSkymion[ξ1, ξ2, γ, {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ]
  ]],
  {{ξ1, 1.32}, 0.01, 10}
, {{ξ2, 0.8}, 0.01, 10}
, {{γ, 0.75}, 0, π}
, {xMax, {5, 10, 15}}
, {{pts, 10}, {5, 10, 20}}
, SaveDefinitions → True
]
```

```

In[ ]:= Manipulate[
  Row[{
    PoincaranaMajorX[
      Map[
        NeellSkyrmion[ξ1, ξ2, γ, {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ],
    PoincaranaSpinX[
      Map[
        NeellSkyrmion[ξ1, ξ2, γ, {#, 0, 0.001}] &,
        Subdivide[0, xMax, pts]
      ]
    ],
    Show[{
      PlotFieldArrows[NeellSkyrmion[ξ1, ξ2, γ, {#1, #2, 0}] &, "major", 5, 1.1],
      PlotFieldEllipses[NeellSkyrmion[ξ1, ξ2, γ, {#1, #2, 0}] &, 0.4, 5, Norm, 1.1]
    }
    , ImageSize → 500
  ]
],
, {{ξ1, 1.32}, 0.01, 10}
, {{ξ2, 0.8}, 0.01, 10}
, {{γ, 0.75}, 0, π}
, {xMax, {5, 10, 15}}
, {{pts, 10}, {5, 10, 20}}
, SaveDefinitions → True
]

```

## Optimization

Same as above

```

In[ ]:= OptimizeSkyrmionNeell[initialParameters_, xMax_, pointNumber_] :=
  Block[{sampledField, costFunction, majorAxisAtEnd, constraint, field},
    field[ξ1_, ξ2_, γ_, x_?NumericQ] := NeellSkyrmion[ξ1, ξ2, γ, {x, 0., 0.}];

    sampledField =
      Table[field[ξ1, ξ2, γ, x], {x, xMax/pointNumber, xMax, xMax/pointNumber}];
    costFunction = Total[Norm[JonesMajorAxisA[#, Norm]] & /@ sampledField];
    majorAxisAtEnd = JonesMajorAxisA[sampledField[[-1]], Norm];
    constraint =  $\frac{\text{majorAxisAtEnd}[[3]]}{\text{Norm}[\text{majorAxisAtEnd}]} + 1$ ;

    FindMaximum[{costFunction, constraint < 0.01,
      10 > ξ1 > 0.1, 10 > ξ2 > 0.1, 0.2 < γ < π/2}, initialParameters]
  ]

```

```

In[ ]:= AbsoluteTiming[
  optimizedSkyrmionNeell =
    OptimizeSkyrmionNeell[{{ξ1, 1.32}, {ξ2, .84}, {γ, .6}}, 5.5, 5]
]

Out[ ]:= {2.71744, {4.99719, {ξ1 → 0.100003, ξ2 → 0.100004, γ → 0.660689}}}

```

Retaking the precomputed outcome:

```
In[ ]:= optimizedSkyrmionNeelL = {4.997189107698044`,  
    {ξ1 → 0.10000267146516169`, ξ2 → 0.10000410457159642`, γ → 0.6606890622945024`}};
```

Visualizing the machine-optimized solution:

```
In[ ]:= {  
    PoincaranaMajorX[  
        Map[  
            NeelLSkyrmion[  
                Apply[Sequence, {ξ1, ξ2, γ} /. optimizedSkyrmionNeelL[[2]], {#, 0, 0.001}] &,  
                Subdivide[0, 5, 20]  
            ]  
        ],  
    PoincaranaSpinX[  
        Map[  
            NeelLSkyrmion[  
                Apply[Sequence, {ξ1, ξ2, γ} /. optimizedSkyrmionNeelL[[2]], {#, 0, 0.001}] &,  
                Subdivide[0, 5, 20]  
            ]  
        ],  
    Show[{  
        PlotFieldArrows[  
            NeelLSkyrmion[Apply[Sequence, {ξ1, ξ2, γ} /. optimizedSkyrmionNeelL[[2]],  
                {#1, #2, 0}] &, "major", 5, 1, 1.1],  
        PlotFieldEllipses[NeelLSkyrmion[Apply[Sequence,  
            {ξ1, ξ2, γ} /. optimizedSkyrmionNeelL[[2]], {#1, #2, 0}] &, 0.4, 5, Norm, 1.1]  
        ]  
    }, ImageSize → 500  
    ]  
    }
```

## Figure Export

```
In[ ]:= DateString[]  
NeelLSkyrmionplot = PlotSkyrmionFigure[  
    NeelLSkyrmion[  
        Apply[Sequence, {ξ1, ξ2, γ} /. optimizedSkyrmionNeelL[[2]], {#1, #2, 0}] &  
        , "major", 7., 5.5];  
DateString[]
```

Out[ ]:= Tue 1 Dec 2020 21:14:22

Out[ ]:= Tue 1 Dec 2020 21:14:50

**NeelLSkyrmionplot**

```
In[ ]:= AbsoluteTiming[FileSize[Export["Neel-L-Skyrmion.pdf",  
    Show[NeelLSkyrmionplot, ImageSize → 247], ImageResolution → 1200]]]
```

Out[ ]:= {9.3186, 1.57164 MB }

## Perfect L-skyrmions

### Figure code

#### skyrmionPlotAxesSmall

```

In[ ]:= skyrmionPlotAxesSmall[rm_, axS_] := Graphics3D[{
  Line[{{-rm, -rm, 0}, {rm, -rm, 0}}],
  Line[{{rm, -rm, 0}, {rm, rm, 0}}],
  Table[{
    Inset[MaTeX[ToString[x], FontSize → 9], {x, -1.075 axS rm, 0}],
    Inset[MaTeX[ToString[x], FontSize → 9], {axS rm, x, 0}]
  ], {x, -3, -rm, -3}],
  Table[{
    Line[{{-x, -rm, 0}, {-x, -.95 rm, 0}}],
    Line[{{x, -rm, 0}, {x, -.95 rm, 0}}],
    Line[{{rm, -x, 0}, {.95 rm, -x, 0}}],
    Line[{{rm, x, 0}, {.95 rm, x, 0}}],
    Inset[MaTeX[ToString[x], FontSize → 9], {x, -axS rm, 0}],
    Inset[MaTeX[ToString[x], FontSize → 9], {axS rm, x, 0}]
  ], {x, 0, rm, 3}],
  Inset[MaTeX["k\\,x", FontSize → 9], {0, -1.55 rm, 0}],
  Inset[MaTeX["k\\,y", FontSize → 9], {1.55 rm, 0, 0}]
}]

```

#### skyrmionPlotColorScaleSmall

```

In[ ]:= skyrmionPlotColorScaleSmall[scAmp_, rm_, lowerLimit_] := Block[{offset = 1.05},
  Graphics3D[{
    Inset[MaTeX["|\\mathbf{E}|", FontSize → 10, Magnification → 0.75],
      {offset rm, offset rm, 5 scAmp / 4}, Scaled[{0.4, 0.5}]],
    Inset[MaTeX[NumberForm[lowerLimit, {1, 2}], FontSize → 10, Magnification → 0.6],
      {offset rm, offset rm, 3 scAmp / 4}],
    Inset[MaTeX["1.00", FontSize → 10, Magnification → 0.6],
      {offset rm, offset rm, 7 scAmp / 4}],
    Lighting → {"Ambient", White},
    ReplaceAll[
      First[
        DensityPlot[y, {x, -.5 rm / 7, .5 rm / 7},
          {y, 0, scAmp}, ColorFunction → Viridis, Frame → False]
      ],
      {x_?AtomQ, y_?AtomQ} =>
        {6 rm / 7, 6 rm / 7, 3 scAmp / 4} + {x Cos[ $\pi$  / 4], x Sin[ $\pi$  / 4], y}
      (*lifting from 2D to 3D*)
    ]
  ]
]

```

#### PlotSkyrmionSimplifiedFigure

Same as PlotSkyrmionFigure, but with edited values of zoom and circleTranslation, a changed epilog, and small-adjusted axes and color scale.

In[#:]=

```
PlotSkyrmionSimplifiedFigure[fieldFunction_, kind_, rm_, sm_, label_] :=
Block[{
  axes, intensityPlot, colorScale, circleLift, arrowsPlot,
  intensityRange, lowerCircleHeight,
  circleTranslation = {0, 0, 11}, zoom = 5 × 1.6 / sm, scAmp = 4, axS = 1.2,
  shrinkWrapSafeLine
},
intensityRange = Last[
  PlotRange[Plot[Norm[fieldFunction[x, 0]], {x, 0,  $\sqrt{2}$  rm}, PlotRange → Full]]];
lowerCircleHeight = scAmp Norm[fieldFunction[sm, 0]] / intensityRange[[2]];

axes = skyrmionPlotAxesSmall[rm, axS];
circleLift =
  skyrmionPlotCircleLift[circleTranslation, lowerCircleHeight, sm, zoom,  $\pi / 4$ ];
arrowsPlot = Map[
  Translate[#, circleTranslation] &,
  PlotFieldArrows[fieldFunction, kind, sm, zoom]
];
intensityPlot =
  plotSkyrmionIntensity[fieldFunction, rm, scAmp, intensityRange[[2]]];
colorScale = skyrmionPlotColorScaleSmall[scAmp, rm,
  intensityRange[[1]] / intensityRange[[2]]];

shrinkWrapSafeLine =
  Graphics3D[{GrayLevel[0.99], Line[{rm, -rm + 1, -1.2}, {rm - 1, -rm, -1.2}]}];

Show[{
  intensityPlot, arrowsPlot, circleLift, axes, colorScale, shrinkWrapSafeLine
},
, BaseStyle → {FontFamily → "Times", FontSize → 28}
, Method → {"ShrinkWrap" → True}
, ViewPoint → {8, -8, 10}
, ViewVertical → {0, 0, 1}
, ImageSize → 0.5 × 246
(*, Epilog → {Inset[Style[label, 65], {0.18, 0.78}]} *)
, Epilog → {
  Inset[MaTeX["\\rm " <> label], {0.15, 0.78}]
}

, Boxed → False
]
]
```

## Figure export

```

In[ ]:= DateString[]
perfectL = Grid[{
  PlotSkyrmionSimplifiedFigure[
    BlochLSkyrmion[ $\pi / 4$ , {#1, #2, 0}] &
    , "major", 7., 4.5, "(a)"],
  PlotSkyrmionSimplifiedFigure[
    NeelLSkyrmion[ $\pi / 4$ , {#1, #2, 0}] &
    , "major", 7., 5.8, "(b)"]
}], Spacings -> {0, 0}];
DateString[]
Out[ ]:= Tue 1 Dec 2020 21:15:00

Out[ ]:= Tue 1 Dec 2020 21:15:12

perfectL

In[ ]:= AbsoluteTiming[
  FileSize[Export["Perfect-L-Skyrmion.pdf", perfectL, ImageResolution -> 1200]]]
Out[ ]:= {14.8313, 0.92539 MB }

```