```
(* Koordinatensysteme, WQ, FS 2013 *)
```

CoordinateTransformData[*t*, *property*]
gives the value of the specified property for the coordinate transformation *t*.

CoordinateTransformData[*t*, *property*, $\{x_1, x_2, \ldots, x_n\}$]
gives the value of the property evaluated at the point $\{x_1, x_2, \ldots, x_n\}$.

```
(* Beispiel Polarkoordinaten 2-dimensional *)

CoordinateTransformData["Cartesian" → "Polar", "Mapping", {x, y}]

CoordinateTransformData["Cartesian" → "Polar", "MappingJacobian", {x, y}]
CoordinateTransformData["Cartesian" → "Polar",
 "MappingJacobianDeterminant", {x, y}]
```

$\left\{\sqrt{x^2+y^2}, \text{ArcTan}[x, y]\right\}$

$\left\{\left\{\frac{x}{\sqrt{x^2+y^2}}, \frac{y}{\sqrt{x^2+y^2}}\right\}, \left\{-\frac{y}{x^2+y^2}, \frac{x}{x^2+y^2}\right\}\right\}$

$\frac{1}{\sqrt{x^2+y^2}}$

(* In der Ausgabe ist eine 'Kurzform' verwendet:
ArcTan[$x$,$y$] gives the arc tangent of $y/x$,
taking into account which quadrant the point($x$,$y$) is in. *)

```
(* The mapping function can be requested from
 CoordinateTransformData and stored for later use. *)
```

In[1]:= map = CoordinateTransformData["Cartesian" → "Polar", "Mapping"]

Out[1]= $\left\{\sqrt{\#1[\![1]\!]^2+\#1[\![2]\!]^2}, \text{ArcTan}[\#1[\![1]\!], \#1[\![2]\!]]\right\}$ &

map[{1, 1}]

Out[3]= $\left\{\sqrt{2}, \frac{\pi}{4}\right\}$

map /@ {{0, 1}, {1, 0}, {1 / 2, 1 / 2}, {3, 1 / 3}}

Out[4]= $\left\{\left\{1, \frac{\pi}{2}\right\}, \{1, 0\}, \left\{\frac{1}{\sqrt{2}}, \frac{\pi}{4}\right\}, \left\{\frac{\sqrt{82}}{3}, \text{ArcTan}\left[\frac{1}{9}\right]\right\}\right\}$

```
CoordinateTransformData["Polar" → "Cartesian", "Mapping", {r, θ}]

mp = CoordinateTransformData["Polar" → "Cartesian", "MappingJacobian", {r, θ}]

imp = CoordinateTransformData[
  {"Polar" → "Cartesian", "Euclidean", 2}, "InverseMappingJacobian", {r, θ}]

CoordinateTransformData["Polar" → "Cartesian",
 "MappingJacobianDeterminant", {r, θ}]
```

$\{r\,\mathrm{Cos}[\theta],\ r\,\mathrm{Sin}[\theta]\}$

$\{\{\mathrm{Cos}[\theta],\ -r\,\mathrm{Sin}[\theta]\},\ \{\mathrm{Sin}[\theta],\ r\,\mathrm{Cos}[\theta]\}\}$

$$\left\{\{\mathrm{Cos}[\theta],\ \mathrm{Sin}[\theta]\},\ \left\{-\frac{\mathrm{Sin}[\theta]}{r},\ \frac{\mathrm{Cos}[\theta]}{r}\right\}\right\}$$

r

```
mp . imp // Simplify
```

{{1, 0}, {0, 1}}

```
(*  Anwendung der Transformation von {x,y} zu {r,θ} *)
pt = CoordinateTransform["Cartesian" → "Polar", {x, y}]
```

$$\left\{\sqrt{x^2+y^2},\ \mathrm{ArcTan}[x,\ y]\right\}$$

```
(* Und zurueck *)
CoordinateTransform["Polar" → "Cartesian", pt]
```

{x, y}

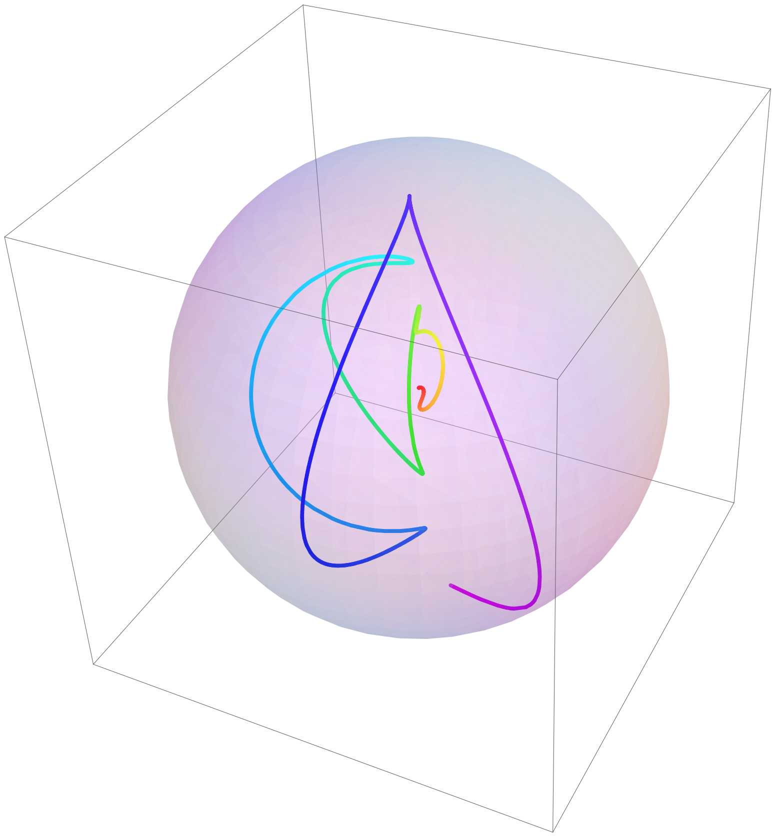
**Example : Convert a curve in non - Cartesian coordinates to a corresponding Cartesian expression for purposes of visualization :**

$$\mathrm{cSpherical}[t\_] := \left\{t,\ \mathrm{Pi}\left(\frac{1+\mathrm{Sin}[16\ t+4\ t\verb|^|2]}{2}\right),\ 1+2\ t+3\ t\verb|^|2\right\}$$

```
cCartesian[t_] = CoordinateTransform["Spherical" → "Cartesian", cSpherical[t]]
```

$$\left\{t\,\mathrm{Cos}\left[1+2\,t+3\,t^2\right]\mathrm{Sin}\left[\frac{1}{2}\pi\left(1+\mathrm{Sin}\left[16\,t+4\,t^2\right]\right)\right],\ t\,\mathrm{Sin}\left[1+2\,t+3\,t^2\right]\mathrm{Sin}\left[\frac{1}{2}\pi\left(1+\mathrm{Sin}\left[16\,t+4\,t^2\right]\right)\right],\ t\,\mathrm{Cos}\left[\frac{1}{2}\pi\left(1+\mathrm{Sin}\left[16\,t+4\,t^2\right]\right)\right]\right\}$$

```
Show[Graphics3D[{Opacity[.25], Sphere[]}], ParametricPlot3D[cCartesian[t],
  {t, 0, 1}, PlotStyle → Thick, ColorFunction → (Hue[.8 #4] &)]]
```



This curve is approximately 11.2 radii in length :

```
NIntegrate[Sqrt[cCartesian'[t].cCartesian'[t]], {t, 0, 1}]
```

11.2049

```
(* in den Sphericals ist es schwieriger, siehe unten: *)
metr = CoordinateChartData[{"Spherical", 3}, "Metric"][{r, θ, φ}] /.
  {r → t, θ -> Pi ((1 + Sin[16 t + 4 t^2])/2)}
```

$\{\{1, 0, 0\}, \{0, t^2, 0\}, \{0, 0, t^2 \mathrm{Sin}[\frac{1}{2}\pi (1 + \mathrm{Sin}[16 t + 4 t^2])]^2\}\}$

```
NIntegrate[Sqrt[cSpherical'[t].metr.cSpherical'[t]], {t, 0, 1}]
```

11.2049

```
(* Koordinaten  --  Karten  *)

CoordinateChartData["Properties"]
```

{AlternateCoordinateNames, CoordinateRangeAssumptions,
Dimension, InverseMetric, Metric, ParameterRangeAssumptions,
ScaleFactors, StandardCoordinateNames, StandardName, VolumeFactor}

```
CoordinateChartData["Polar", "ScaleFactors", {r, θ}]
```

{1, r}

```
CoordinateChartData[{"Polar", 2}, "ScaleFactors", {r, θ}]
```

{1, r}

```
CoordinateChartData["Polar", "Metric", {r, θ}]
CoordinateChartData["Polar", "InverseMetric", {r, θ}]
```

$\{\{1, 0\}, \{0, r^2\}\}$

$\{\{1, 0\}, \{0, \frac{1}{r^2}\}\}$

```
vf = CoordinateChartData["Polar", "VolumeFactor", {r, θ}]
```

r

$\int_0^R \int_0^{2\pi} vf \, d\theta \, dr$

$\pi R^2$

```
CoordinateChartData["Polar", "StandardCoordinateNames", {r, θ}]
```

{r, θ}

```
% // InputForm
```

{"r", "θ"}

```
ToExpression /@ % // InputForm
```

{r, θ}

```
test = CoordinateChartData[{"Polar", 2}, "CoordinateRangeAssumptions"]
```

$\#1[[1]] > 0 \;\&\&\; -\pi < \#1[[2]] \leq \pi \;\&$

```
test[{3, π / 2}] (* gueltiger Punkt *)
```

True

```
test[{-3, π / 2}] (* ungueltiger Punkt *)
```

False

```
(* Verwendung der Metrik, Beispiel Zylinder-Koordinaten,
die auf einer Helix verwendet werden koennen.
  Give the components of the (covariant) metric as a matrix: *)
```

```
CoordinateChartData[{"Cylindrical", 3}, "Metric"][{r, θ, z}]
```

$\{\{1, 0, 0\}, \{0, r^2, 0\}, \{0, 0, 1\}\}$

```
(* By particularizing to a curve,
the metric can be used to compute the differential arc length *)
```

```
metric[t_] = CoordinateChartData[{"Cylindrical", 3}, "Metric"][{r[t], θ[t], z[t]}]
```

$\{\{1, 0, 0\}, \{0, r[t]^2, 0\}, \{0, 0, 1\}\}$

```
helix[t_] = {a, b t, c t}

helix'[t] . metric[t].helix'[t] /. r[t] → a
```

{a, b t, c t}

$a^2 b^2 + c^2$

```
(* Allgemeine Anwendung:
  A function to compute the differential arc
   length of a curve in a particular coordinate system: *)
ds[curve_List, t_, chart_] := Module[{metric, tangent},
  metric = CoordinateChartData[chart, "Metric", curve];
  tangent = D[curve, t];
  Sqrt[tangent . metric . tangent] ⅆt
 ]
(* The differential arc length of a general curve in polar coordinates: *)

ds[{r[t], θ[t]}, t, "Polar"]
```

$dt \sqrt{r'[t]^2 + r[t]^2 \theta'[t]^2}$

```
(* The differential arc length of a
 helix expressed in cylindrical coordinates: *)
ds[{R, u, u}, u, {"Cylindrical", 3}]
```

$\sqrt{1 + R^2}\ du$

```
(* This can now be integrated: *)
```

$s[t\_] := \int_0^t \sqrt{1 + R^2}\ du$

```
s[t]  (* Die wahre Bogenlaenge ist somit *)
```

$\sqrt{1 + R^2}\ t$

```
(* Analoge 3-dimensionale Transformation *)

CoordinateTransformData["Cartesian" → "Spherical", "Mapping", {x, y, z}]
```

$\left\{\sqrt{x^2 + y^2 + z^2}, \text{ArcTan}\left[z, \sqrt{x^2 + y^2}\right], \text{ArcTan}[x, y]\right\}$

```
CoordinateTransformData["Spherical" → "Cartesian", "Mapping", {r, θ, φ}]
```

{r Cos[φ] Sin[θ], r Sin[θ] Sin[φ], r Cos[θ]}

```
CoordinateTransformData["Spherical" → "Cartesian", "Mapping", {r, θ, φ}]
CoordinateTransformData["Cartesian" → "Polar", "MappingJacobian", {x, y}]
CoordinateTransformData["Spherical" → "Cartesian", "Mapping", {r, θ, φ}]
```

{r Cos[φ] Sin[θ], r Sin[θ] Sin[φ], r Cos[θ]}

$$\left\{\left\{\frac{x}{\sqrt{x^2+y^2}}, \frac{y}{\sqrt{x^2+y^2}}\right\}, \left\{-\frac{y}{x^2+y^2}, \frac{x}{x^2+y^2}\right\}\right\}$$

{r Cos[φ] Sin[θ], r Sin[θ] Sin[φ], r Cos[θ]}

```
vf = CoordinateChartData[{"Spherical", 3}, "VolumeFactor", {r, θ, φ}]
```

$r^2$ Sin[θ]

$$\int_0^R \int_0^\pi \int_0^{2\pi} vf \, d\varphi \, d\theta \, dr$$

$$\frac{4\pi R^3}{3}$$

```
(* Allgemeine Uebersicht zu Koordinatensystemen in der Ebene *)
CoordinateChartData[{ All, 2}]
```

{{{Bipolar, {ạ}}, Euclidean, 2}, {Cartesian, Euclidean, 2},
{{Confocal, {ạ, ḅ}}, Euclidean, 2}, {{Elliptic, {ạ}}, Euclidean, 2},
{Polar, Euclidean, 2}, {PlanarParabolic, Euclidean, 2}}

```
(* Allgemeine Uebersicht zu Koordinatensystemen im 3-
 dimensionalen Euklidischen Raum *)
CoordinateChartData[{ All, 3}]
```

{{{BipolarCylindrical, {ạ}}, Euclidean, 3},
{{Bispherical, {ạ}}, Euclidean, 3}, {Cartesian, Euclidean, 3},
{CircularParabolic, Euclidean, 3}, {{Confocal, {ạ, ḅ, c̣}}, Euclidean, 3},
{{ConfocalParaboloidal, {ạ, ḅ}}, Euclidean, 3}, {{Conical, {ḅ, c̣}}, Euclidean, 3},
{Cylindrical, Euclidean, 3}, {{EllipticCylindrical, {ạ}}, Euclidean, 3},
{Hyperspherical, Euclidean, 3}, {{OblateSpheroidal, {ạ}}, Euclidean, 3},
{ParabolicCylindrical, Euclidean, 3}, {{ProlateSpheroidal, {ạ}}, Euclidean, 3},
{Spherical, Euclidean, 3}, {{Toroidal, {ạ}}, Euclidean, 3}}

```
(* Ein gegebenes Vektorfeld in andere Koordinaten umrechnen *)
```

**TransformedField[**$t$**,** $f$**, {**$x_1$**,** $x_2$**, …,** $x_n$**} -> {**$y_1$**,** $y_2$**, …,** $y_n$**}]**
**(***
**uses the coordinate transformation** $t$ **to transform the scalar,**
**vector, or tensor field** $f$ **from coordinates** $x_i$ **to** $y_i$**. *)**

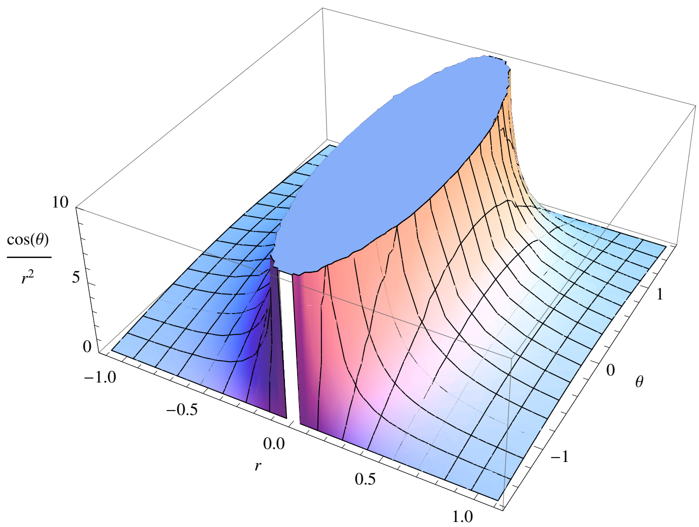
```
(* Beispiel: An electric dipole of dipole moment
  p located at the origin and aligned with the z axis has
  the following electric potential in spherical coordinates:
```

In[23]:= `Vs = (p Cos[θ]) / r^2`

Out[23]= $\frac{p\,\mathrm{Cos}[\theta]}{r^2}$

In[28]:= `Plot3D[Cos[θ]/r^2, {r, -1.1, 1.1}, {θ, -Pi / 2, Pi / 2}, PlotRange → {0, 10}, AxesLabel → {r, θ, Vs /. p → 1}]`

Out[28]=




Compute the corresponding expression in Cartesian coordinates :

In[24]:= `Vc = TransformedField[ "Spherical" → "Cartesian", Vs, {r, θ, φ} → {x, y, z}]`

Out[24]= $\frac{p\,z}{(x^2+y^2+z^2)^{3/2}}$

Derive the dipole electric field in spherical coordinates :
 !! Der Gradient kann einfach als Grad angesetzt werden,
wenn man noch das verwendete Koordinatensystem angibt !!

In[25]:= `Es = -Grad[Vs, {r, θ, φ}, "Spherical"]`

Out[25]= $\left\{\frac{2\,p\,\mathrm{Cos}[\theta]}{r^3}, \frac{p\,\mathrm{Sin}[\theta]}{r^3}, 0\right\}$

Transform this expression to Cartesian coordinates :

In[26]:= `Ec = TransformedField["Spherical" → "Cartesian", Es, {r, θ, φ} → {x, y, z}]`

Out[26]= $\left\{\frac{3\,p\,x\,z}{(x^2+y^2+z^2)^{5/2}}, \frac{3\,p\,y\,z}{(x^2+y^2+z^2)^{5/2}}, -\frac{p\,(x^2+y^2)}{(x^2+y^2+z^2)^{5/2}}+\frac{2\,p\,z^2}{(x^2+y^2+z^2)^{5/2}}\right\}$

The same expression is obtained by differentiating the Cartesian potential function :

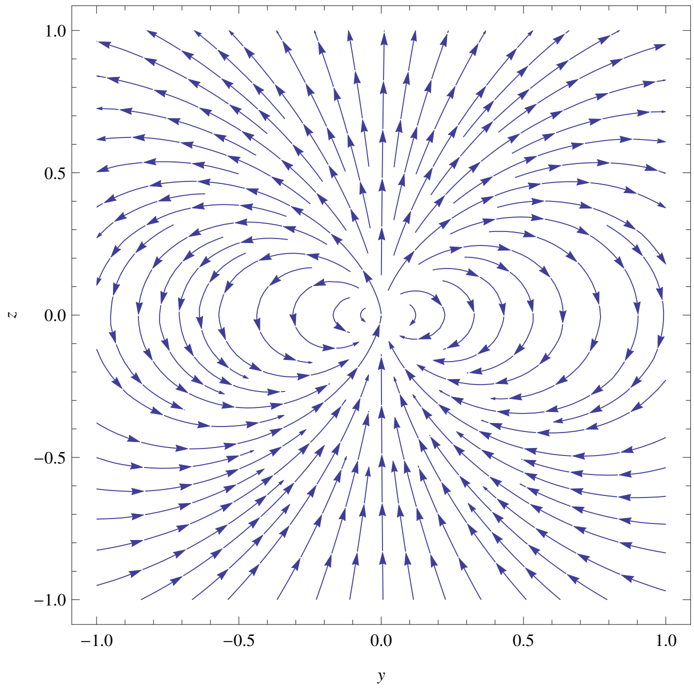
`Ec == -Grad[Vc, {x, y, z}] // Simplify`

`True`

Plot the lines of force in the *yz* plane :

In[27]:=
```
StreamPlot[Rest[Ec /. {p → 1, x → 0}] // Evaluate,
 {y, -1, 1}, {z, -1, 1}, FrameLabel → {y, z}]
```

Out[27]=




```
(* Rest verwendet die erste Komponente nicht;
damit sind wir fuer Ec wirklich in der yz - Ebene *)
```

```
(* This transforms a rank-2 tensor
  from Cartesian to polar coordinates.  *)
```

```
mat = {{x ^2 + y^2, 0}, {0, x^2 y^2}}
```

Out[5]= $\{\{x^2 + y^2, 0\}, \{0, x^2 y^2\}\}$

```
TransformedField["Cartesian" → "Polar", mat, {x, y} → {r, θ}] // Simplify
```

Out[19]= $\left\{\left\{r^2 \mathrm{Cos}[\theta]^2 \left(\mathrm{Cos}[\theta]^2 + \mathrm{Sin}[\theta]^2 + r^2 \mathrm{Sin}[\theta]^4\right), -\frac{1}{16} r^2 \left(8 - r^2 + r^2 \mathrm{Cos}[4\,\theta]\right) \mathrm{Sin}[2\,\theta]\right\}, \left\{-\frac{1}{16} r^2 \left(8 - r^2 + r^2 \mathrm{Cos}[4\,\theta]\right) \mathrm{Sin}[2\,\theta], r^2 \mathrm{Sin}[\theta]^2 \left(\mathrm{Cos}[\theta]^2 + r^2 \mathrm{Cos}[\theta]^4 + \mathrm{Sin}[\theta]^2\right)\right\}\right\}$

```
(* man beachte, dass dies keine Diagonalmatrix mehr ist ! *)
```

Using Map,
it is possible to transform the matrix as a
 list of two vectors or as a matrix of four scalar fields.
  The results are quite different in each case.

In[17]:=
```
Map[TransformedField[ "Cartesian" → "Polar", #, {x, y} → {r, θ}] &, mat] // Expand
TrigReduce[%]
```

Out[17]= $\{\{r^2 \text{Cos}[\theta]^3 + r^2 \text{Cos}[\theta] \text{Sin}[\theta]^2, -r^2 \text{Cos}[\theta]^2 \text{Sin}[\theta] - r^2 \text{Sin}[\theta]^3\}, \{r^4 \text{Cos}[\theta]^2 \text{Sin}[\theta]^3, r^4 \text{Cos}[\theta]^3 \text{Sin}[\theta]^2\}\}$

Out[18]= $\{\{r^2 \text{Cos}[\theta], -r^2 \text{Sin}[\theta]\}, \{\frac{1}{16}(2 r^4 \text{Sin}[\theta] + r^4 \text{Sin}[3\theta] - r^4 \text{Sin}[5\theta]), \frac{1}{16}(2 r^4 \text{Cos}[\theta] - r^4 \text{Cos}[3\theta] - r^4 \text{Cos}[5\theta])\}\}$

```
(* hier versagt bei der Vereinfachung auch TrigReduce:
   -r^2 Cos[θ]^2 Sin[θ]-r^2 Sin[θ]^3    = -r^2 Sin[θ]
    r^4 Cos[θ]^2 Sin[θ]^3 = r^4 Sin[θ] 1/8 (1-Cos[4 θ])
    r^4 Cos[θ]^3 Sin[θ]^2 = r^4 Cos[θ] 1/8 (1-Cos[4 θ])
     mit *)
TrigReduce[Cos[θ]^2 Sin[θ]^2]
```

Out[21]= $\frac{1}{8}(1 - \text{Cos}[4\theta])$

In[13]:=
```
Map[TransformedField[ "Cartesian" → "Polar", #, {x, y} → {r, θ}] &, mat, {2}] //
 Simplify
```

Out[13]= $\{\{r^2, 0\}, \{0, r^4 \text{Cos}[\theta]^2 \text{Sin}[\theta]^2\}\}$

```
(* Dies bekommt man also, wenn man direkt komponentenweise in mat die
 Koordinatentransformation einsetzt. *)
```