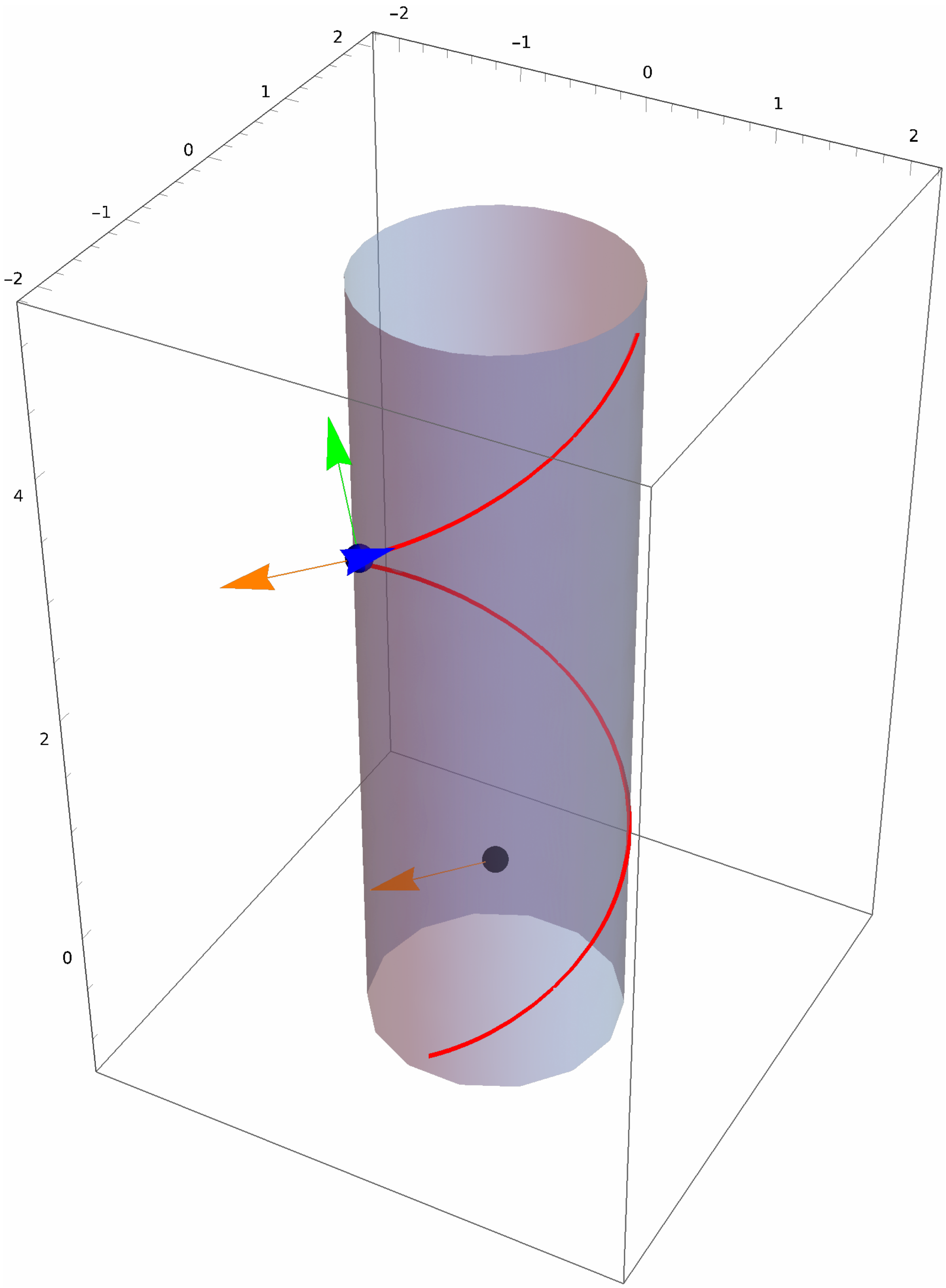
```
In[1]:= (* === 全体の描画内容 ===
1. S¹×ℝの円柱（xy平面の単位円をz軸方向に延ばした図形）を描画する。
2. 円柱上に、弧長パラメータsで定義された曲線γ(s)（ヘリックス）を描画する。
3. 曲線上の任意の点p(x,y,z)を描画する（pはx²+y²=1を満たす）。
4. 点pのxy平面への射影q=(x,y,0)を求め、原点からqへの位置ベクトルΓを描画する。
5. 同じ位置ベクトルΓを、点pを始点として描画し、色を変えて強調するものである。 *)

(* --- ヘリックス曲線γ(s)の定義 ---
γ(s) = {Cos(α s), Sin(α s), β s} と定める。
弧長パラメータとして単位速さとなるように、α² + β² = 1となる。
ここでは、α = 0.8, β = 0.6 と定める。 *)
Clear[gamma, s]; gamma[s_] := {Cos[0.8 s], Sin[0.8 s], 0.6 s};

In[2]:= (* --- 曲線上の点pの定義 ---
点pは曲線γ上の特定のパラメータ値 s0 に対応する点である。
ここでは s0 = 5 としている。 *)
s0 = 5; p = gamma[s0];

In[3]:= (* --- 点pのxy平面への射影と位置ベクトルΓの定義 ---
点p = (x, y, z) であるので、その射影 q は (x, y, 0) である。
原点から q への位置ベクトルΓは、単に q と同一である。 *)
q = {p[[1]], p[[2]], 0}; vecGamma = q; (* 位置ベクトルΓ *)



(* なお、先に作った位置ベクトルΓを点pを始点として描画する場合は、
  その矢印は点pから p + vecGamma へ向かう。 *)
pShiftedVec = p + vecGamma;

(* --- S¹×ℝの円柱の描画 ---
円柱は、xy平面の単位円 {Cos[w], Sin[w]} を z 軸方向に -3 ≤ z ≤ 3 まで延ばしたものである。 *)
cylinder = ParametricPlot3D[
{Cos[w], Sin[w], z},
{w, 0, 2 Pi},
{z, -5, 5},
Mesh → None,
PlotStyle → Directive[LightGray, Opacity[0.5]]
];

(* --- 曲線γ(s)の描画 ---
s の範囲は 0 ≤ s ≤ 10 として、曲線γ(s)を描画する。 *)
curve = ParametricPlot3D[gamma[s], {s, -2, 8}, PlotStyle → {Thick, Red}];
```

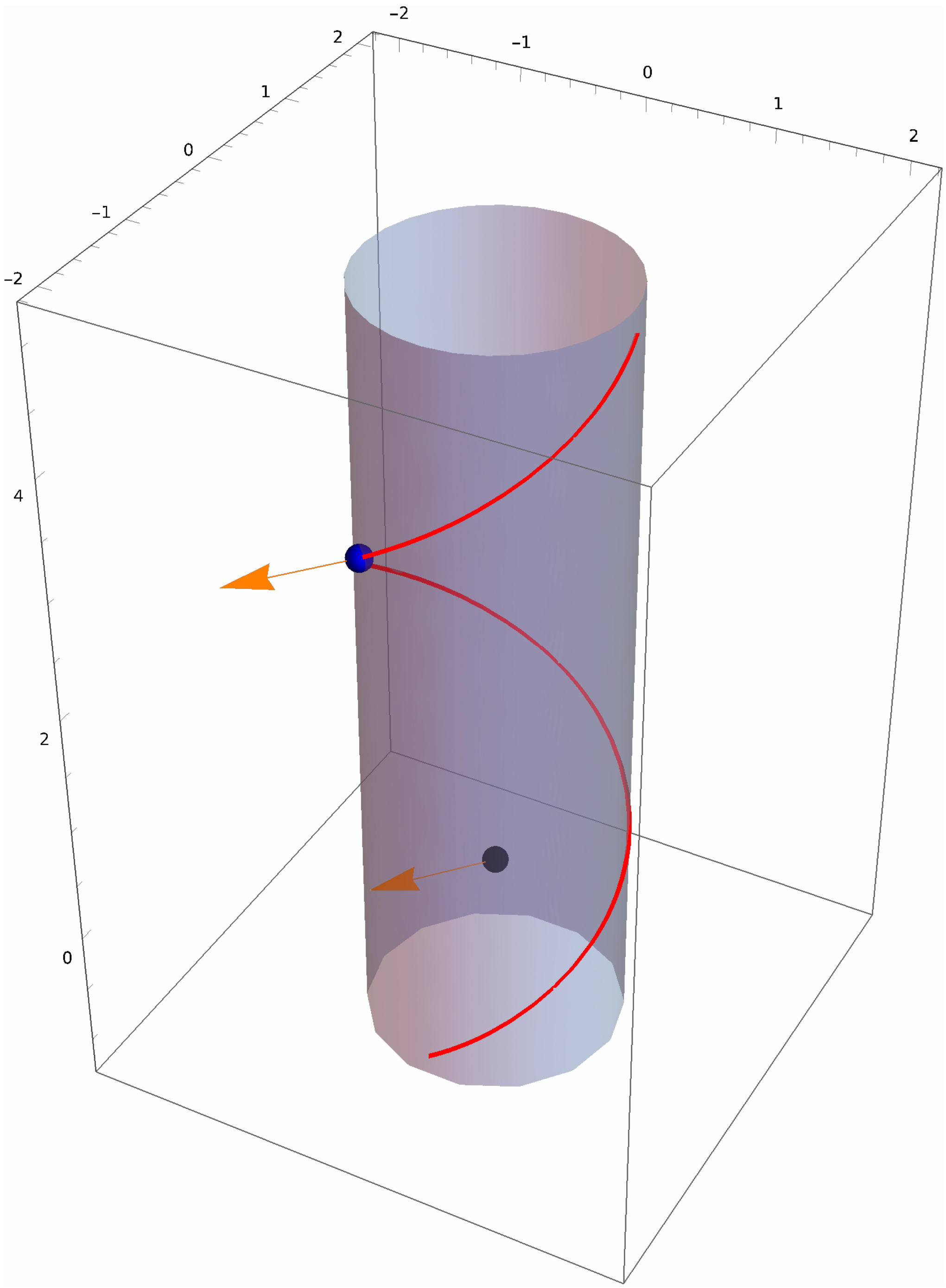
```
(* --- 点pの描画 ---
点pは Graphics3D の Sphere を用いて描画する。
サイズは適宜調整している。 *)
pointP = Graphics3D[{Blue, Sphere[p, 0.1]}];

(* --- 原点から点qへの位置ベクトルΓの描画 ---
Arrow を用いて、原点 (0,0,0) から点qへの矢印を描画する。 *)
arrowOriginToQ = Graphics3D[{Orange, Arrow[{{0, 0, 0}, q}]}];

(* --- 点pを始点とした位置ベクトルΓの描画 ---
先に定義した位置ベクトルΓを、点pを始点として再描画する。
色を Magenta にして強調している。 *)
arrowPToPplusVec = Graphics3D[{Orange, Arrow[{p, p + vecGamma}]}];

(* === 以下、追加部分 === *)
(* 原点 (0,0,0) を視覚的に示すために、原点に小さな Sphere を描画する。 *)
originPoint = Graphics3D[{Black, Sphere[{0, 0, 0}, 0.1]}];

(* 点 p における接ベクトル e(s。) の計算と描画。
曲線 γ(s) の接ベクトルは微分 γ'(s) であり、s。における接ベクトルを正規化して長さ1にする。 *)
e = Normalize[D[gamma[s], s] /. s → s0];
arrowTangent = Graphics3D[{Blue, Arrow[{p, p + e}]}];

(* 位置ベクトル Γ と接ベクトル e に直交する単位ベクトル n の計算と描画。
n は、
   点 p の射影 q（すなわち vecGamma）と接ベクトル e の外積をとり正規化することで求める。 *)
n = Normalize[Cross[vecGamma, e]];
arrowNormal = Graphics3D[{Green, Arrow[{p, p + n}]}];

(* --- すべての要素を同一画面内に表示 ---
Show を用いて、円柱、曲線、点、矢印、原点表示、接ベクトル、法線を重ね合わせて描画する。 *)
Show[
cylinder,
curve,
pointP,
arrowOriginToQ,
arrowPToPplusVec,
originPoint,
arrowTangent, arrowNormal,
Axes → True,
Boxed → True,
PlotRange → {{-2, 2}, {-2, 2}, {-1, 5}},
ImageSize → Large
]
```

Out[18]=



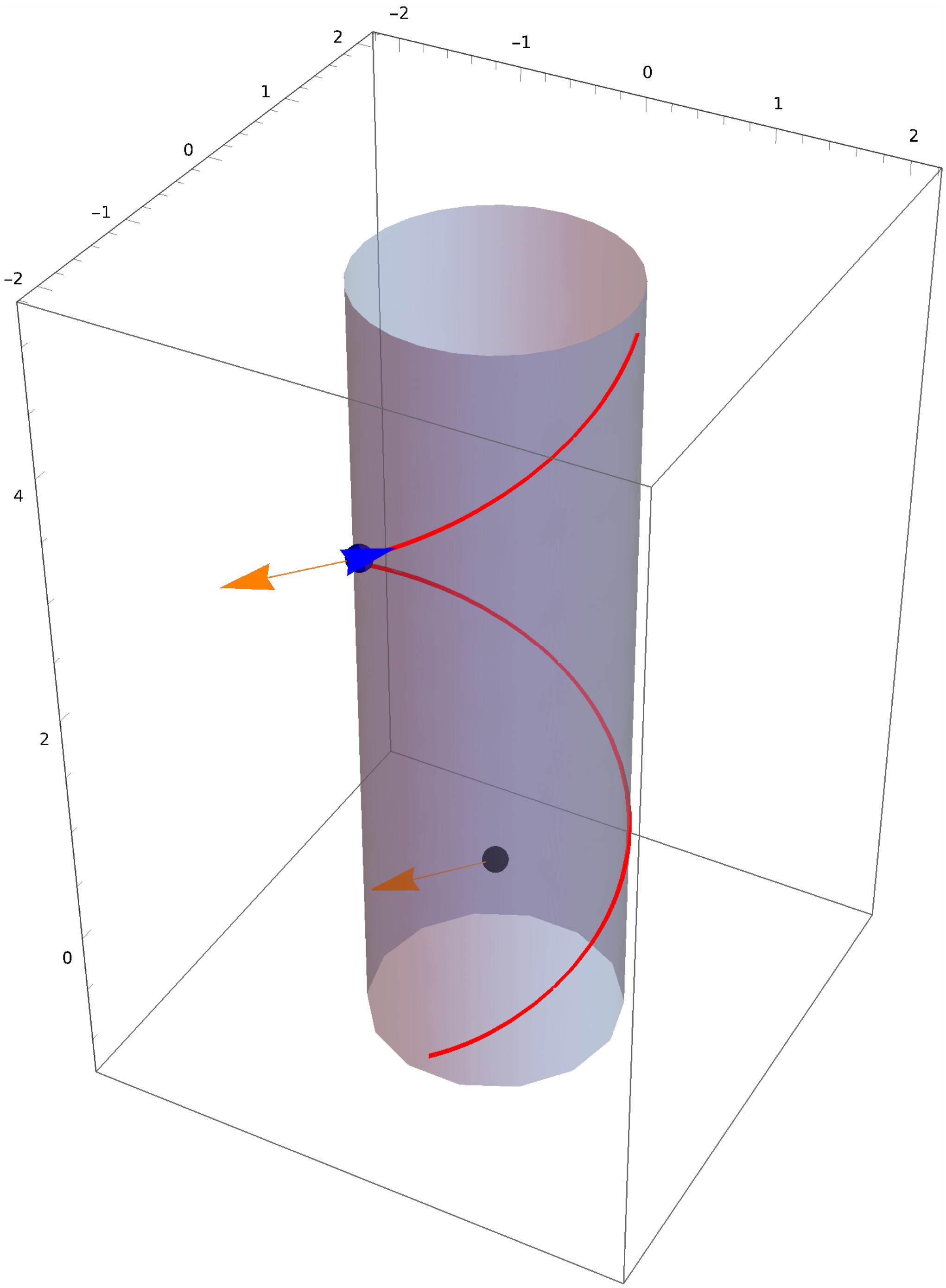
In[19]:=
```
(*Γのみ描画*)
Show[
cylinder,
curve,
pointP,
arrowOriginToQ,
arrowPToPplusVec,
originPoint,
(*arrowTangent, arrowNormal,*)
Axes → True,
Boxed → True,
PlotRange → {{-2, 2}, {-2, 2}, {-1, 5}},
ImageSize → Large
]
```

Out[19]=



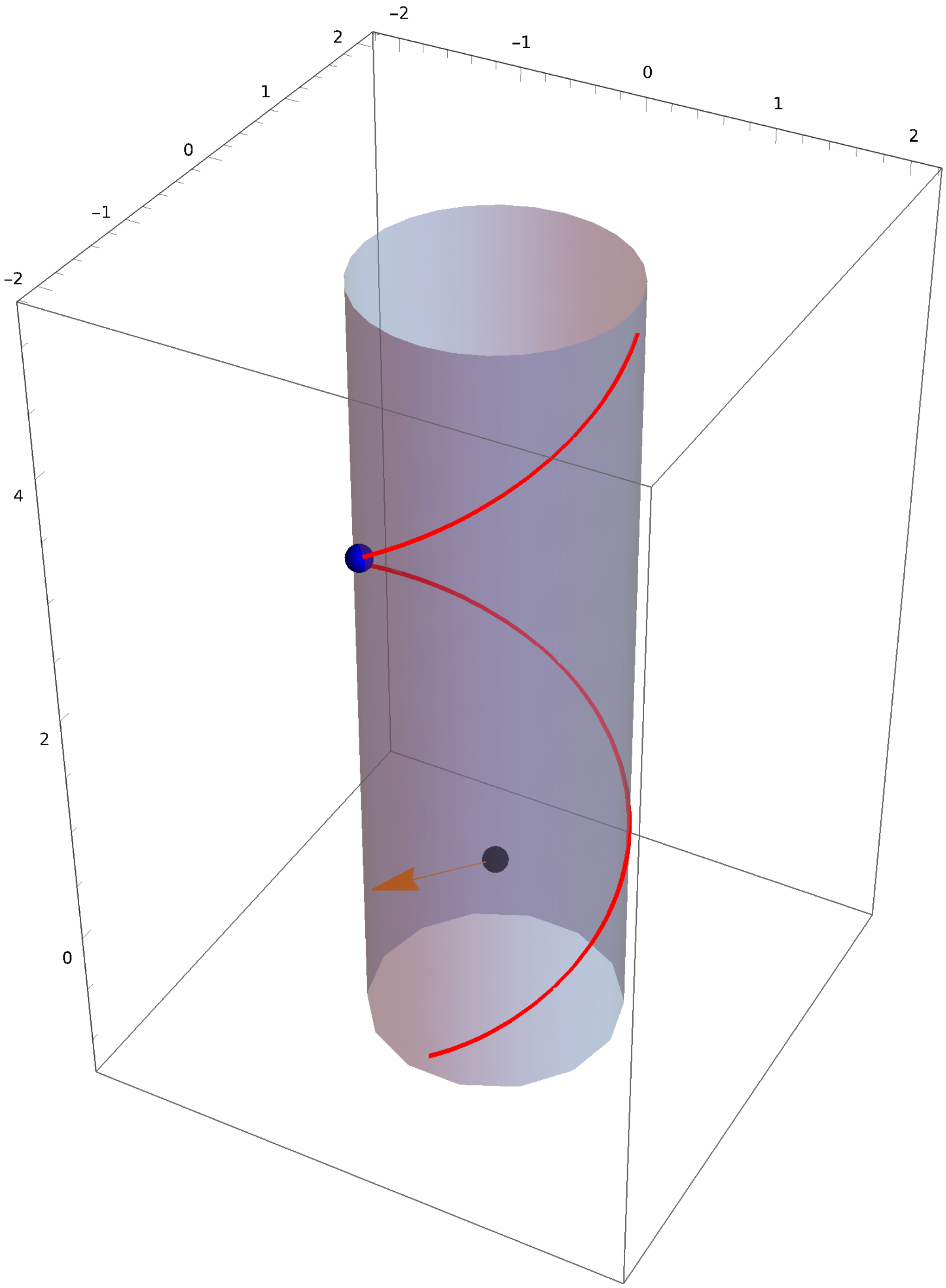
In[20]:=
```
(*Γ、eを描画*)
Show[
cylinder,
curve,
pointP,
arrowOriginToQ,
arrowPToPplusVec,
originPoint,
arrowTangent, (* arrowNormal,*)
Axes → True,
Boxed → True,
PlotRange → {{-2, 2}, {-2, 2}, {-1, 5}},
ImageSize → Large
]
```

Out[20]=



```
In[24]:= (*ベクトルを描画していない場合*)
Show[
cylinder,
curve,
pointP,
arrowOriginToQ,
(*arrowPToPplusVec,*)
originPoint,
(*arrowTangent, arrowNormal,*)
Axes → True,
Boxed → True,
PlotRange → {{-2, 2}, {-2, 2}, {-1, 5}},
ImageSize → Large
]
```

Out[24]=



In[21]:= `normOriginToQ = Norm[q] // Simplify`

Out[21]=

`1.`

In[22]:= `normPToPplusVec = Norm[(p + vecGamma) - p] // Simplify`

Out[22]=

`1.`