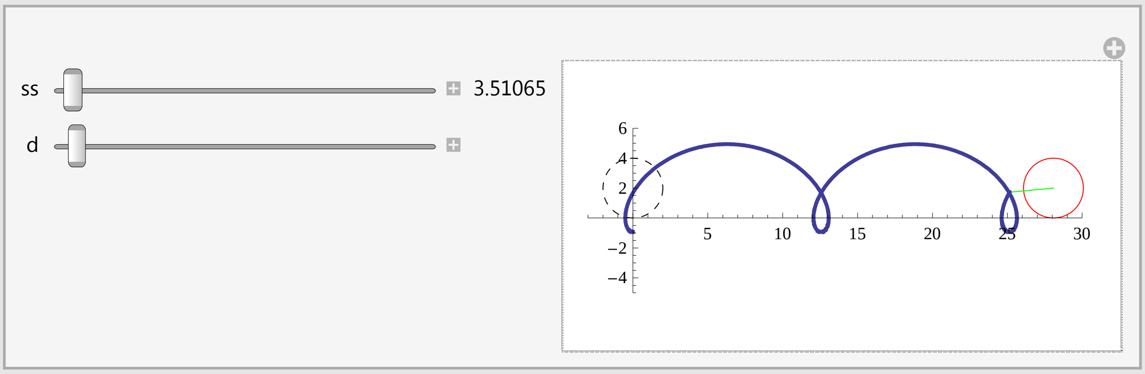
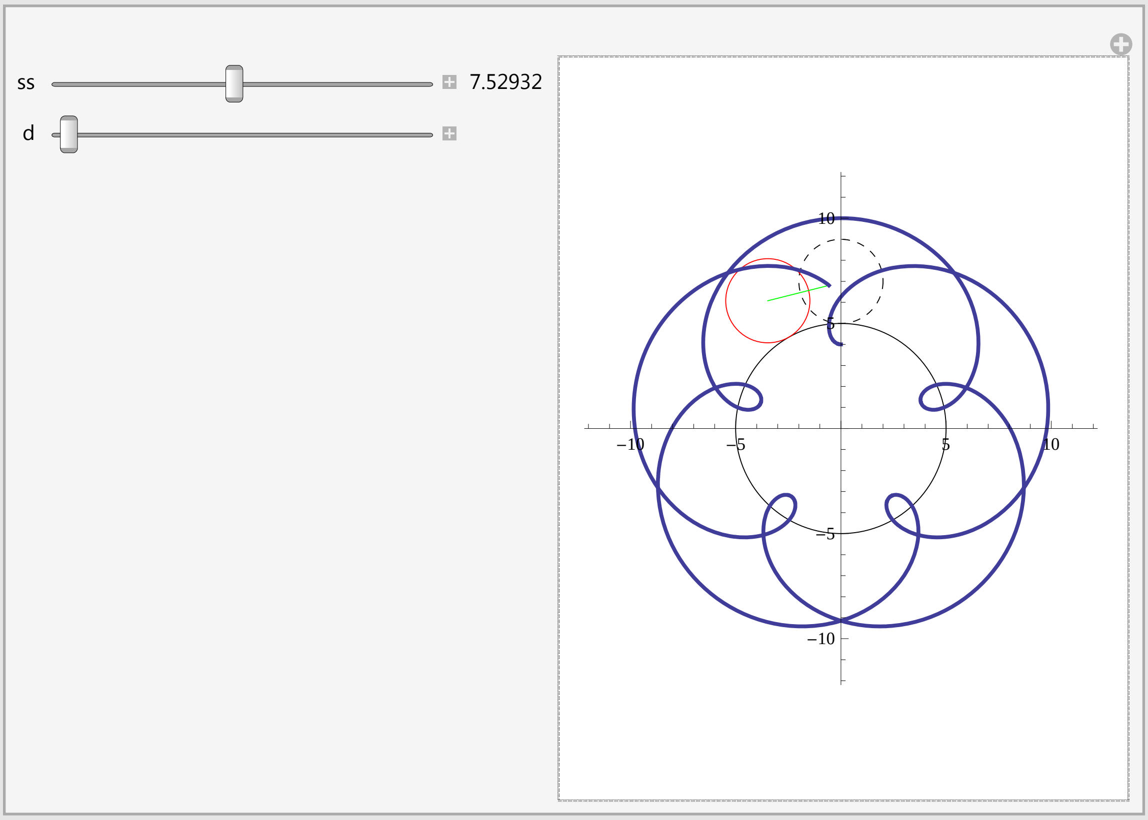
Лекция 4. Классическая дифференциальная геометрия.

Кривые (циклоида, эпициклоида, гипоциклоида (кривая Штейнера))

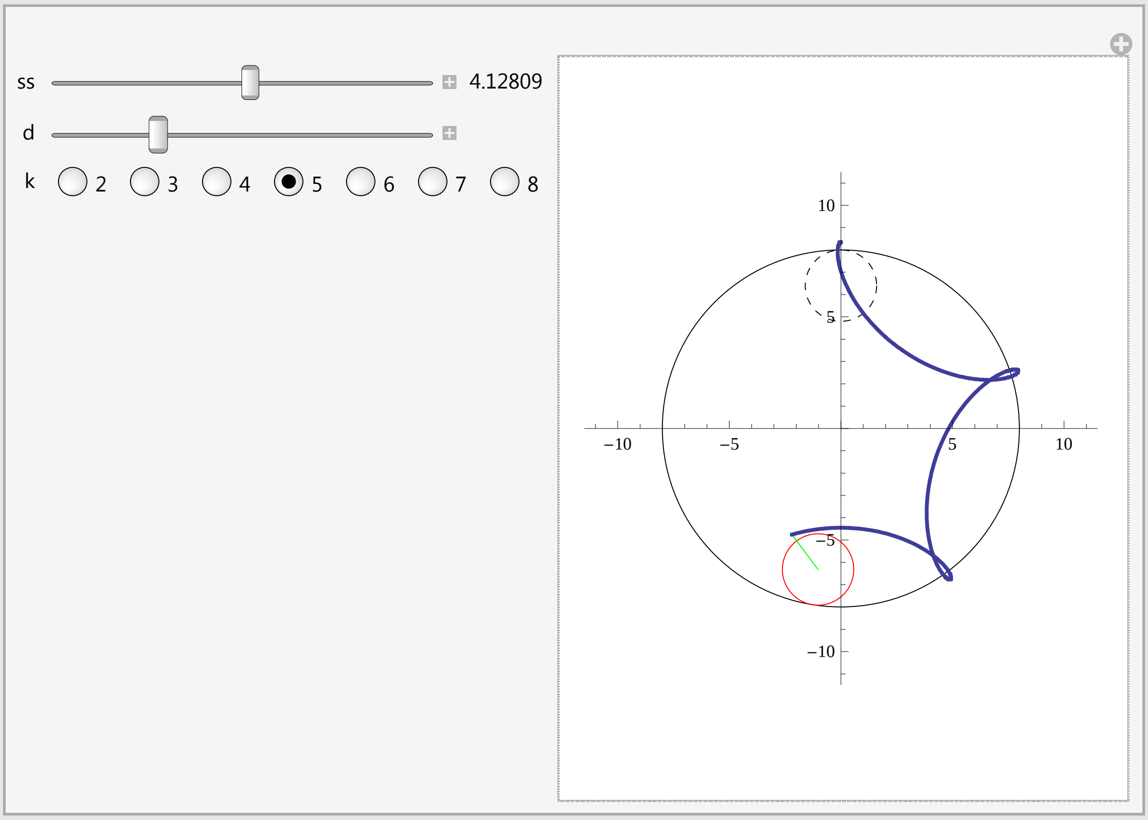
```
Manipulate[ r = 2; ϕ = 4; R = 10;
 Show[{
   Graphics[{{Dashed, Circle[{0, r}, r]}, Red, Circle[{r ϕ ss, r}, r], Green,
     Line[{{r ϕ ss, r}, {r ϕ ss + d Sin[ϕ ss], r + d Cos[ϕ ss]}}]}, Axes → True],
   ParametricPlot[{r ϕ s + d Sin[ϕ s], r + d Cos[ϕ s]}, {s, 0, ss},
    PlotStyle → Thick, PlotRange → {{-3, 30}, {-5, 6}}]
  }, PlotRange → {{-3, 30}, {-5, 6}}]
 ], {ss, 0.01, r 2πR/(ϕ r), Appearance → "Labeled"}, {d, -3, 3}]
```



```
Manipulate[R = 5; r = 2; ψ = 4;
 Show[{
   Graphics[{Circle[{0, 0}, R], {Dashed, Circle[{0, R + r}, r]},
     Red, Circle[(R + r) {Sin[r/R ψ ss], Cos[r/R ψ ss]}, r], Green,
     Line[{(R + r) {Sin[r/R ψ ss], Cos[r/R ψ ss]}, (R + r) {Sin[r/R ψ ss], Cos[r/R ψ ss]} +
        d {Sin[(r/R + 1) ψ ss], Cos[(r/R + 1) ψ ss]}}]}, Axes → True],
   ParametricPlot[(R + r) {Sin[r/R ψ s], Cos[r/R ψ s]} +
     d {Sin[(r/R + 1) ψ s], Cos[(r/R + 1) ψ s]}, {s, 0, ss}, PlotStyle → Thick]},
  PlotRange → R + 2 r + 0.2 + 3],
 {ss, 0.01, r (2 π R)/(ψ r), Appearance → "Labeled"}, {d, -3, 3}]
```



```
Manipulate[R = 8; r = R / k; ψ = 4;
 Show[{
   Graphics[{Circle[{0, 0}, R], {Dashed, Circle[{0, R - r}, r]},
     Red, Circle[(R - r) {Sin[r/R ψ ss], Cos[r/R ψ ss]}, r], Green,
     Line[{(R - r) {Sin[r/R ψ ss], Cos[r/R ψ ss]}, (R - r) {Sin[r/R ψ ss], Cos[r/R ψ ss]} -
        d {Sin[(r/R - 1) ψ ss], Cos[(r/R - 1) ψ ss]}}]}, Axes → True],
   ParametricPlot[(R - r) {Sin[r/R ψ s], Cos[r/R ψ s]} - d {Sin[(r/R - 1) ψ s], Cos[(r/R - 1) ψ s]},
    {s, 0, ss}, PlotStyle → Thick]},
  PlotRange → R + 3.5],
 {ss, 0.01, r (2 π R)/(ψ r), Appearance → "Labeled"},
 {d, -4, 4}, {k, {2, 3, 4, 5, 6, 7, 8}, ControlType → RadioButton}]
```

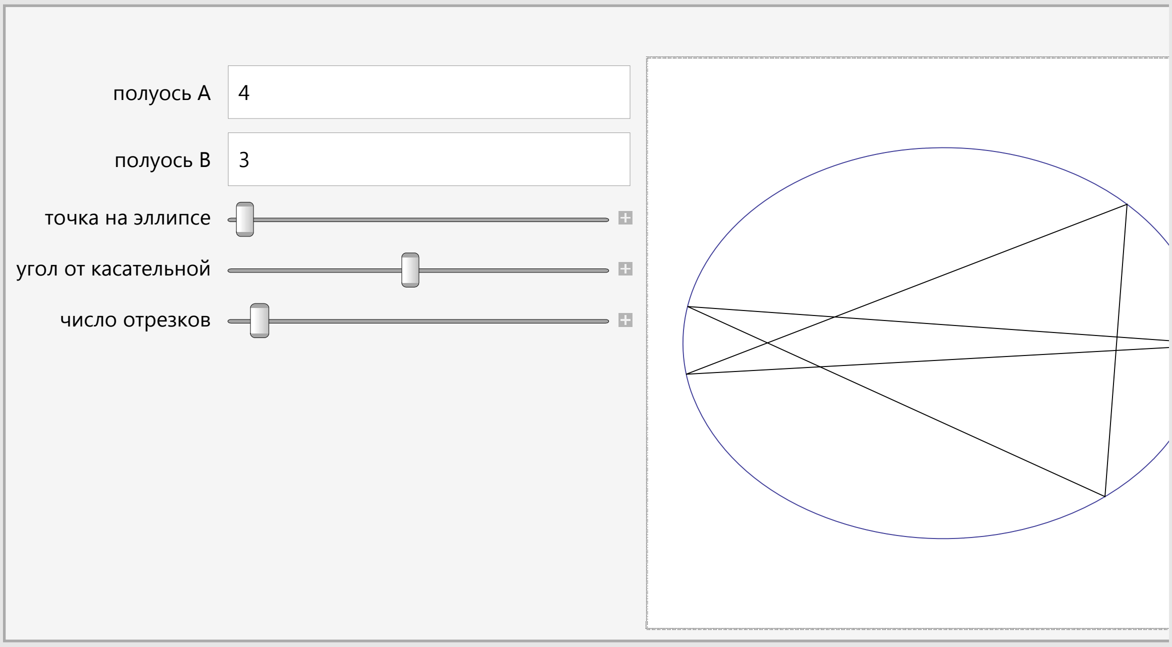


Биллиард в эллипсе

In[10]:=

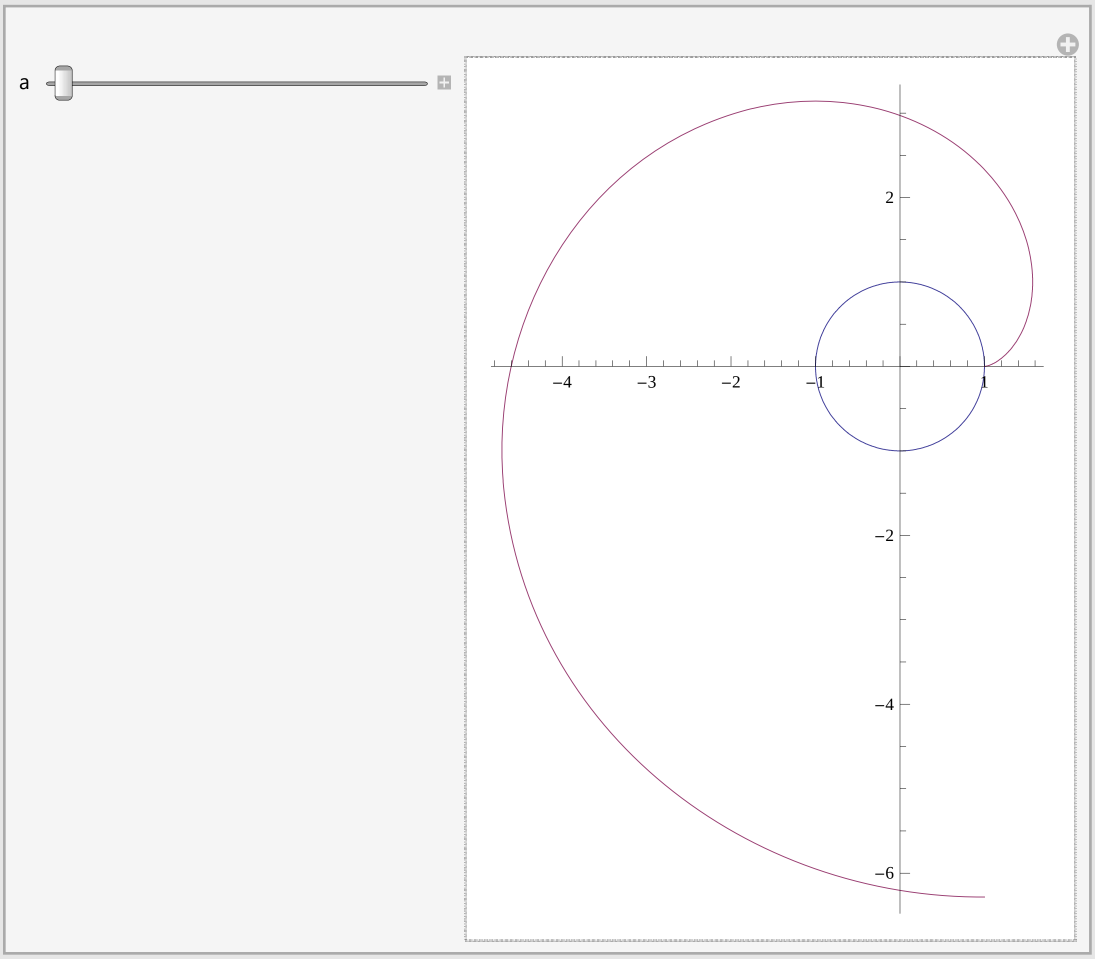
```
f[A_, B_, ϕ_] := {A Cos[ϕ], B Sin[ϕ]};
Tf[A_, B_, ϕ_] := Normalize[D[f[A, B, t], t] /. t → ϕ];
SOLUTION[A_, B_, ϕ_, α_] := FullSimplify[NSolve[x^2/A^2 + y^2/B^2 == 1 &&
    f[A, B, ϕ] + λ RotationMatrix [α].Tf[A, B, ϕ] == {x, y} && λ ≠ 0, {x, y, λ}]];
NextPoint[A_, B_, ϕ_, α_] := {SOLUTION[A, B, ϕ, α][[1, 1, 2]],
   SOLUTION[A, B, ϕ, α][[1, 2, 2]]};

NewPhi[A_, B_, ϕ_, α_] :=
  N[ArcTan[1/A SOLUTION[A, B, ϕ, α][[1, 1, 2]], 1/B SOLUTION[A, B, ϕ, α][[1, 2, 2]]]];
NewAl[A_, B_, ϕ_, α_] :=
  N[ArcCos[((NextPoint[A, B, ϕ, α] - f[A, B, ϕ])).Tf[A, B, NewPhi[A, B, ϕ, α]] /
      (Norm[NextPoint[A, B, ϕ, α] - f[A, B, ϕ]] Norm[Tf[A, B, NewPhi[A, B, ϕ, α]]])]];
FF[V_, n_] := If[Mod[n[[1]], 2] == 0, f[V[[1]], V[[2]], V[[3]]],
   f[V[[1]], V[[2]], V[[3]]]];
TRAJECTORY[A_, B_, ϕ_, α_, NN_] := (PTSϕα1 = {{ϕ, α}};
   Do[PTSϕα1 = Append[PTSϕα1, {NewPhi[A, B, Last[PTSϕα1][[1]], Last[PTSϕα1][[2]]],
       NewAl[A, B, Last[PTSϕα1][[1]], Last[PTSϕα1][[2]]]}], {NN}];
   PTSxy1 := MapIndexed[FF, {A, B, #[[1]]} & /@ PTSϕα1]);
Manipulate[TRAJECTORY[A, B, ϕ, α, NN]; Show[ParametricPlot[f[A, B, t], {t, 0, 2 π}],
  ParametricPlot[f1[A, B, t], {t, 0, 2 π}], Graphics[Line[PTSxy1]],
  Axes → False, AspectRatio → Automatic], {{A, 4, "полуось A"}},
 {{B, 3, "полуось B"}}, {{ϕ, 0, "точка на эллипсе"}, 0, 2 π},
 {{α, 1.5, "угол от касательной"}, 0.001, π - 0.001},
 {{NN, 5, "число отрезков"}, 1, 100, 1}]
```

Out[18]=



Эвольвента

```
Manipulate[
 γ[t_] := {a Cos[t], a Sin[t]};
 s[t_] := NIntegrate[Norm[γ'[τ]], {τ, 0, t}];
 δ[t_] := γ[t] - γ'[t]/Norm[γ'[t]] s[t];
 ParametricPlot[{γ[t], δ[t]}, {t, 0, 2 π}],
 {{a, 1}, 1, 5}]
```

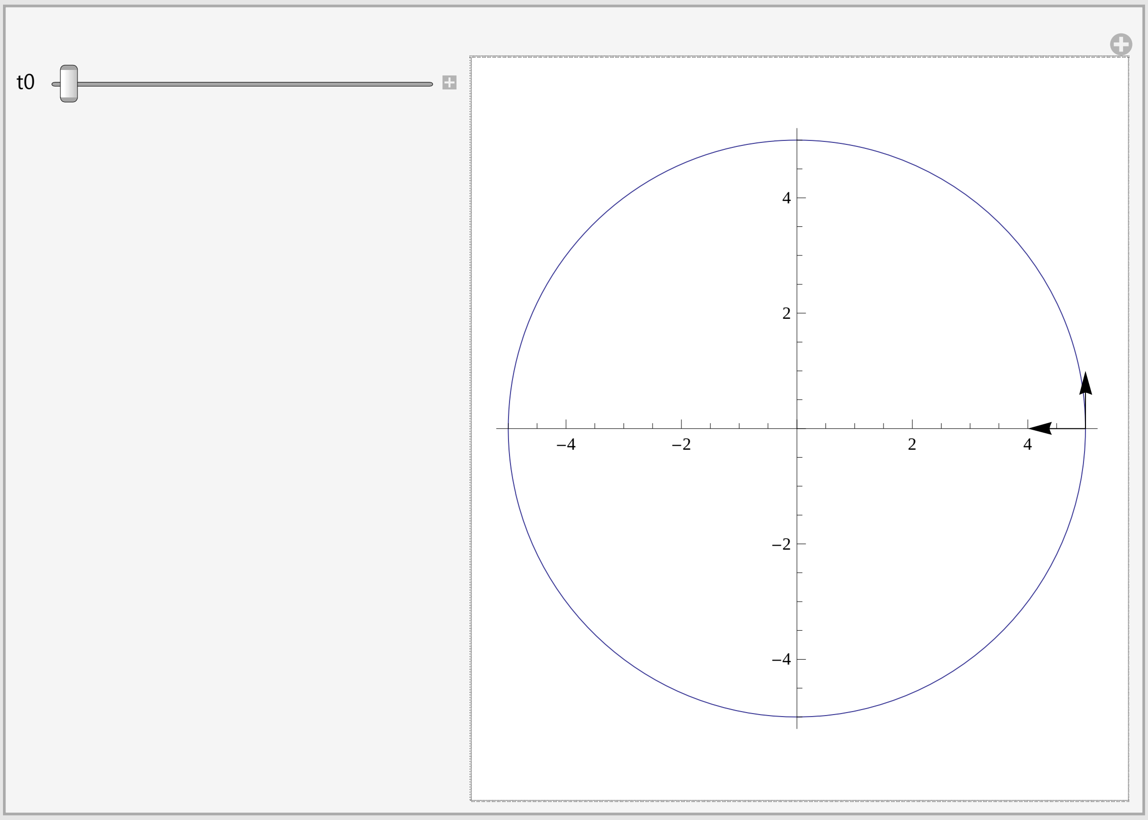


Визуализация репера Френе

In[19]:=

```
Manipulate[γ[t_] := 5 {Cos[t], Sin[t]};
 v[t_] := γ'[t] / Norm[γ'[t]];
 n[t_] := (γ''[t] - (v[t].γ''[t]) v[t]) / Norm[γ''[t] - (v[t].γ''[t]) v[t]];
 Show[{ParametricPlot[γ[t], {t, 0, 2 π}], Graphics[{
     Arrow[{γ[t0], γ[t0] + v[t0]}], Arrow[{γ[t0], γ[t0] + n[t0]}]
    }]
  }],
 {t0, 0, 2 π}
]
```

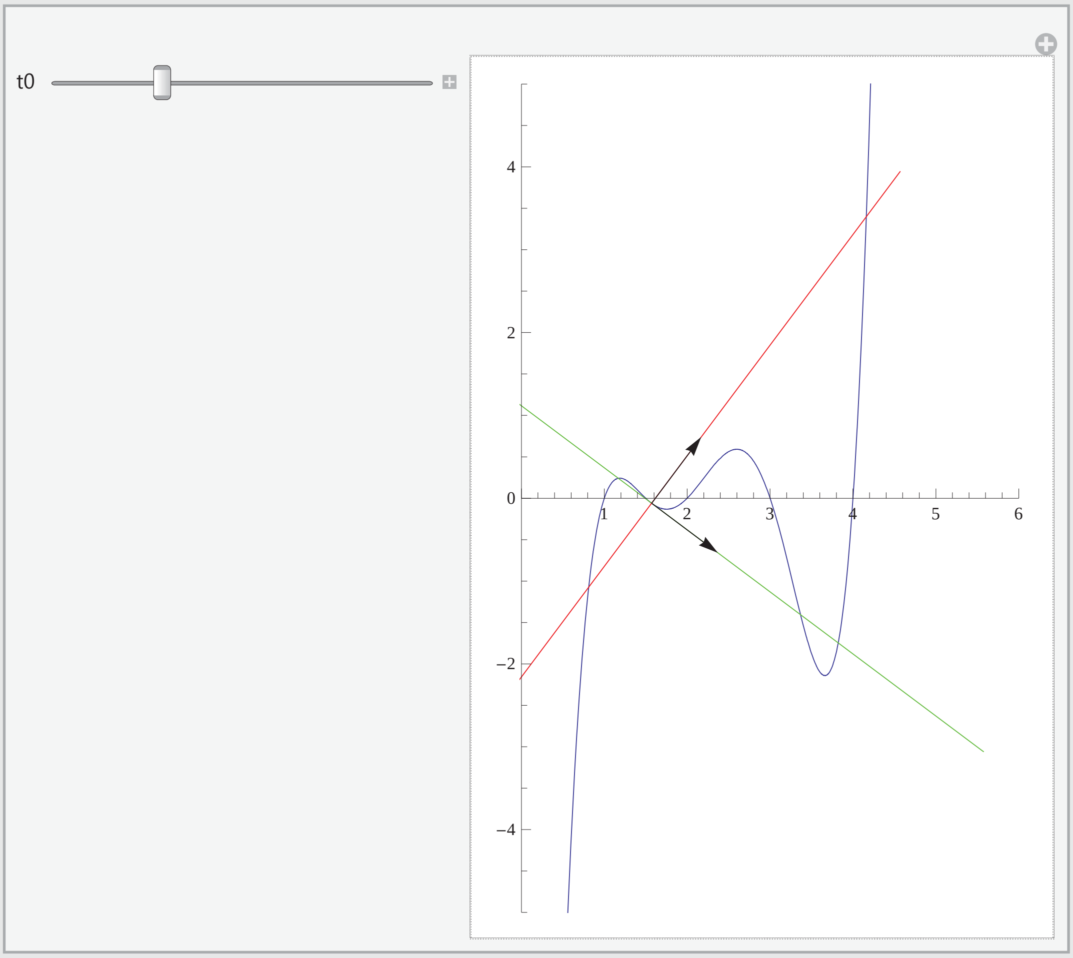
Out[19]=



In[20]:=

```
Manipulate[γ[t_] := {t, (t - 1) (t - 1.5) (t - 2) (t - 3) (t - 4)};
 v[t_] := γ'[t] / Norm[γ'[t]];
 n[t_] := (γ''[t] - (v[t].γ''[t]) v[t]) / Norm[γ''[t] - (v[t].γ''[t]) v[t]];
 Show[{ParametricPlot[γ[t], {t, 0.5, 4.5}, PlotRange → {{0, 6}, {-5, 5}}],
   ParametricPlot[
    {γ[t0] + k v[t0], γ[t0] + k n[t0]}, {k, -5, 5}, PlotStyle → {Green, Red}],
   Graphics[{
     Arrow[{γ[t0], γ[t0] + v[t0]}], Arrow[{γ[t0], γ[t0] + n[t0]}]
    }]
  }],
 {{t0, π/2}, 0.5, 4.5}
]
```

Out[20]=

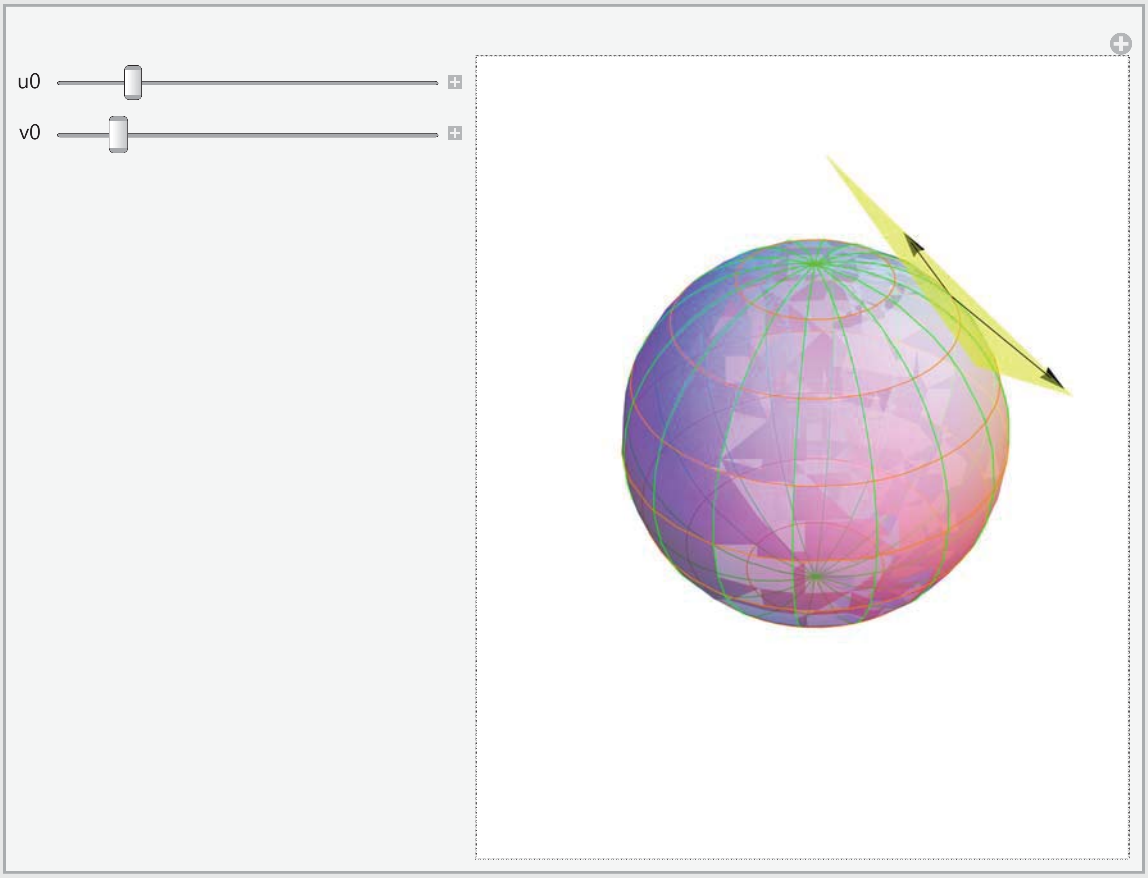


Визуализация канонического базиса поверхности

In[23]:=

```
Clear[r, r1, r2, u, v, n, uu, vv];
Manipulate[r[u_, v_] := {Sin[u] Cos[v], Sin[u] Sin[v], Cos[u]};
 r1[u_, v_] := D[r[uu, v], uu] /. uu → u;
 r2[u_, v_] := D[r[u, vv], vv] /. vv → v;
 n[u_, v_] := Cross[r1[u, v], r2[u, v]] / Norm[Cross[r1[u, v], r2[u, v]]];
 Show[{
   ParametricPlot3D[r[u, v], {u, 0, 2 π}, {v, 0, 2 π}, MeshStyle → {Orange, Green},
    Boxed → False, Axes → None, PlotStyle → {Opacity[0.5]}],
   (*ParametricPlot3D[{r[u0,v0]+x r1[u0,v0]+y r2[u0,v0]},{x,-1,1},
    {y,-1,1},Mesh->None,PlotStyle→{Opacity[0.5]}],*)
   ContourPlot3D[{x, y, z}.n[u0, v0] == r[u0, v0].n[u0, v0], {x, -2, 2},
    {y, -2, 2}, {z, -2, 2}, Mesh → None, ContourStyle → {Yellow, Opacity[0.5]}],
   Graphics3D[{
     Arrow[{r[u0, v0], r[u0, v0] + r1[u0, v0]}],
     Arrow[{r[u0, v0], r[u0, v0] + r2[u0, v0]}]
    }]
  }],
 {{u0, π/4}, π/8, π - π/8, π/8}, {{v0, π/4}, 0, 2 π, π/8}]
```

Out[24]=



Визуализация изменения координат касательного вектора

In[25]:=

```
r[u_, v_] := {Cosh[u] Cos[v], Cosh[u] Sin[v], u};
r1[u_, v_] := D[r[uu, v], uu] /. uu → u;
r2[u_, v_] := D[r[u, vv], vv] /. vv → v;
g11[u_, v_] := r1[u, v].r1[u, v];
g12[u_, v_] := r1[u, v].r2[u, v];
g22[u_, v_] := r2[u, v].r2[u, v];
G[u_, v_] := ( g11[u, v]  g12[u, v]
               g12[u, v]  g22[u, v] );
Manipulate[
 γ[t_] := {t, a t + b}; (*координаты кривой в координатах на поверхности*)
 γr[t_] := r[u, v] /. {u → γ[t][[1]], v → γ[t][[2]]};
 (*координаты кривой в координатах объемлющего пространства*)
 v[t_] := γ'[t]; (*координаты касательного вектора в координатах на поверхности*)
 w[t_] := Transpose[{r1[u, v], r2[u, v]}].v[t] /. {u → γ[t][[1]], v → γ[t][[2]]};
 (*координаты касательного вектора в координатах объемлющего пространства*)
 e1[t_] := r1[u, v] /. {u → γ[t][[1]], v → γ[t][[2]]};
 e2[t_] := r2[u, v] /. {u → γ[t][[1]], v → γ[t][[2]]};
 α[t_] :=
  ArcCos[(v[t].G[u, v].{1, 0}) / (Sqrt[v[t].G[u, v].v[t]] Sqrt[{1, 0}.G[u, v].{1, 0}]) /. {u → γ[t][[1]], v → γ[t][[2]]}] //
    FullSimplify; (*угол между кривой и первой координатной линией*)
 Column[{α[t0], v[t0],
    Show[{
      ParametricPlot3D[r[u, v],
       {u, -1, 1}, {v, 0, 2 π}, Boxed → False, Mesh → None, Axes → None,
       PlotStyle → {Red, Opacity[0.5]}, PlotRange → {{-2, 2}, {-2, 2}, {-1, 1}}],
      ParametricPlot3D[γr[t], {t, -1, 1}, Boxed → False, Mesh → None,
       Axes → None, PlotStyle → {Green, Thick}],
      Graphics3D[{Arrow[{γr[t0], γr[t0] + e1[t0]}], Arrow[{γr[t0], γr[t0] + e2[t0]}],
        Yellow, Arrow[{γr[t0], γr[t0] + w[t0]}]}]
     }]
   }],
 {t0, -1, 1}, {{a, 1}, 0, 10}, {b, 0, 5}]
```

Out[32]=

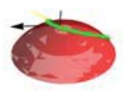
t0

a

b

$$\text{ArcCos}\left[\sqrt{\frac{2}{3-\text{Cos}[2]}}\right]$$

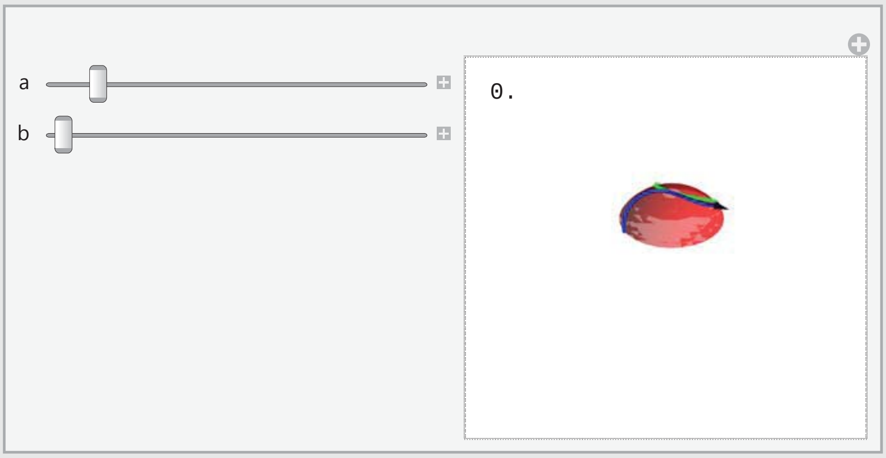
{1, 1}



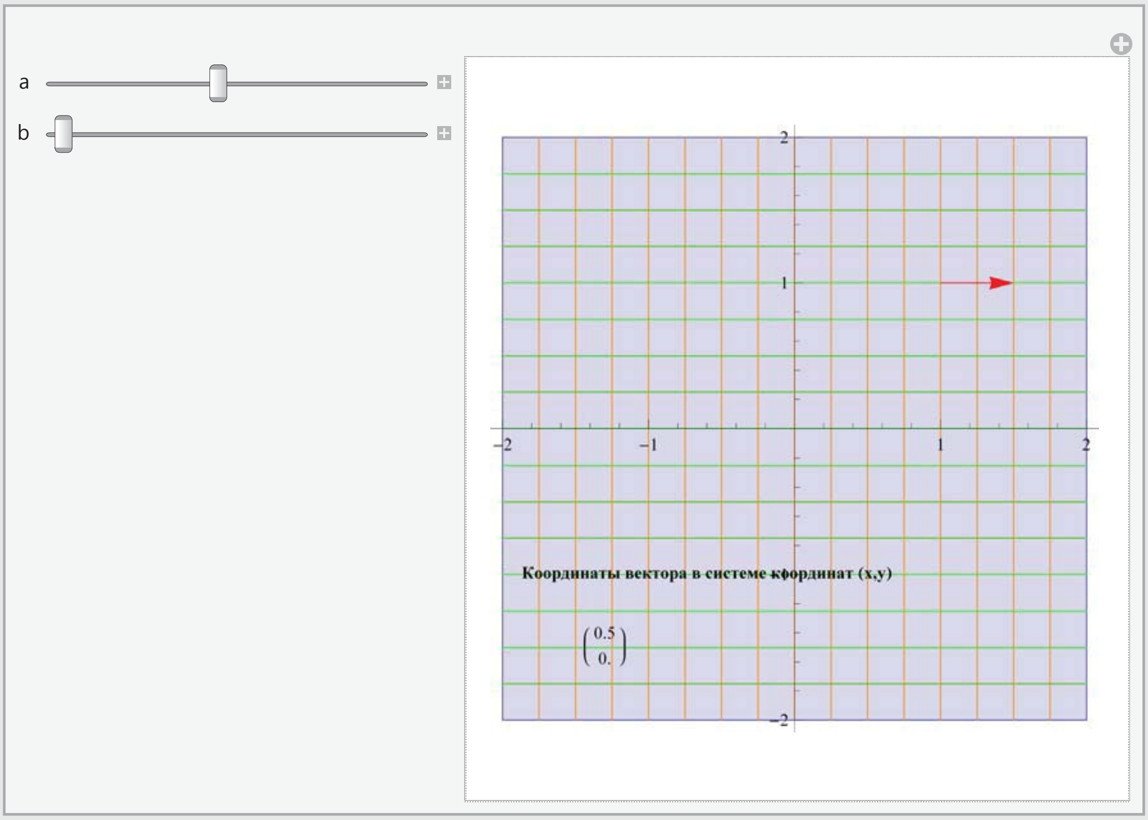
In[33]:=

```
r[u_, v_] := {Cosh[u] Cos[v], Cosh[u] Sin[v], u};
r1[u_, v_] := D[r[uu, v], uu] /. uu → u;
r2[u_, v_] := D[r[u, vv], vv] /. vv → v;
g11[u_, v_] := r1[u, v].r1[u, v];
g12[u_, v_] := r1[u, v].r2[u, v];
g22[u_, v_] := r2[u, v].r2[u, v];
G[u_, v_] := ( g11[u, v]  g12[u, v]
               g12[u, v]  g22[u, v] );
Manipulate[
 γ1[t_] := {t, a t + b}; (*координаты первой кривой в координатах на поверхности*)
 γ1r[t_] := r[u, v] /. {u → γ1[t][[1]], v → γ1[t][[2]]};
 γ2[t_] := {t, t^2}; (*координаты второй кривой в координатах на поверхности*)
 γ2r[t_] := r[u, v] /. {u → γ2[t][[1]], v → γ2[t][[2]]};
 (*координаты второй кривой в координатах объемлющего пространства*)
 v1[t_] := γ1'[t]; (*координаты касательного вектора
  первой кривой в координатах на поверхности*)
 w1[t_] := Transpose[{r1[u, v], r2[u, v]}].v1[t] /. {u → γ1[t][[1]], v → γ1[t][[2]]};
 (*координаты касательного вектора первой
  кривой в координатах объемлющего пространства*)
 v2[t_] := γ2'[t]; (*координаты касательного вектора
  второй кривой в координатах на поверхности*)
 w2[t_] := Transpose[{r1[u, v], r2[u, v]}].v2[t] /. {u → γ2[t][[1]], v → γ2[t][[2]]};
 (*координаты касательного вектора второй
  кривой в координатах объемлющего пространства*)
 intpt = NSolve[γ1[t] == γ2[t], t];
 t0 = t /. intpt[[1]];
 α[t_] := ArcCos[ v1[t].G[u, v].v2[t] / (Sqrt[v1[t].G[u, v].v1[t]] Sqrt[v2[t].G[u, v].v2[t]]) /.
     {u → γ1[t][[1]], v → γ1[t][[2]]}] // FullSimplify; (*угол между кривыми*)
 Column[{α[t0],
   Show[{
     ParametricPlot3D[r[u, v],
      {u, -1, 1}, {v, 0, 2 π}, Boxed → False, Mesh → None, Axes → None,
      PlotStyle → {Red, Opacity[0.5]}, PlotRange → {{-2, 2}, {-2, 2}, {-1, 1}}],
     ParametricPlot3D[{γ1r[t], γ2r[t]}, {t, -1, 1}, Boxed → False,
      Mesh → None, Axes → None, PlotStyle → {{Green, Thick}, {Blue, Thick}}],
     Graphics3D[{Arrow[{γ1r[t0], γ1r[t0] + w1[t0]}],
       Arrow[{γ2r[t0], γ2r[t0] + w2[t0]}]}]
    }]
  }],
 {{a, 1}, 0, 10}, {b, 0, 5}]
```

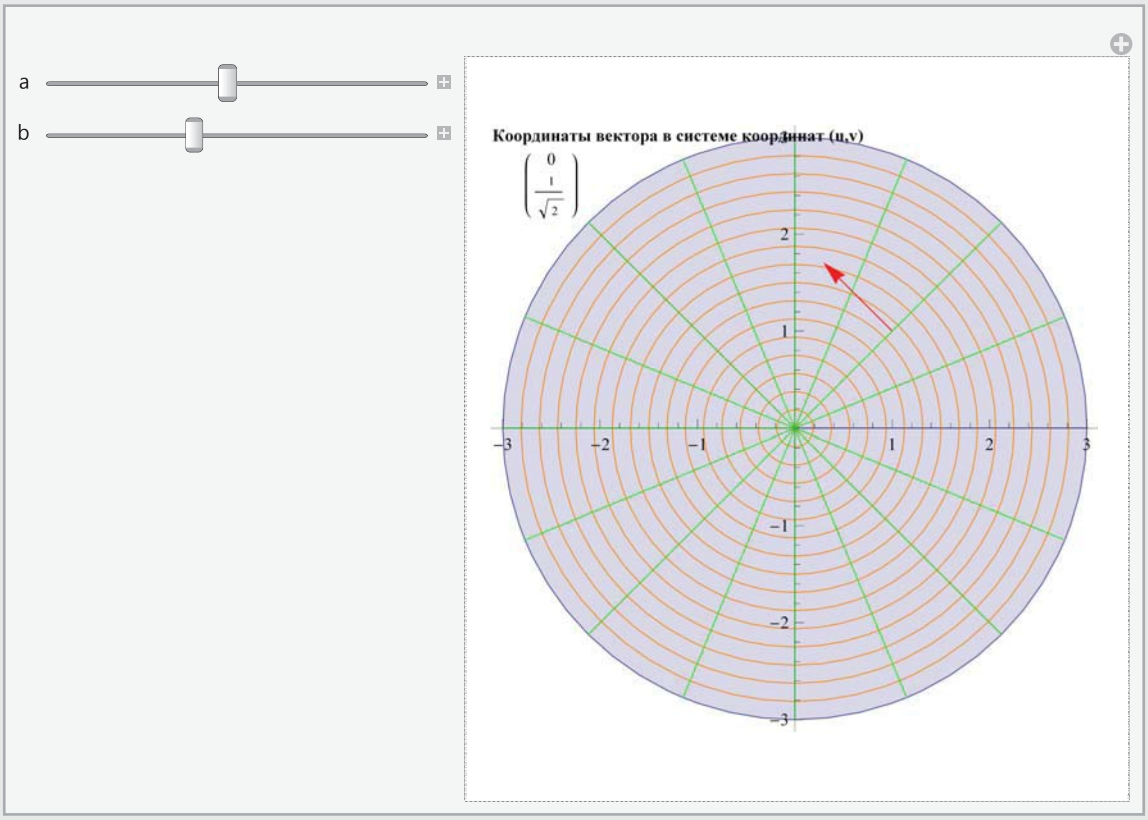
Out[40]=




```
Manipulate[
 P = {1, 1};
 ξ[a_, b_] := {a Cos[b], a Sin[b]};
 Show[{ParametricPlot[{x, y}, {x, -2, 2}, {y, -2, 2}, Frame → False,
     MeshStyle → {Orange, Green}],
   Graphics[{Red, Arrow[{P, P + ξ[a, b]}], Black,
     Text[ξ[a, b] // MatrixForm, {-1.3, -1.5}], Text[Style[
       "Координаты вектора в системе координат (x,y)", Bold], {-0.6, -1}]}]}],
 {{a, 0.5}, 0.1, 1}, {b, 0, 2
   π}]
```



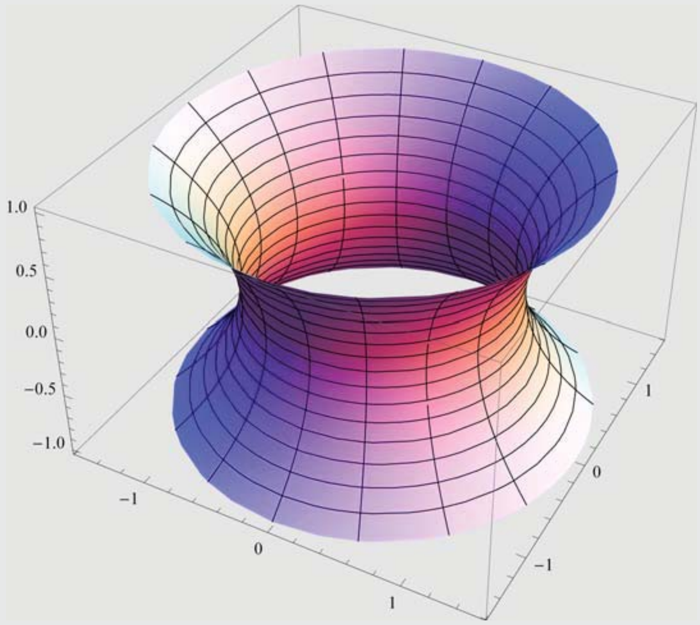
```
Manipulate[u[x_, y_] := Sqrt[x^2 + y^2];
 v[x_, y_] := ArcTan[y/x];
 f[x_, y_] := {u[x, y], v[x, y]};
 invf[u_, v_] := {u Cos[v], u Sin[v]};
 ux[x_, y_] := D[u[xx, y], xx] /. xx → x;
 uy[x_, y_] := D[u[x, yy], yy] /. yy → y;
 vx[x_, y_] := D[v[xx, y], xx] /. xx → x;
 vy[x_, y_] := D[v[x, yy], yy] /. yy → y;
 j[x_, y_] := ( ux[x, y]  uy[x, y]
                vx[x, y]  vy[x, y] );
 P = {1, 1};
 ξ[a_, b_] := {a Cos[b], a Sin[b]};
 η[{x_, y_}, a_, b_] := j[x, y].ξ[a, b];
 Show[{ParametricPlot[invf[u, v], {u, 0, 3},
    {v, 0, 2 π}, Frame → False, MeshStyle → {Orange, Green}],
   Graphics[{Red, Arrow[{P, P + ξ[a, b]}], Black,
     Text[η[P, a, b] // MatrixForm, {-2.5, 2.5}], Text[
      Style["Координаты вектора в системе координат (u,v)", Bold], {-1.2, 3}]}]}],
 {{a, 1}, 0.1, 2}, {{b, 3π/4}, 0, 2 π}]
```



Визуализация зависимости от параметра

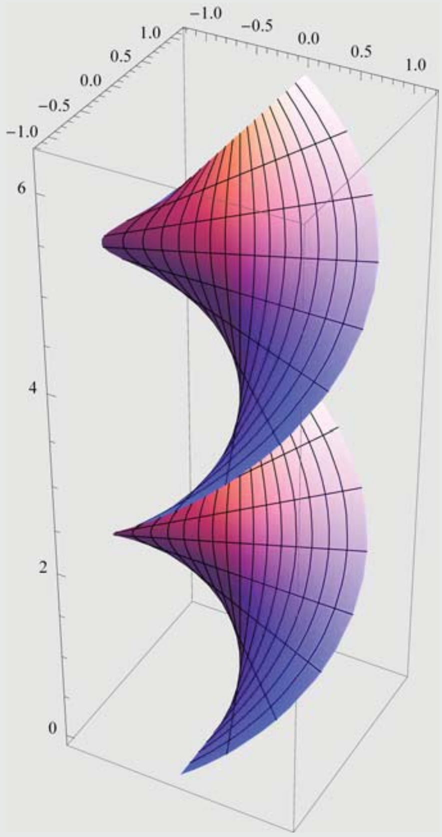
Катеноид (минимальная поверхность вращения)

```
ParametricPlot3D[{Cosh[u] Cos[v], Cosh[u] Sin[v], u}, {u, -1, 1}, {v, 0, 2 π}]
```



Геликоид (минимальная линейчатая поверхность)

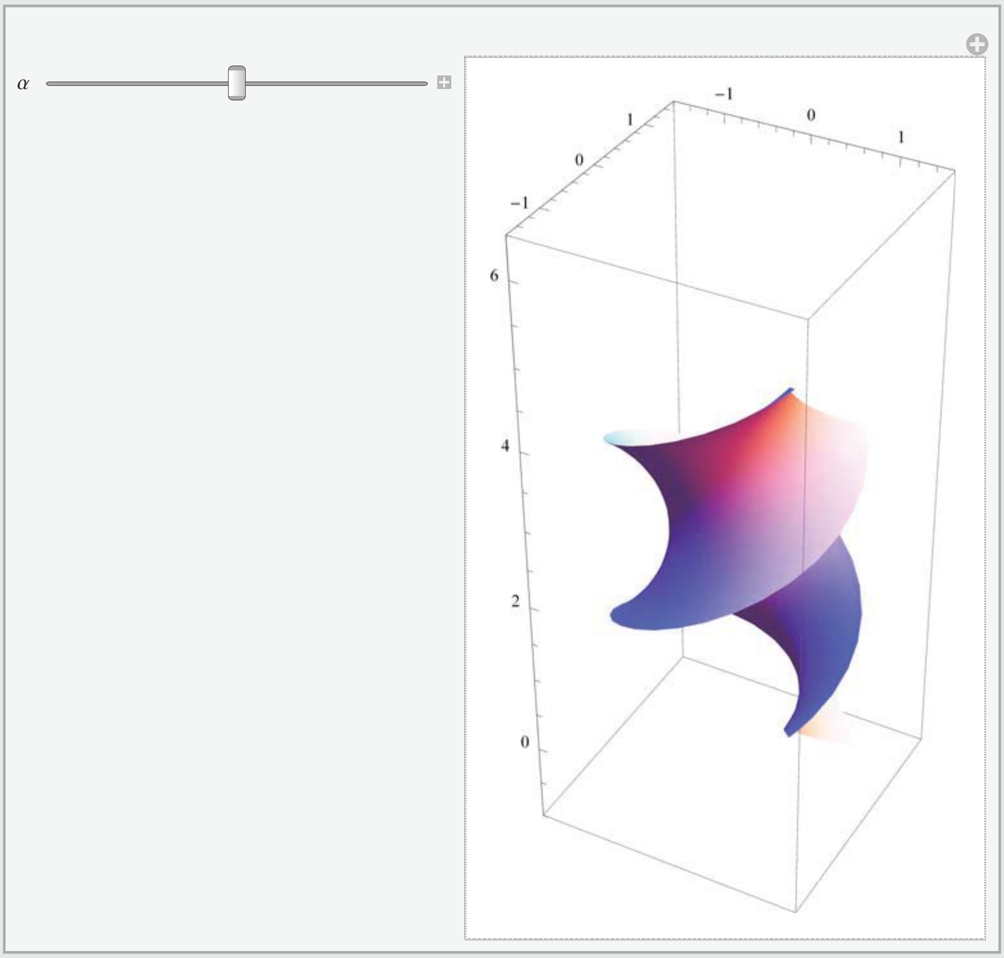
```
ParametricPlot3D[{Sinh[u] Sin[v], -Sinh[u] Cos[v], v}, {u, -1, 1}, {v, 0, 2 π}]
```



Оказывается, что можно так изогнуть один виток геликоида, что он примет форму катеноида с разрезом по меридиану. Более того, все промежуточные поверхности минимальны.

(Здесь PlotRange описывает отрезки изменения координат {x,y,z} точек поверхности; Mesh - координатная сетка на поверхности, в данном случае Mesh → None означает, что сетка отсутствует; PlotPoints - количество точек на каждой координатной кривой: чем больше точек, тем более гладко выглядит поверхность, хотя прорисовка замедляеся; α - параметр изгибания)

```
Manipulate[ParametricPlot3D[Cos[α] {Cosh[u] Cos[v], Cosh[u] Sin[v], u} +
    Sin[α] {Sinh[u] Sin[v], -Sinh[u] Cos[v], v}, {u, -1, 1}, {v, 0, 2 π},
   PlotRange → {{-1.6, 1.6}, {-1.6, 1.6}, {-1, 6.5}}, Mesh → None,
   PlotPoints → 30],
 {α, 0, π/2}]
```



Проведем вычисления, которые покажут, что это изометричная деформация, причем в классе минимальных поверхностей.

```
RR[u_, v_] :=
 Cos[α] {Cosh[u] Cos[v], Cosh[u] Sin[v], u} + Sin[α] {Sinh[u] Sin[v], -Sinh[u] Cos[v], v}
RU[u_, v_] := D[RR[uu, v], uu] /. uu → u;
RV[u_, v_] := D[RR[u, vv], vv] /. vv → v;
RUU[u_, v_] := D[RR[uu, v], uu, uu] /. uu → u;
RUV[u_, v_] := D[RR[uu, vv], uu, vv] /. {uu → u, vv → v};
RVV[u_, v_] := D[RR[u, vv], vv, vv] /. vv → v;
```

Вычисляем первую квадратичную форму.

```
EE[u_, v_] := RU[u, v].RU[u, v] // Simplify
FF[u_, v_] := RU[u, v].RV[u, v] // Simplify
GG[u_, v_] := RV[u, v].RV[u, v] // Simplify
EE[u, v]
FF[u, v]
GG[u, v]
```

$\text{Cosh}[u]^2$

0

$\text{Cosh}[u]^2$

Видно, что она не зависит от параметра изгибания,
значит при всех значениях параметра изгибания получаются изометричные поверхности.

Теперь вычислим нормаль. Обратите внимание на специфику использования команды Simplify при нормировке вектора нормали.

```
nn1[u_, v_] := Normalize[Cross[RU[u, v], RV[u, v]]];
nn1[u, v]
nn2[u_, v_] := Simplify[Normalize[Cross[RU[u, v], RV[u, v]]]];
nn2[u, v]
nn3[u_, v_] := Cross[RU[u, v], RV[u, v]] // Simplify
nn3[u, v]
Sqrt[nn3[u, v].nn3[u, v]] // Simplify
Len[u_, v_] :=
  Simplify[Sqrt[nn3[u, v].nn3[u, v]], Element[u, Reals] && Element[v, Reals]];
Len[u, v]
nn[u_, v_] := nn3[u, v] / Len[u, v];
nn[u, v]
```

```
{(-Cos[v] Cos[α]^2 Cosh[u] - Cos[v] Cosh[u] Sin[α]^2) /
   (√(Abs[-Cos[v] Cos[α]^2 Cosh[u] - Cos[v] Cosh[u] Sin[α]^2]^2 +
        Abs[-Cos[α]^2 Cosh[u] Sin[v] - Cosh[u] Sin[v] Sin[α]^2]^2 +
        Abs[Cos[v]^2 Cos[α]^2 Cosh[u] Sinh[u] + Cos[α]^2 Cosh[u] Sin[v]^2 Sinh[u] +
            Cos[v]^2 Cosh[u] Sin[α]^2 Sinh[u] + Cosh[u] Sin[v]^2 Sin[α]^2 Sinh[u]]^2)),
  (-Cos[α]^2 Cosh[u] Sin[v] - Cosh[u] Sin[v] Sin[α]^2) /
   (√(Abs[-Cos[v] Cos[α]^2 Cosh[u] - Cos[v] Cosh[u] Sin[α]^2]^2 +
        Abs[-Cos[α]^2 Cosh[u] Sin[v] - Cosh[u] Sin[v] Sin[α]^2]^2 +
        Abs[Cos[v]^2 Cos[α]^2 Cosh[u] Sinh[u] + Cos[α]^2 Cosh[u] Sin[v]^2 Sinh[u] +
            Cos[v]^2 Cosh[u] Sin[α]^2 Sinh[u] + Cosh[u] Sin[v]^2 Sin[α]^2 Sinh[u]]^2)),
  (Cos[v]^2 Cos[α]^2 Cosh[u] Sinh[u] + Cos[α]^2 Cosh[u] Sin[v]^2 Sinh[u] +
     Cos[v]^2 Cosh[u] Sin[α]^2 Sinh[u] + Cosh[u] Sin[v]^2 Sin[α]^2 Sinh[u]) /
   (√(Abs[-Cos[v] Cos[α]^2 Cosh[u] - Cos[v] Cosh[u] Sin[α]^2]^2 +
        Abs[-Cos[α]^2 Cosh[u] Sin[v] - Cosh[u] Sin[v] Sin[α]^2]^2 +
        Abs[Cos[v]^2 Cos[α]^2 Cosh[u] Sinh[u] + Cos[α]^2 Cosh[u] Sin[v]^2 Sinh[u] +
            Cos[v]^2 Cosh[u] Sin[α]^2 Sinh[u] + Cosh[u] Sin[v]^2 Sin[α]^2 Sinh[u]]^2))}
```

```
{-Cos[v] Cosh[u] / √(Abs[Cos[v] Cosh[u]]^2 + Abs[Cosh[u] Sin[v]]^2 + Abs[Cosh[u] Sinh[u]]^2),
 -Cosh[u] Sin[v] / √(Abs[Cos[v] Cosh[u]]^2 + Abs[Cosh[u] Sin[v]]^2 + Abs[Cosh[u] Sinh[u]]^2),
 Cosh[u] Sinh[u] / √(Abs[Cos[v] Cosh[u]]^2 + Abs[Cosh[u] Sin[v]]^2 + Abs[Cosh[u] Sinh[u]]^2)}
```

```
{-Cos[v] Cosh[u], -Cosh[u] Sin[v], Cosh[u] Sinh[u]}
```

```
√(Cosh[u]^4)
```

```
Cosh[u]^2
```

```
{-Cos[v] Sech[u], -Sech[u] Sin[v], Tanh[u]}
```

Получили разумное выражение для нормали - самая последняя формула. Все предыдущие - неудачные попытки объяснить *Mathematica*, что мы хотим упростить ответ.
Ищем теперь вторую квадратичную форму.

```
LL[u_, v_] := RUU[u, v].nn[u, v] // Simplify
MM[u_, v_] := RUV[u, v].nn[u, v] // Simplify
NN[u_, v_] := RVV[u, v].nn[u, v] // Simplify
LL[u, v]
MM[u, v]
NN[u, v]
```

-Cos[α]

-Sin[α]

Cos[α]

```
KK[u_, v_] := (LL[u, v] NN[u, v] - MM[u, v]^2) / (EE[u, v] GG[u, v] - FF[u, v]^2 )
HH [u_, v_] := (EE[u, v] NN[u, v] - 2 FF[u, v] MM [u, v] + GG[u, v] LL[u, v]) /
  (EE[u, v] GG[u, v] - FF[u, v]^2)
KK[u, v] // FullSimplify
HH[u, v]
```

-Sech[u]^4

0

Локсодромия

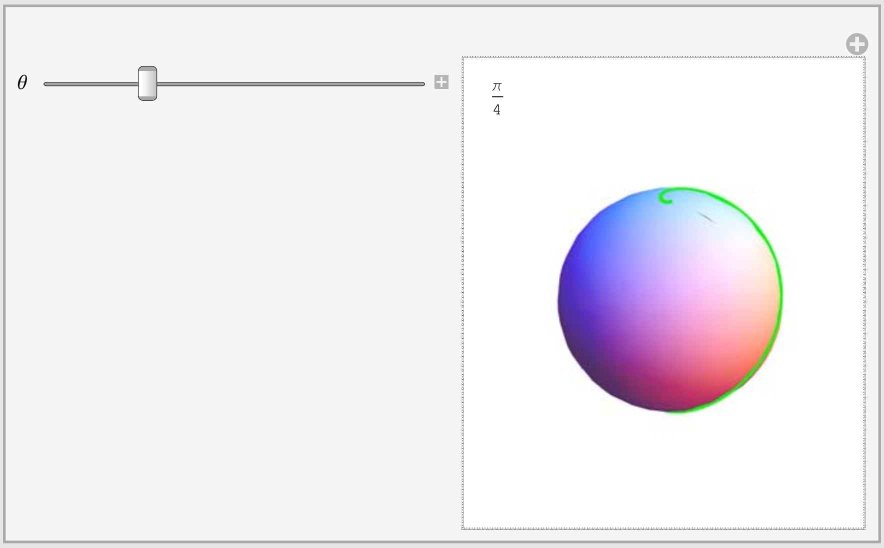
In[1]:=

```
angle[r_, γ_][t_] := Module[{u, v, ru, rv},
  ru = D[r[u, v], u];
  rv = D[r[u, v], v];
  g = ( ru.ru  ru.rv
        ru.rv  rv.rv );
  ArcCos[ γ'[t].g.{1, 0} / (Sqrt[γ'[t].g.γ'[t]] Sqrt[{1, 0}.g.{1, 0}]) ] /.
    {u → γ[t][[1]], v → γ[t][[2]]} // FullSimplify
 ] (*угол между кривой и второй координатной линией*)
```

In[2]:=

```
Manipulate[sphere[u_, v_] := {Sin[u] Cos[v], Sin[u] Sin[v], Cos[u]};
 lok[t_] := {t, Tan[θ] Log[Tan[π/2 - t/2]]}; (*уравнение локсодромии на сфере*)
 loksph[t_] := sphere[u, v] /. {u → lok[t][[1]], v → lok[t][[2]]};
 Column[{angle[sphere, lok][t], Show[{ParametricPlot3D[sphere[u, v],
      {u, 0, π}, {v, 0, 2 π}, Boxed → False, Mesh → None, Axes → None],
     ParametricPlot3D[loksph[t], {t, 0, π}, PlotStyle → {Green, Thick}]
    }]}],
 {{θ, π/4}, 0, π}]
```

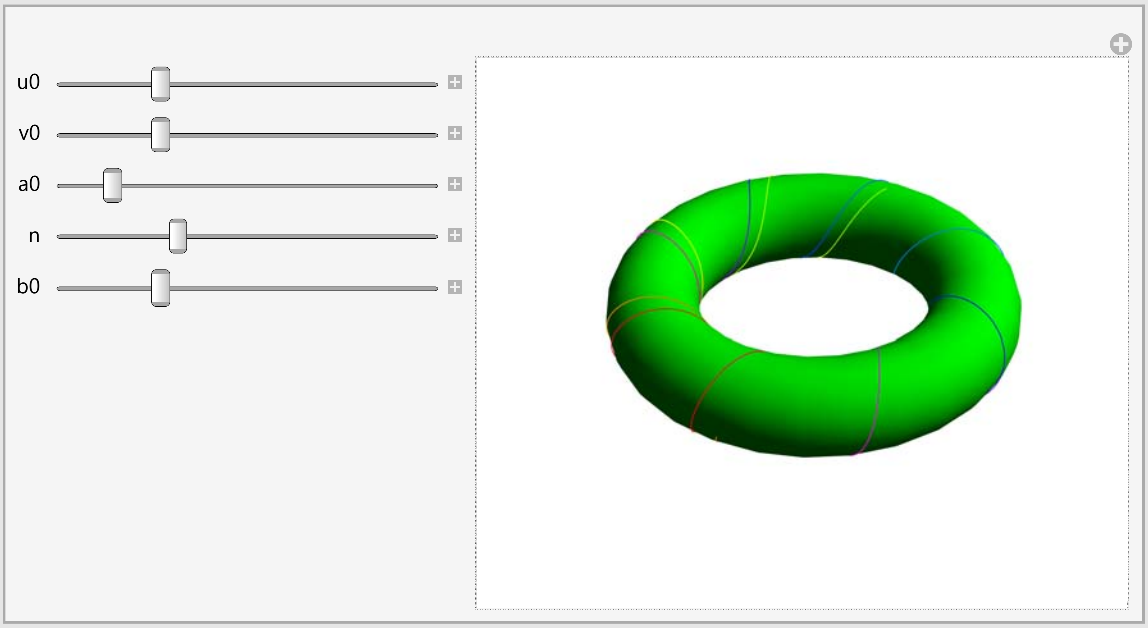
Out[2]=



In[3]:=

```
Clear[T, H, G, a, b, λ, θ, e1, e2, r, dr, sol]
Manipulate[

 G[u_, v_, a_, b_] := {(a + b Cos[u]) Cos[v], (a + b Cos[u]) Sin[v], b Sin[u]};
 GaussianCurvatureOfTor[a_, b_, λ_, θ_] := Cos[λ] / (b (a + b Cos[λ]));

 e1[u_, v_, a_, b_] := {-b Sin[u[t]] Cos[v[t]], -b Sin[u[t]] Sin[v[t]], b Cos[u[t]]};
 e2[u_, v_, a_, b_] := {- (a + b Cos[u[t]]) Sin[v[t]], (a + b Cos[u[t]]) Cos[v[t]], 0};
r[u_, v_, a_, b_] :=
   {(a + b Cos[u[t]]) Cos[v[t]], (a + b Cos[u[t]]) Sin[v[t]], b Sin[u[t]]};
dr[u_, v_, a_, b_] := D[r[u, v, a, b], t];

sol = NDSolve[{dr[u, v, a0, b0].e2[u, v, a0, b0] == Sin[n],
     (*угол между локсодромой и меридианом, а также параллелью*)
     dr[u, v, a0, b0].e1[u, v, a0, b0] == Cos[n], u[0] == u0, v[0] == v0}
     (*начальная точка*), {u, v}, {t, 0, 100}];

 Show[{
   ParametricPlot3D[G[u, v, a0, b0],
    {u, 0, 2 * Pi}, {v, 0, 2 * Pi}, Mesh → None, PlotStyle → Green],
   ParametricPlot3D[G[u, v, a0, b0] /. {u → u[t], v → v[t]} /. sol[[1]], {t, 0, 5},
    ColorFunctionScaling → False, ColorFunction → Function[{x, y, z, u}, Hue[0.3 * a0 *
          b0 * GaussianCurvatureOfTor[a0, b0, u, v] /. {u → u[t], v → v[t]} /. sol[[1]]]]
  },
  Boxed → False, Axes → False
 ],
 {{u0, Pi / 2}, 0, 2 Pi, Pi / 10}, {{v0, Pi / 2}, 0, 2 Pi, Pi / 10},
 {{a0, 0.7}, 0.6, 1.5, 0.1}, {{n, 3 Pi / 10}, 0, Pi, Pi / 10}, {{b0, 0.2}, 0.1, 0.5, 0.05}
]
```

Out[4]=

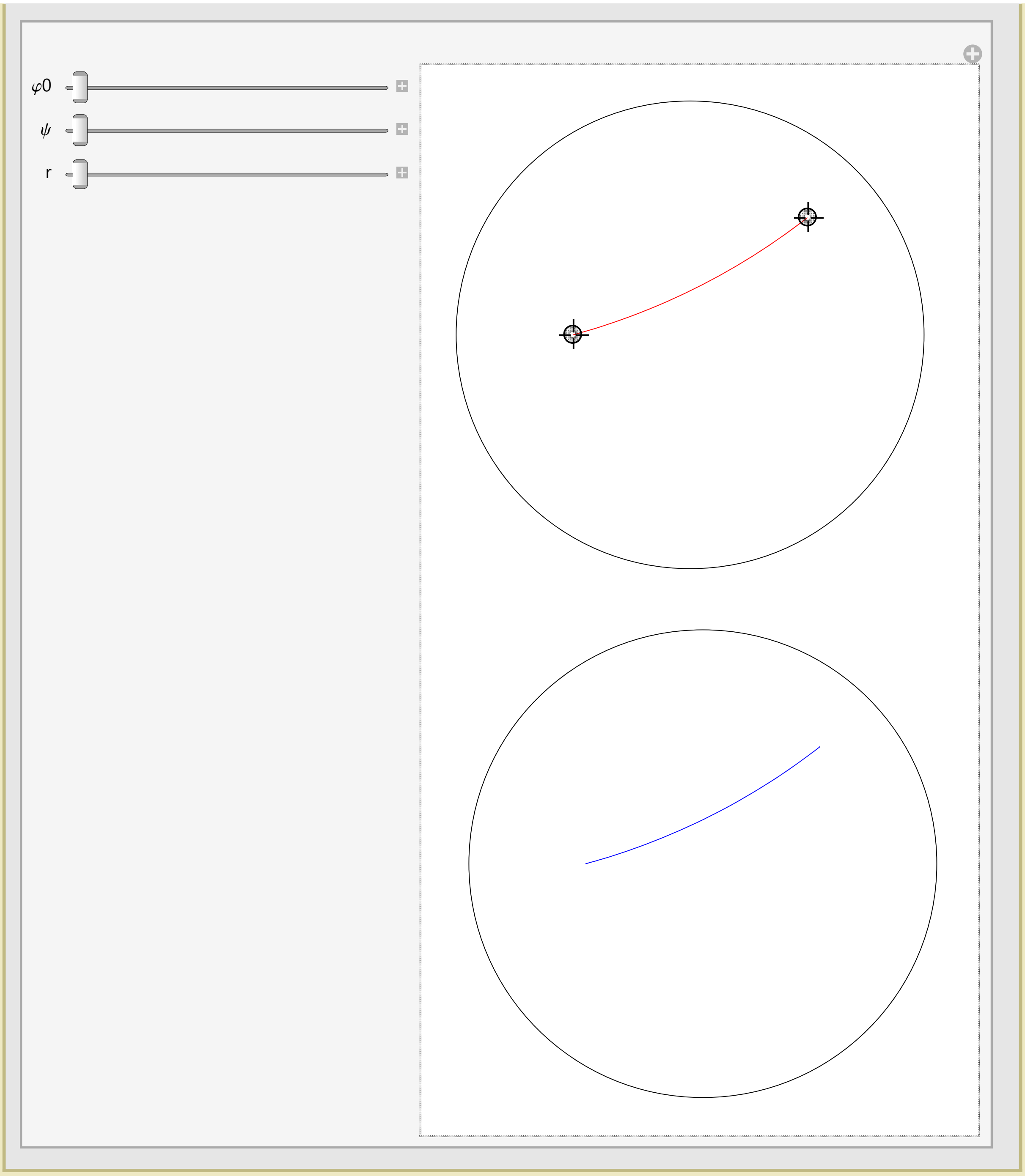


Движения в модели единичного круга и в модели верхней полуплоскости

In[1]:=

```
Manipulate[g[z_, φ_, z00_] := e^(i φ) (z - z00)/(1 - z Conjugate[z00]);
ginv[z_, φ_, z0_] := e^(-i φ) (z + e^(i φ) z0)/(1 + z e^(-i φ) Conjugate[z0]);
 line[z1_, z2_, t_] := ginv[t g[z2, 0, z1], 0, z1];
 CirPoly[p_, Clr_] := Module[{},
   Show[{
     Graphics[Circle[]],
     Table[ParametricPlot[With[{z = line[Complex[Sequence @@ p[[i]]],
            Complex[Sequence @@ p[[If[i == Length[p], 1, i + 1]]]], t]},
         {Re[z], Im[z]}], {t, 0, 1}, PlotStyle → Directive[Clr]],
       {i, If[Length[p] == 2, 1, Length[p]]}]
     }, ImageSize → {300, 300}]
  ];
 z0 = r (Cos[ψ] + i Sin[ψ]);
 gtable[p_, φ_, z00_] := Table[{Re[g[Complex[p[[i]][[1]], p[[i]][[2]]], φ, z00]],
    Im[g[Complex[p[[i]][[1]], p[[i]][[2]]], φ, z00]]}, {i, 1, Length[p]}];
 Row[{CirPoly[p, Red], CirPoly[gtable[p, φ0, z0], Blue]
  }],
 {φ0, 0, 2 π}, {ψ, 0, 2 π}, {r, 0, 0.999},
 {{p, {{-1 / 2, 0}, {1 / 2, 1 / 2}}}, Locator, LocatorAutoCreate → True}]
```

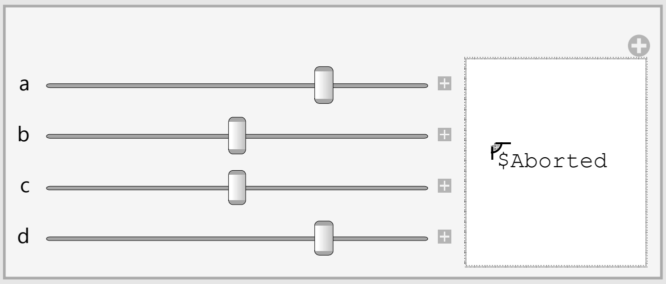
Out[1]=



In[2]:=

```
Manipulate[h[z_, {a_, b_, c_, d_}] := If[a d - b c > 0, (a z + b)/(c z + d), (-a z - b)/(c z + d)];
 linepl[z1_, z2_, t_] := Module[{p1, p2, med, hp1, hp2, cent, r, hp, s, tt},
   p1 = {Re[z1], Im[z1]};
   p2 = {Re[z2], Im[z2]};
   med = (p1 + p2)/2;
   intpt = Solve[med〚2〛 + s (p1〚1〛 - p2〚1〛) == 0, s, Reals];
   s0 = If[Length[intpt] == 1, s /. intpt〚1〛, ⅇ^π]; (*если точки лежат на
    одной вертикальной прямой, то присвоим s0 значение, например, ⅇ^π *)
   cent = {med〚1〛 + s0 (p2〚2〛 - p1〚2〛), 0}; (*центр окружности-прямой*)
   r = Sqrt[(p2 - cent).(p2 - cent)] ; (*радиус окружности-прямой*)
   hp1 = (z1 - (cent〚1〛 + r))/(z1 - (cent〚1〛 - r)) // FullSimplify;
   (* используем преобразование, переводящее окружность,
   проходящую через точки z1 и z2, в вертикальную прямую*)
   hp2 = (z2 - (cent〚1〛 + r))/(z2 - (cent〚1〛 - r)) // FullSimplify;
   hp[tt_] := (-(hp1 + tt (hp2 - hp1)) (cent〚1〛 - r) + (cent〚1〛 + r))/(1 - (hp1 + tt (hp2 - hp1)));
   (*обратное преобразование*)
   If[Length[intpt] == 1, {Re[hp[t]], Im[hp[t]]}, p1 + t (p2 - p1)]
  ];
 PlPoly[p_, Clr_] := Module[{},
   Show[{
     Table[ParametricPlot[With[{z = linepl[Complex[Sequence @@ p〚i〛],
            Complex[Sequence @@ p〚If[i == Length[p], 1, i + 1]〛], t]},
         {Re[z], Im[z]}], {t, 0, 1}, PlotStyle → Directive[Clr]],
       {i, If[Length[p] == 2, 1, Length[p]]}]
    }, ImageSize → {300, 300}, PlotRange → {{-5, 5}, {0, 5}}]
  ];
 gtable[p_, {a_, b_, c_, d_}] := Table[{Re[h[Complex[p〚i〛〚1〛, p〚i〛〚2〛], {a, b, c, d}]],
    Im[h[Complex[p〚i〛〚1〛, p〚i〛〚2〛], {a, b, c, d}]]}, {i, 1, Length[p]}];
 Row[{PlPoly[p, Red], PlPoly[gtable[p, {a, b, c, d}], Blue]}],
 {{a, 1}, -2, 2, 0.1}, {{b, 0}, -1, 1, 0.1}, {{c, 0}, -1, 1, 0.1},
 {{d, 1}, -2, 2, 0.1}, {{p, {{0, 1}, {2, 1}}}, Locator, LocatorAutoCreate → True}]
```

Out[2]=



Скрученная полоса

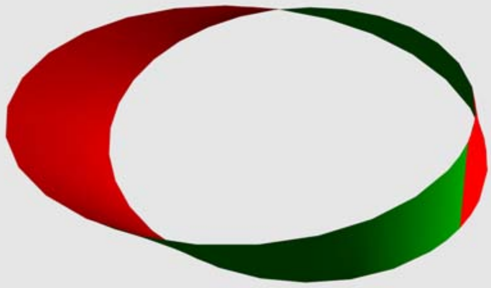
In[3]:=

```
(*Визуализируем скрученную полосу*)
k = 0;
Row[{"Выберите степень скрученности:", Setter[Dynamic[k], 0, "0"],
   Setter[Dynamic[k], 1, "1"], Setter[Dynamic[k], 2, "2"],
   Setter[Dynamic[k], 3, "3"], Setter[Dynamic[k], 4, "4"]}, " "]
Dynamic[x = Cos[2 * fi * Pi] * (1 - teta * Sin[fi * k * Pi]);
 y = Sin[2 * fi * Pi] * (1 - teta * Sin[fi * k * Pi]);
 z = teta * Cos[fi * k * Pi];
 ParametricPlot3D[{x, y, z}, {fi, 0, 1}, {teta, -0.25, 0.25}, Mesh → None,
  Boxed → False, Axes → False, PlotStyle → {FaceForm[Red, Green]}]]
```

Out[4]=

Выберите степень скрученности: 0 1 2 3 4

Out[5]=



In[6]:=

```
(*Визуализируем кратное разрезание*)
W = 0.25; H = 0.3; mode = 0; k = 0; tau = 0;
Row[{"Выберите степень скрученности:", Setter[Dynamic[k], 0, "0"],
   Setter[Dynamic[k], 1, "1"], Setter[Dynamic[k], 2, "2"],
   Setter[Dynamic[k], 3, "3"], Setter[Dynamic[k], 4, "4"]}, " "]
Row[{"Степень разрезания:", Slider[Dynamic[mode], {0, 2, 0.1}]}, " "]

x = Cos[2 * fi * Pi] * (1 - teta * Sin[skr * fi * Pi]);
y = Sin[2 * fi * Pi] * (1 - teta * Sin[skr * fi * Pi]);
z = teta * Cos[skr * fi * Pi];
v2 = Normalize[D[{x, y, z}, teta]] * W;
v1 = Normalize[D[{x, y, z}, fi]] * W;
n = Cross[v1, v2] * H / (W^2);

Dynamic[ src = {x, y, z} /. {skr → k};
 v20 = If[mode > 0, Min[mode, 1] * (v2 /. {tau → 0, skr → k}), {0, 0, 0}];
 v21 = If[mode > 1, (mode - 1) * (v2 /. {tau → W / 2, skr → k}), {0, 0, 0}];
 v22 = If[mode > 1, (mode - 1) * (v2 /. {tau → -W / 2, skr → k}), {0, 0, 0}];
 Show[{ParametricPlot3D[src + v20 / 2, {fi, 0, 1}, {teta, 0, W / 2}, Mesh → None,
     PlotStyle → {FaceForm[Green, Red]}], ParametricPlot3D[src - v20 / 2,
     {fi, 0, 1}, {teta, -W / 2, 0}, Mesh → None, PlotStyle → {FaceForm[Green, Red]}],
    ParametricPlot3D[src + v20 / 2 + v21, {fi, 0, 1}, {teta, W / 2, W}, Mesh → None,
     PlotStyle → {FaceForm[Green, Red]}], ParametricPlot3D[src - v20 / 2 - v22,
     {fi, 0, 1}, {teta, -W, -W / 2}, Mesh → None, PlotStyle → {FaceForm[Green, Red]}]},
   PlotRange → All, Boxed → False, Axes → False]]
```

Out[7]=

Выберите степень скрученности: 0 1 2 3 4

Out[8]=

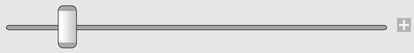
Степень разрезания:

Out[15]=

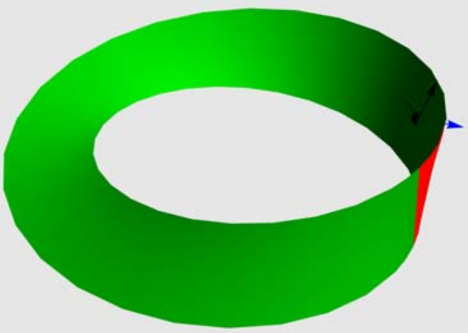
In[16]:=
```
(*Визуализируем тест на ориентируемость*)
W = 0.25; H = 0.3; k = 0; tau = 0;
Row[{"Выберите степень скрученности:", Setter[Dynamic[k], 0, "0"],
   Setter[Dynamic[k], 1, "1"], Setter[Dynamic[k], 2, "2"],
   Setter[Dynamic[k], 3, "3"], Setter[Dynamic[k], 4, "4"]}, " "]
Manipulator[Dynamic[tau], {0, 1}]
Dynamic[x = Cos[2 * fi * Pi] * (1 + teta * Sin[fi * k * Pi]);
 y = Sin[2 * fi * Pi] * (1 + teta * Sin[fi * k * Pi]);
 z = teta * Cos[fi * k * Pi];
 P0 = {x, y, z} /. {teta → 0, fi → tau};
 v1 = {-Sin[2 * Pi * tau], Cos[2 * Pi * tau], 0};
 v2 = {Cos[2 * Pi * tau] * Sin[k * Pi * tau],
    Sin[2 * Pi * tau] * Sin[k * Pi * tau], Cos[k * Pi * tau]};
 v1 = Normalize[v1] * W;
 v2 = Normalize[v2] * W;
 n = Cross[v1, v2] * H / (W^2);
 Show[ParametricPlot3D[{x, y, z}, {fi, 0, 1}, {teta, -W, W},
    Mesh → None, PlotRange → {{-1 - H, 1 + H}, {-1 - H, 1 + H}, {-H, H}},
    Boxed → False, Axes → False, PlotStyle → {FaceForm[Red, Green]}],
   Graphics3D[{PointSize[W / 12], Arrowheads[W / 10], Black, Point[P0],
     Arrow[{P0, P0 + v1}], Arrow[{P0, P0 + v2}], Blue, Arrow[{P0, P0 + n}]}]]]
```

Out[17]= Выберите степень скрученности: 0 1 2 3 4

Out[18]=



Out[19]=



КОНЕЦ ЧЕТВЕРТОЙ ЛЕКЦИИ