Homework 3

Section I: Synthesis Problem #1

1.1. Find optimal time

In[393]:=

```
Clear[μ, ν, x1, x2, θ]
bereinige
μ = -3; ν = 6;
x1 = 0.5; x2 = 0.5;
s = NSolve[{2 μ θ - 1 + Exp[-2 μ θ] == 2 μ^2 (x1^2 + x2^2) && θ ≥ 0}, θ];
    löse numerisch          Exponentialfunktion
T = s[[1]][[1]][[2]];
Print["Optimal time is ", T]
gib aus
```

```
Optimal time is 0.421327
```

1.2. Find function $\theta(x_1, x_2)$

Since solving the equation explicitly is fairly difficult, we do the following instead:

1. Define the grid of values of $\{x_1{}^{(i)}, x_2{}^{(i)}\}_i$.

2. Interpolate the polynomial through this set of points.

In[399]:=

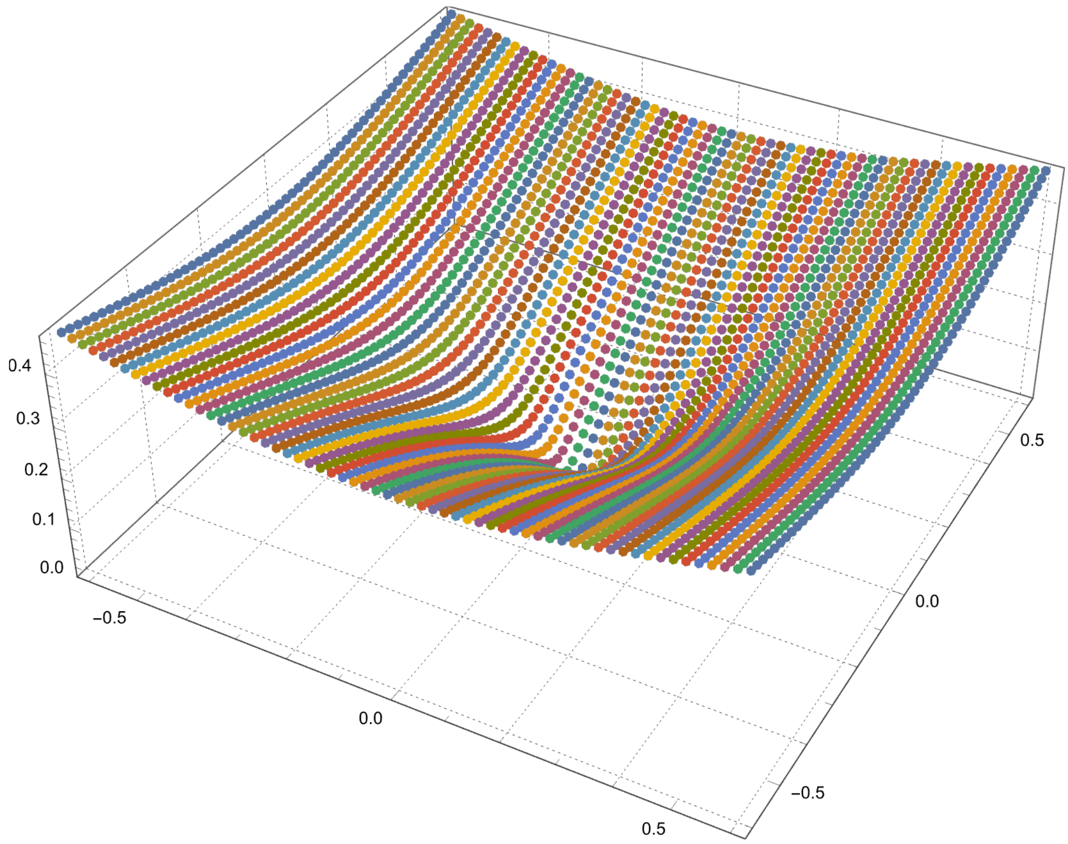
```
Clear[x1, x2]
bereinige
t = Table[
    Tabelle
    {x1, x2, NSolve[{2 μ θ - 1 + Exp[-2 μ θ] == 2 μ^2 (x1^2 + x2^2) && θ ≥ 0}, θ][[1]][[1]][[2]]},
                löse numerisch          Exponentialfunktion
    {x1, -0.6, 0.6, 0.02},
    {x2, -0.6, 0.6, 0.02}
   ];
```

In[402]:=

```
ListPointPlot3D[t,
  PlotTheme → "Detailed",
  PlotStyle → Directive[PointSize[0.01]],
  ImageSize → 550]
```

listenbezogenes 3D-Streudiagramm
Thema der graphischen Darstellung
Darstellungsstil | Anweisung | Punktgröße
Bildgröße

Out[402]=



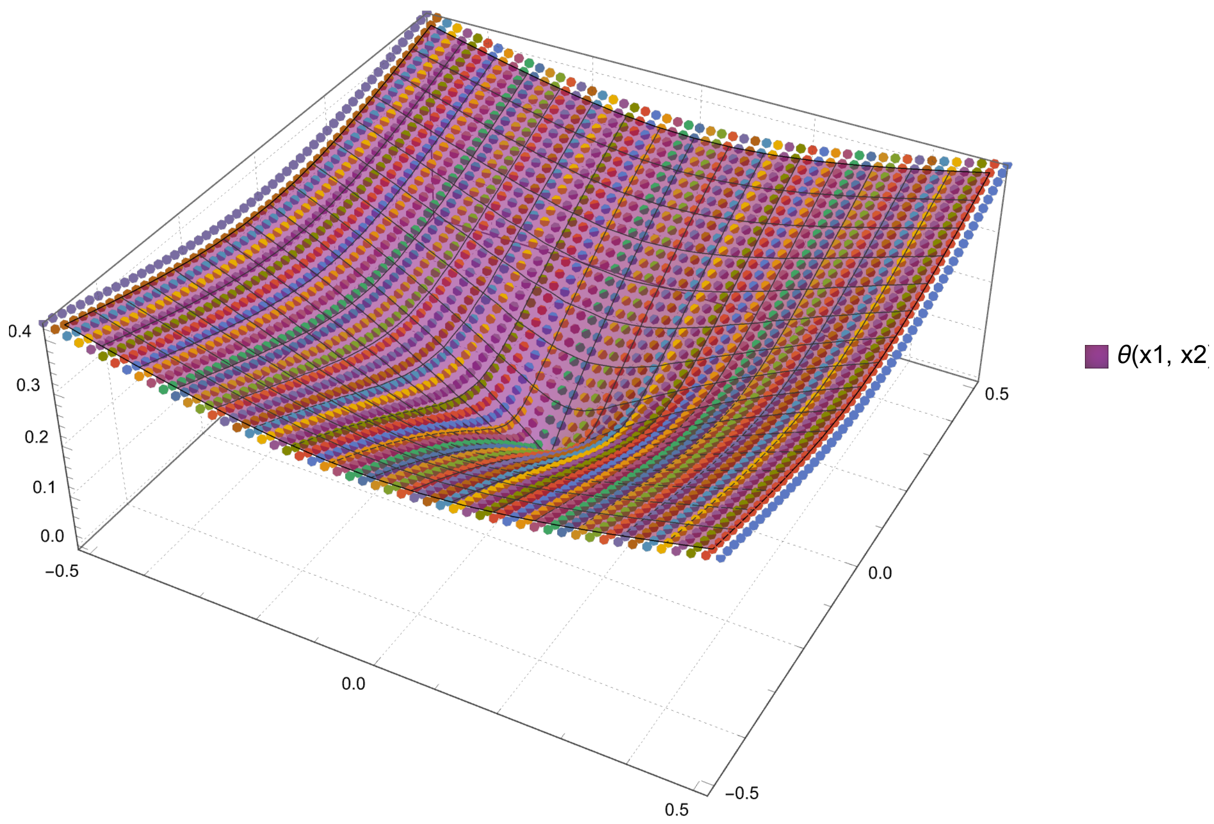
Now, flatten the array and conduct the interpolation operation.

In[403]:=

```
ft = Flatten[t];
θ[x1_, x2_] = Interpolation[
   Table[{{ft⟦i⟧, ft⟦i + 1⟧}, ft⟦i + 2⟧}, {i, 1, Length[ft], 3}], {x1, x2}];
Show[
 Plot3D[θ[x1, x2], {x1, -0.5, 0.5}, {x2, -0.5, 0.5},
  PlotTheme → "Detailed", PlotStyle → Directive[Purple, Opacity[0.5]]],
 ListPointPlot3D[t, PlotStyle → Directive[PointSize[0.01]]],
 ImageSize → 550
]
```

ebne ein · Interpolation · Tabelle · Länge · zeige an · stelle Funktion graphisch in 3D dar · Thema der graphischen Darstell··· · Darstellungsstil · Anweisung · lila · Deckkraft · listenbezogenes 3D-Streu·· · Darstellungsstil · Anweisung · Punktgröße · Bildgröße

Out[405]=



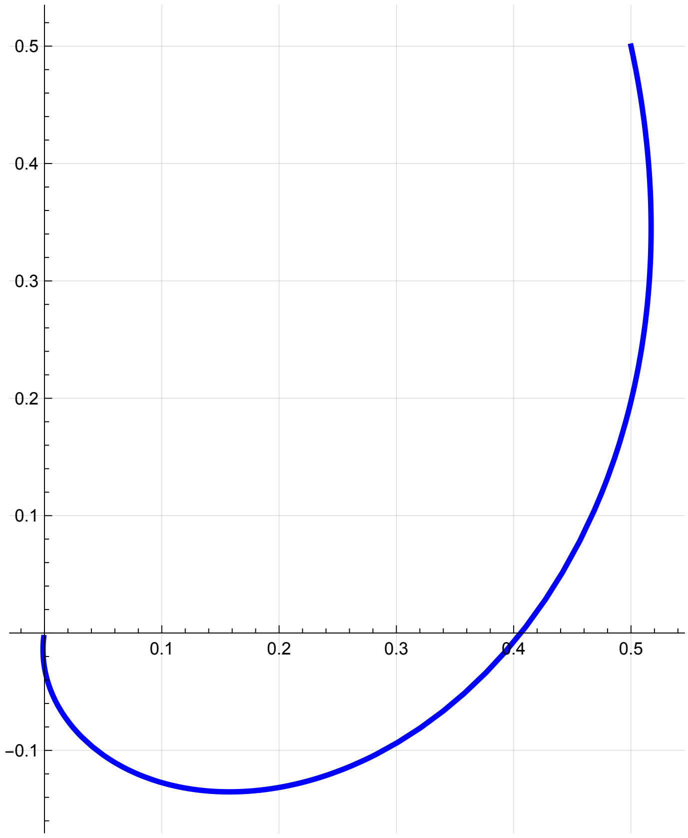
1.3. Find the trajectory

First, define the controlling function

In[423]:=

```
Clear[x1, x2, t]
⌊bereinige
u1[x1_, x2_] = -θ[x1, x2] x1 / (x1^2 + x2^2);
u2[x1_, x2_] = -θ[x1, x2] x2 / (x1^2 + x2^2);
s = NDSolve[{x1'[t] == μ x1[t] + ν x2[t] + u1[x1[t], x2[t]],
    ⌊löse Differentialgleichung numerisch
     x2'[t] == -ν x1[t] + μ x2[t] + u2[x1[t], x2[t]],
     x1[0] == 0.5, x2[0] == 0.5},
    {x1[t], x2[t]}, {t, T}];
Show[
⌊zeige an
 ParametricPlot[Evaluate[{x1[t], x2[t]} /. s], {t, 0, T},
 ⌊parametrische Darste· ⌊werte aus
  GridLines → Automatic,
  ⌊Gitternetzlinien ⌊automatisch
  PlotStyle → Directive[Blue, Thickness[0.008]]]
  ⌊Darstellungsstil ⌊Anweisung ⌊blau ⌊Dicke
]
```

Out[427]=

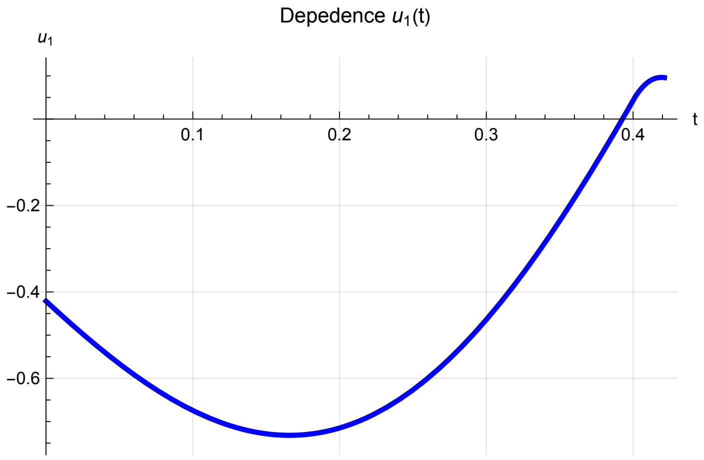


Additionally, building the control functions $u_1(t)$ and $u_2(t)$

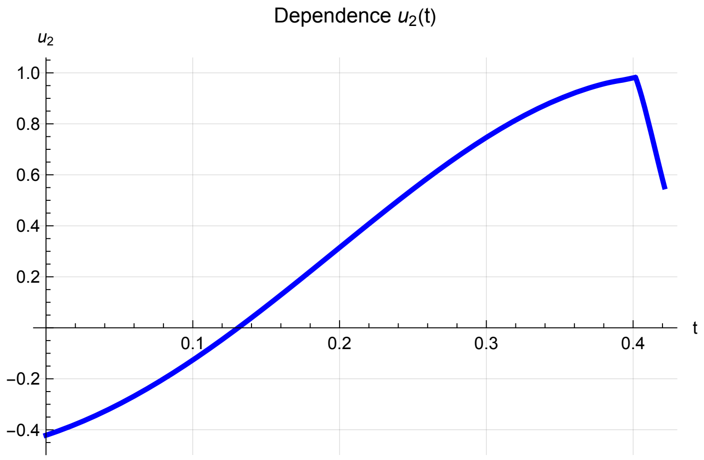
In[428]:=

```
Show[
zeige an
 Plot[Evaluate[u1[x1[t], x2[t]] /. s], {t, 0, T},
 stell··· werte aus
  GridLines → Automatic,
  Gitternetzlinien automatisch
  PlotStyle → Directive[Blue, Thickness[0.008]]],
  Darstellungsstil Anweisung blau Dicke
 PlotLabel → "Depedence u₁(t)",
 Beschriftung der Graphik
 AxesLabel → {"t", "u₁"}
 Achsenbeschriftungen
]
Show[
zeige an
 Plot[Evaluate[u2[x1[t], x2[t]] /. s], {t, 0, T},
 stell··· werte aus
  GridLines → Automatic,
  Gitternetzlinien automatisch
  PlotStyle → Directive[Blue, Thickness[0.008]]],
  Darstellungsstil Anweisung blau Dicke
 PlotLabel → "Dependence u₂(t)",
 Beschriftung der Graphik
 AxesLabel → {"t", "u₂"}
 Achsenbeschriftungen
]
Show[
zeige an
 Plot[Evaluate[{u1[x1[t], x2[t]]^2 + u2[x1[t], x2[t]]^2} /. s], {t, 0, T},
 stell··· werte aus
  GridLines → Automatic,
  Gitternetzlinien automatisch
  PlotStyle → Directive[Blue, Thickness[0.008]]],
  Darstellungsstil Anweisung blau Dicke
 PlotLabel → "Dependence u₁^2(t)+u₂^2(t)",
 Beschriftung der Graphik
 AxesLabel → {"t", "u₁^2(t)+u₂^2(t)"}
 Achsenbeschriftungen
]
```

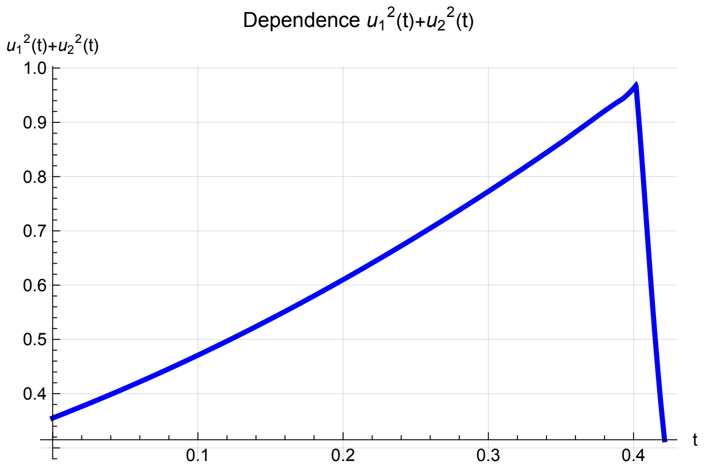
Out[428]=



Out[429]=



Out[430]=



Notice: Notice that indeed $u_1^2 + u_2^2 \leq 1$ for all t.

Section II: Synthesis Problem #2

2.1. Region plotting

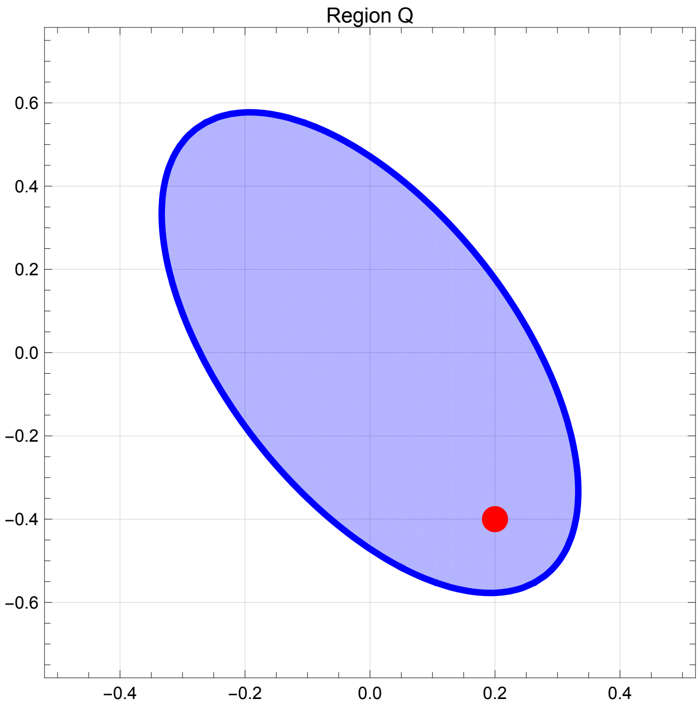
We need to plot the region $3\,x_1^2 + 2\,x_1\,x_2 + x_2^2 \le 2/9$ and pick a random point inside.

In[442]:=

```
x10 = 0.2; x20 = -0.4;
Show[
 RegionPlot[{3 x1^2 + 2 x1 x2 + x2^2 ≤ 2/9}, {x1, -0.5, 0.5}, {x2, -0.75, 0.75},
  GridLines → Automatic,
  PlotStyle → Directive[Blue, Opacity[0.3]],
  BoundaryStyle → Directive[Blue, Thickness[0.01]]],
 ListPlot[{{x10, x20}}, PlotStyle → Directive[Red, PointSize[0.04]]],
 PlotLabel → "Region Q"
]
```

zeige an | graphische Darstellung einer Region | Gitternetzlinien | automatisch | Darstellungsstil | Anweisung | blau | Deckkraft | Stil der Randregion | Anweisung | blau | Dicke | listenbezogene Graphik | Darstellungsstil | Anweisung | rot | Punktgröße | Beschriftung de··· | Region

Out[443]=



2.2. Finding θ at given point

In[444]:=

```
s = NSolve[{2/9 θ^4 - θ^2 x20^2 - 2 θ x10 x20 - 3 x10^2 == 0, θ ≥ 0}, θ];
  löse numerisch

θ0 = s[[1]][[1]][[2]];
Print["The value of θ at given point is ", θ0]
gib aus
```

```
The value of θ at given point is 0.80924
```

2.3. Solving Differential Equation

Note: We adjust the parameter *T* to make the destination point at distance less than 0.01 to the center. Solve the equation first.

In[447]:=

```
Clear[t]
bereinige
ϵ = 0.005; δ = 0.01; T = 1.74;
s = NDSolve[{x1'[t] == x2[t] + ϵ (-x1[t]^2/x3[t]^2 - 2 x1[t] × x2[t]/x3[t] + x2[t]^2),
  löse Differentialgleichung numerisch
    x2'[t] == -x1[t]/x3[t]^2 - 2 x2[t]/x3[t] + δ (x1[t] × x2[t]/x3[t]^2 + 2 x2[t]^2/x3[t] + x1[t]^2),
    x3'[t] ==
     -(x1[t]^2 + x2[t]^2 x3[t]^2)/(6 x1[t]^2 + 3 x1[t] × x2[t] × x3[t] + x3[t]^2 x2[t]^2) + ((-3 + x3[t]^3) x1[t]^3 +
        x3[t] (-8 + x3[t]^3) x1[t]^2 x2[t] - 2 x3[t]^2 x1[t] x2[t]^2 - x3[t]^3 x2[t]^3) /
      (100 x3[t] (6 x1[t]^2 + 3 x1[t] × x2[t] × x3[t] + x3[t]^2 x2[t]^2)),
    x1[0] == x10, x2[0] == x20, x3[0] == θ0}, {x1[t], x2[t], x3[t]}, {t, T}];
Evaluate[{x1[t]^2 + x2[t]^2 ≤ 0.01^2} /. s] /. {t → T}
werte aus
```

Out[450]=

```
{{True}}
```

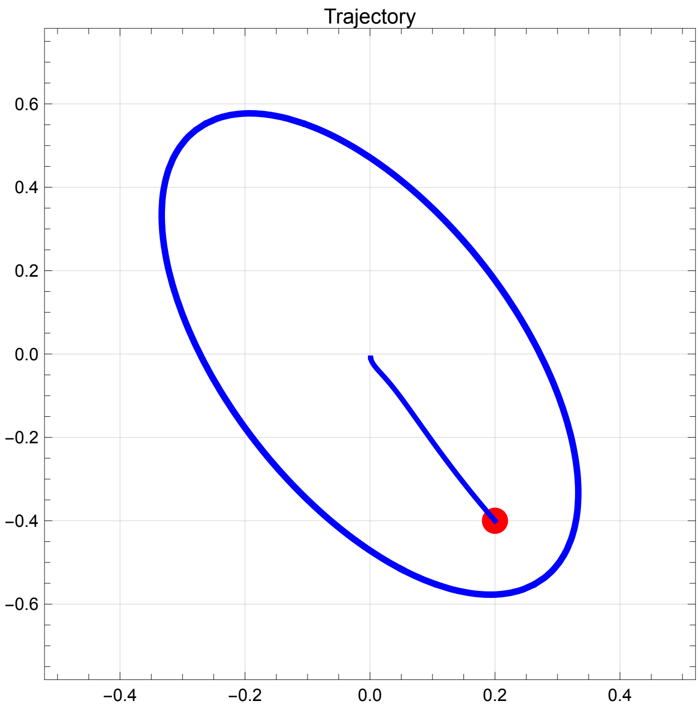
Now, building the trajectory

In[451]:=

```
Show[
 RegionPlot[{3 x1^2 + 2 x1 x2 + x2^2 ≤ 2/9}, {x1, -0.5, 0.5}, {x2, -0.75, 0.75},
  GridLines → Automatic,
  PlotStyle → Directive[Opacity[0.0]],
  BoundaryStyle → Directive[Blue, Thickness[0.01]]],
 ListPlot[{{x10, x20}}, PlotStyle → Directive[Red, PointSize[0.04]]],
 ParametricPlot[Evaluate[{x1[t], x2[t]} /. s], {t, 0, T},
  GridLines → Automatic,
  PlotStyle → Directive[Blue, Thickness[0.008]]],
 PlotLabel → "Trajectory"
]
```

zeige an | graphische Darstellung einer Region | Gitternetzlinien | automatisch | Darstellungsstil | Anweisung | Deckkraft | Stil der Randregion | Anweisung | blau | Dicke | listenbezogene Graphik | Darstellungsstil | Anweisung | rot | Punktgröße | parametrische Darste· | werte aus | Gitternetzlinien | automatisch | Darstellungsstil | Anweisung | blau | Dicke | Beschriftung der Graphik

Out[451]=



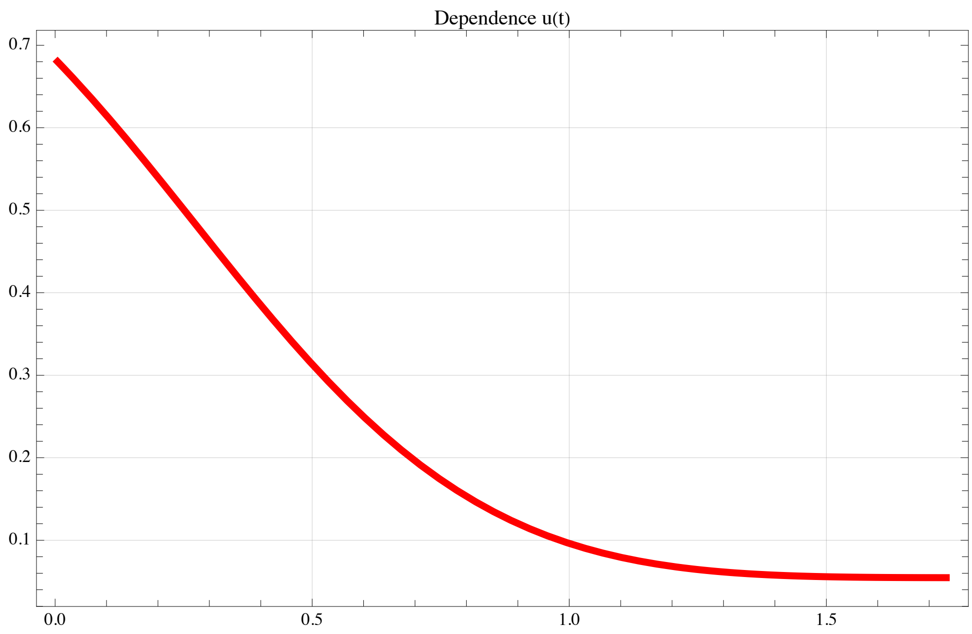
Plot the control function and controlability derivative:

In[457]:=

```
Plot[Evaluate[(-x1[t] / x3[t]^2 - 2 x2[t] / x3[t]) /. s],
 {t, 0.0, T}, PlotRange → All, PlotTheme → "Scientific",
 PlotStyle → Directive[Red, Thickness[0.0075]], GridLines → Automatic,
 ImageSize → 500, PlotLabel → "Dependence u(t)", AxesLabel → {"t", "u(t)"}]
```

stell··· werte aus Koordinatenbe·· alle Thema der graphischen Darstellung Darstellungsstil Anweisung rot Dicke Gitternetzlinien automatisch Bildgröße Beschriftung der Graphik Achsenbeschriftungen

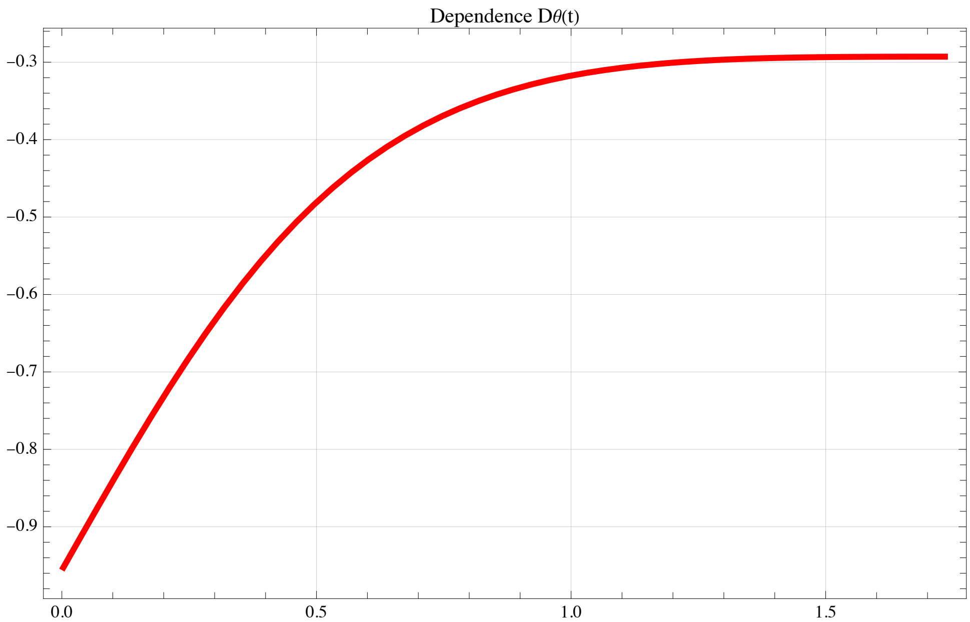
Out[457]=



In[460]:=

```
θd[x1_, x2_, x3_] = - (x1^2 + x2^2 x3^2)/(6 x1^2 + 3 x1 x2 x3 + x3^2 x2^2) +
    ((-3 + x3^3) x1^3 + x3 (-8 + x3^3) x1^2 x2 - 2 x3^2 x1 x2^2 - x3^3 x2^3)/(100 x3 (6 x1^2 + 3 x1 x2 x3 + x3^2 x2^2));
Plot[Evaluate[θd[x1[t], x2[t], x3[t]] /. s],
⌊stell··· ⌊werte aus
 {t, 0.0, T}, PlotRange → All, PlotTheme → "Scientific",
              ⌊Koordinatenbe·· ⌊alle ⌊Thema der graphischen Darstellung
 PlotStyle → Directive[Red, Thickness[0.0065]], GridLines → Automatic,
 ⌊Darstellungsstil ⌊Anweisung ⌊rot ⌊Dicke ⌊Gitternetzlinien ⌊automatisch
 ImageSize → 500, PlotLabel → "Dependence Dθ(t)", AxesLabel → {"t", "Dθ(t)"}]
 ⌊Bildgröße ⌊Beschriftung der Graphik ⌊Achsenbeschriftungen
```

Out[461]=



2.4. Trajectory, Control Function, Derivative

Since I am not totally sure about the trajectory and control function from the last subsection, I decided to do everything from scratch, using control function $u(x) = -\frac{x_1}{\theta^2(x_1,x_2)} - \frac{2\,x_2}{\theta(x_1,x_2)}$.

First, approximating the function $\theta(x_1, x_2)$, as with the first question.

In[235]:=

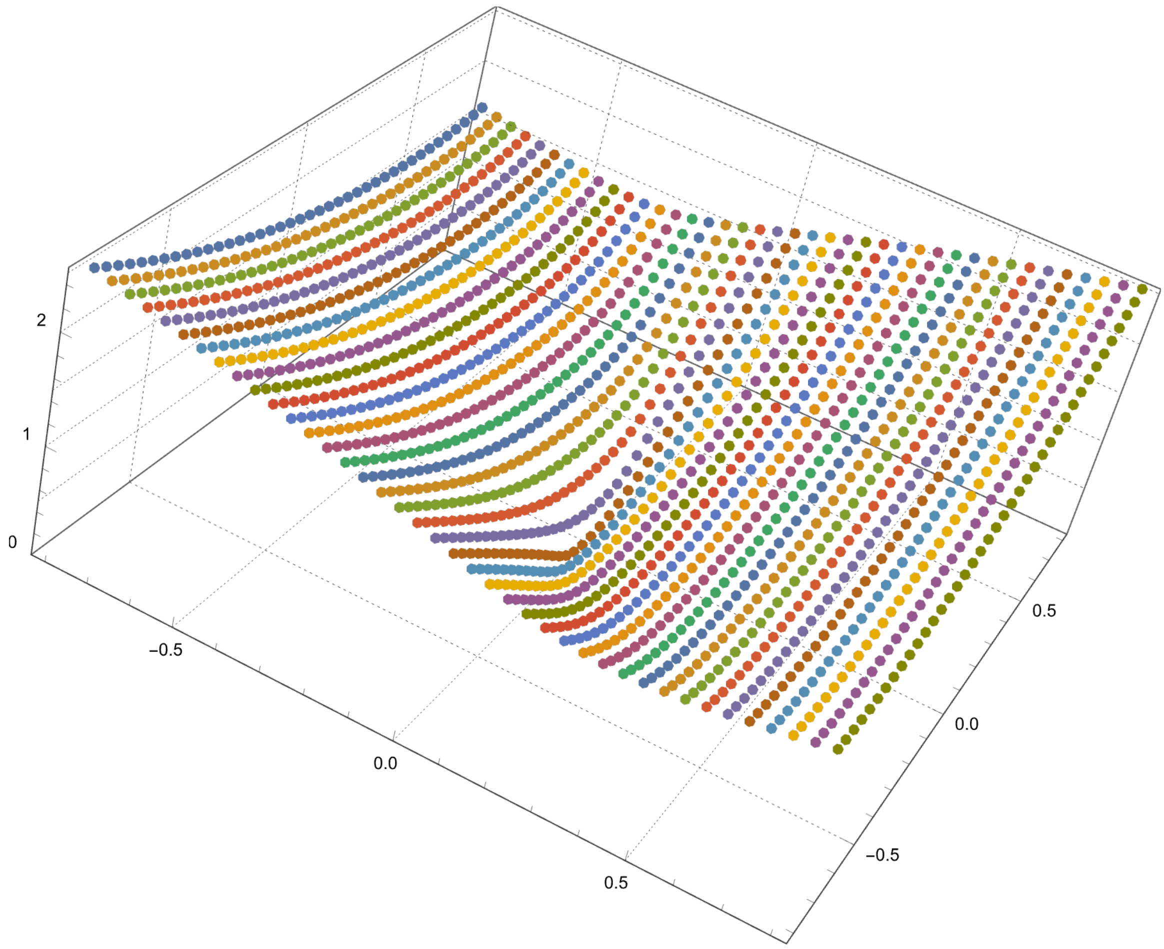
```
Clear[x1, x2]
⌊bereinige
tab = Table[
      ⌊Tabelle
   {x1, x2, NSolve[{2/9 θ^4 - x2^2 θ^2 - 2 θ x1 x2 - 3 x1^2 == 0 && θ ≥ 0}, θ][[1]][[1]][[2]]},
            ⌊löse numerisch
   {x1, -0.8, 0.8, 0.041},
   {x2, -0.8, 0.8, 0.041}
  ];
```

In[465]:=

```
ListPointPlot3D[tab,
 PlotTheme → "Detailed",
 PlotStyle → Directive[PointSize[0.01]],
 ImageSize → 600]
```

listenbezogenes 3D-Streudiagramm | Thema der graphischen Darstellung | Darstellungsstil | Anweisung | Punktgröße | Bildgröße

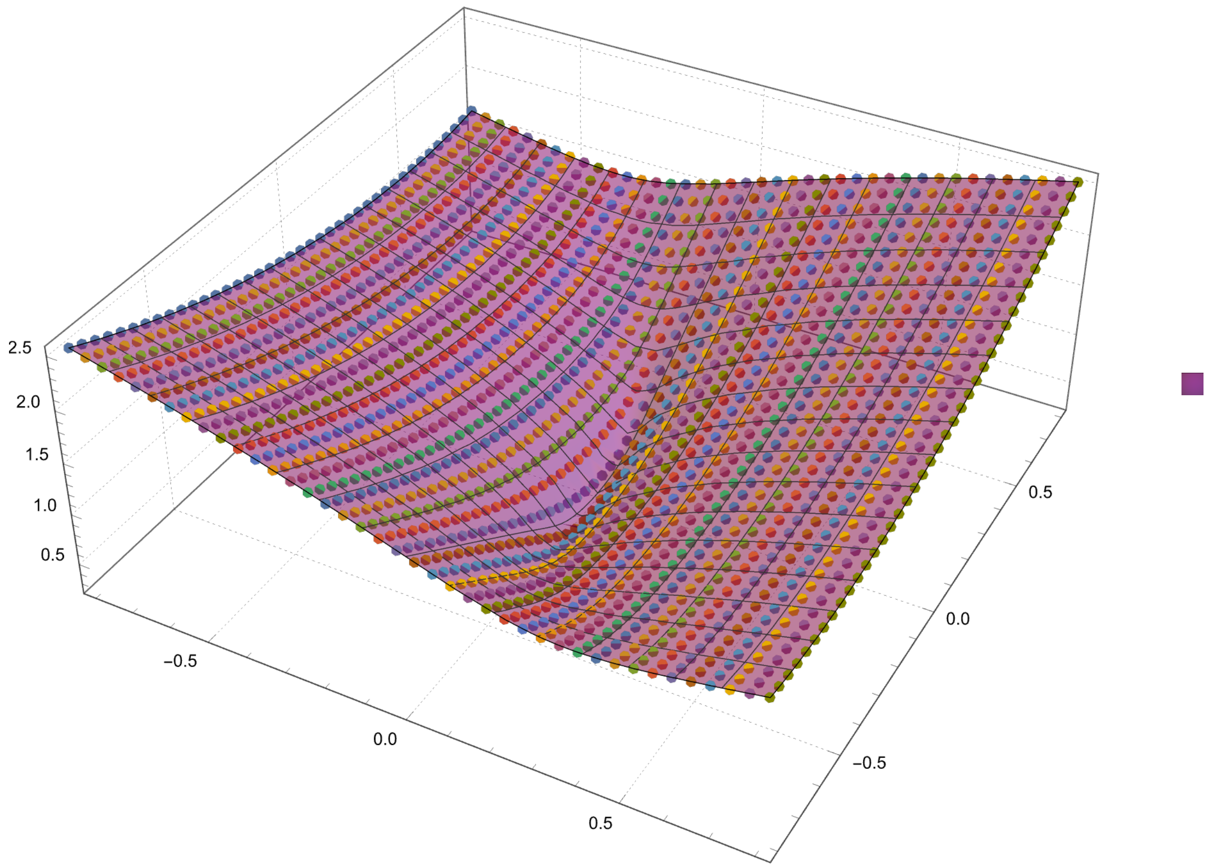
Out[465]=



In[462]:=

```
ft = Flatten[tab];
     ⌊ebne ein
θ[x1_, x2_] = Interpolation[
                  ⌊Interpolation
   Table[{{ft⟦i⟧, ft⟦i + 1⟧}, ft⟦i + 2⟧}, {i, 1, Length[ft], 3}], {x1, x2}];
  ⌊Tabelle                                          ⌊Länge
Show[
⌊zeige an
 Plot3D[θ[x1, x2], {x1, -0.8, 0.8}, {x2, -0.8, 0.8},
 ⌊stelle Funktion graphisch in 3D dar
  PlotTheme → "Detailed", PlotStyle → Directive[Purple, Opacity[0.5]]],
  ⌊Thema der graphischen Darstell⋯ ⌊Darstellungsstil ⌊Anweisung ⌊lila ⌊Deckkraft
 ListPointPlot3D[tab, PlotStyle → Directive[PointSize[0.01]]],
 ⌊listenbezogenes 3D-Streudia⋯ ⌊Darstellungsstil ⌊Anweisung ⌊Punktgröße
 ImageSize → 600
 ⌊Bildgröße
]
```

Out[464]=



Solving the differential equation!

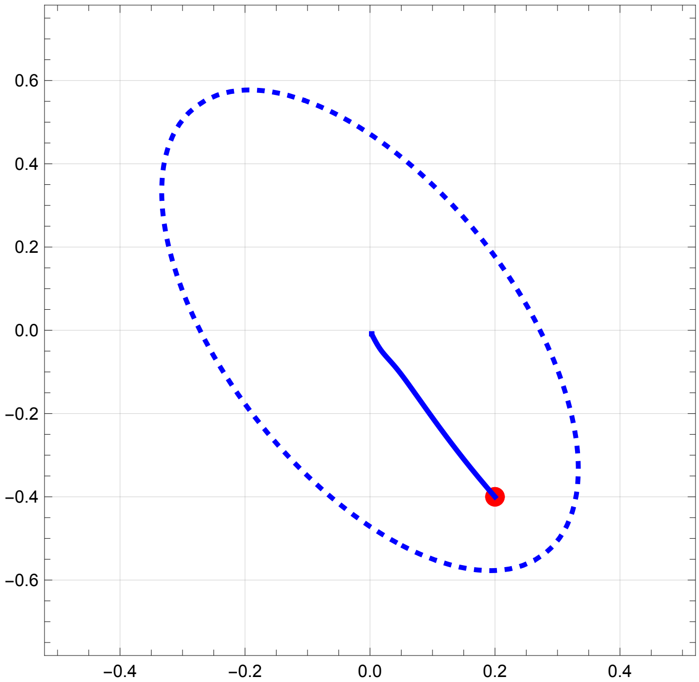
In[466]:=

```
u[x1_, x2_] = - x1/θ[x1, x2]^2 - 2 x2/θ[x1, x2];
s = NDSolve[{x1'[t] == x2[t] + ϵ (x2[t]^2 + x1[t] × u[x1[t], x2[t]]),
    [löse Differentialgleichung numerisch
   x2'[t] == u[x1[t], x2[t]] + δ (x1[t]^2 - x2[t] × u[x1[t], x2[t]]),
   x1[0] == x10, x2[0] == x20}, {x1, x2}, {t, T}]
Show[
[zeige an

 RegionPlot[{3 x1^2 + 2 x1 x2 + x2^2 ≤ 2/9}, {x1, -0.5, 0.5}, {x2, -0.75, 0.75},
 [graphische Darstellung einer Region

  GridLines → Automatic,
  [Gitternetzlinien [automatisch
  PlotStyle → Directive[Opacity[0.0]],
  [Darstellungsstil [Anweisung [Deckkraft
  BoundaryStyle → Directive[Blue, Thickness[0.0075], Dashed]],
  [Stil der Randregion [Anweisung [blau [Dicke [gestrichelt
 ListPlot[{{x10, x20}}, PlotStyle → Directive[Red, PointSize[0.03]]],
 [listenbezogene Graphik [Darstellungsstil [Anweisung [rot [Punktgröße
 ParametricPlot[Evaluate[{x1[t], x2[t]} /. s], {t, 0, T},
 [parametrische Darste· [werte aus
  GridLines → Automatic,
  [Gitternetzlinien [automatisch
  PlotStyle → Directive[Blue, Thickness[0.008]]]
  [Darstellungsstil [Anweisung [blau [Dicke
]
```

Out[467]=

```
{{x1 → InterpolatingFunction[ Domain: {{0., 1.74 }} Output: scalar ],
  x2 → InterpolatingFunction[ Domain: {{0., 1.74 }} Output: scalar ]}}
```

Out[468]=

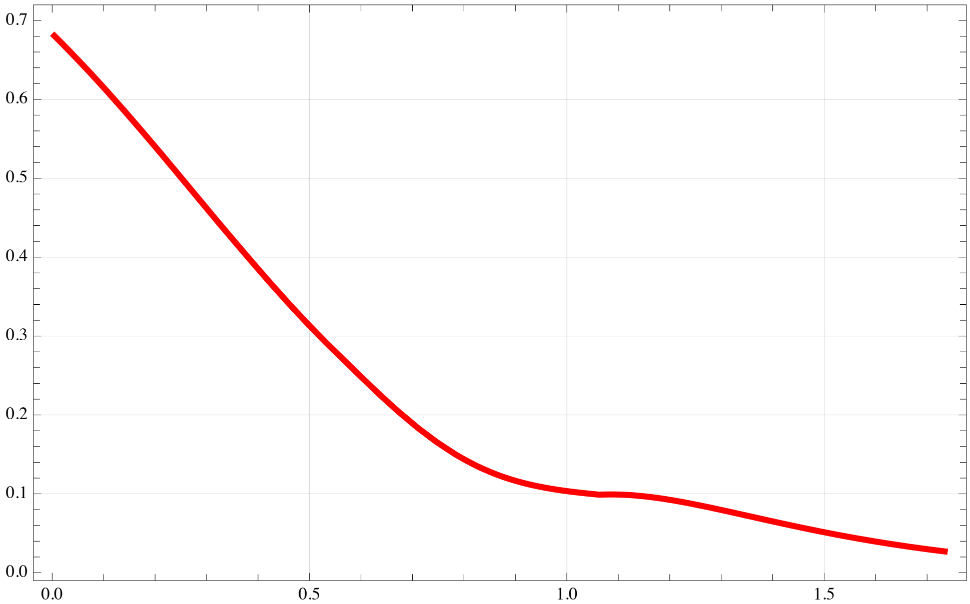


In[469]:=

```
Plot[Evaluate[u[x1[t], x2[t]] /. s],
 {t, 0.0, T}, PlotRange → All, PlotTheme → "Scientific",
 PlotStyle → Directive[Red, Thickness[0.0065]], GridLines → Automatic,
 ImageSize → 500]
```

stell··· werte aus Koordinatenbe·· alle Thema der graphischen Darstellung Darstellungsstil Anweisung rot Dicke Gitternetzlinien automatisch Bildgröße

Out[469]=



Controlability Function Derivative

In[470]:=

```
Plot[Evaluate[{D[θ[x1[t], x2[t]], t]} /. s],
 {t, 0.0, T}, PlotRange → All, PlotTheme → "Scientific",
 PlotStyle → Directive[Red, Thickness[0.0065]], GridLines → Automatic,
 ImageSize → 500]
```

stell··· werte aus leite ab Koordinatenbe·· alle Thema der graphischen Darstellung Darstellungsstil Anweisung rot Dicke Gitternetzlinien automatisch Bildgröße

Out[470]=