```
In[122]:= Clear["Global`*"]

In[123]:= (*Funktionen*)
GiveX[l_, y_] := Extract[l, Position[l, {xs_, ys_} /; ys == y]][[1, 1]]
GiveX[l_, Range_List] :=
 Extract[l, Position[l, {xs_, ys_} /; ys >= Range[[1]] && ys <= Range[[2]]]][[1, 1]]
GiveY[l_, x_] := Extract[l, Position[l, {xs_, ys_} /; xs == x]][[1, 2]]
GiveY[l_, Range_List] :=
 Extract[l, Position[l, {xs_, ys_} /; xs >= Range[[1]] && xs <= Range[[2]]]][[1, 2]]
GiveXY[l_, z_] := Extract[l, Position[l, {xs_, ys_, zs_} /; zs == z]][[1]][[{1, 2}]]
FindMaximumList[l_] := {GiveX[l, Max[l]], Max[l]}(*Annahme: y>x*)
CheckEq[l1_List, l2_List, n_] := Sequence[]
CheckEq[l1_List, l2_List, n_] :=
 l1[[n]] /; (l1[[n, 2]] ≥ l2[[n, 2]] && l1[[n, 2]] ≤ l2[[n + 5, 2]]) ||
   (l1[[n, 2]] ≤ l2[[n, 2]] && l1[[n, 2]] ≥ l2[[n + 5, 2]])
CheckEqPoints[l1_List, l2_List, start_: 1] := Flatten[
  Table[{CheckEq[l1, l2, n]}, {n, start, Length[l1] - 5}] /. {} → Sequence[], 1]

In[132]:= (*Lösung der DGL*)
s1[L_, r_, h_, x_] :=
 NDSolve[{i[L, x] θ''[t] == M L1[L, x] h φ''[t] Sin[φ[t] + θ[t]] + M L1[L, x]
       h φ'[t]^2 Cos[φ[t] + θ[t]] - m L2[L, x] r ψ''[t] Cos[ψ[t] - θ[t]] +
      m L2[L, x] r ψ'[t]^2 Sin[ψ[t] - θ[t]] + (V[L, x] - λ[t] L2[L, x]) Cos[θ[t]],
    h φ''[t] == L1[L, x] θ''[t] Sin[φ[t] + θ[t]] + L1[L, x] θ'[t]^2 Cos[φ[t] + θ[t]] -
      g Sin[φ[t]], r ψ''[t] == -L2[L, x] θ''[t] Cos[ψ[t] - θ[t]] -
      L2[L, x] θ'[t]^2 Sin[ψ[t] - θ[t]] - (g + λ[t] / m) Cos[ψ[t]],
    r Sin[ψ[t]] + L2[L, x] (Sin[θ[t]] - Sin[θ0]) == 0, θ[0] == θ0, θ'[0] == 0,
    φ[0] == 0, φ'[0] == 0, ψ[0] == -Pi, ψ'[0] == 0}, {θ, φ, ψ, λ},
   {t, Tm1, Tp1}, Method → {"IndexReduction" -> Automatic}] // Quiet
(*Schlinge in der Luft*)
s2[L_, r_, h_, x_] :=
 NDSolve[{i[L, x] θ''[t] == M L1[L, x] h φ''[t] Sin[φ[t] + θ[t]] + M L1[L, x]
      h φ'[t]^2 Cos[φ[t] + θ[t]] - m L2[L, x] r ψ''[t] Cos[ψ[t] - θ[t]] +
     m L2[L, x] r ψ'[t]^2 Sin[ψ[t] - θ[t]] + V[L, x] Cos[θ[t]], h φ''[t] ==
    L1[L, x] θ''[t] Sin[φ[t] + θ[t]] + L1[L, x] θ'[t]^2 Cos[φ[t] + θ[t]] - g Sin[φ[t]],
   r ψ''[t] == -L2[L, x] θ''[t] Cos[ψ[t] - θ[t]] - L2[L, x] θ'[t]^2 Sin[ψ[t] - θ[t]] -
     g Cos[ψ[t]], θ[t1] == θ1, θ'[t1] == θs1, φ[t1] == φ1,
   φ'[t1] == φs1, ψ[t1] == ψ1, ψ'[t1] == ψs1}, {θ, φ, ψ}, {t, Tm2, Tp2}]
```

In[134]:=
```
Pv[L_, r_, h_, x_, t1_: 0, t2_: 0] := Block[{S}, S = s2[L, r, h, x];
  Plot[Evaluate[Sqrt[(L2[L, x] θ'[t])^2 + (r ψ'[t])^2 +
       2 L2[L, x] ψ'[t] θ'[t] Cos[ψ[t] - θ[t]]] /. S], {t, Tm2, Tp2},
   PlotRange → All, GridLines → {{{t1, Red}, {t2, Green}}, None},
   PlotLabel → "Geschossgeschwindigkeit", AxesLabel → {"t", "v(t)"}]]
Ppsi[L_, r_, h_, x_, t1_: 0, t2_: 0] := Block[{S}, S = s2[L, r, h, x];
  Plot[Evaluate[ψ[t] /. S], {t, Tm1, Tp1}, PlotRange → All, GridLines → {{{t1, Red},
      {t2, Green}}, None}, PlotLabel → "Psi", AxesLabel → {"t", "ψ(t)"}]]
Pphi[L_, r_, h_, x_, t1_: 0, t2_: 0] := Block[{S}, S = s2[L, r, h, x];
  Plot[Evaluate[φ[t] /. S], {t, Tm1, Tp1}, PlotRange → All, GridLines → {{{t1, Red},
      {t2, Green}}, None}, PlotLabel → "Phi", AxesLabel → {"t", "ψ(t)"}]]
Pth[L_, r_, h_, x_, t1_: 0, t2_: 0] := Block[{S}, S = s2[L, r, h, x];
  Plot[Evaluate[θ[t] /. S], {t, Tm1, Tp1}, PlotRange → All, GridLines → {{{t1, Red},
      {t2, Green}}, None}, PlotLabel → "Theta", AxesLabel → {"t", "θ(t)"}]]
Plam[L_, r_, h_, x_, t1_: 0] := Block[{S}, S = s1[L, r, h, x];
  Plot[Evaluate[λ[t] /. S], {t, Tm1, Tp1}, PlotRange → All, GridLines →
    {{{t1, Red}}, None}, PlotLabel → "Lambda", AxesLabel → {"t", "λ(t)"}]]
```

In[139]:=
```
(*Listen und Funktionen für das erste Intervall*)
lamList1[L_, r_, h_, x_] := Block[{S}, S = s1[L, r, h, x];
  Table[{t, (λ[t] /. S)[[1]]}, {t, Tm1, Tp1, (Tp1 - Tm1) / n}]]
tlv[L_, r_, h_, x_] := GiveX[lamList1[L, r, h, x], {-0.1, 0}]
SetInitial2[L_, r_, h_, x_] := Block[{S}, S = s1[L, r, h, x];
  θ1 = (θ[t1] /. S)[[1]];
  θs1 = (θ'[t1] /. S)[[1]];
  φ1 = (φ[t1] /. S)[[1]];
  φs1 = (φ'[t1] /. S)[[1]];
  ψ1 = (ψ[t1] /. S)[[1]];
  ψs1 = (ψ'[t1] /. S)[[1]];
 ]
```

In[142]:=
```
(*Listen und Funktionen für das zweite Intervall*)
thList2[L_, r_, h_, x_] := Block[{S},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  Table[{t, (θ[t] /. S)[[1]]}, {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
psiList2[L_, r_, h_, x_] := Block[{S},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  Table[{t, (ψ[t] /. S)[[1]]}, {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
phiList2[L_, r_, h_, x_] := Block[{S},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  Table[{t, (φ[t] /. S)[[1]]}, {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
(*Betrag der Geschossgeschwindigkeit*)
vList[L_, r_, h_, x_] := Block[{S},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  Table[
   {t, Re[(Sqrt[(L2[L, x] θ'[t])^2 + (r ψ'[t])^2 + 2 L2[L, x] ψ'[t] θ'[t] Cos[ψ[t] -
           θ[t]]] /. S)[[1]]]}, {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
(*x-Komponente der Geschossgeschwindigkeit*)
vxList[L_, r_, h_, x_] := Block[{S},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  Table[{t, (-L2[L, x] Sin[θ[t]] θ'[t] - r Sin[ψ[t]] ψ'[t] /. S)[[1]]},
   {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
(*y-Komponente der Geschossgeschwindigkeit*)
vyList[L_, r_, h_, x_] := Block[{S},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  Table[{t, (L2[L, x] Cos[θ[t]] θ'[t] + r Cos[ψ[t]] ψ'[t] /. S)[[1]]},
   {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
(*Integrierte Wurfweite*)
sWurf[vx_, vy_, h_] := vx (vy + Sqrt[2 g h + vy^2]) / g
sWurfList[L_, r_, h_, x_] := Block[{S, vx, vy, H},
  SetInitial2[L, r, h, x];
  S = s2[L, r, h, x];
  vx = ((-L2[L, x] Sin[θ[t]] θ'[t] - r Sin[ψ[t]] ψ'[t]) /. S)[[1]];
  vy = ((L2[L, x] Cos[θ[t]] θ'[t] + r Cos[ψ[t]] ψ'[t]) /. S)[[1]];
  H = ((L Sin[θ[t]] + h) /. S)[[1]];
  Table[{t, Re[sWurf[vx, vy, H]]}, {t, Tm2, Tp2, (Tp2 - Tm2) / n}]]
tmax[L_, r_, h_, x_] := FindMaximumList[Abs[sWurfList[L, r, h, x]]][[1]]
smax[L_, r_, h_, x_] := GiveY[sWurfList[L, r, h, x], tmax[L, r, h, x]]
vmax[L_, r_, h_, x_] := GiveY[vList[L, r, h, x], tmax[L, r, h, x]]
```

In[153]:=
```
(*reales Trägheitsmoment eines spitz zulaufenden Balkens*)
iw[h_, Δ_] :=
  (h * (a * (4 * b^3 + 3 * b^2 * d + 2 * b * d^2 + d^3 + 8 * b * h^2 + 12 * d * h^2 - 40 *
          (b + d) * h * Δ + 40 * (2 * b + d) * Δ^2) +
       c * (b^3 + 2 * b^2 * d + 3 * b * d^2 + 4 * d^3 + 12 * b * h^2 + 48 * d * h^2 -
          40 * (b + 3 * d) * h * Δ + 40 * (b + 2 * d) * Δ^2)) * ρ) / 240
(*Abmessungen des Wurfarms*)
a = 0.1;
b = 0.1;
c = 0.05;
d = 0.05;
```

In[158]:=
```
(*Parameter*)
M = 50.; (*Masse des Gegengewichts*)
m = 0.5; (*Masse des Geschosses*)
Tm1 = 0.0;(*Zeitintervall Schlinge auf dem Boden*)
Tp1 = 0.3;
Tm2 = 0.0;
(*Zeitintervall Wurf, Tm2 wird später automatisch bestimmt Tm2→t1*)
Tp2 = 1.0;
θ0 = -Pi / 4; (*Startwinkel des Wurfarms,
wenn keine Einschränkung aus Geometrie*)
n = 2000; (*Anzahl der Auswertungspunkte*)
(*Konstanten*)
g = 9.81;
(*Dichte Holz*)
ρ = 670.;
```

In[165]:=
```
(*Startwerte*)
L0 = 3.5 (*Länge des Wurfarms*)
r0 = 2.16 (*Länge der Schlinge*)
h0 = 0.9 (*Länge der Aufhängung des Gegengewichts*)
x0 = 0.7 (*Abstand Aufhängung des Wurfarms zum SP, guter Start: L0/4*)
```

Out[165]= 3.5

Out[166]= 2.16

Out[167]= 0.9

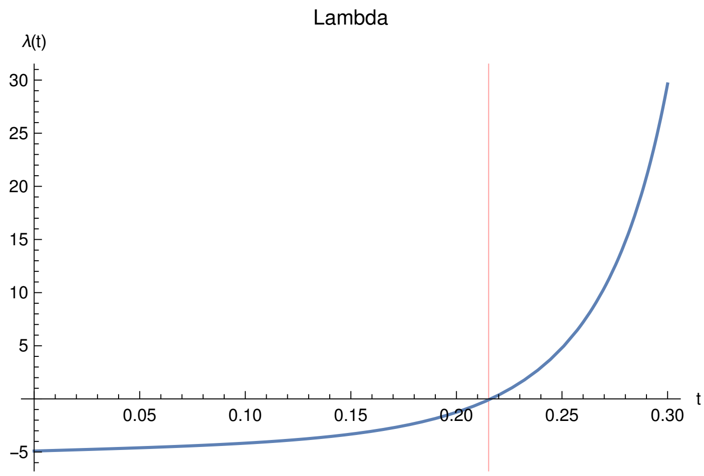
Out[168]= 0.7

In[169]:=
```
(*abgeleitete Parameter*)
L1[L_, x_] := L / 2 - x
L2[L_, x_] := L / 2 + x
μ[L_] := L d^2 ρ
V[L_, x_] := M g L1[L, x] - m g L2[L, x] - μ[L] g x
i[L_, x_] := iw[L, x] + m L2[L, x]^2 + M L1[L, x]^2
```

In[174]:=
```
(*Löse zuerst die erste DGL, um den Zeitpunkt zu finden,
wann das Geschoss den Boden verlässt → λ=0*)
t1 = tlv[L0, r0, h0, x0]
Plam[L0, r0, h0, x0, t1]
(*finde dann die Anfangsbedingungen für die zweite DGL*)
SetInitial2[L0, r0, h0, x0]
θ1
θs1
φ1
φs1
ψ1
ψs1
```

Out[174]= 0.21525

Out[175]=



Out[177]= -0.324362

Out[178]= 2.6478

Out[179]= 0.00365473

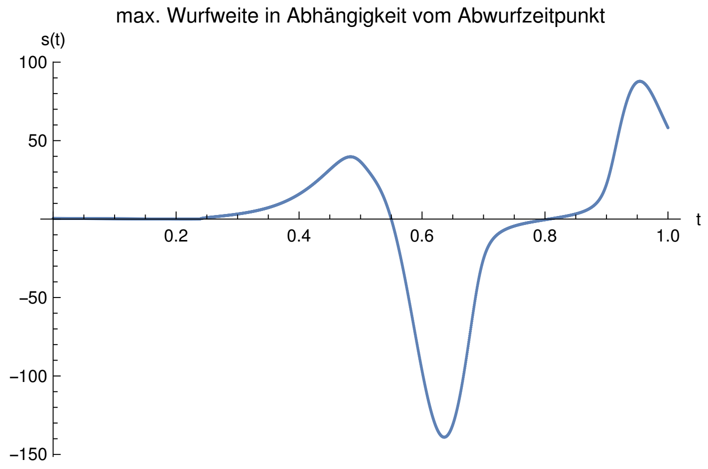
Out[180]= 0.338941

Out[181]= -2.68538

Out[182]= 3.17099

In[183]:=
```
ListPlot[sWurfList[L0, r0, h0, x0], PlotRange → All,
 PlotLabel → "max. Wurfweite in Abhängigkeit vom Abwurfzeitpunkt",
 AxesLabel → {"t", "s(t)"}]
```

Out[183]=

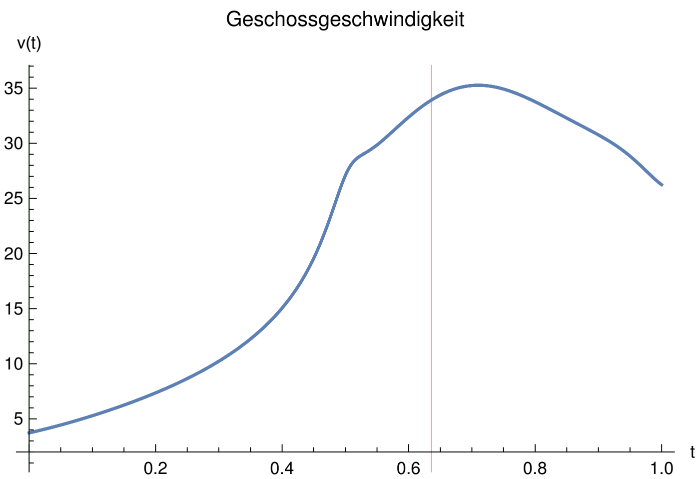


In[184]:=
```
t2 = tmax[L0, r0, h0, x0]
smax[L0, r0, h0, x0]
Show[Pv[L0, r0, h0, x0, t2], ListPlot[vList[L0, r0, h0, x0]]]
Show[Pth[L0, r0, h0, x0, t2], ListPlot[thList2[L0, r0, h0, x0]]]
Show[Ppsi[L0, r0, h0, x0, t2], ListPlot[psiList2[L0, r0, h0, x0]]]
```

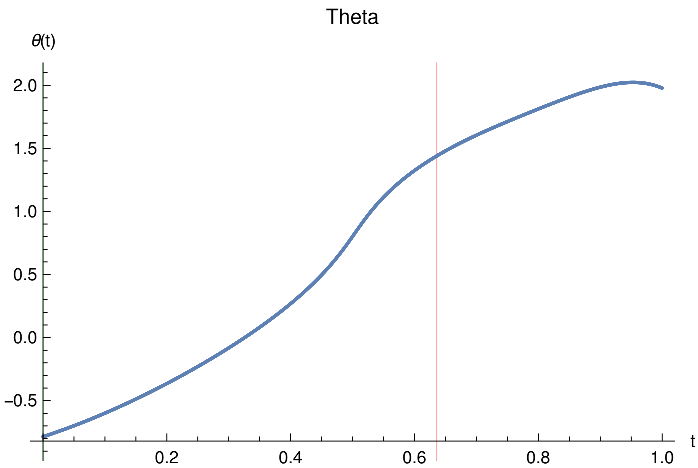
Out[184]= 0.636

Out[185]= -139.029

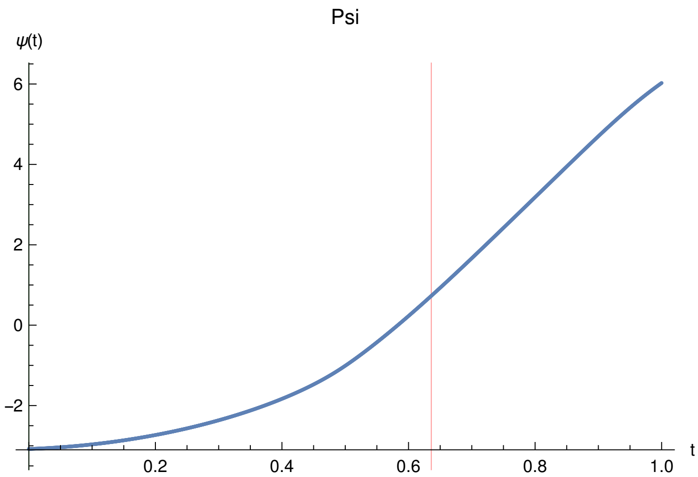
Out[186]=



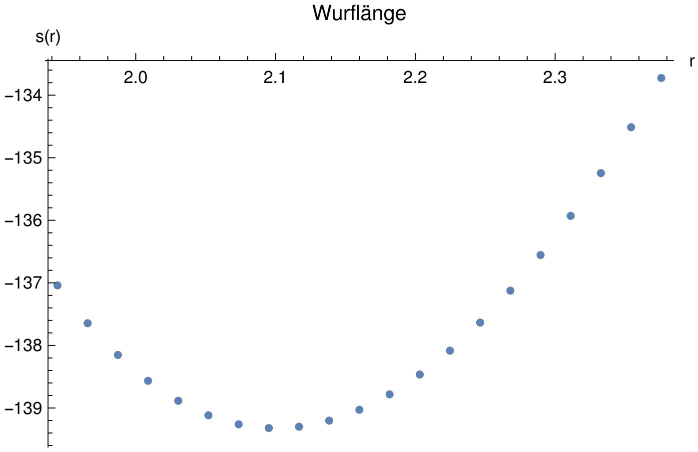
Out[187]=

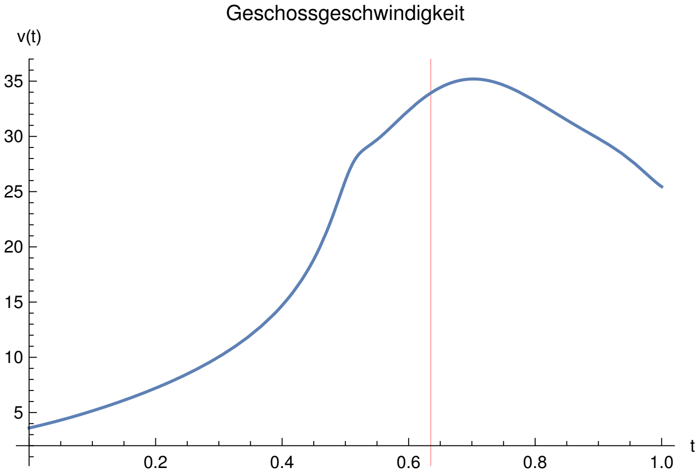


Out[188]=



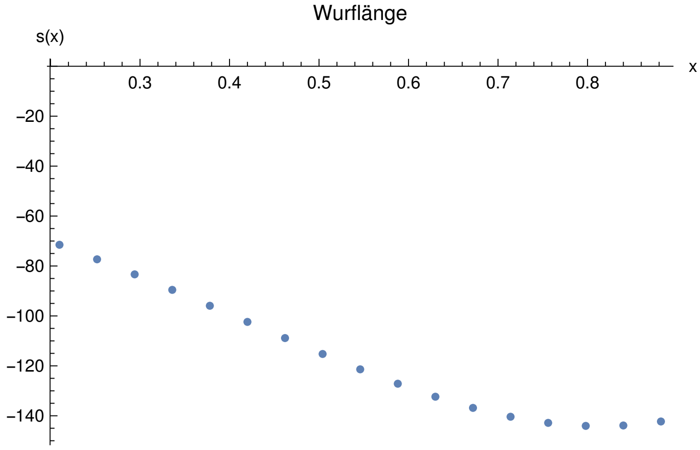
In[189]:=
```
(*1D Analyse in r*)
smaxr = Table[{r, smax[L0, r, h0, x0]}, {r, 0.9 r0, 1.1 r0, 0.1 r0 / 10}];
ListPlot[smaxr, PlotLabel → "Wurflänge", AxesLabel → {"r", "s(r)"}]
rmax = FindMaximumList[Abs[smaxr]][[1]];
r0 = rmax;
t2 = tmax[L0, r0, h0, x0];
Pv[L0, r0, h0, x0, t2]
Print["Abwurfzeitpunkt t2 = ", t2]
Print["neuer Startwert für L0 = ", L0]
Print["neuer Startwert für r0 = ", r0]
Print["neuer Startwert für h0 = ", h0]
Print["neuer Startwert für x0 = ", x0]
Print["vmax = ", vmax[L0, r0, h0, x0]]
Print["smax = ", smax[L0, r0, h0, x0]]
```

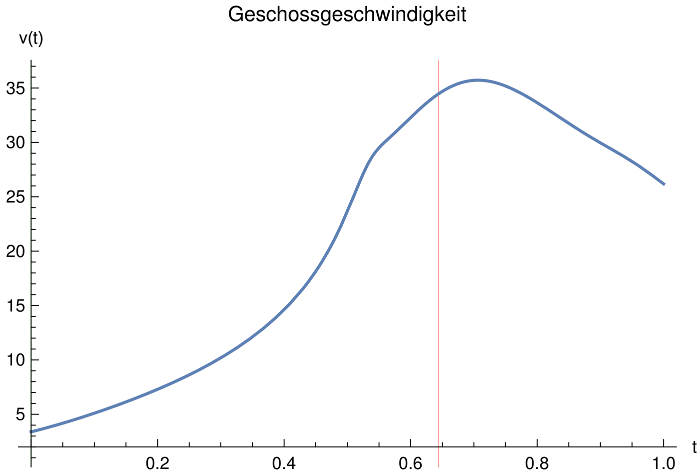
Out[190]=


Out[194]=



```
Abwurfzeitpunkt t2 = 0.635
neuer Startwert für L0 = 3.5
neuer Startwert für r0 = 2.0952
neuer Startwert für h0 = 0.9
neuer Startwert für x0 = 0.7
vmax = 33.9247
smax = -139.321
```

In[202]:=
```
(*1D Analyse in x*)
smaxx = Table[{x, smax[L0, r0, h0, x]}, {x, 0.3 x0, 1.3 x0, 0.6 x0 / 10}];
ListPlot[smaxx, PlotLabel → "Wurflänge", AxesLabel → {"x", "s(x)"}]
xmax = FindMaximumList[Abs[smaxx]][[1]];
x0 = xmax;
t2 = tmax[L0, r0, h0, x0];
Pv[L0, r0, h0, x0, t2]
Print["Abwurfzeitpunkt t0 = ", t2]
Print["neuer Startwert für L0 = ", L0]
Print["neuer Startwert für r0 = ", r0]
Print["neuer Startwert für h0 = ", h0]
Print["neuer Startwert für x0 = ", x0]
Print["vmax = ", vmax[L0, r0, h0, x0]]
Print["smax = ", smax[L0, r0, h0, x0]]
```

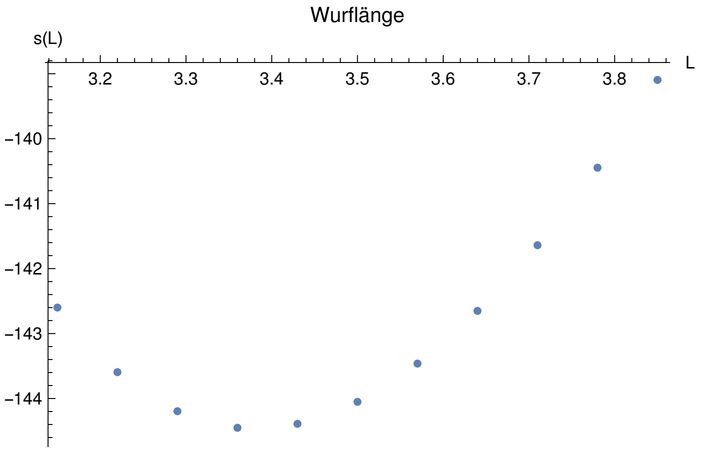
Out[203]=


Out[207]=


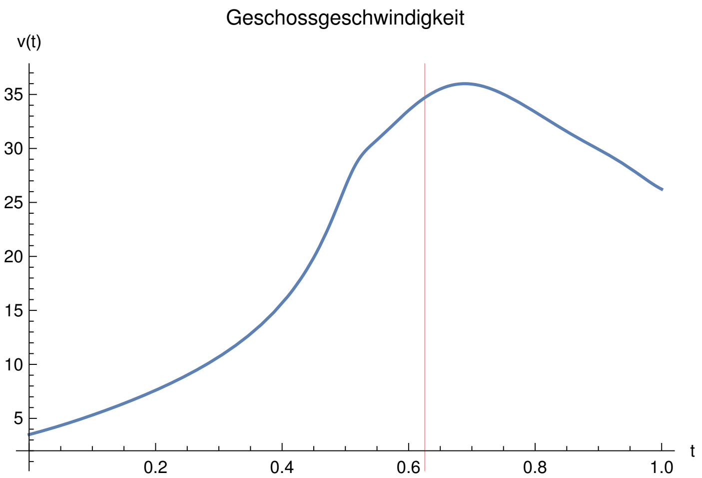

```
Abwurfzeitpunkt t0 = 0.644
neuer Startwert für L0 = 3.5
neuer Startwert für r0 = 2.0952
neuer Startwert für h0 = 0.9
neuer Startwert für x0 = 0.798
vmax = 34.4571
smax = -144.051
```

In[215]:=
```
(*1D Analyse in L*)
smaxL = Table[{L, smax[L, r0, h0, x0]}, {L, 0.9 L0, 1.1 L0, 0.2 L0 / 10}];
ListPlot[smaxL, PlotLabel → "Wurflänge", AxesLabel → {"L", "s(L)"}]
Lmax = FindMaximumList[Abs[smaxL]][[1]];
L0 = Lmax;
t2 = tmax[L0, r0, h0, x0];
Pv[L0, r0, h0, x0, t2]
Print["Abwurfzeitpunkt t0 = ", t2]
Print["neuer Startwert für L0 = ", L0]
Print["neuer Startwert für r0 = ", r0]
Print["neuer Startwert für h0 = ", h0]
Print["neuer Startwert für x0 = ", x0]
Print["vmax = ", vmax[L0, r0, h0, x0]]
Print["smax = ", smax[L0, r0, h0, x0]]
```

Out[216]=



Out[220]=




```
Abwurfzeitpunkt t0 = 0.6255
neuer Startwert für L0 = 3.36
neuer Startwert für r0 = 2.0952
neuer Startwert für h0 = 0.9
neuer Startwert für x0 = 0.798
vmax = 34.6981
smax = -144.451
```

In[228]:=
```
(*Notiz: Optimierung über h konvergiert nicht. h sollte
  deshalb so groß wie technisch möglich gewählt werden.*)
(*1D Analyse in h*)
(*smaxh=Table[{h,smax[L0,r0,h,x0]},{h,0.9h0,1.1 h0,0.2h0/10}];
ListPlot[smaxh,PlotLabel→"Wurflänge",AxesLabel→{"h","s(h)"}]
 hmax=FindMaximumList[Abs[smaxh]][[1]];
h0=hmax;
t2=tmax[L0,r0,h0,x0];
Pv[L0,r0,h0,x0,t2]
 Print["Abwurfzeitpunkt t0 = ",t2]
 Print["neuer Startwert für L0 = ",L0]
 Print["neuer Startwert für r0 = ",r0]
 Print["neuer Startwert für h0 = ",h0]
 Print["neuer Startwert für x0 = ",x0]
 Print["vmax = ",vmax[L0,r0,h0,x0]]
 Print["smax = ",smax[L0,r0,h0,x0]]*)
```

In[229]:=
```
(*Vektoren*)
r1List1[L_, r_, h_, x_] := Block[{S}, S = s1[L, r, h, x];
  Table[{((L2[L, x] Cos[θ[t]]) /. S)[[1]],
    ((L2[L, x] Sin[θ[t]]) /. S)[[1]]}, {t, Tm1, t1, 0.01}]]
r1List2[L_, r_, h_, x_] := Block[{S}, S = s2[L, r, h, x];
  Table[{((L2[L, x] Cos[θ[t]]) /. S)[[1]],
    ((L2[L, x] Sin[θ[t]]) /. S)[[1]]}, {t, t1, t2, 0.01}]]
r1List[L_, r_, h_, x_] := Join[r1List1[L, r, h, x], r1List2[L, r, h, x]]
r2List1[L_, r_, h_, x_] := Block[{S}, S = s1[L, r, h, x];
  Table[{((L2[L, x] Cos[θ[t]] + r Cos[ψ[t]]) /. S)[[1]],
    ((L2[L, x] Sin[θ[t]] + r Sin[ψ[t]]) /. S)[[1]]}, {t, Tm1, t1, 0.01}]]
r2List2[L_, r_, h_, x_] := Block[{S}, S = s2[L, r, h, x];
  Table[{((L2[L, x] Cos[θ[t]] + r Cos[ψ[t]]) /. S)[[1]],
    ((L2[L, x] Sin[θ[t]] + r Sin[ψ[t]]) /. S)[[1]]}, {t, t1, t2, 0.01}]]
r2List[L_, r_, h_, x_] := Join[r2List1[L, r, h, x], r2List2[L, r, h, x]]
r3List1[L_, r_, h_, x_] := Block[{S}, S = s1[L, r, h, x];
  Table[{((-L1[L, x] Cos[θ[t]]) /. S)[[1]],
    ((-L1[L, x] Sin[θ[t]]) /. S)[[1]]}, {t, Tm1, t1, 0.01}]]
r3List2[L_, r_, h_, x_] := Block[{S}, S = s2[L, r, h, x];
  Table[{((-L1[L, x] Cos[θ[t]]) /. S)[[1]],
    ((-L1[L, x] Sin[θ[t]]) /. S)[[1]]}, {t, t1, t2, 0.01}]]
r3List[L_, r_, h_, x_] := Join[r3List1[L, r, h, x], r3List2[L, r, h, x]]
r4List1[L_, r_, h_, x_] := Block[{S}, S = s1[L, r, h, x];
  Table[{((-L1[L, x] Cos[θ[t]] - h Sin[φ[t]]) /. S)[[1]],
    ((-L1[L, x] Sin[θ[t]] - h Cos[φ[t]]) /. S)[[1]]}, {t, Tm1, t1, 0.01}]]
r4List2[L_, r_, h_, x_] := Block[{S}, S = s2[L, r, h, x];
  Table[{((-L1[L, x] Cos[θ[t]] - h Sin[φ[t]]) /. S)[[1]],
    ((-L1[L, x] Sin[θ[t]] - h Cos[φ[t]]) /. S)[[1]]}, {t, t1, t2, 0.01}]]
r4List[L_, r_, h_, x_] := Join[r4List1[L, r, h, x], r4List2[L, r, h, x]]
Manipulate[
 Show[ListPlot[{r2List[L0, r0, h0, x0], r1List[L0, r0, h0, x0]}, PlotRange →
    {{-L2[L0, x0] - r0, L2[L0, x0] + r0}, {-L2[L0, x0] - r0, L2[L0, x0] + r0}},
   Frame → True, ImageSize → {600, 600}, AspectRatio → 1],
  Graphics[Line[{r4List[L0, r0, h0, x0][[Round[n]]],
     r3List[L0, r0, h0, x0][[Round[n]]], {0, 0},
     r1List[L0, r0, h0, x0][[Round[n]]], r2List[L0, r0, h0, x0][[Round[n]]]}]]],
 {n, 1, Length[r1List[L0, r0, h0, x0]]}]
```

Out[241]=