

#DO NOT Run Complete Notebook at once might takes hours to run the full code

```
Needs["DifferentialEquations`NDSolveProblems`"];
Needs["DifferentialEquations`NDSolveUtilities`"];
Needs["DifferentialEquations`InterpolatingFunctionAnatomy`"];
Needs["GUIKit`"];
(*RangeOfProjectile[Throwing Arm length a, weight arm length b,
  Weight pendulum length c, height of pivot d, length of sling l, mass 1 m1,
  mass to be thrown m2, release angle trigger rA, show log 1 or 0]
This function takes the variables of the trebuchet and
provides all the significant values in the output.
  this implementation of the process as a function Simplifies
  iteration process and makes the optimization faster
  slog values allows you to print the values of variables in current
  iteration the logging if switched off by default and can be
  enabled by passing value 1*)
iniTheta = 170 Degree;
initialAlfa = 10 Degree;

RangeOfProjectile[a_, b_, c_, d_, l_, m1_, m2_, rA_, Slog_: 0] := (designDelay = 5;
  
$$\phi[t_] = \theta[t] - \text{ArcCos}[(d + a \cos[\theta[t]]) / l];$$

  
$$x1[t_] = b \sin[\theta[t]] - c \sin[\theta[t] + \gamma[t]]; \quad$$

  
$$y1[t_] = -b \cos[\theta[t]] + c \cos[\theta[t] + \gamma[t]]; \quad$$

  
$$x2[t_] = -a \sin[\theta[t]] + l \sin[\theta[t] - \phi[t]]; \quad$$

  
$$y2[t_] = a \cos[\theta[t]] - l \cos[\theta[t] - \phi[t]]; \quad$$

  Te1 =
  
$$\frac{1}{2} m1 ((D[x1[t], t]^2) + (D[y1[t], t]^2)) + \frac{1}{2} m2 ((D[x2[t], t]^2) + (D[y2[t], t]^2));$$

  V1 = m1 g y1[t] + m2 g y2[t];
  L1 = Te1 - V1;

Eqn1 = {D[D[L1, \theta'[t]], t] - D[L1, \theta[t]] = 0,
  D[D[L1, \gamma'[t]], t] - D[L1, \gamma[t]] = 0, \theta[0] = iniTheta, \theta'[0] = 0, \gamma[0] = initialAlfa,
```

```

 $\gamma'[0] == 0, \text{WhenEvent}[\theta[t] == \text{ArcCos}\left[\frac{1-d}{a}\right] + 0.00001, \text{"StopIntegration"}]\};$ 

sol1 = NDSolve[Eqn1, {y, \theta}, {t, 0, 2}];
end = Last[First[InterpolatingFunctionDomain[First[\theta /. sol1]]]];
datat = Table[Flatten[{t, Evaluate[{\theta[t]} /. sol1]}], {t, 0, end, 0.0005}];
datag = Table[Flatten[{t, Evaluate[{\gamma[t]} /. sol1]}], {t, 0, end, 0.0005}];
dataa = Table[Flatten[{t, Evaluate[{\phi[t]} /. sol1]}], {t, 0, end, 0.0005}];

thE = First[\theta[end] /. sol1];
thvE = First[\theta'[end] /. sol1];
gmE = First[\gamma[end] /. sol1];
gmvE = First[\gamma'[end] /. sol1];
alfE = First[\phi[end] /. sol1];
alfEV = First[\phi'[end] /. sol1];
x2[t_] := -a Sin[\theta[t]] + l Sin[\theta[t] - \alpha[t]];
y2[t_] := a Cos[\theta[t]] - l Cos[\theta[t] - \alpha[t]];

Te =
 $\frac{1}{2} m_1 ((D[x_1[t], t]^2) + (D[y_1[t], t]^2)) + \frac{1}{2} m_2 ((D[x_2[t], t]^2) + (D[y_2[t], t]^2));$ 
V = m1 g y1[t] + m2 g y2[t];
L = Te - V;
Eqn2 = {D[D[L, \alpha'[t]], t] - D[L, \alpha[t]] == 0,
        D[D[L, \theta'[t]], t] - D[L, \theta[t]] == 0, D[D[L, \gamma'[t]], t] - D[L, \gamma[t]] == 0,
        \theta[end] == thE, \theta'[end] == thvE, \gamma[end] == gmE, \gamma'[end] == gmvE, \alpha[end] == alfe,
        \alpha'[end] == alfeV, \text{WhenEvent}[\alpha[t] == rA, \text{"StopIntegration"}]\};

sol2 = NDSolve[Eqn2, {y, \theta, \alpha}, {t, end, 20}];
end2 = Last[First[InterpolatingFunctionDomain[First[\theta /. sol2]]]];

datat2 = Table[Flatten[{t, Evaluate[{\theta[t]} /. sol2]}], {t, end, end2, 0.0005}];
datag2 = Table[Flatten[{t, Evaluate[{\gamma[t]} /. sol2]}], {t, end, end2, 0.0005}];
dataa2 = Table[Flatten[{t, Evaluate[{\alpha[t]} /. sol2]}], {t, end, end2, 0.0005}];

SOL = {{\theta \rightarrow \text{Interpolation}[\text{Join}[datat, datat2]],
        \gamma \rightarrow \text{Interpolation}[\text{Join}[datag, datag2]], \alpha \rightarrow \text{Interpolation}[\text{Join}[dataa, dataa2]]]};
theta = First[\theta[end2 - 0.001] /. SOL];
thetaDot = First[\theta'[end2 - 0.001] /. SOL];
alfa = First[\alpha[end2 - 0.001] /. SOL];
alfaDot = First[\alpha'[end2 - 0.001] /. SOL];

Relseangle =
ArcTan[(First[y2'[end2 - 0.001] /. SOL]) / (First[x2'[end2 - 0.001] /. SOL])];

```

```

ReleaseVelocity = ((-1 Cos[alfa - theta] (alfaDot - thetaDot) - a Cos[theta] thetaDot)^2 +
    (1 Sin[alfa - theta] (alfaDot - thetaDot) - a Sin[theta] thetaDot)^2)^(1/2);
If[Slog > 0, Print[{a, b, c, d, l, m1, m2, rA}], klp = a(*do nothing*)];
{solution → SOL, range → (ReleaseVelocity^2 * Sin[2 * RelseaseAngle]) / g,
 releaseangle → RelseaseAngle / Degree, relVel → ReleaseVelocity, Plot_α → Plot[
   α[t] /. SOL, {t, 0, end2}, AxesLabel → {"t(Seconds)", "α[t]"}, PlotRange → Full],
 Plot_θ → Plot[θ[t] /. SOL, {t, 0, end2}, AxesLabel → {"t(Seconds)", "θ[t]"}, PlotRange → Full],
 Plot_γ → Plot[γ[t] /. SOL, {t, 0, end2}, AxesLabel → {"t(Seconds)", "γ[t]"}, PlotRange → Full],
 Plot_dα → Plot[α'[t] /. SOL, {t, 0, end2},
   AxesLabel → {"t(Seconds)", "α'[t]"}, PlotRange → Full],
 Plot_dθ → Plot[θ'[t] /. SOL, {t, 0, end2},
   AxesLabel → {"t(Seconds)", "θ'[t]"}, PlotRange → Full],
 Plot_dγ → Plot[γ'[t] /. SOL, {t, 0, end2},
   AxesLabel → {"t(Seconds)", "γ'[t]"}, PlotRange → Full],
 Para_P → ParametricPlot[{{First[x2[t] /. SOL], First[y2[t] /. SOL]},
   {First[x1[t] /. SOL], First[y1[t] /. SOL]}},
   {t, 0, end2}, AxesLabel → {"x", "y"}, PlotLegends → {"m2", "m1"}]}

}

g = 9.81;
a = 4;
b = 1;
c = 0.5;
d = 5;
l = 4;
m1 = 133;
m2 = 1;

rA = 2.75;

```

Initial Solution

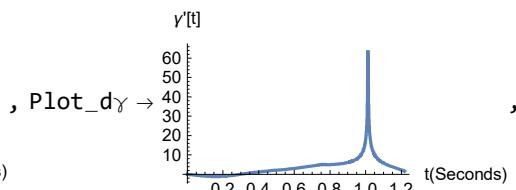
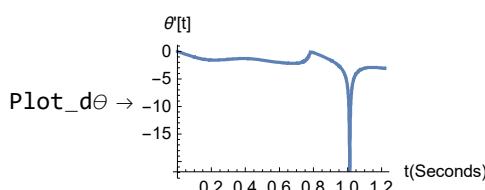
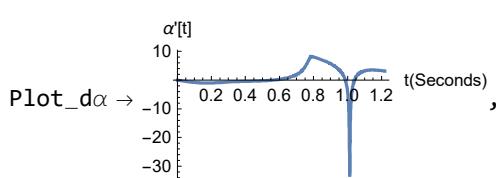
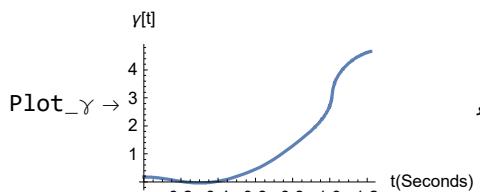
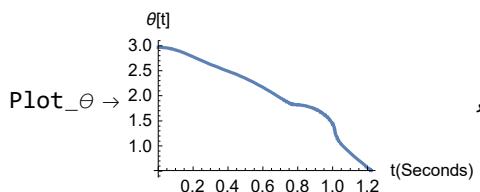
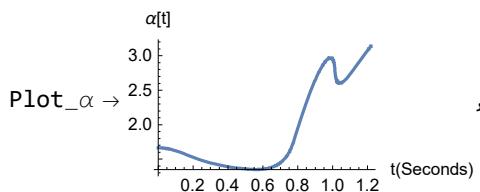
```
Solution = RangeOfProjectile[4, b, c, d, 4, m1, m2, 3.14]
```

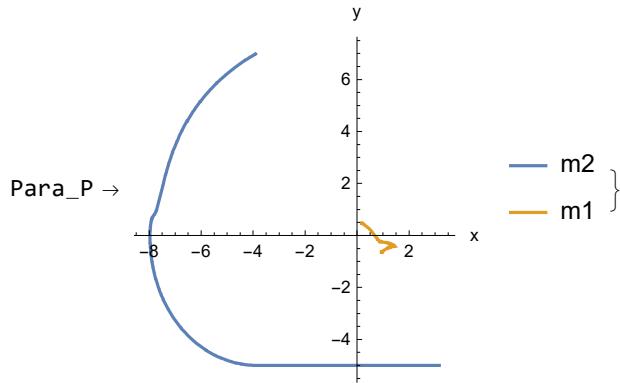
{solution → {θ → InterpolatingFunction[ Domain: {{0., 1.22}}], Output: scalar},

$\gamma \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \oplus \quad \text{Domain: } \{\{0., 1.22\}\} \\ \text{Output: scalar} \end{array} \right],$

$\alpha \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \oplus \quad \text{Domain: } \{\{0., 1.22\}\} \\ \text{Output: scalar} \end{array} \right] \},$

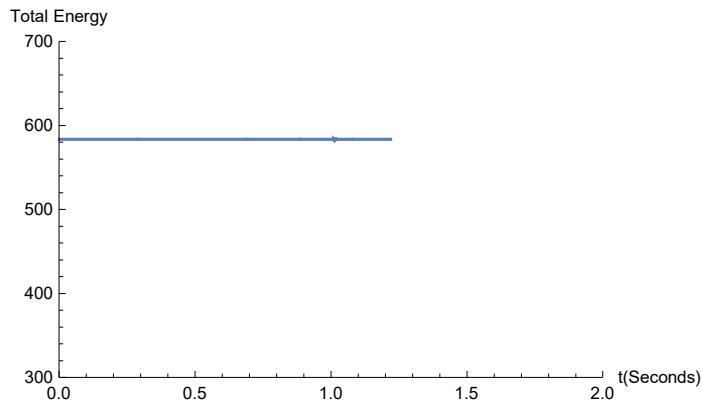
range $\rightarrow 121.193$, releaseangle $\rightarrow 29.579$, relVel $\rightarrow 37.2118$,



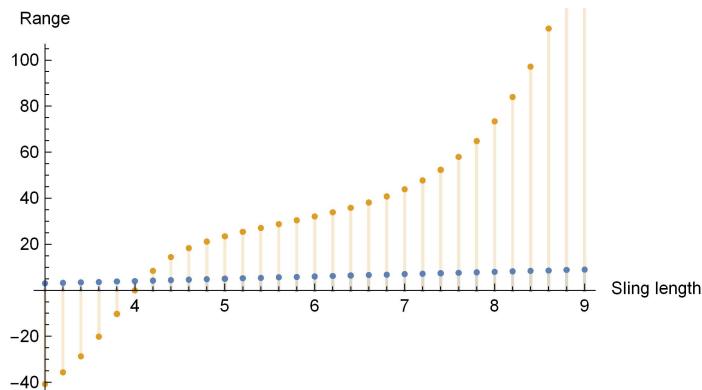


Validation

```
Plot[Te + V /. SOL, {t, 0, end2},
AxesLabel → {"t(Seconds)", "Total Energy"}, PlotRange → {{0, 2}, {300, 700}}]
```



```
DiscretePlot[{x, range /. RangeOfProjectile[6, b, c, d, x, m1, m2, 3.13]},
{x, 3, 9, 0.2}, AxesLabel → {"Sling length", "Range"}]
```



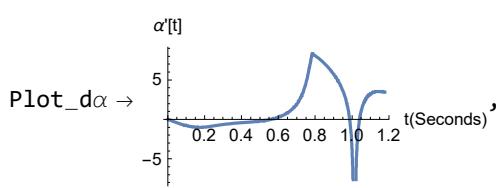
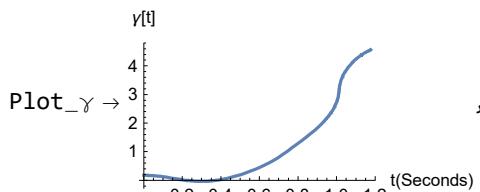
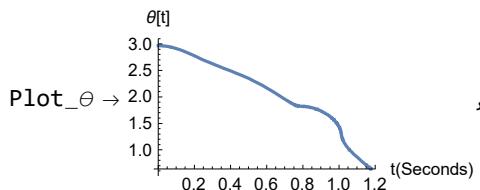
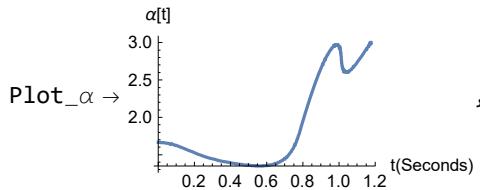
```
(*Calling function RangeOfProjectile[Throwing Arm length a, weight arm length b,
Weight pendulum length c, height of pivot d, length of sling l, mass 1 m1,
mass to be thrown m2, release angle trigger rA, Show log 1 or 0]*)
Solution = RangeOfProjectile[4, b, c, d, 4, m1, m2, 3]
```

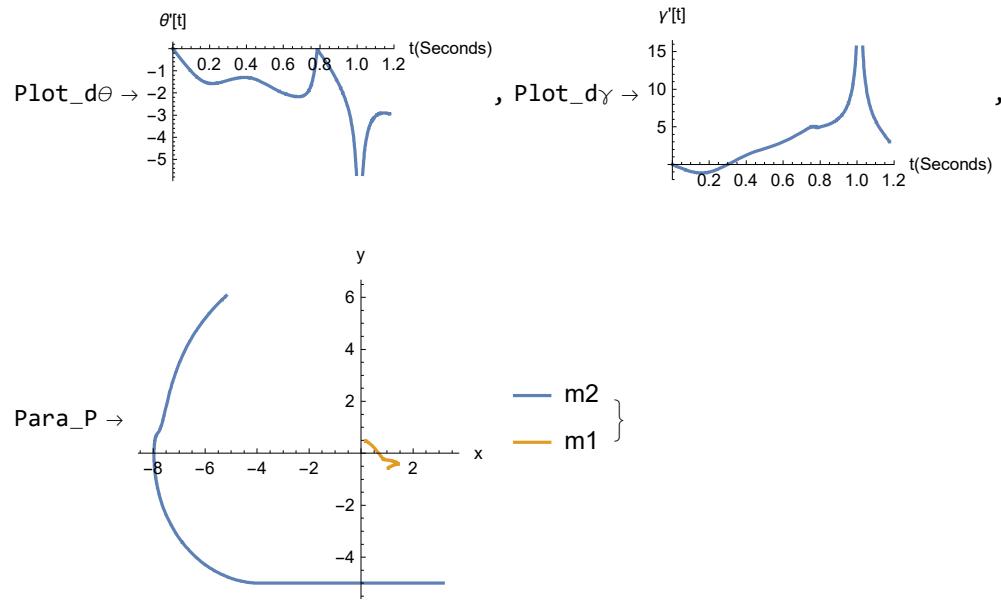
{solution → {θ → InterpolatingFunction [+  Domain: {{0., 1.18}}],

γ → InterpolatingFunction [+  Domain: {{0., 1.18}}],

α → InterpolatingFunction [+  Domain: {{0., 1.18}}] }},

range → 140.957, releaseangle → 42.3589, relVel → 37.265,

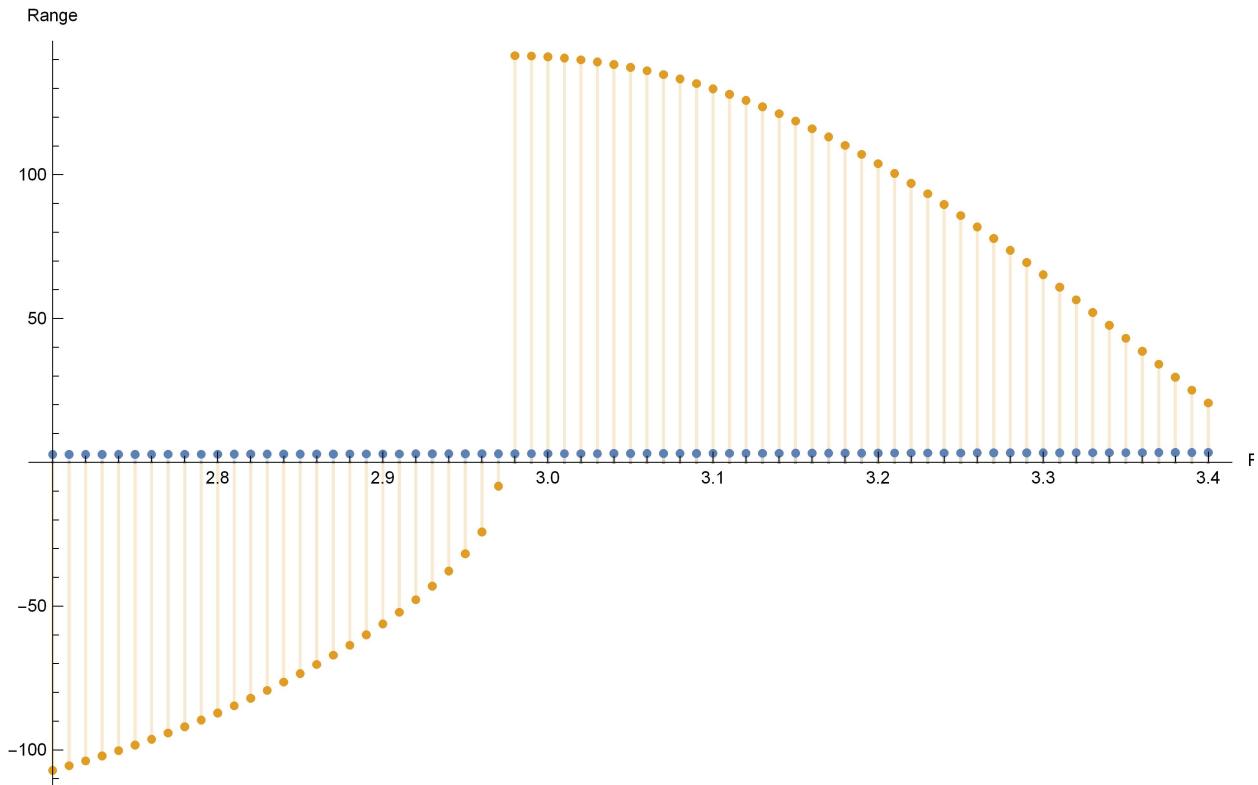




```
(*Calling function RangeOfProjectile[Throwing Arm length a, weight arm length b,
Weight pendulum length c, height of pivot d, length of sling l, mass 1 m1,
mass to be thrown m2, release angle trigger rA, Show log 1 or 0]*)
```

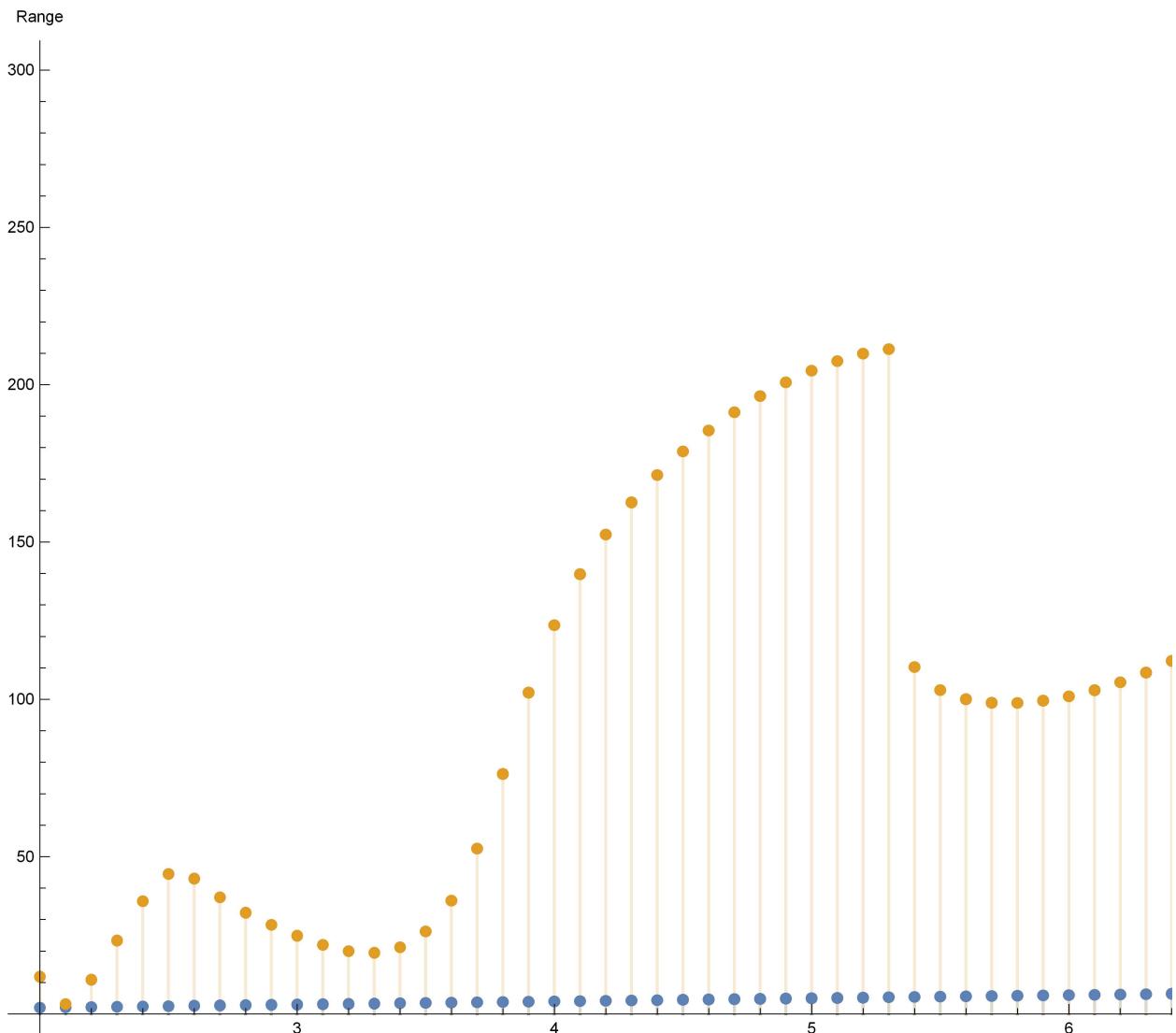
Release Angle iteration

```
DiscretePlot[{x, range /. RangeOfProjectile[4, b, c, d, 4, m1, m2, x, 1]},  
{x, 2.7, 3.4, 0.01}, AxesLabel -> {"Release Angle", "Range"}]
```



Sling length iteration

```
(*RangeOfProjectile[Throwing Arm length a, weight arm length b,  
Weight pendulum length c, height of pivot d, length of sling l,  
mass 1 m1, mass to be thrown m2, release angle trigger rA]*)  
DiscretePlot[{x, range /. RangeOfProjectile[4, b, c, d, x, m1, m2, 3.13]},  
{x, 2, 8, 0.1}, AxesLabel -> {"Sling length", "Range"}]
```

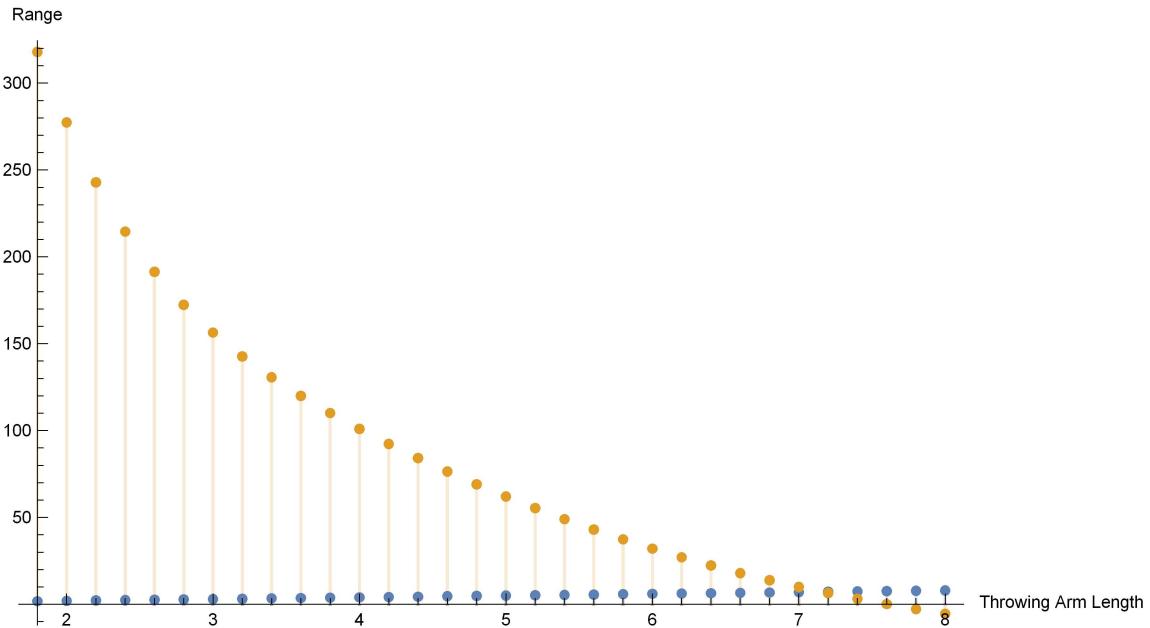


Throwing arm length iteration

```
(*RangeOfProjectile[Throwing Arm length a, weight arm length b,
Weight pendulum length c, height of pivot d, length of sling l,
mass 1 m1, mass to be thrown m2, release angle trigger rA]*)  

DiscretePlot[{x, range /. RangeOfProjectile[x, b, c, d, 6, m1, m2, 3.13, 1]},  

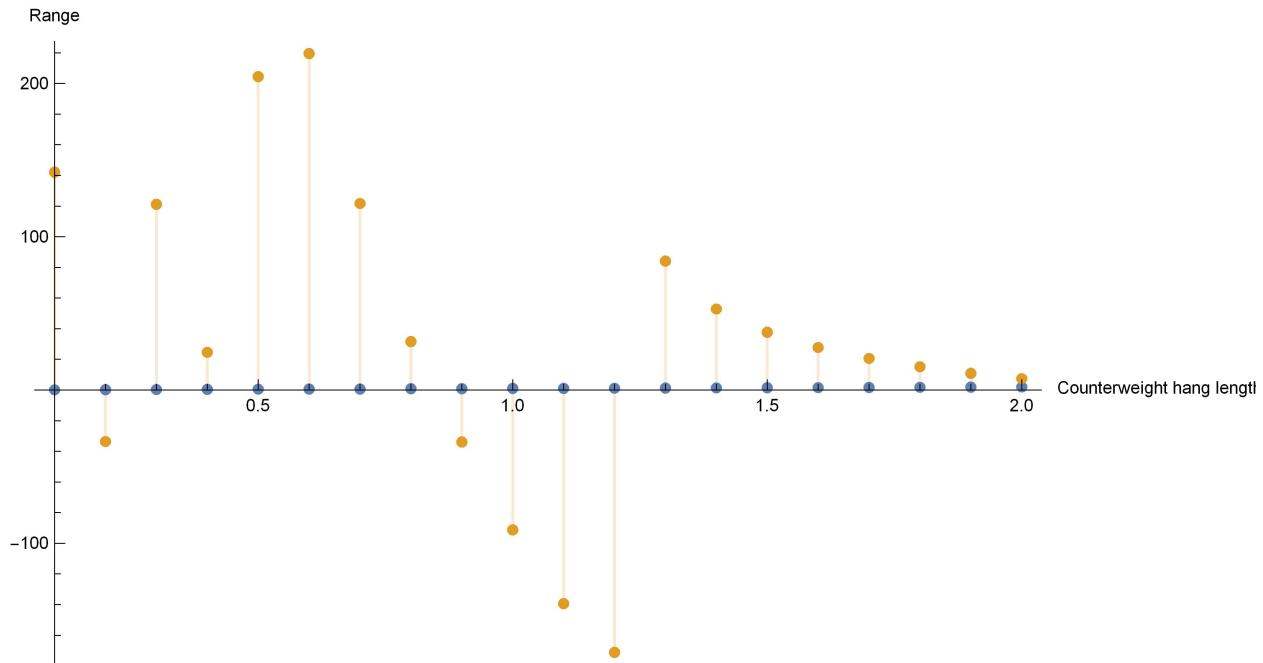
{x, 1.8, 8, 0.2}, AxesLabel -> {"Throwing Arm Length", "Range"}]
```



pendulum length iteration

```
(*RangeOfProjectile[Throwing Arm length a, weight arm length b,
Weight pendulum length c, height of pivot d, length of sling l,
mass 1 m1, mass to be thrown m2, release angle trigger rA]*)
```

```
DiscretePlot[{x, range /. RangeOfProjectile[4, b, x, d, 5, m1, m2, 3.13]}, {x, 0.1, 2, 0.1},  
AxesLabel -> {"Counterweight hang length", "Range"}, PlotRange -> Full]
```



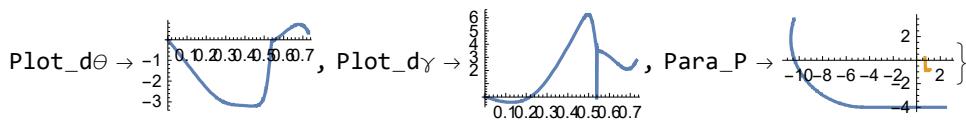
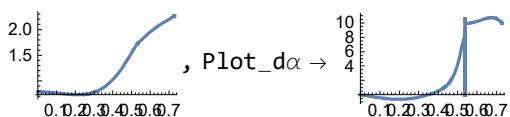
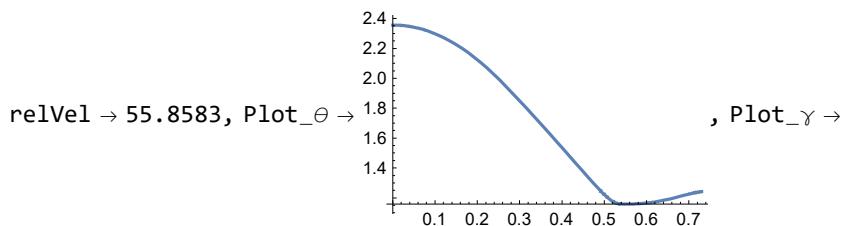
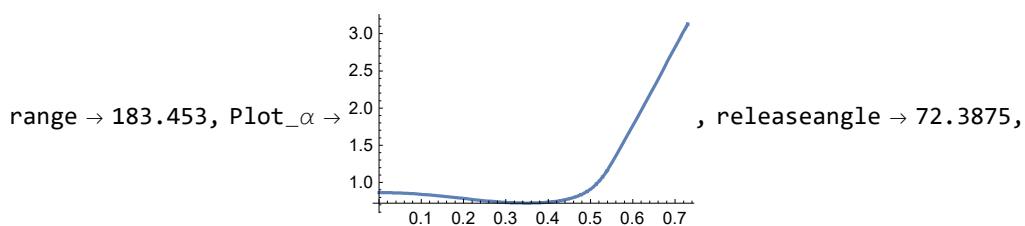
Test at choice of value from iteration

```
RangeOfProjectile[5, b, c, d, 6, m1, m2, 3.13]
```

$\{ \text{solution} \rightarrow \{ \{ \theta \rightarrow \text{InterpolatingFunction} [\text{GraphIcon}, \text{Domain: } \{ \{ 0., 0.73 \} \}, \text{Output: scalar}] \},$

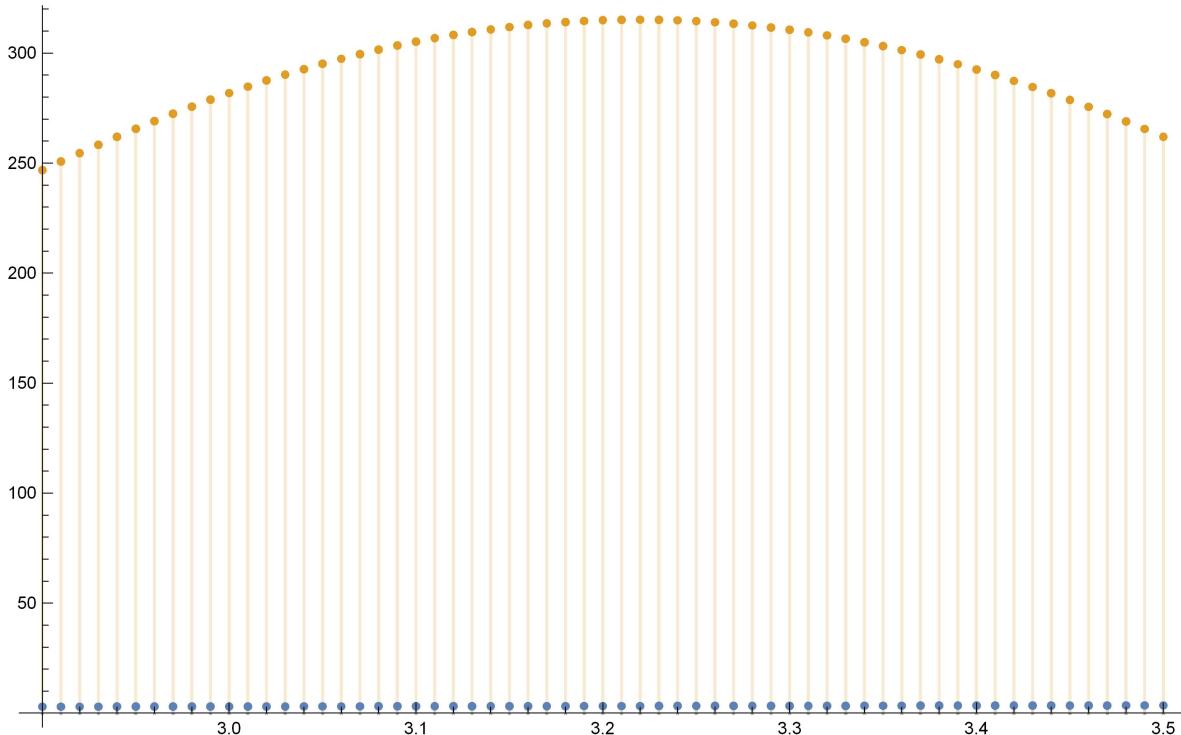
$\gamma \rightarrow \text{InterpolatingFunction} [\text{GraphIcon}, \text{Domain: } \{ \{ 0., 0.73 \} \}, \text{Output: scalar}] \},$

$\alpha \rightarrow \text{InterpolatingFunction} [\text{GraphIcon}, \text{Domain: } \{ \{ 0., 0.73 \} \}] \} \},$



release angle iteration again

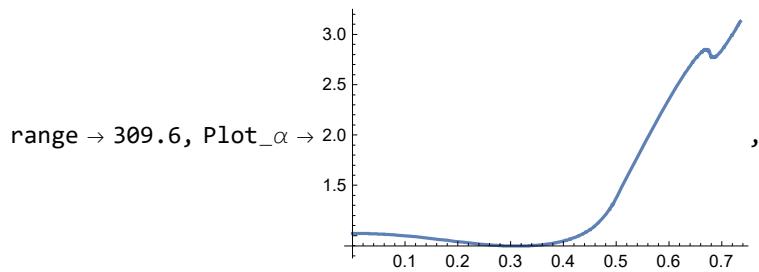
```
DiscretePlot[{x, range /. RangeOfProjectile[4, b, 0.4, d, 5, m1, m2, x]}, {x, 2.9, 3.5, 0.01}, AxesLabel -> {"Release Angle", "Range"}]
```



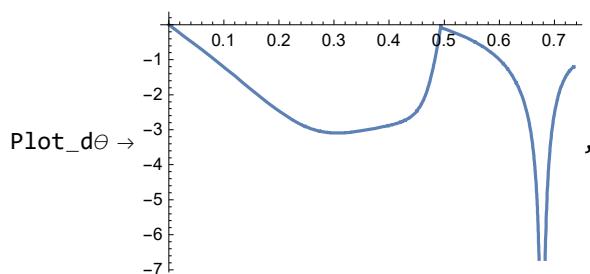
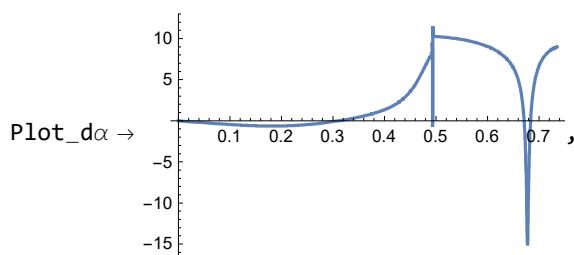
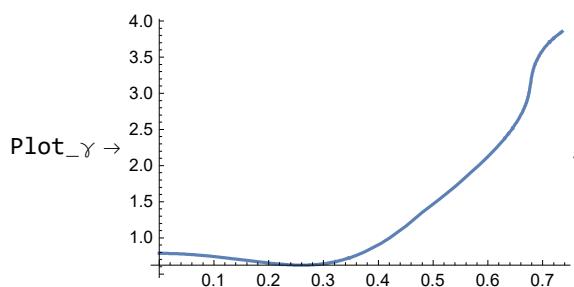
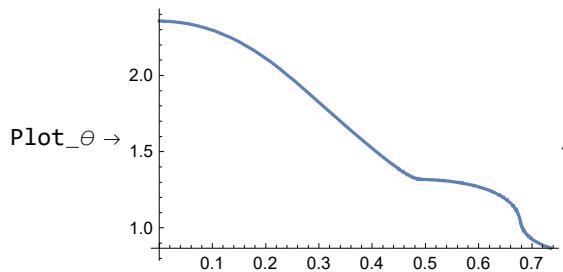
Test at choice of value from iteration

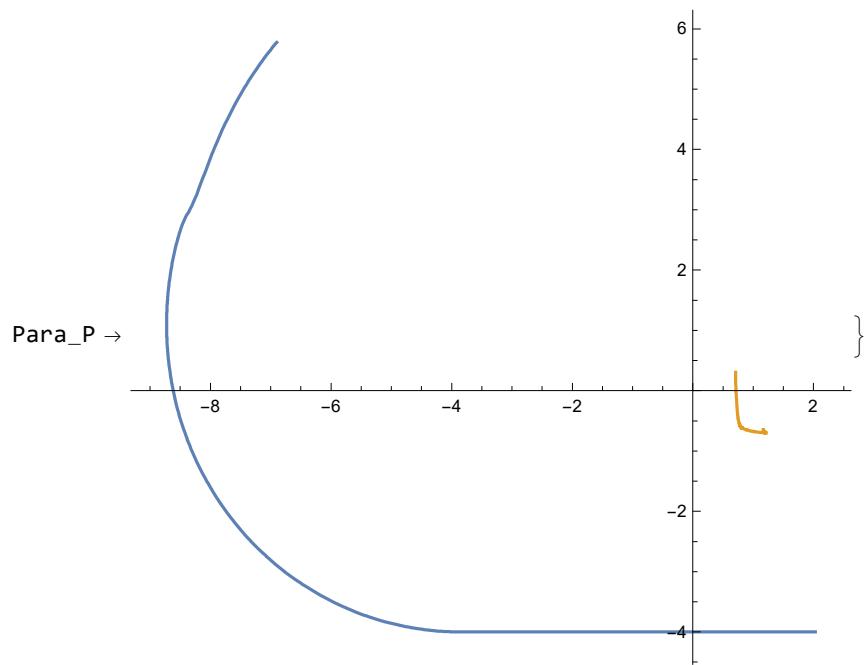
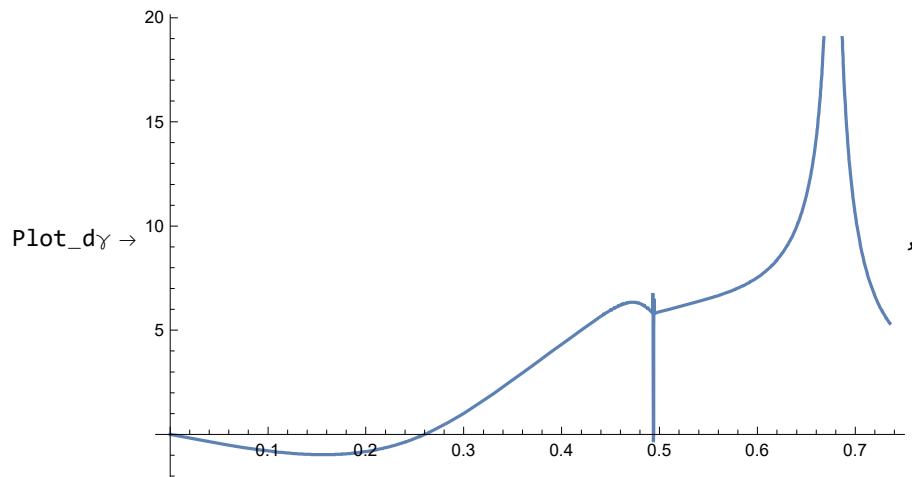
```
RangeOfProjectile[4, b, 0.4, d, 5, m1, m2, 3.13]
```

$\{ \text{solution} \rightarrow \{ \{ \theta \rightarrow \text{InterpolatingFunction} [\text{+ } \text{GraphIcon}, \text{Domain: } \{ \{ 0., 0.735 \} \}, \text{Output: scalar}] \},$
 $\gamma \rightarrow \text{InterpolatingFunction} [\text{+ } \text{GraphIcon}, \text{Domain: } \{ \{ 0., 0.735 \} \}, \text{Output: scalar}] \},$
 $\alpha \rightarrow \text{InterpolatingFunction} [\text{+ } \text{GraphIcon}, \text{Domain: } \{ \{ 0., 0.735 \} \}, \text{Output: scalar}] \} \}$



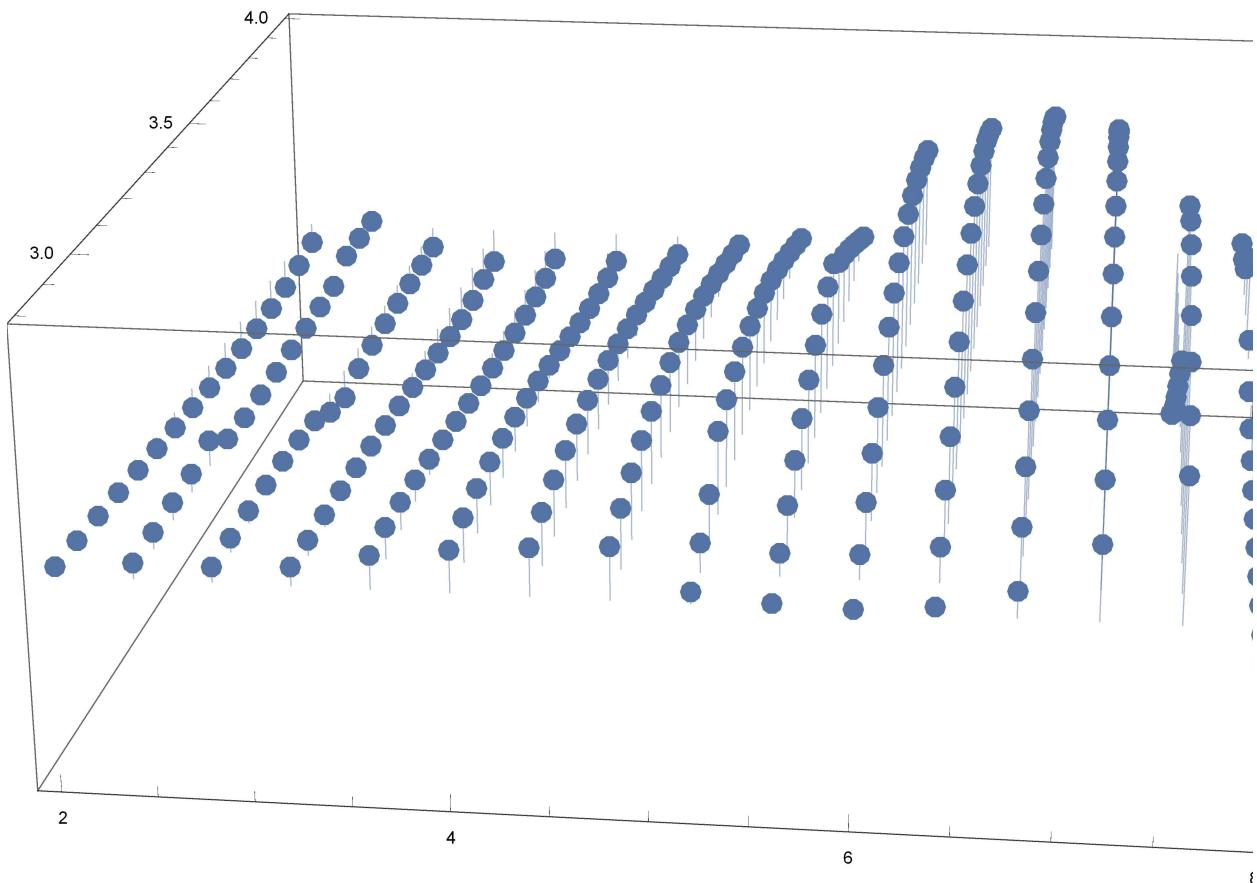
releaseangle → 50.8574, relVel → 55.6937,



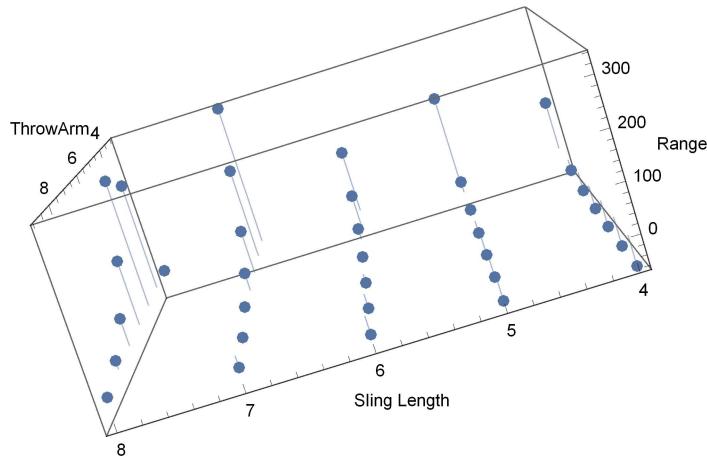


2D iteration across release angle and sling length

```
DiscretePlot3D[range /. RangeOfProjectile[4, b, 0.4, d, x, m1, m2, y, 1], {x, 2, 8, 0.4},  
{y, 2.8, 4, 0.08}, AxesLabel -> {"Sling Length", "Release Angle", "Range"}]
```



```
DiscretePlot3D[range /. RangeOfProjectile[y, b, 0.4, d, x, m1, m2, 3.13, 1],
{y, 4, 8, 1}, {x, 2, 6, 1}, AxesLabel -> {"Sling Length", "ThrowArm", "Range"}]
```



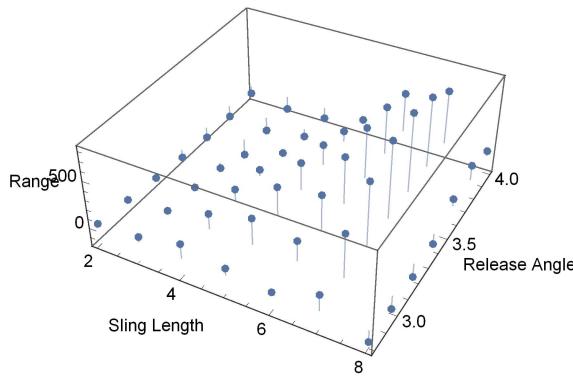
```
DiscretePlot3D[range /. RangeOfProjectile[4, b, 0.4, d, x, m1, m2, y], {x, 2, 8, 1},
{y, 2.8, 4, 0.2}, AxesLabel -> {"Sling Length", "Release Angle", "Range"}]
```

... NDSolve: At t == 0.6499320637752753` step size is effectively zero; singularity or stiff system suspected.

... NDSolve: At t == 0.6499320637752753` step size is effectively zero; singularity or stiff system suspected.

... NDSolve: At t == 0.6499320637752753` step size is effectively zero; singularity or stiff system suspected.

... General: Further output of NDSolve::ndsz will be suppressed during this calculation.



```
data3d = Table[range /. RangeOfProjectile[z, b, 0.4, d, x, m1, m2, y, 1],
{z, 2, 6, 0.5}, {y, 2.8, 3.3, 0.1}, {x, 2, 6, 0.5}]
```

Animation

```
Clear[pen]
```

```

pen[s_] := Module[{g = g, a = 6.4, b = b, c = 0.1,
  d = d, l = 8, m1 = m1, m2 = m2, tmax = 20, ptA, ptB, t, ptm1, ptm2},
  x1[t] = b Sin[θ[t]] - c Sin[θ[t] + γ[t]];
  y1[t] = -b Cos[θ[t]] + c Cos[θ[t] + γ[t]];
  x2[t] := -a Sin[θ[t]] + l Sin[θ[t] - α[t]];
  y2[t] := a Cos[θ[t]] - l Cos[θ[t] - α[t]];
  ptA = {First[-a Sin[θ[s]] /. SOL], First[a Cos[θ[s]] /. SOL]};
  ptB = {First[b Sin[θ[s]] /. SOL], First[-b Cos[θ[s]] /. SOL]};
  ptm1 = {First[x1[s] /. SOL], First[y1[s] /. SOL]};
  ptm2 = {First[x2[s] /. SOL], First[y2[s] /. SOL]};
  Show[Graphics[{Thickness[.01], Line[{{0, 0}, ptA}], PointSize[.03], Black,
    Point[ptA], Thickness[.01], Line[{{0, 0}, ptB}], PointSize[.03], Black,
    Point[ptB], Gray, PointSize[.06], Point[{0, 0}], Black, PointSize[.04],
    Point[{0, 0}], White, PointSize[.02], Point[{0, 0}], Thickness[.005], Blue,
    Line[{ptB, ptm1}], PointSize[.05], Blue, Point[ptm1], Thickness[.005],
    Red, Line[{ptA, ptm2}], PointSize[.02], Red, Point[ptm2]}], Axes → False,
  Ticks → None, AxesStyle → GrayLevel[.5], PlotRange → {{-20, 20}, {-20, 20}}]]

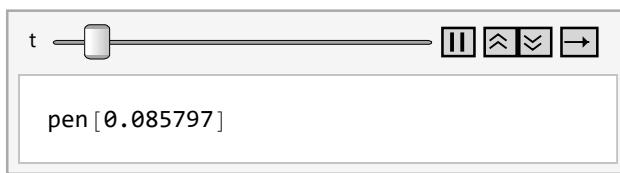
```

$\text{ptA} = \{-a \cos[\theta[s] - 90 \text{ Degree}], -a \sin[\theta[s] - 90 \text{ Degree}]\} /. \text{SOL}[[1]]$

$\left\{ -4 \sin[\text{InterpolatingFunction}\left[\begin{array}{c} \text{+} \\ \text{-->} \end{array} \right] \text{Domain: } \{0., 0.775\} \text{Output: scalar}] [s] \right\}$

$\left\{ 4 \cos[\text{InterpolatingFunction}\left[\begin{array}{c} \text{+} \\ \text{-->} \end{array} \right] \text{Domain: } \{0., 0.775\} \text{Output: scalar}] [s] \right\}$

$\text{mov} = \text{Animate}[\text{pen}[t], \{t, 0, \text{end2}\}, \text{AnimationRate} \rightarrow 0.15, \text{RefreshRate} \rightarrow 50]$



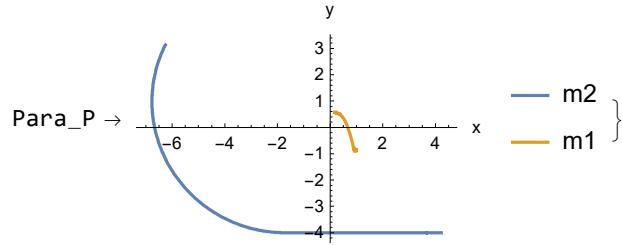
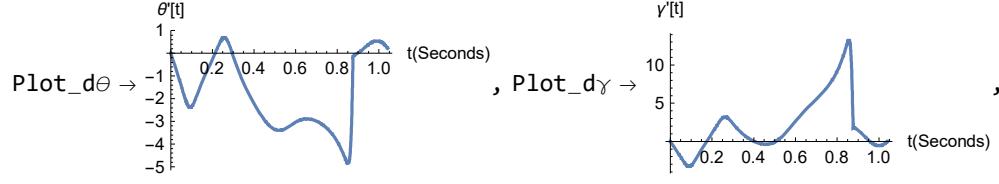
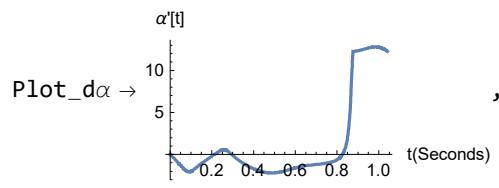
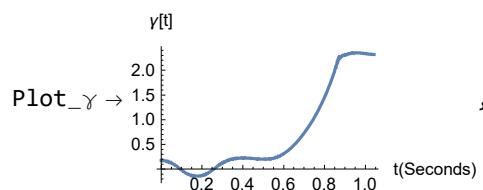
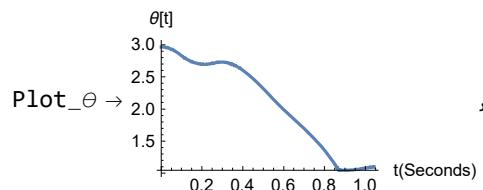
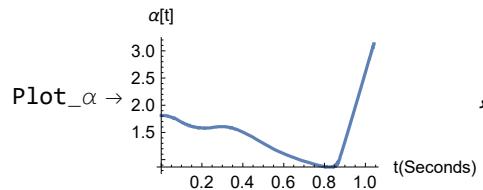
$\text{RangeOfProjectile}[2, b, 0.4, 4, 5, m1, m2, 3.13]$

$\left\{ \text{solution} \rightarrow \left\{ \theta \rightarrow \text{InterpolatingFunction}\left[\begin{array}{c} \text{+} \\ \text{-->} \end{array} \right] \text{Domain: } \{0., 1.04\} \text{Output: scalar} \right\}, \right.$

$\left. \gamma \rightarrow \text{InterpolatingFunction}\left[\begin{array}{c} \text{+} \\ \text{-->} \end{array} \right] \text{Domain: } \{0., 1.04\} \text{Output: scalar} \right\},$

$\left. \alpha \rightarrow \text{InterpolatingFunction}\left[\begin{array}{c} \text{+} \\ \text{-->} \end{array} \right] \text{Domain: } \{0., 1.04\} \text{Output: scalar} \right\} \right\},$

range → 284.205, releaseangle → 64.2954, relVel → 59.7245,



a = 2; c = 0.4; d = 4; l = 5;

data3dr2

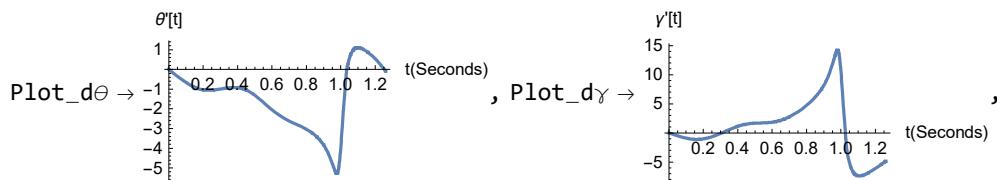
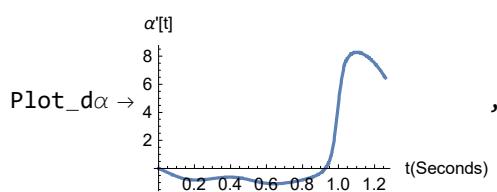
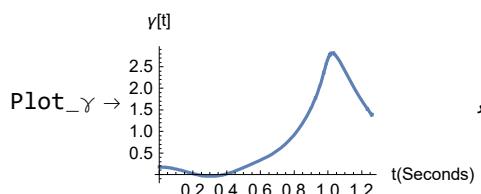
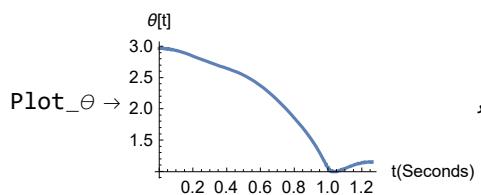
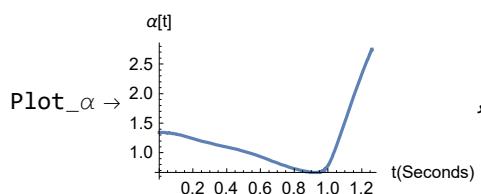
RangeOfProjectile[5.5, b, 0.4, d, 8, m1, m2, 2.75,]

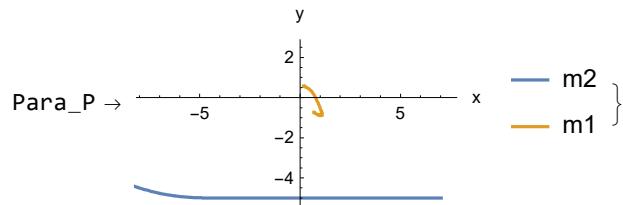
$\{ \text{solution} \rightarrow \{\{\theta \rightarrow \text{InterpolatingFunction}[\text{GraphIcon}, \text{Domain: } \{0., 1.26\}], \text{Output: scalar}\}\},$

$\gamma \rightarrow \text{InterpolatingFunction}[\text{GraphIcon}, \text{Domain: } \{0., 1.26\}], \text{Output: scalar}],$

$\alpha \rightarrow \text{InterpolatingFunction}[\text{GraphIcon}, \text{Domain: } \{0., 1.26\}], \text{Output: scalar}\}] \}$

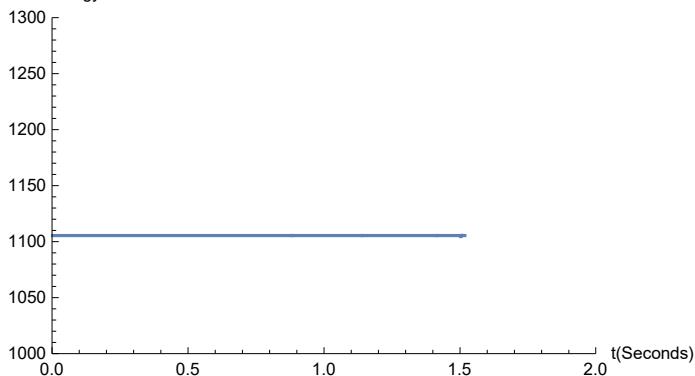
range $\rightarrow 14.9923$, releaseangle $\rightarrow 88.4887$, relVel $\rightarrow 52.8125$,





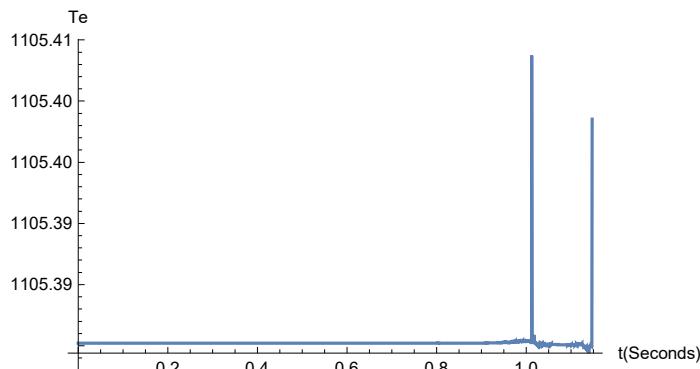
```
Plot[Te + V /. SOL, {t, 0, end2},
AxesLabel → {"t(Seconds)", "Total Energy"}, PlotRange → {{0, 2}, {1000, 1300}}]
```

Total Energy

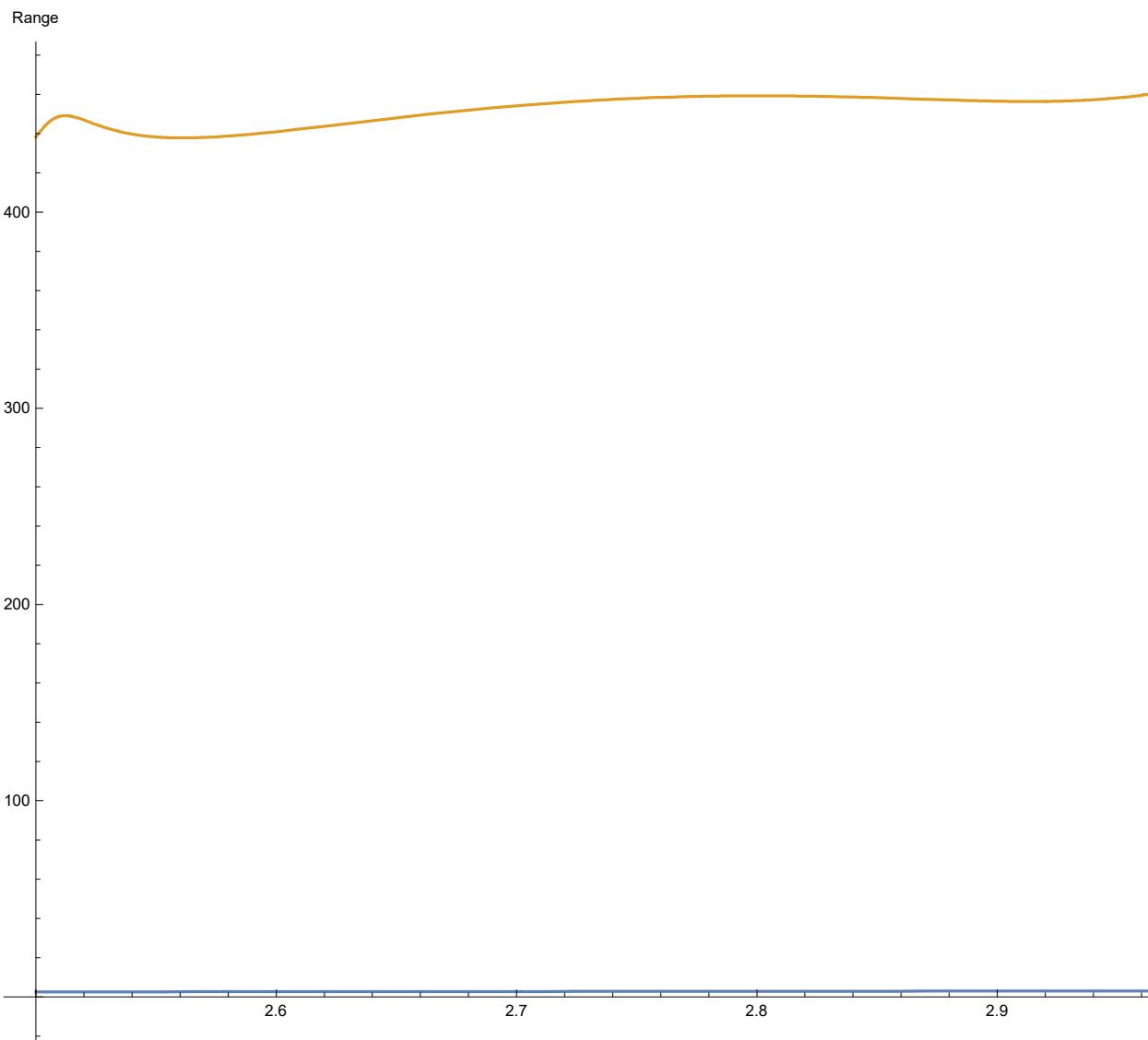


```
DiscretePlot[{x, range /. RangeOfProjectile[6.4, b, 0.1, d, 8, m1, m2, x]},
{x, 2, 3.15, 0.01}, AxesLabel → {"Release Angle", "Range"}]
```

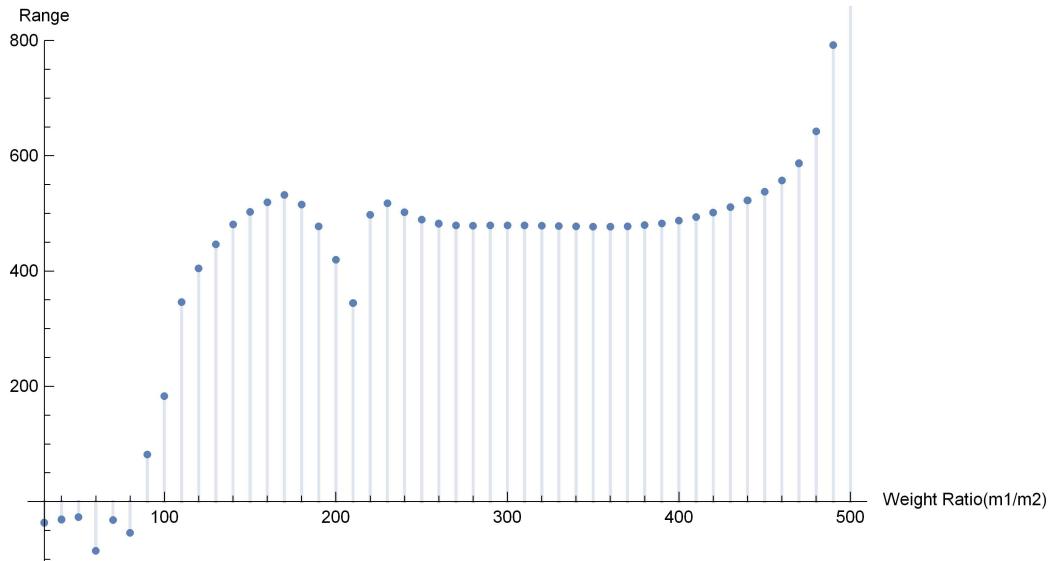
```
Plot[Te1 + V1 /. sol1, {t, 0, end}, AxesLabel → {"t(Seconds)", "Te"}, PlotRange → Full]
```



```
Plot[{x, range /. RangeOfProjectile[6.4, b, 0.1, d, 8, m1, m2, x, 1]},  
{x, 2.5, 3.15}, AxesLabel -> {"Release Angle", "Range"}]
```



```
DiscretePlot[{range /. RangeOfProjectile[6.4, b, 0.1, d, 8, x, 1, 2.83, 1]}, {x, 30, 500, 10}, AxesLabel -> {"Weight Ratio(m1/m2)", "Range"}]
```

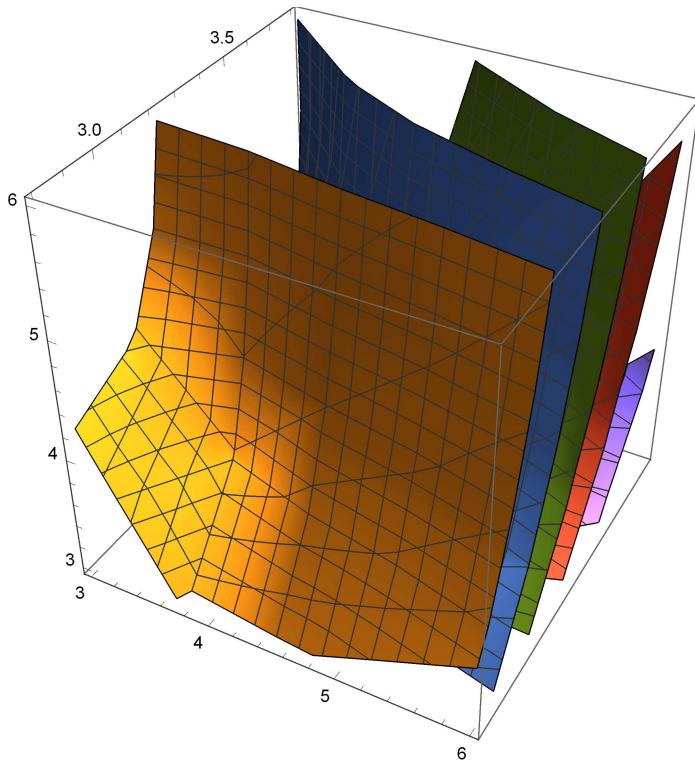


sol1

```
{γ → InterpolatingFunction[ Domain: {{0., 2.}}  
Output: scalar],  
θ → InterpolatingFunction[ Domain: {{0., 2.}}  
Output: scalar]} } }
```

3-D and 4-D Iterations and Visualization

```
data3dr = Table[range /. RangeOfProjectile[z, b, 0.4, d, x, m1, m2, y, 1],
{z, 3, 6, 0.75}, {y, 2.8, 3.4, 0.15}, {x, 3, 6, 0.75}]
ListContourPlot3D[data3dr, DataRange -> {{3, 6}, {2.8, 3.4}, {3, 6}}, Contours -> 5]
```



(*DO not run the next cell Use the text data from below,
it takes more than 2 Hours to run*)

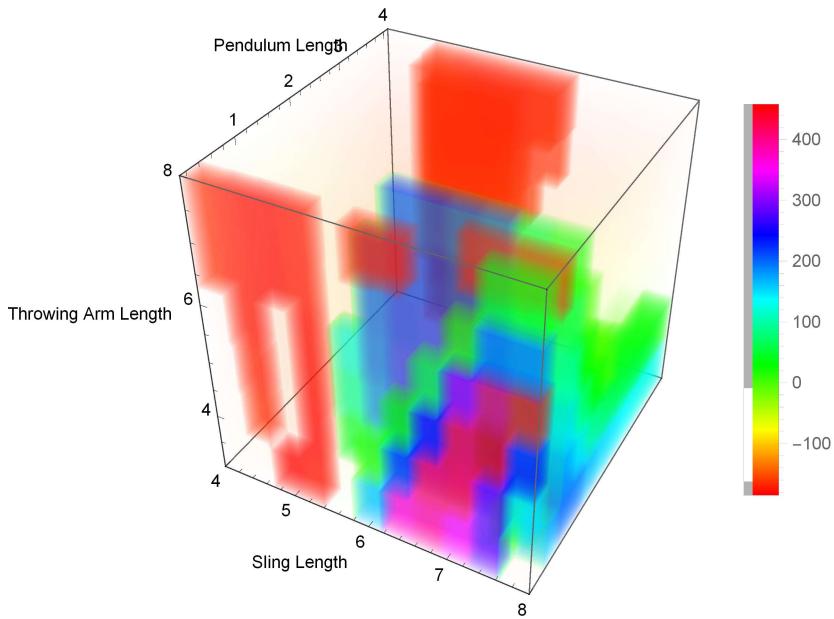
```
data3dr2 = Table[range /. RangeOfProjectile[z, b, y, d, x, m1, m2, u, 1],
{z, 4, 8, 0.8}, {y, 0.1, 4.1, 0.4}, {x, 3, 8, 0.5}, {u, 2.75, 3.15, 0.08}]
```

{ ... 1 ... }

[large output](#) [show less](#) [show more](#) [show all](#) [set size limit...](#)

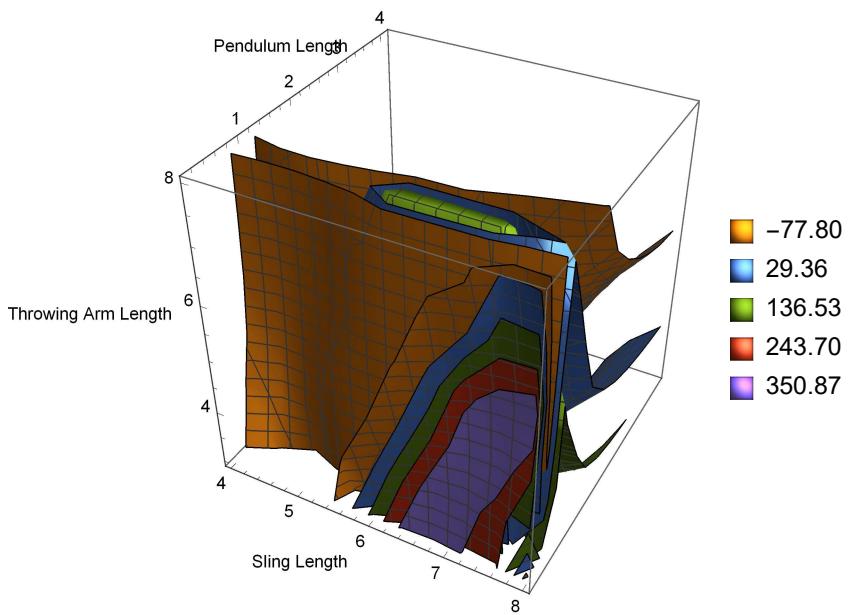
```
Labeled[ListDensityPlot3D[data3dr2[[All, All, All, 1]],  
  DataRange -> {{4, 8}, {0.1, 4.1}, {3, 8}}, ColorFunction -> Hue, PlotLegends -> Automatic,  
  AxesLabel -> {"Sling Length", "Pendulum Length", "Throwing Arm Length"}],  
 "Density Plot - Range of projectile at release angle 2.75", Top]
```

Density Plot – Range of projectile at release angle 2.75

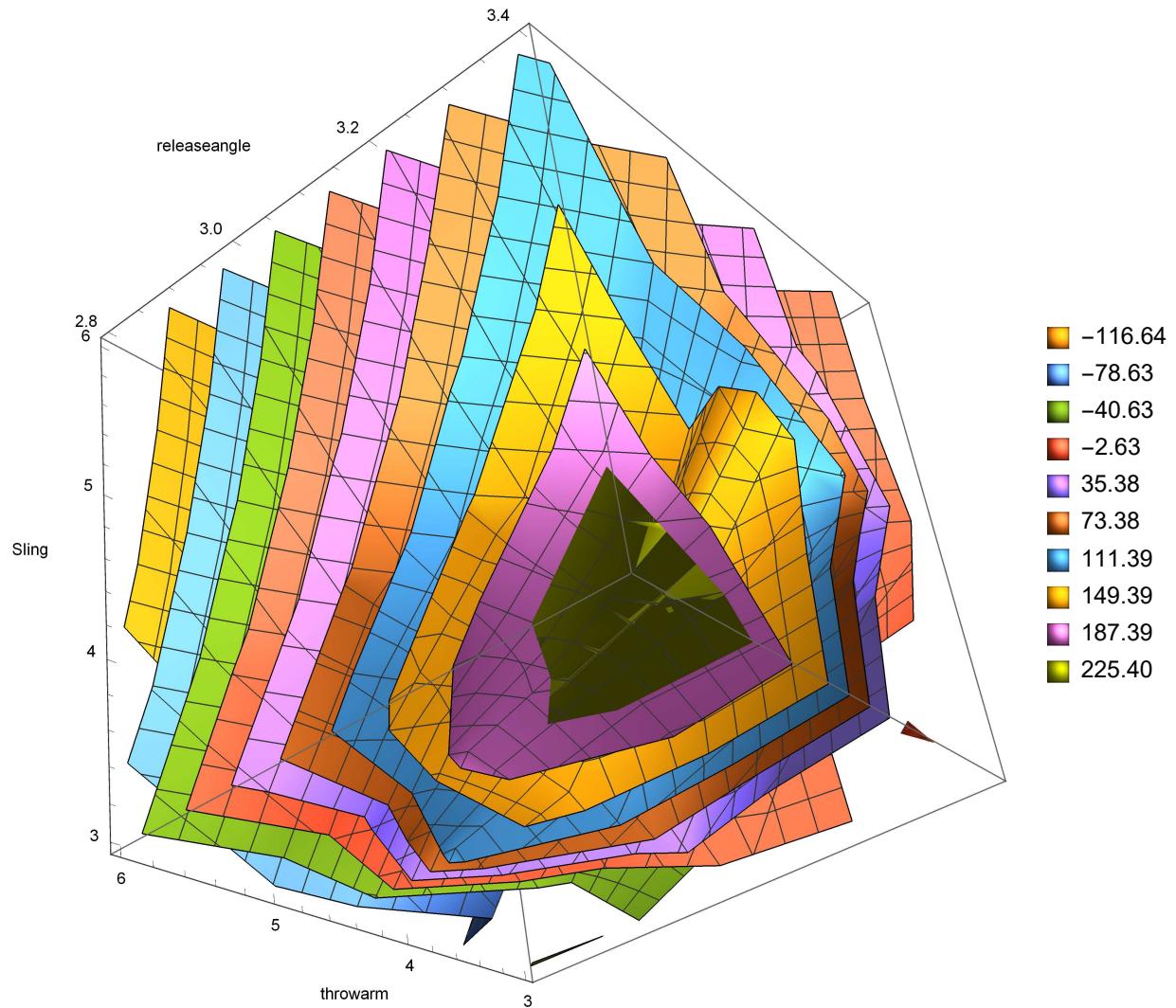


```
Labeled[ListContourPlot3D[data3dr2[[All, All, All, 1]],  
  DataRange -> {{4, 8}, {0.1, 4.1}, {3, 8}}, Contours -> 5, PlotLegends -> "Expressions",  
  AxesLabel -> {"Sling Length", "Pendulum Length", "Throwing Arm Length"}],  
 "Contour Plot - Range of projectile at release angle 2.75", Top]
```

Contour Plot – Range of projectile at release angle 2.75



```
ListContourPlot3D[data3dr, DataRange -> {{3, 6}, {2.8, 3.4}, {3, 6}}, Contours -> 10,  
PlotLegends -> "Expressions", AxesLabel -> {"throwarm", "releaseangle", "Sling"}]
```



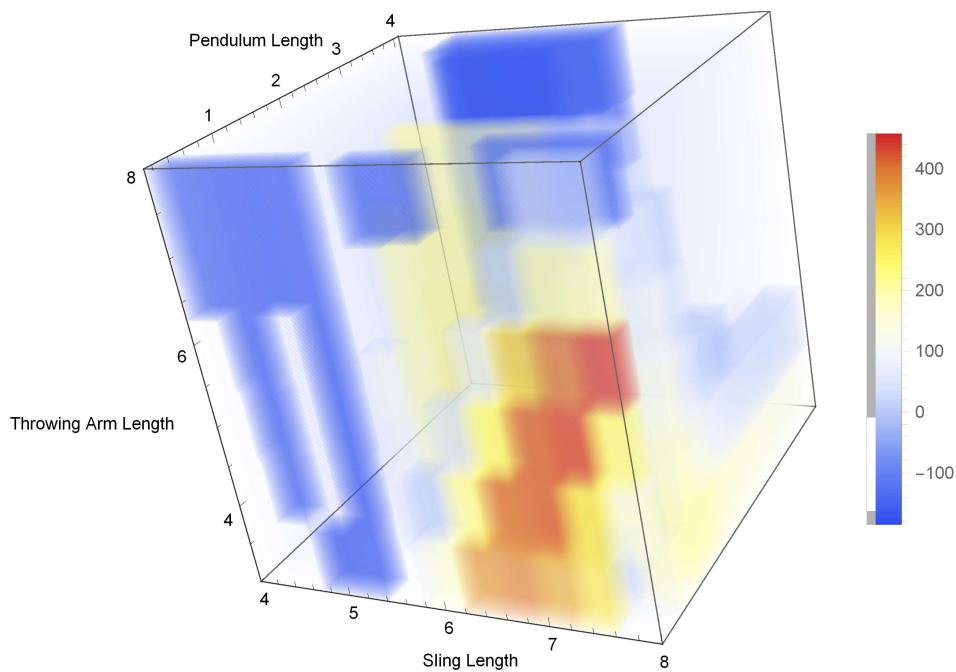
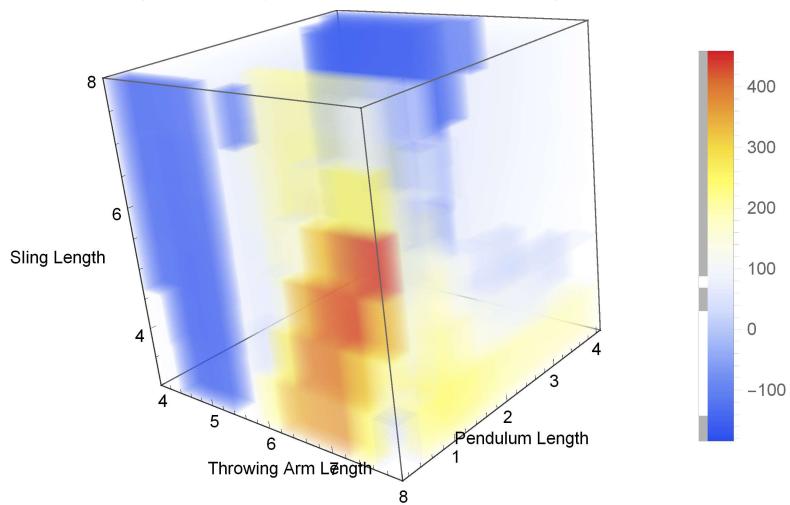
```
Export["test3.dat", data3dr2]
```

test3.dat

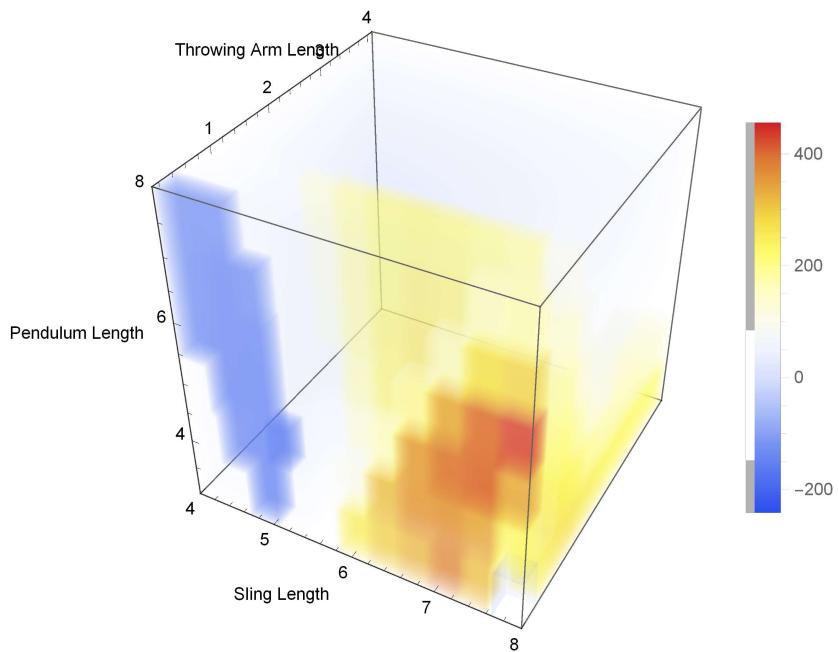
```

Labeled[
  ListDensityPlot3D[data3dr2[[All, All, All, 1]], DataRange → {{4, 8}, {0.1, 4.1}, {3, 8}},
  ColorFunction → "TemperatureMap", PlotLegends → Automatic,
  AxesLabel → {"Sling Length", "Pendulum Length", "Throwing Arm Length"}],
  "Range of projectile at release angle 2.75", Top]
Labeled[ListDensityPlot3D[data3dr2[[All, All, All, 2]]],
  DataRange → {{4, 8}, {0.1, 4.1}, {3, 8}},
  PlotLegends → Automatic, ColorFunction → "TemperatureMap",
  AxesLabel → {"Throwing Arm Length", "Pendulum Length", "Sling Length"}],
  "Range of projectile at release angle 2.83", Top]
Labeled[ListDensityPlot3D[data3dr2[[All, All, All, 3]]],
  DataRange → {{4, 8}, {0.1, 4.1}, {3, 8}},
  PlotLegends → Automatic, ColorFunction → "TemperatureMap",
  AxesLabel → {"Sling Length", "Throwing Arm Length", "Pendulum Length"}],
  "Range of projectile at release angle 2.91", Top]
Labeled[ListDensityPlot3D[data3dr2[[All, All, All, 4]]],
  DataRange → {{4, 8}, {0.1, 4.1}, {3, 8}},
  PlotLegends → Automatic, ColorFunction → "TemperatureMap",
  AxesLabel → {"Throwing Arm Length", "Pendulum Length", "Sling Length"}],
  "Range of projectile at release angle 2.99", Top]
Labeled[ListDensityPlot3D[data3dr2[[All, All, All, 5]]],
  DataRange → {{4, 8}, {0.1, 4.1}, {3, 8}},
  PlotLegends → Automatic, ColorFunction → "TemperatureMap",
  AxesLabel → {"Throwing Arm Length", "Pendulum Length", "Sling Length"}],
  "Range of projectile at release angle 3.07", Top]
Labeled[ListDensityPlot3D[data3dr2[[All, All, All, 6]]],
  DataRange → {{4, 8}, {0.1, 4.1}, {3, 8}},
  PlotLegends → Automatic, ColorFunction → "TemperatureMap",
  AxesLabel → {"Throwing Arm Length", "Pendulum Length", "Sling Length"}],
  "Range of projectile at release angle 3.15", Top]

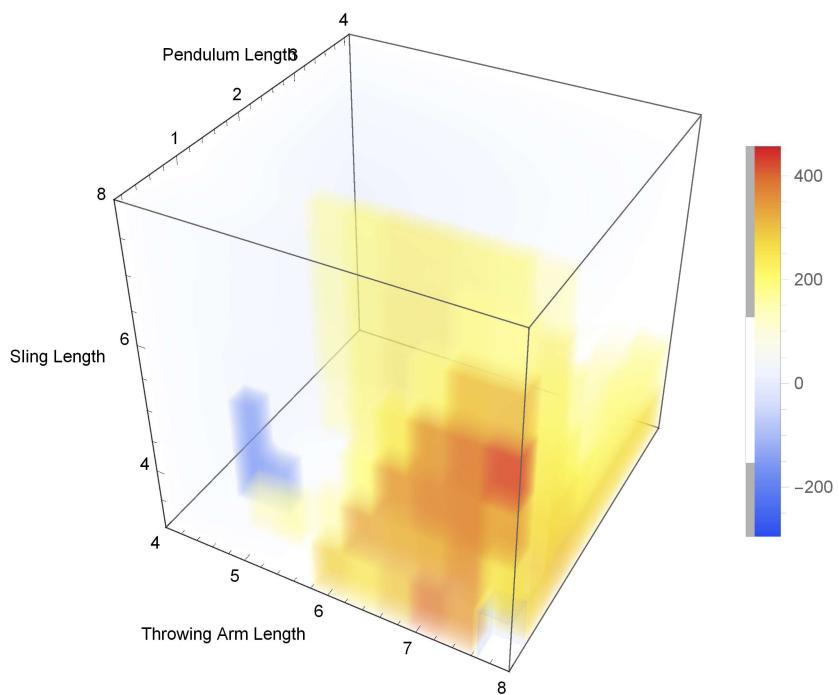
```

"Range of projectile at release angle 2.75"**"Range of projectile at release angle 2.83"**

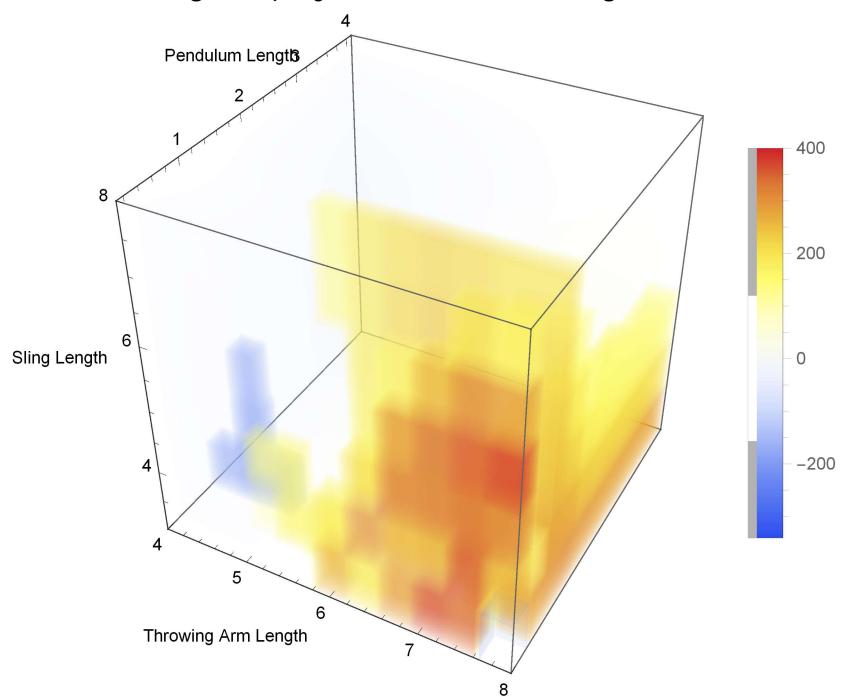
Range of projectile at release angle 2.91



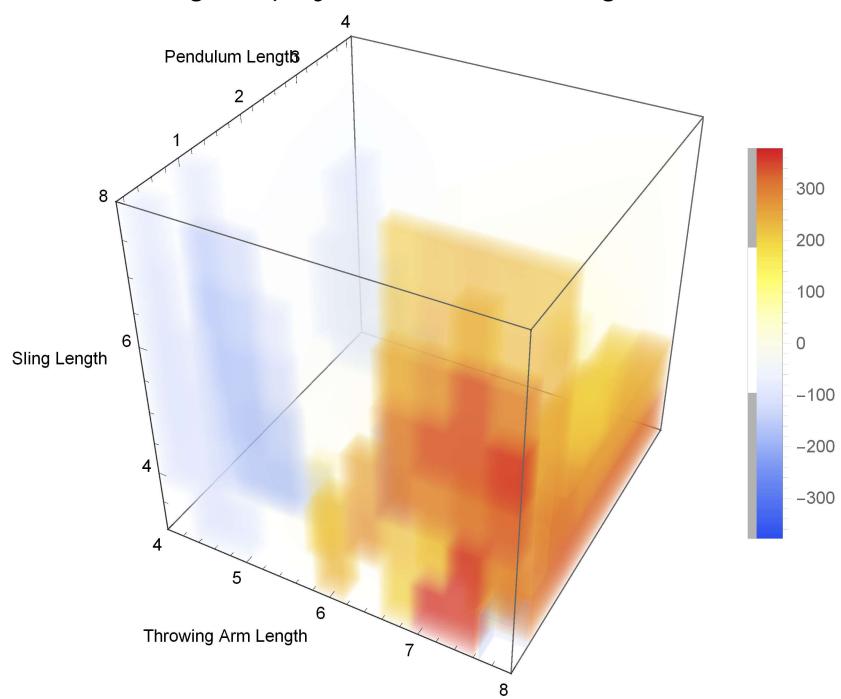
Range of projectile at release angle 2.99



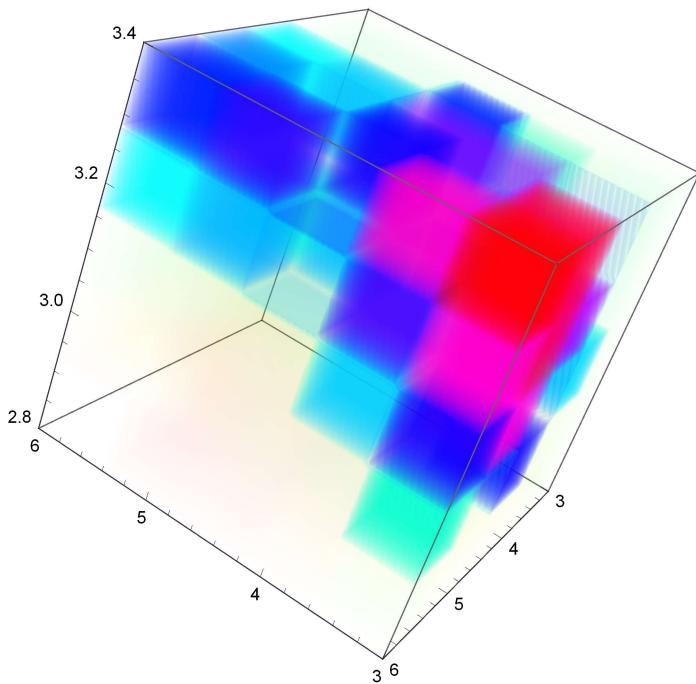
Range of projectile at release angle 3.07



Range of projectile at release angle 3.15



```
ListDensityPlot3D[data3dr, DataRange -> {{, 6}, {2.8, 3.4}, {3, 6}},
ColorFunction -> Hue, PlotLegends -> "Expressions"]
```



Final Optimum Parameters and Run

$a = 6.4$, $b = 1$, $c = 0.1$, $d = 5$, $l = 8$, $m_1/m_2 = 133$, Release angle = 2.83

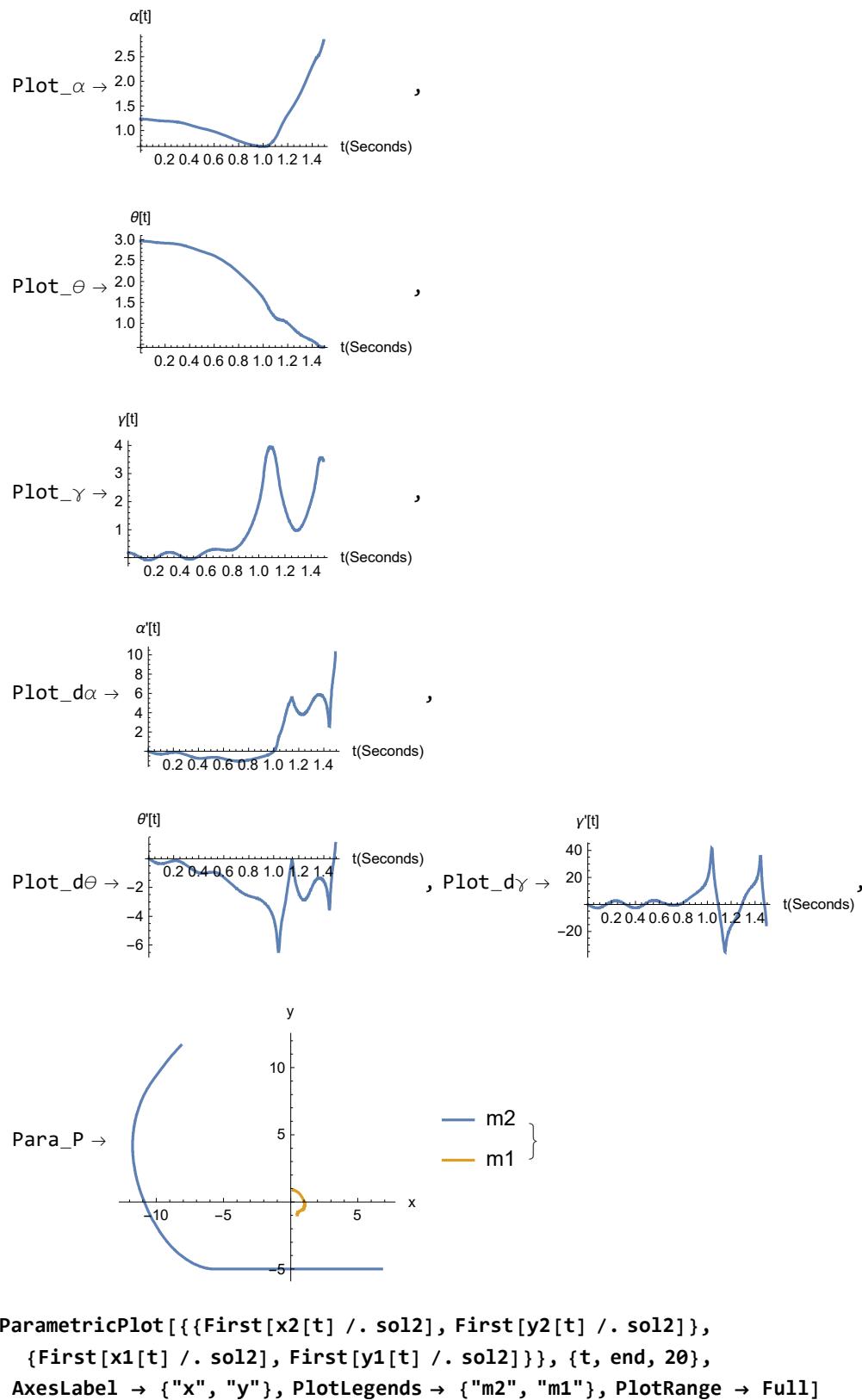
```
RangeOfProjectile[6.4, b, 0.1, d, 8, m1, m2, 2.83]
```

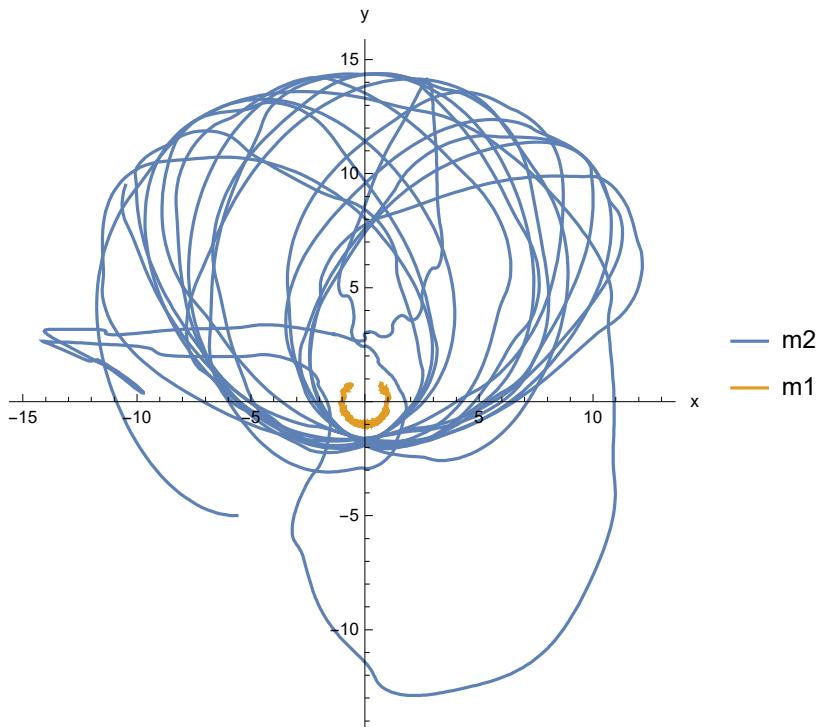
{solution → {θ → InterpolatingFunction[ Domain: {{0., 1.49}}], Output: scalar}}

γ → InterpolatingFunction[ Domain: {{0., 1.49}}], Output: scalar]

α → InterpolatingFunction[ Domain: {{0., 1.49}}], Output: scalar]}

range → 458.928, releaseangle → 45.0032, relVel → 67.0976,





{ ... 1 ... }

large output

show less

show more

show all

set size limit...