(* Ground Zero Model; Web: www.911history.de
As of: 2019-01-04; Contents: Creative Commons *)

(* Power Source: Uranium fission; Chain reaction inhibitor: Thorium;

Cooling agent: liquid granite (natural convection);
Final ignition: pressure-induced phase separation uranium/thorium mix *)

(* shows for WTC2: 'glacier' valley and hot melt-
funnel after eruption and a progressive (100 seconds) 125 kt Energy Input,

followed by a final 25 kt nuclear fizzle reaction (6 seconds) *)
(* shows for WTC1: screenshot of the dynamics during the nuclear plasma eruption,
acceleration and direction defined by the melt-
funnel structure / pit below freight elevator 50 *)
(* shows for WTC7: creation of the eruptive melt-funnel and blocking of the
eruption due to ingressing water (cooling of the liquefied granite) *)

(* Ignition (nuclear fizzle) is naturally
achieved by: 'void coefficient of reactivity'
changes from negative to positive under extreme pressure,
as soon as the boiling uranium/thorium mixture has burned it's way deep
enough down into the rock, with a heavy liquid granite dome above *)

(* As soon as the boiling stops under pressure,
liquid thorium and liquid uranium can separate: https://
en.wikipedia.org/wiki/Phase_(matter) *)

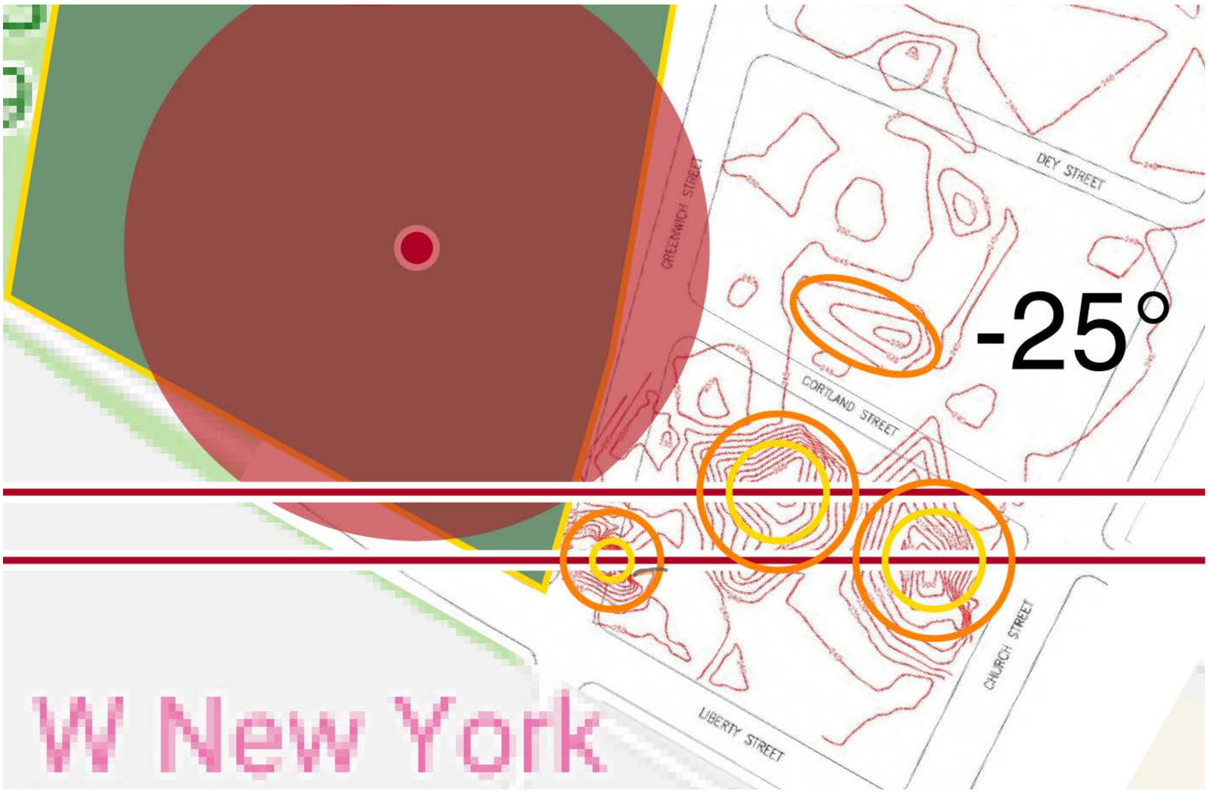
(* Uranium: Melting point: 1132 °C | Boiling point: 4131 °C *)
(* Thorium: Melting point: 1750 °C | Boiling point: 4788 °C *)
(* Granite: Melting point: 1250 °C |
Boiling point: 2200 - 2900 °C [Silicon dioxide, as one of many components]*)

(* Theory: Silverstein Glacier Valley is the result of
a displaced [far reaching] convection flow of liquid granite,
a solidified testimony of the energy flow / energy landscape *)
(* Destruction of the Twin Towers is the result of x-
and gamma-radiation induced weakening of the steel core,
followed by a short-lived nuclear plasma eruption, upshot: 350 m *)

(* Map: overlay of Google maps and elevation data of the glacier valley;

source1 (photos): https://www.nytimes.com/2008/09/22/nyregion/22rocks.html *)
(* Map: overlay of Google maps and elevation data of the glacier valley;

source2 (elevations):
https://dspace.sunyconnect.suny.edu/bitstream/handle/1951/47900/moss-09.pdf *)
(* NOTE: the valley may have indeed existed BEFORE the nuclear event, but was
most likely transformed by it *)




```
rockbed =
 Polygon[{{-500, -500, -200}, {-500, 500, -200}, {500, 500, -200}, {500, -500, -200}}]
 Vieleck
soil = Cuboid[{-500, -500, 75}, {500, 500, 90}]
       Quader
river = Triangle[{{-500, -100, 90}, {-415, 500, 90}, {-500, 500, 90}}]
        Dreieck
bathtub =
 Line[{{-52.5, 312.5, 100}, {-62.5, 255, 100}, {-65, 227.5, 100}, {-85, 97.5, 100}
 Linie
   , {-92.5, 62.5, 100}, {-105, -12.5, 100}, {32, -87.5, 100}, {50, -27.5, 100},
   {75, 115, 100}, {90, 180, 100}, {112.5, 222, 100}, {-52.5, 312, 100}}]

WTC1 = GeometricTransformation[Cuboid[{-44.5, 105, 100}, {19.5, 169, 517}],
       geometrische Transformation  Quader
  RotationTransform[-29 Degree, {0, 0, 1}, {-12.5, 137, 275}]]
                        Grad

Antenna = Cylinder[{{-12.5, 137, 517}, {-12.5, 137, 627}}, 3]
          Zylinder
WTC2 = GeometricTransformation[ Cuboid[{-32, 32, 100}, {32, -32, 515}],
       geometrische Transformation   Quader
  RotationTransform[-29 Degree, {0, 0, 1}, {0, 0, 275}]]
                        Grad
WTC7base = Polygon[{{65, 337.5, 100}, {175, 277.5, 100},
           Vieleck
   {137.5, 232.5, 100}, {52.5, 280, 100}, {65, 337.5, 100}}]
WTC7roof = Polygon[{{65, 337.5, 290}, {175, 277.5, 290},
           Vieleck
   {137.5, 232.5, 290}, {52.5, 280, 290}, {65, 337.5, 290}}]
```

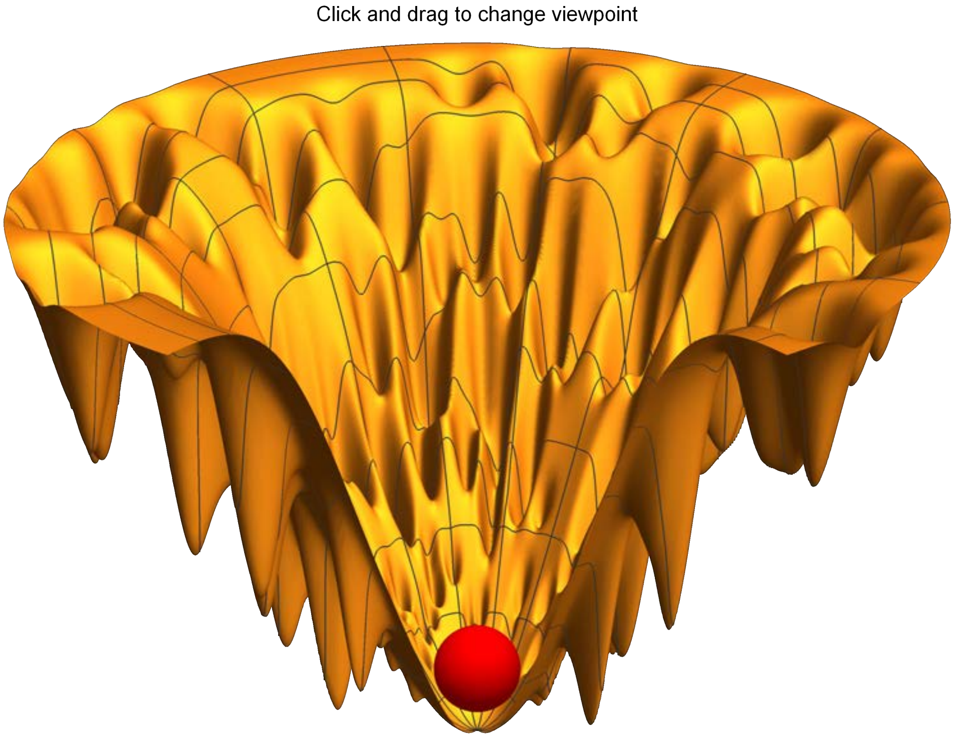
```
WTC7roofBoxHigh = GeometricTransformation[Cuboid[{115, 252.5, 290}, {155, 287.5, 300}],
                  geometrische Transformation  Quader
  RotationTransform[-29 Degree, {0, 0, 1}, {135, 270, 300}]]
  Rotationstransformation     Grad
WTC7roofBoxLow = GeometricTransformation[Cuboid[{77.5, 285, 290}, {122.5, 310, 295}],
                 geometrische Transformation  Quader
  RotationTransform[-29 Degree, {0, 0, 1}, {100, 297.5, 295}]]
  Rotationstransformation     Grad
WTC7north = Polygon[{{65, 337.5, 100}, {175, 277.5, 100},
            Vieleck
    {175, 277.5, 290}, {65, 337.5, 290}, {65, 337.5, 100}}]
WTC7south = Polygon[{{137.5, 232.5, 100}, {52.5, 280, 100},
            Vieleck
    {52.5, 280, 290}, {137.5, 232.5, 290}, {137.5, 232.5, 100}}]
WTC7east = Polygon[{{65, 337.5, 100}, {52.5, 280, 100},
           Vieleck
    {52.5, 280, 290}, {65, 337.5, 290}, {65, 337.5, 100}}]
WTC7west = Polygon[{{175, 277.5, 100}, {137.5, 232.5, 100},
           Vieleck
    {137.5, 232.5, 290}, {175, 277.5, 290}, {175, 277.5, 100}}]
Z1 = Sphere[{-12.5, 137, 0}, 50]
     Kugel
Z2 = Sphere[{0, 0, 0}, 2]
     Kugel
Z7 = Sphere[{145, 272.5, 0}, 25]
     Kugel
F1 = Sphere[{-12.5, 137, 0}, 25]
     Kugel
F2  = Sphere[{0, 0, 0}, 50]
     Kugel
F7 = Sphere[{145, 272.5, 0}, 15]
     Kugel
aE1 = 6; bE1 = 4; cE1 = 2;
aJ1 = .5; bJ1 = .5;

Air1 =  Cylinder[{{-12.5, 137, 455}, {-12.5, 137, 510}}, 32]
        Zylinder
Air2 = Cylinder[{{0, 0, 420}, {0, 0, 510}}, 32]
       Zylinder
Air7 =  Cylinder[{{145, 272.5, 125}, {145, 272.5, 275}}, 20]
        Zylinder

chimney1 = Cylinder[{{-12.5, 137, 75}, {-12.5, 137, 415}}, 5]
           Zylinder
chimney2 = Cylinder[{{0, 0, 75}, {0, 0, 415}}, 5]
           Zylinder
chimney7  =  Cylinder[{{145, 272.5, 75}, {145, 272.5, 150}}, 5]
             Zylinder
cone1 = Cone[{{-12.5, 137, 475}, {-12.5, 137, 0}}, 10]
        Kegel
cone2 = Cone[{{0, 0, 475}, {0, 0, 0}}, 10]
        Kegel
cone7 = Cone[{{145, 272.5, 350}, {145, 272.5, 0}}, 10]
        Kegel
```

```
Kegel
valley1OLD = Ellipsoid[{50, -80, 75}, {12.5, 12.5, 20}]
                 Ellipsoid
e1 = 0.625; (* definition of ellipsoid 1 for PotHole 1; x and y radius are equal*)
valley1 = 75 - Sqrt[400 - (x - 50) ^2 / e1^2 - (y + 80) ^2 / e1^2]
                  Quadratwurzel

valley2OLD = Sphere[{92.5, -62.5, 75}, 20]
                Kugel
valley2 = 75 - Sqrt[400 - (x - 92.5) ^2 - (y + 62.5) ^2]
                  Quadratwurzel
valley3OLD = Sphere[{132.5, -80, 75}, 20]
                 Kugel
valley3 = 75 - Sqrt[400 - (x - 132.5) ^2 - (y + 80) ^2]
                  Quadratwurzel
valley4OLD = GeometricTransformation[Ellipsoid[{115, -20, 75}, {40, 20, 10}],
                geometrische Transformation    Ellipsoid
  RotationTransform[-25 Degree, {0, 0, 1}, {115, -20, 75}]]
  Rotationstransformation        Grad
e4x = 4; (* definition of ellipsoid 4 for PotHole 4; x and y radius*)
e4y = 2; (* definition of ellipsoid 4 for PotHole 4; x and y radius*)
valley4 = 75 - Sqrt[100 - (x - 115) ^2 / e4x^2 - (y + 20) ^2 / e4y^2]
                  Quadratwurzel

fixpoint = Sphere[{500, 500, 825}, 0.1]
             Kugel
```

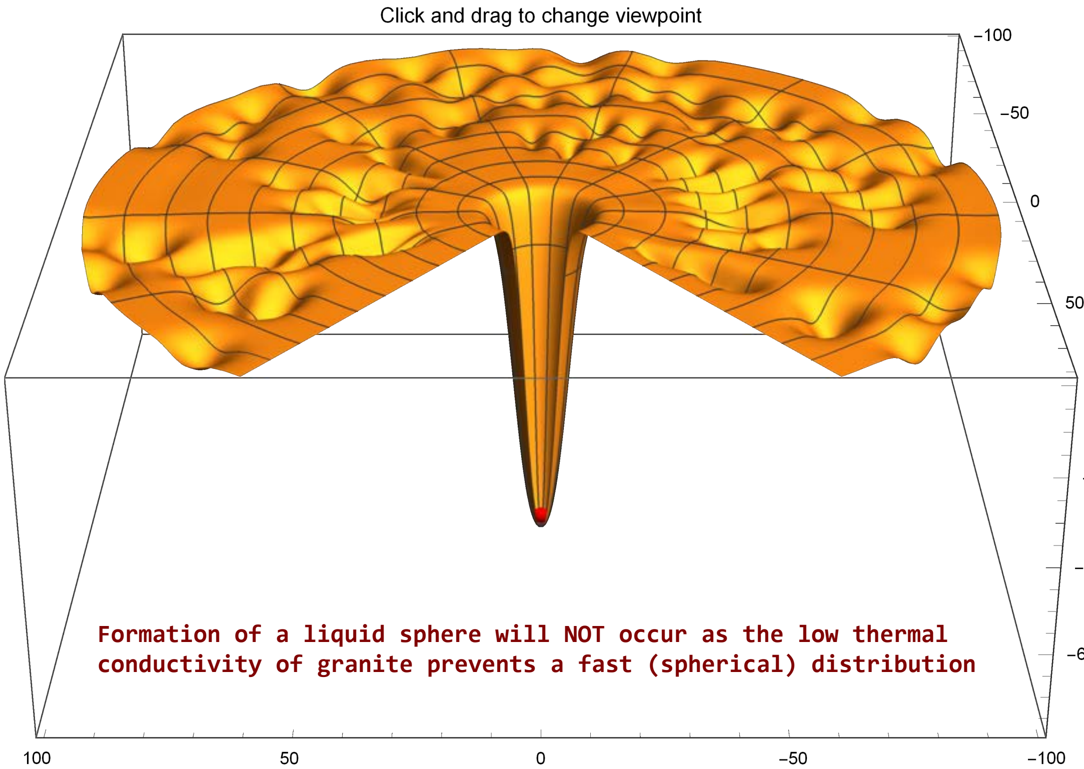
https://dspace.sunyconnect.suny.edu/bitstream/handle/1951/47900/moss-09.pdf

```
(* Original title of the mathematical funnel
 model: frustrated energy landscapes (folding funnels);
Source/Credit: Terry Oas (oas@duke.edu )*)
(* Source/Credit: https://oaslab.com/drawing_funnels.html  *)
```

(*



At ~ 4000 °C the uranium/thorium sphere liquefies the surrounding rock, sinking down and creating a funnel; liquefied granite acts a natural cooling agent



*)

```
(*  Set general appearance of funnel *)
     weise zu
SeedRandom[5];
sähe Zufall
(*to get the same funnel every time, leave seed fixed; to get a new funnel,
change this number, to get a random funnel, remove number from []*)
Nmesh = 10;
(*Number of mesh lines drawn on funnel, the more lines,
  Zahl
the slower rendering and interactive rotation*)
HiCurve = 16; (*Curvature of the upper part of funnel, try 2*)
LoCurve = 8; (*Curvature of the lower part of funnel, try 8*)
BottomRadius = 0.2;  (*the radius (0-1.5) of the part of
 the funnel with no local minima to make a smooth native well*)
MiddleRadius = 1.5;  (*part of funnel where the number and size
 of minima changes from the Lo region to the Hi region*)
TopRadius = 0.5;
(*maximum radius where local minima occur.  To avoid
 having the rim of funnel cut through minima, make < 1.5*)
FunnelCutOut = 0.1 π;  (*angular size (in radians) of arc
 cut out of funnel to allow side view of interior*)
StartAngle = 0.25 π;  (*lowest angle where minima occur.  To avoid
  cutting through local wells with cut-out, make > FunnelCutOut*)
StopAngle = 1.75 π;
(*highest angle where minima occur.  To avoid cutting through local wells with cut-
 out, make < 2π- FunnelCutOut*)

(*Define appearance of local minima on Lo region of funnel*)
NhiMinima = 10;  (*number of local minima, try 100*)
HiMinimaScale = {1.6, 2.6};  (*range of minima well depths*)
HiMinimaRadialWidth = 0.08;  (*width of minima in radial direction*)
HiMinimaTangentWidth = 0.001;  (*width of minima pepindicular to radius*)

(*Define appearance of local minima on Hi region of funnel*)
NloMinima = 40;  (*number of local minima, try 20*)
LoMinimaScale = {0.2, 0.3};  (*range of minima well depths*)
LoMinimaRadialWidth = 0.01;  (*width of minima in radial direction*)
LoMinimaTangentWidth = 0.001;  (*width of minima pepindicular to radius*)

MinimaHi = 0;
MinimaLo = 0;

a = 65;  (* increasing top view area *)
b = 3.5;  (* increasing suddenness of chasm *)
c = 38;  (* increasing depth of the main funnel *)
d = 75;  (* increasing vertical position (height) of the entire funnel; '
 Zero Point' is 75 m below bedrock *)
        Punkt

Do[MinimaHi += Random[Real, HiMinimaScale]
iteriere            zufällig  reell
```

$$\left(e^{-\frac{\left(\frac{\theta-\text{RandomReal}[\{\text{StartAngle},\text{StopAngle}\}]}{2\pi}\right)^2}{\text{HiMinimaTangentWidth}}}\right)\left(e^{-\frac{(z-\text{RandomReal}[\{\text{MiddleRadius},\text{TopRadius}\}])^2}{\text{HiMinimaRadialWidth}}}\right), \{\text{NhiMinima}\}];$$

```
Do[MinimaLo += Random[Real, LoMinimaScale] (ⅇ^(-((θ-RandomReal[{StartAngle,StopAngle}])/(2 π))^2/LoMinimaTangentWidth))
iteriere          zufällig  reell

    (ⅇ^(-(z-RandomReal[{BottomRadius,MiddleRadius}])^2/LoMinimaRadialWidth)), {NloMinima}];
```

$$\texttt{Do}\left[\texttt{MinimaLo} \mathrel{+}= \texttt{Random}[\texttt{Real}, \texttt{LoMinimaScale}] \left(e^{-\frac{\left(\frac{\theta-\texttt{RandomReal}[\{\texttt{StartAngle},\texttt{StopAngle}\}]}{2\pi}\right)^2}{\texttt{LoMinimaTangentWidth}}}\right) \left(e^{-\frac{(z-\texttt{RandomReal}[\{\texttt{BottomRadius},\texttt{MiddleRadius}\}])^2}{\texttt{LoMinimaRadialWidth}}}\right), \{\texttt{NloMinima}\}\right];$$

```
(* Parameter *)
(* Density of granite: ro = 2620 kg/m³ | Heat capacity of granite: c =
  790 J/kgK | Enthalpie of fusion of granite: H = 420000 J/kg  *)
(* Thermal conductivity of granite: lambda = 2.8 Watt/MeterKelvin;
Source: https://de.wikipedia.org/wiki/Wärmeleitfähigkeit *)

(* Question 1: how many kg of granite at 20 °C can be
  heated up to 1220 °C  when very slowly absorbing 150 kt ? *)
(* Solution 1: Heat Q = 150 kt = 150 x 4.18 10^12 J | Delta T = 1200 °K *)
(* c = Q/(mass x Delta T)  | mass =
  Q/(c x Delta T) =  6.27 10^14 /(790 x 1200) = 6.6 x 10^8 kg *)

(* Question 2: this corresponds to which volume ? *)
(* Solution 2: ro = m/V  | V = 6.6 x 10^8 kg / (2620 kg/m³) = 252440 m³ *)

(* Question 3: this corresponds to which radius of a sphere ? *)
(* Solution 3: V =
 4/3 * π r^3  | radiusHeatZone2[t_]:=CubeRoot[(0.75*V)/π; = 252440 m³] = 39 m*)
                                      Kubikwurzel

(* Question 4: how big is the sphere when the granite is in addition liquefied ? *)
(* Solution 4: ___________Use Enthalpie of fusion of granite: H =
 420000 J/kg  _______________________*)



(* Energy-Input WTC2 (PreFizzle): 75 kt in 90 min (= 0.0138889 kt per second *)
          Eingabe
(* 0.0138889 kt per second = 5.8 10^10 Joule per second (58.000 MW) *)
(* 1 kt = 4.18 10^12 J *)

(* k1 = damping factor (loss by radiation); k1 ≤ 1) *)
(* k2 = damping factor (loss by fusion enthalpie); k2 ≤ 1)*)

(* Energy-Input WTC2 (Total): 150 kt in 90 min (= 0.02777 kt per second *)
          Eingabe      Gesamtsumme

k1 = 0.95; k2 = 0.95; Q = 11.6*^10; t = 0.5;

slowEnergyInput2[t_] := k1 * Q * t / 4.18*^12;

radiusHeatZone2Temp100[t_] := CubeRoot[(0.75 * k2 * k1 * Q * t) / (π * 790 * 100 * 2620)];
                              Kubikwurzel
radiusHeatZone2Temp1200[t_] := CubeRoot[(0.75 * k2 * k1 * Q * t) / (π * 790 * 1200 * 2620)];
                               Kubikwurzel
```

```
(* defines the graphical Slider *)
                                   Schieberegler
(* DynamicModule[{t=0.5},
   dynamisches Modul
 {Slider[Dynamic[t]],Dynamic[Round[slowEnergyInput2[5400*t]]{kt }] *)
  Schieb… dynamisch       dynamisch runde


Show[
zeige an
 Graphics3D[{Blue, river, LightBlue, chimney1, chimney7, Black, fixpoint, Black, Thick,
 3D-Graphik  blau        hellblau                          schwarz           schwarz dick
   Dashed, bathtub, Yellow, rockbed, LightPurple, Opacity[.5], WTC7base, WTC7roof,
   gestrichelt      gelb             helles Lila  Deckkraft
   WTC7south, WTC7north, WTC7east, WTC7west, Red, Z1, Z2, Z7, Yellow, F1, F7, Green,
                                             rot              gelb            grün
   Opacity[.1], soil, Gray, Antenna, WTC7roofBoxHigh, WTC7roofBoxLow, WTC1, Orange,
   Deckkraft          grau                                                 orange
   valley4OLD , Blue, Opacity[.25], cone1, cone7, LightCyan, Air1}, Axes → True],
                blau  Deckkraft                   helles Blaugrün   Achsen  wahr
 (* ContourPlot3D[{x^2+y^2== (z+75)^2 },{x,-200,200},{y,-200,200},
    3D-Konturgraphik
  {z,0,200},ContourStyle→Opacity[.1]], *)
            Konturenstil  Deckkraft

 ContourPlot3D[(x + 12.5)^2 / aE1^2 + (y - 137)^2 / bE1^2 + (z - 430)^2 / cE1^2 == 80,
 3D-Konturgraphik
  {x, -500, 500}, {y, -500, 500}, {z, -500, 500}, Mesh → 5, PlotPoints → 100],
                                                  Gitternetz Anzahl der Punkte in der Graphik
 ContourPlot3D[{(x + 12.5)^2 / aJ1^2 + (y - 137)^2 / bJ1^2 == (480 - z)},
 3D-Konturgraphik
  {x, -300, 300}, {y, -300, 300}, {z, 75, 500}, Mesh → 3, PlotPoints → 100],
                                                Gitternetz Anzahl der Punkte in der Graphik

 (* Glacier Valley; PotHole 1; OLD=Ellipsoid[{50,-80,75},{12.5,12.5,20}] *)
                                   Ellipsoid
 Plot3D[valley1, {x, 37.5, 62.5}, {y, -92.5, -67.5}, Mesh → 3, PlotPoints → 100],
 stelle Funktion graphisch in 3D dar                 Gitternetz Anzahl der Punkte in der Graphik

 (* Glacier Valley; PotHole 2; OLD = Sphere[{92.5,-62.5,75},20] *)
                                     Kugel
 Plot3D[valley2, {x, 72.5, 112.5}, {y, -82.5, -42.5}, Mesh → 3, PlotPoints → 100],
 stelle Funktion graphisch in 3D dar                  Gitternetz Anzahl der Punkte in der Graphik
 (* Glacier Valley; PotHole 3; OLD = Sphere[{132.5,-80,75},20] *)
                                     Kugel
 Plot3D[valley3, {x, 112.5, 152.5}, {y, -100, -60}, Mesh → 3, PlotPoints → 100],
 stelle Funktion graphisch in 3D dar                Gitternetz Anzahl der Punkte in der Graphik

 (* Glacier Valley; PotHole 4;
 OLD = GeometricTransformation[Ellipsoid[{115,-20,75},{40,20,10}],
       geometrische Transformation Ellipsoid
   RotationTransform[-25 Degree,{0,0,1},{115,-20,75}]] *)
   Rotationstransformation  Grad
```

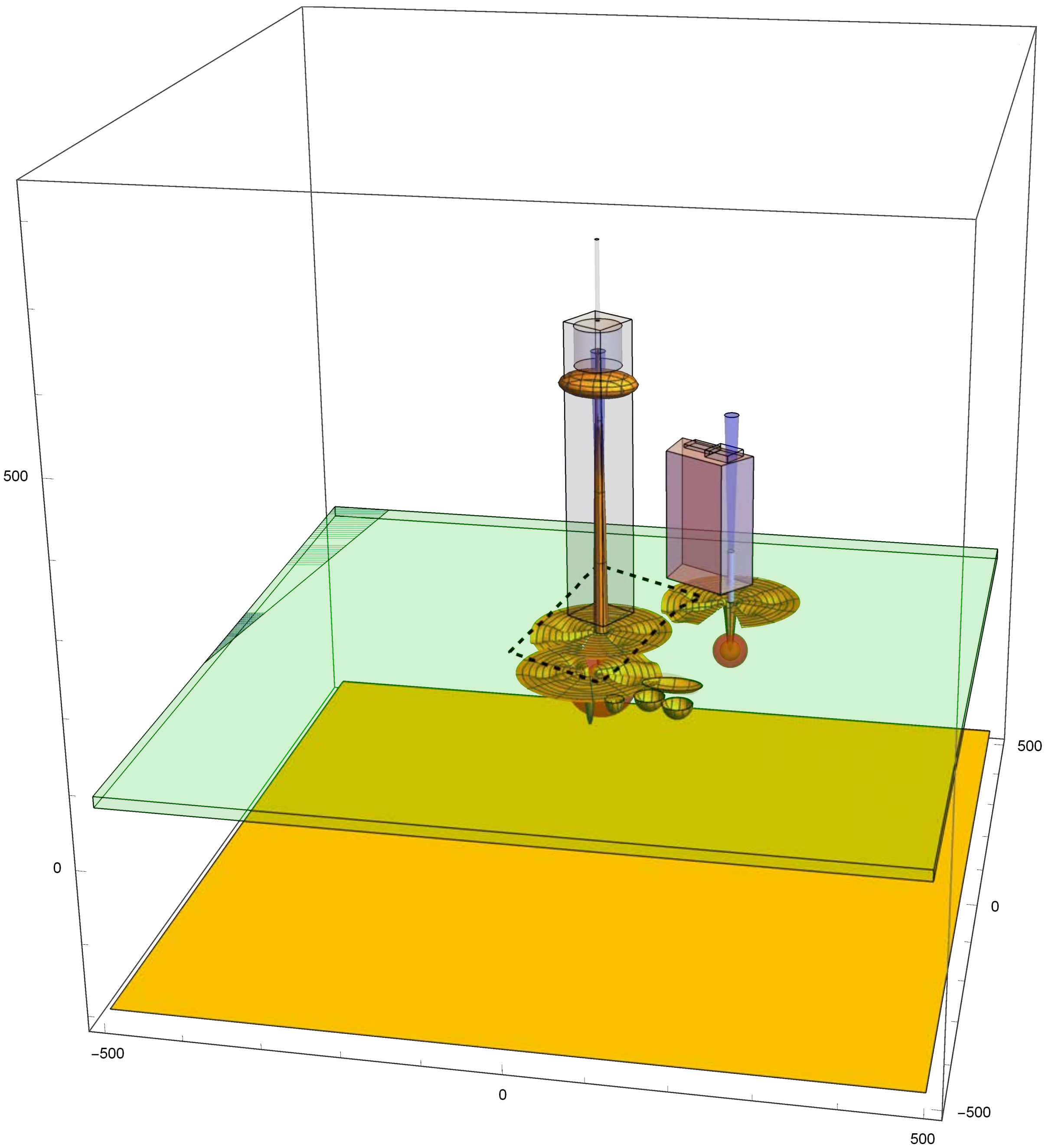
```
[Rotationstransformation      [Grad
Plot3D[valley4, {x, 75, 155}, {y, 0, -40}, Mesh → 3, PlotPoints → 100],
[stelle Funktion graphisch in 3D dar                 [Gitternetz  [Anzahl der Punkte in der Graphik


FunnelWTC1 = ParametricPlot3D[{-12.5 - a * z * Sin[θ], 137 - a * z * Cos[θ], (d +
                [parametrische 3D-Darstellung        [Sinus                [Kosinus
     -1.5 * c * e^(-HiCurve (b * z)^3) - .5 * c * e^(-LoCurve (b * z)^2) - 4 * MinimaHi - 4 * MinimaLo)},
  {θ, 0, 2 π}, {z, 0, 1.5}, PlotPoints → 120, Mesh → Nmesh,
                            [Anzahl der Punkte in ⋯ [Gitternetz
  ImageSize → Large, ViewPoint → {0, 3, 1.5}, Axes → True, Boxed → True,
  [Bildgröße   [groß   [Ansichtspunkt              [Achsen [wahr [einger⋯ [wahr
  PlotRange → All, PlotLabel → "Click and drag to change viewpoint"],
  [Koordinatenb⋯ [alle  [Beschriftung der Graphik


FunnelWTC2 = ParametricPlot3D[{a * z * Sin[θ], a * z * Cos[θ], (d +
                [parametrische 3D-Darstellung  [Sinus       [Kosinus
     -1.5 * c * e^(-HiCurve (b * z)^3) - .5 * c * e^(-LoCurve (b * z)^2) - 4 * MinimaHi - 4 * MinimaLo)},
  {θ, FunnelCutOut, 2 π - FunnelCutOut}, {z, 0, 1.5}, PlotPoints → 120, Mesh → Nmesh,
                                                   [Anzahl der Punkte in ⋯ [Gitternetz
  ImageSize → Large, ViewPoint → {0, 3, 1.5}, Axes → True, Boxed → True,
  [Bildgröße   [groß   [Ansichtspunkt              [Achsen [wahr [einger⋯ [wahr
  PlotRange → All, PlotLabel → "Click and drag to change viewpoint"],
  [Koordinatenb⋯ [alle  [Beschriftung der Graphik


FunnelWTC7 = ParametricPlot3D[{145 - a * z * Sin[θ], 272.5 - a * z * Cos[θ], (d +
                [parametrische 3D-Darstellung     [Sinus                 [Kosinus
     -1.5 * c * e^(-HiCurve (b * z)^3) - .5 * c * e^(-LoCurve (b * z)^2) - 4 * MinimaHi - 4 * MinimaLo)},
  {θ, FunnelCutOut, 2 π - FunnelCutOut}, {z, 0, 1.5}, PlotPoints → 120, Mesh → Nmesh,
                                                   [Anzahl der Punkte in ⋯ [Gitternetz
  ImageSize → Large, ViewPoint → {0, 3, 1.5}, Axes → True, Boxed → True,
  [Bildgröße   [groß   [Ansichtspunkt              [Achsen [wahr [einger⋯ [wahr
  PlotRange → All, PlotLabel → "Click and drag to change viewpoint"]
  [Koordinatenb⋯ [alle  [Beschriftung der Graphik




] (* end Show *)
          [zeige an
(*      }]  end DynamicModule *)
              [dynamisches Modul
```

Resulting radius of a hot sphere at 150 kt input yield:

red = yield [kt]
orange = size of sphere with 1200 °C
blue = size of sphere with 100 °C

```
(* defines the graphical Slider *)
                          Schieberegler
DynamicModule[{t = 0.5},
dynamisches Modul
 {Slider[Dynamic[t]], Dynamic[Round[slowEnergyInput2[5400 * t]] {kt}]}]
  Schieb··· dynamisch    dynamisch runde

(* curves only *)
Plot[{slowEnergyInput2[t], radiusHeatZone2Temp1200[t], radiusHeatZone2Temp100[t]},
stelle Funktion graphisch dar
 {t, 0, 5400}, PlotStyle → {Red, Orange, Blue}]
               Darstellungsstil rot  orange  blau
```

For the interactive 3D model [Wolfram Mathematica] see:
http://www.911history.de/aaannxyz_ch08_en.html



Resulting radius of a hot sphere at 150 kt input yield:

Out[689]=

red = yield [kt]
orange = size of sphere with 1200 °C
blue = size of sphere with 100 °C

Out[690]= {slider, {71 kt}}

Out[691]=