

---

## Koda, ki izračuna točke odboja, dokler svetloba ne uide

```
ClearAll[ogledaloVIP]
ogledaloVIP[koef_, d_, tocka1_, vektor1_] :=
Module[{tocka = tocka1, vektor = vektor1, odboji = 0, tocke = {}, a, b, c, t1,
  t2, gradient, proj}, While[(Norm[tocka1 - tocka] ≥ 0.101) || (odboji == 0),
  a = Sum[koef[[i]] * vektor[[i]]^2, {i, Length[vektor]}];
  b = Sum[koef[[i]] * 2 * vektor[[i]] * tocka[[i]], {i, Length[vektor]}];
  c = Sum[koef[[i]] * tocka[[i]]^2, {i, Length[tocka]}] - d;
  {t1, t2} = (-b + {-1, 1} * (b^2 - 4 * a * c)^0.5) / (2 * a);

  If[Abs[t1] < 0.0001, tocka = tocka + t2 * vektor, tocka = tocka + t1 * vektor];

  gradient = 2 * koef * tocka;
  proj = (Dot[vektor, gradient] / Dot[gradient, gradient]) * gradient;
  vektor = 2 * proj - vektor;
  AppendTo[tocke, Round[tocka, 0.001]];
  odboji += 1;
  odboji < 100];
Print["Število odbojev: ", odboji];

tocke]
```

## Koda, ki grafično prikaže prejšnje

In[508]:=

```
(*naslednja parametra tvorita: koef[[1]]*x^2+koef[[2]]* y^2+koef[[3]]*z^2 == d*)
koef = {1, 2, 3};
d = 25;

contourData = ContourPlot3D[koef[[1]] * x^2 + koef[[2]] * y^2 + koef[[3]] * z^2 == 25,
  {x, -5, 5}, {y, -5, 5}, {z, -5, 5}, MeshFunctions -> {#3 &},
  MeshStyle -> {Opacity[0.01], Directive[Opacity[0.5]]}, PlotPoints -> 50,
  ContourStyle -> Directive[Opacity[0.1]], PlotTheme -> "Classic"];
povrsinaKonture = First[contourData /. {Surface -> g_ -> g}];

(*Funkcija za risanje odbojev.*)
risanjeSTockami[n_, dodatnaTocka_, smerniVektor_] :=
Module[{crte, tockeZaUporabo, dodatnaCrta, tocke},
  tocke = ogledaloVIP[koef, d, dodatnaTocka, smerniVektor];
  tockeZaUporabo = Take[tocke, n];
  crte = Table[Line[{tockeZaUporabo[[i]], tockeZaUporabo[[i + 1]]}],
    {i, 1, Length[tockeZaUporabo] - 1}];
  dodatnaCrta = Line[{dodatnaTocka, tockeZaUporabo[[1]]}];
  tockeInCrtePlot = Graphics3D[{{Red, Sphere[#, 0.1] & /@ tockeZaUporabo},
    {Blue, crte}, {Green, dodatnaCrta}, {Black, Sphere[dodatnaTocka, 0.1]}}];
  Show[Graphics3D[{povrsinaKonture, Opacity[0.3]}], tockeInCrtePlot];

(*Ustvari manipulacijo za prilagajanje števila točk, in smernega vektorja*)
vrh = Sqrt[d / koef[[3]]];
Manipulate[risanjeSTockami[n, {0, 0, vrh}, {a, b, c}],
  {n, 1, Min[60, Length[tocke]], 1, Appearance -> "Labeled"}, {a, -2, 2},
  {b, -2, 2}, {c, -2, 2}]
```

Out[514]=

