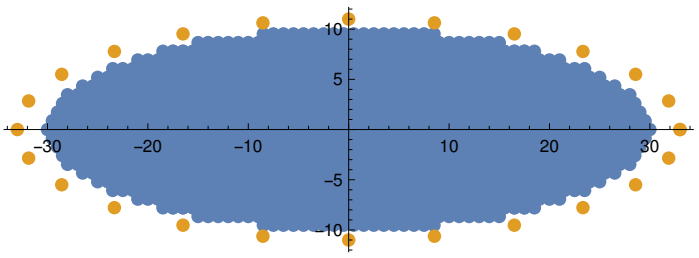
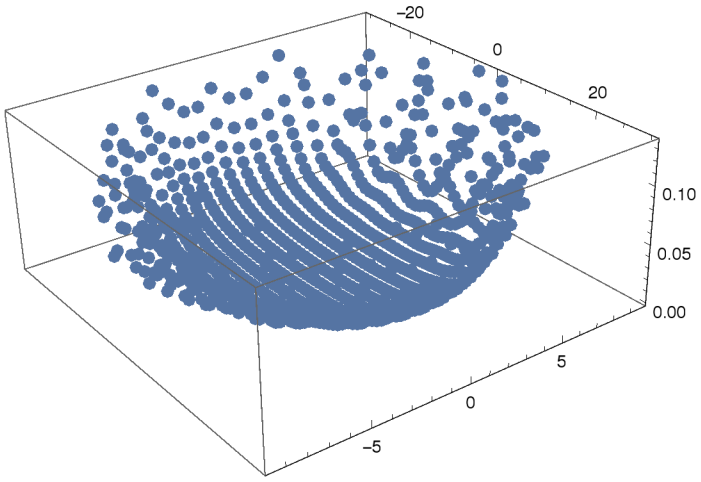
```
hex[m_Integer] := hex[m] =
    N@{{1 / n, 0}, {-1 / n, 0}, {-1 / (2 n), Sqrt[3] / (2 n)}, {1 / (2 n), -Sqrt[3] / (2 n)},
       {1 / (2 n), Sqrt[3] / (2 n)}, {-1 / (2 n), -Sqrt[3] / (2 n)}, {0., 0.}} /. n :> 2^m;
gen[z_List, m_Integer: 8] := gen[z, m] = Table[hex[m][[j]] + #, {j, 7}] & /@ z // Union //
     Flatten[#, 1] &;
cgen = Compile[{z, m}, gen[z, m]];
prehexgen[z_List, g_Integer: 1, m_Integer: 8] :=
  hexgen[z, g, m] = Nest[Union@cgen[#, m] &, z, g];
hexgen[z_List, g_Integer: 1, m_Integer: 8, k_Integer: 1] := hexgen[z, g, m, k] =
    (Level[(Total[prehexgen[z, g, m] /. {x1_, x2_} :> pt[N[{x1, x2}]]] /.
          {n___ pt[x_] → pt[x]}), 1]) /. pt[x_] → k x;

hexgen[{0, 0}, 4, 2, 10];

q2[x_List, y_List] :=
  q2[x, y] = (Total@Table[Times[N[1 / 2], (x[[j]] - y[[j]])^2], {j, 1, 2}]);
INT[n_Integer] := INT[n] = N@Range[0, 1, N[1 / (8 n)]];
X[1] := X[1] = (hexgen[{{0, 0}}, 32, 3, 8]) //.
    {{x1_Real, x2_Real} :> Nothing /; (x1 / (3 × 10))^2 + (x2 / 10)^2 > 1};
(*Rejection method for generating uniform sample of the ellipsoid source.*)
Y[1] := Y[1] = (INT[3]) /. {x_Real :> {(1.1) 3 × 10 Cos[2 Pi x], (1.1) 10 Sin[2 Pi x]}};
(*parameterization of the boundary of the ellipsoid*)
Print@Length[Y[1]];
Print@ListPlot[{X[1], Y[1]}, AspectRatio → Automatic,
   PlotStyle -> PointSize[Large], PlotRange → All];
```

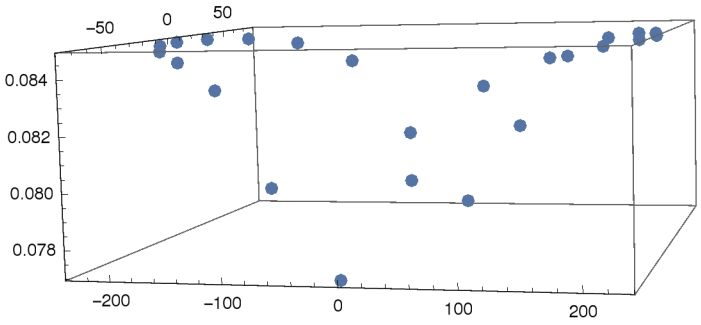
```
25
```



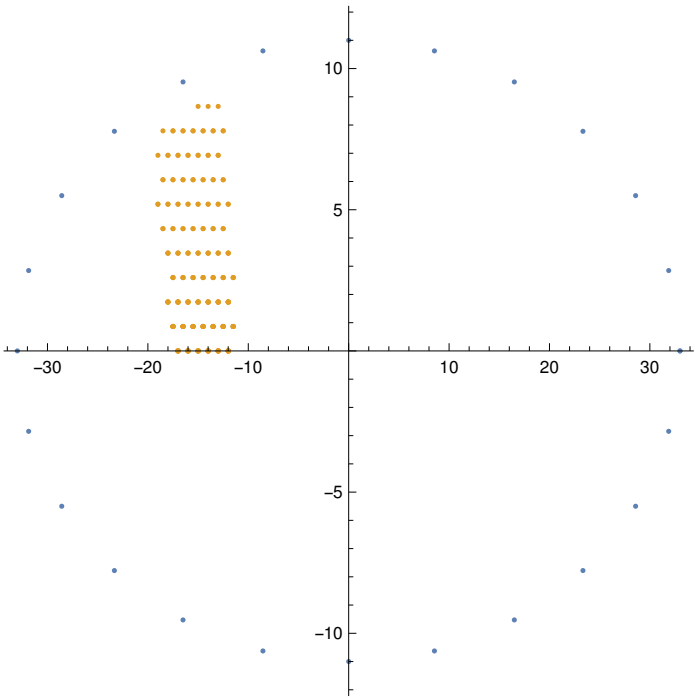
```
TC[{x1_Real, x2_Real}] := TC[{x1, x2}] = Times[Length[Y[1]]^(0),
    Total[Y[1] /. {{y1_Real, y2_Real} :> q2[{x1, x2}, {y1, y2}]^(-3 / 2)}]];
TC1[{x1_Real, x2_Real}] := Times[Length[Y[1]]^(0),
   Total[Y[1] /. {{y1_Real, y2_Real} :> q2[{x1, x2}, {y1, y2}]^(-1)}]];
(*TC is the total potential, depending on source {x1,x2}
 and summed over target Y[1]*)
(*Print@ListPointPlot3D[X[1]/.{{x1_Real, x2_Real}:>{x1,x2,TC[{x1,x2}]}},
   PlotStyle->PointSize[Large]];*)
B[{x1_Real, x2_Real}, {y1_Real, y2_Real}] :=
  B[{x1, x2}, {y1, y2}] =  1 / 2 q2[{x1, x2}, {y1, y2}]^(-3 / 2);
(*Discount at target point {y1,y2} *)
R[{x1_Real, x2_Real}, {y1_Real, y2_Real}] :=
  R[{x1, x2}, {y1, y2}] = TC[{x1, x2}] - B[{x1, x2}, {y1, y2}];
(*R=TC-B is repulsion cost: total minus discount*)

(*Print@ListPointPlot3D[X[1]/.{{x1_Real, x2_Real}:>{x1,x2,R[{x1,x2},{1.,0.}]}},
   AspectRatio→1,PlotStyle->PointSize[Large], PlotRange→Full];*)

r[{y1_Real, y2_Real}] :=
  r[{y1, y2}] = {x1_Real, x2_Real} :> t[{y1, y2}] v_{x1,x2}[R[{x1, x2}, {y1, y2}]];
(*psi[1,{y1,y2}]:=*)
p[1, {y1_Real, y2_Real}] := p[1, {y1, y2}] = Total@(X[1] /. r[{y1, y2}]);

p[2, {y1_Real, y2_Real}] :=
  p[2, {y1, y2}] = (p[1, {y1, y2}] //. {n1___ t[y_] v_{x1_Real,x2_Real}[a_Real] +
        n2___ t[y_] v_{x3_Real,x4_Real}[b_Real] :> t[y] v_{x1,x2}[a] /; a < b,
      t[y_] v_x_[ComplexInfinity] → 0, t[y_] v_x_[Indeterminate] → 0});

p[2, Y[1][[3]]];

ListPointPlot3D[X[1] /. {x1_Real, x2_Real} → {x1, x2, TC[{x1, x2}]},
 PlotStyle -> PointSize[Large], AspectRatio → Automatic, PlotRange → Automatic]
(*X[1]/.{r[Y[1][[1]]]};
X[1]*)
```



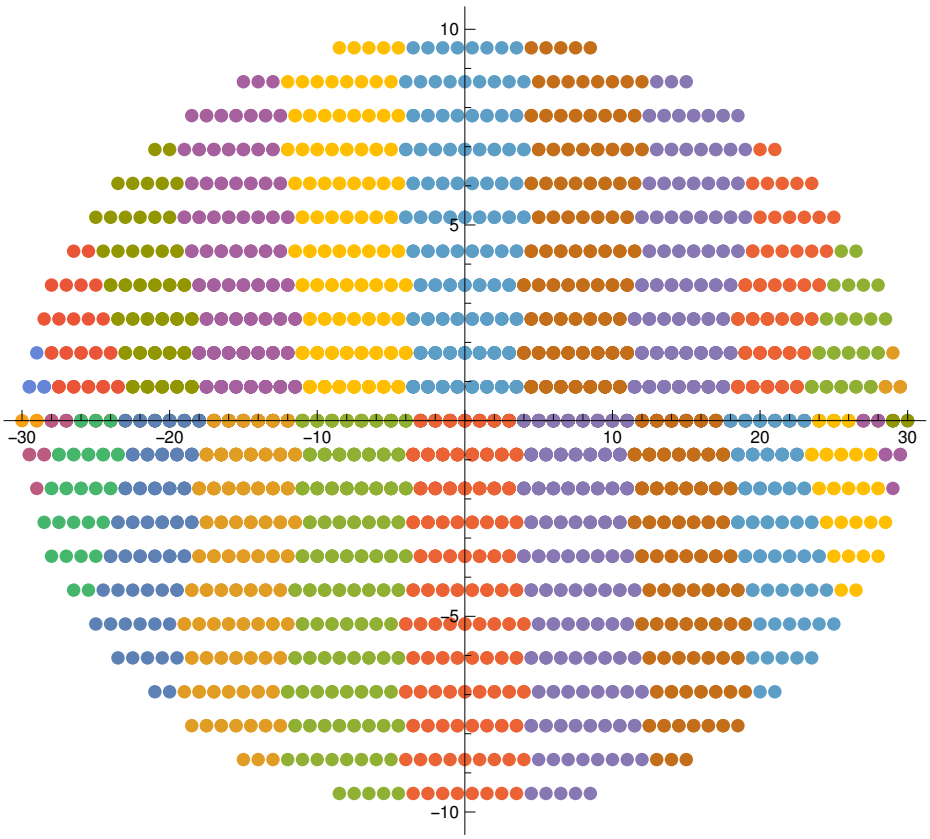

```
p[3, {y1_Real, y2_Real}] :=
  p[3, {y1, y2}] = (p[2, {y1, y2}] /. {t[y_List] v_x_List[a_Real] :> a});

psi[{y1_Real, y2_Real}] :=
  psi[{y1, y2}] = (p[2, {y1, y2}] /. t[y_List] v_x_List [a_Real] :> {y1, y2, a});

psival[{y1_Real, y2_Real}] :=
  psival[{y1, y2}] = (p[2, {y1, y2}] /. t[y_List] v_x_List [a_Real] :> a);

YY := (Y[1] /. {y1_Real, y2_Real} :> psi[{y1, y2}]);

YY;

ListPointPlot3D[YY, PlotStyle -> PointSize[Large] ]
```



```
(*rule[{x1_Real, x2_Real}]:=rule[{x1,x2}]=
   {y1_Real, y2_Real}:>s[{x1,x2}]v_{y1,y2}[R[{x1,x2}, {y1,y2}]-psival[{y1,y2}]];*)
rule[{x1_Real, x2_Real}] := rule[{x1, x2}] =
   {y1_Real, y2_Real} :> s[{x1, x2}] v_{y1,y2}[R[{x1, x2}, {y1, y2}]];
(*Evaluation rule. Does evaluation time increase/decrease when we replace →
 with :> ?? *)
Y[2, {x1_Real, x2_Real}] := Y[2, {x1, x2}] = Total@(Y[1] /. rule[{x1, x2}]);
(*replace every target variable with evaluation at {x1,x2}*)
Y[3, {x1_Real, x2_Real}] :=
  Y[3, {x1, x2}] = (Y[2, {x1, x2}] //. {n1___ s[x_] v_{y1_Real,y2_Real}[a_Real] +
        n2___ s[x_] v_{y3_Real,y4_Real}[b_Real] :> s[x] v_{y1,y2}[a] /; a < b}) ;
(* pairwise comparison over target Y to find the argmin y with
 respect to source point x*)
Y[3, X[1][[1]]];

X[2] := X[2] = X[1] /. {x1_Real, x2_Real} :> Y[3, {x1, x2}];
X[3] := X[3] = (Total@X[2] //. {n___ s[x_] v_y_[a_Real] :> s[x] v[y]});
sub[{y1_Real, y2_Real}] :=
  sub[{y1, y2}] = (Level[Coefficient[X[3], v[{y1, y2}]], 1]) /. {s[x_] :> x};

ListPlot[{Y[1], sub[Y[1][[9]]]}, AspectRatio → 1, PlotRange → Full]

(*sub[Y[1][[1]]]*)
```



```
cells := (Y[1] /. {y1_Real, y2_Real} :> sub[{y1, y2}] /; Length[sub[{y1, y2}]] > 0);
ListPlot[cells, AspectRatio → Full, PlotStyle -> PointSize[Large]]
```




```
Y[1][[1]]
```

$\{4.2423 \times 10^{-16}, -1.\}$

```
Plot[RandomReal[{0, 1}, 61]]
```

Plot::argr : Plot called with 1 argument; 2 arguments are expected. ≫

```
Plot[RandomReal[{0, 1}, 61]]
```

```
Length[Y[1]]
```

9

```
INT[4]
```

```
{0., 0.03125, 0.0625, 0.09375, 0.125, 0.15625, 0.1875, 0.21875,
 0.25, 0.28125, 0.3125, 0.34375, 0.375, 0.40625, 0.4375, 0.46875,
 0.5, 0.53125, 0.5625, 0.59375, 0.625, 0.65625, 0.6875, 0.71875,
 0.75, 0.78125, 0.8125, 0.84375, 0.875, 0.90625, 0.9375, 0.96875, 1.}
```

```
Plot[{(1 - x^2)^(-1), (1 - x)^(-2) + (-1 - x)^(-2) }, {x, -3, 3}]
```