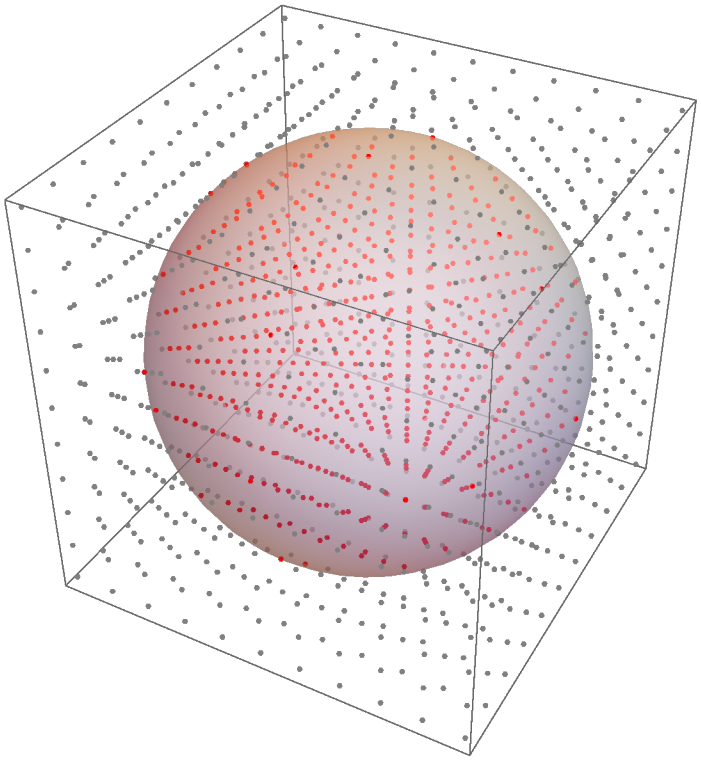
In[*]:=
```
(* Finds the lattice points in an escribed hypercube *)
escribedhypercubelatticepoints[c_, r_] :=
  Tuples[Range[Ceiling[# - r], Floor[# + r]] & /@ c];

(* Solves the 1D case of finding lattice points in a range *)
shrink[pair_] := If[pair[[1]] > pair[[2]],
   {pair[[2]] // Ceiling, pair[[1]] // Floor},
   {pair[[1]] // Ceiling, pair[[2]] // Floor}
  ];

(* Solves the quadratic equation when adding a new dimension to a hypersphere,
giving the range of values the new dimension has for a particular point *)
solve[c_, r_, knowns_] :=
  First[c] + # * Sqrt[r^2 - Total[(knowns - Rest[c])^2]] & /@ {-1, 1};

(* Add all the points over the range for the new dimension
 at a lower dimension point, none if the range is empty *)
joinwithlower[range_, lowerdimpoint_] :=
  If[Length[range] == 0, {}, Prepend[lowerdimpoint, #] & /@ range];

(* Module for listing hypersphere lattice points *)
hyperspherelatticepoints[c_, r_] := Module[{lowerdimpoints, ranges},
   (* Base case *)
   If[Length[c] == 1,
    Return[Tuples[Range @@ shrink[{Last[c] - r, Last[c] + r}], 1]]
   ];
   (* Recursive work to find values for subsequent dimensions *)
   lowerdimpoints = hyperspherelatticepoints[Rest[c], r];
   (* Solve for the current dimension *)
   ranges = Range @@ shrink[solve[c, r, #]] & /@ lowerdimpoints;

   (* Merge the points into a single list *)
   Catenate[
    MapIndexed[joinwithlower[#1, lowerdimpoints[[#2 // First]]] &, ranges, 1]]
  ];

(* Module for counting hypersphere lattice points *)
hyperspherelatticepointscount[c_, r_] := Module[{lowerdimpoints, counts},
   (* Base case *)
   If[Length[c] == 1,
    Return[First@Differences[shrink[{Last[c] - r, Last[c] + r}]]]
   ];
   (* Can't avoid scaling memory usage with hyper surface area,
```

```
    need to call enumeration module *)
    lowerdimpoints = hyperspherelatticepoints[Rest[c], r];
    (* Sum and output the counts *)
    counts = First@Differences[shrink[solve[c, r, #]]] & /@ lowerdimpoints;
    Total[counts]
   ];

(* Input *)
circle = {{0, 0, 0}, 5};
(* Whether to graph the solution in 2D or 3D *)
display = True;
Print["Number of points: ", hyperspherelatticepointscount @@ circle];
solution = hyperspherelatticepoints @@ circle;
If[Length[First@circle] == 2 && display, Print@Graphics[
     {Red, Disk @@ circle, Black, Point /@ escribedhypercubelatticepoints @@ circle,
      White, Large // PointSize, Point /@ solution}]];
If[Length[First@circle] == 3 && display, Print@
    Graphics3D[{Opacity@.5, Sphere @@ circle, Opacity@1, Gray,
      Point /@ escribedhypercubelatticepoints @@ circle, Red, Point /@ solution}]];
(* Some checks for verifying the solution, can be removed for real use *)
Print["Are all points in the hypersphere? ",
  AllTrue[solution, Norm[# - circle[[1]]] ≤ circle[[2]] &]];
Print["Are they unique? ", DuplicateFreeQ[solution]];
Print["Are they integers? ", AllTrue[solution, IntegerPart[#] == # &]];
Print["Is the solution the same as solving naively with the escribed hypercube? ",
  Sort@solution == Sort@Select[escribedhypercubelatticepoints @@ circle,
     Norm[# - circle[[1]]] ≤ circle[[2]] &]];
```

```
Number of points: 434
```



Are all points in the hypersphere? True

Are they unique? True

Are they integers? True

Is the solution the same as solving naively with the escribed hypercube? True