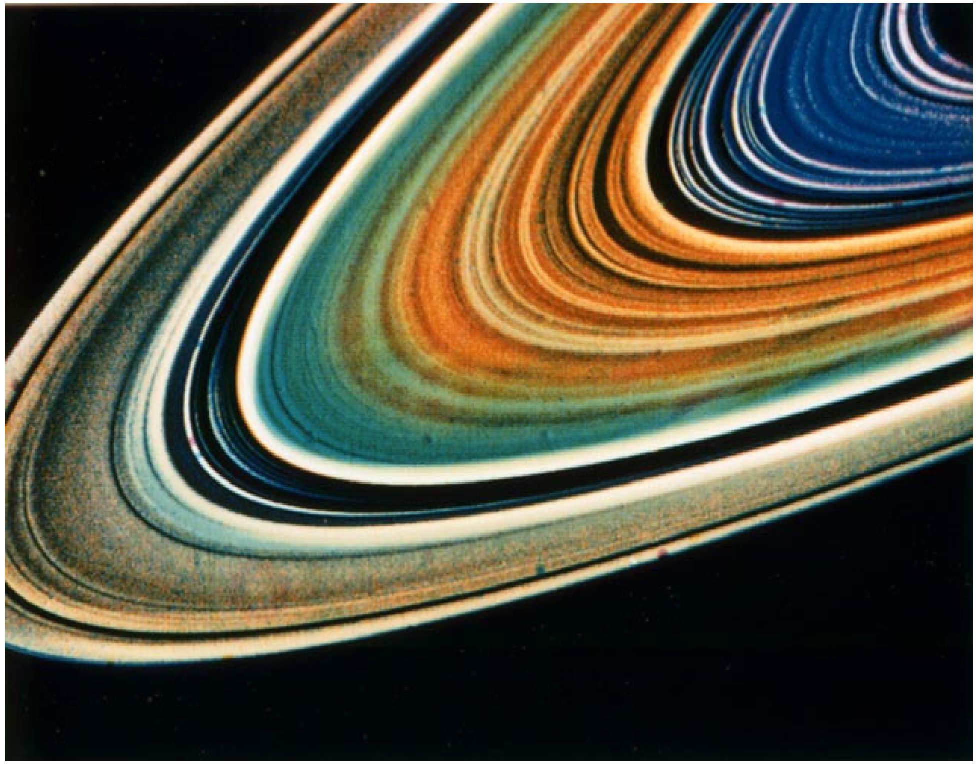
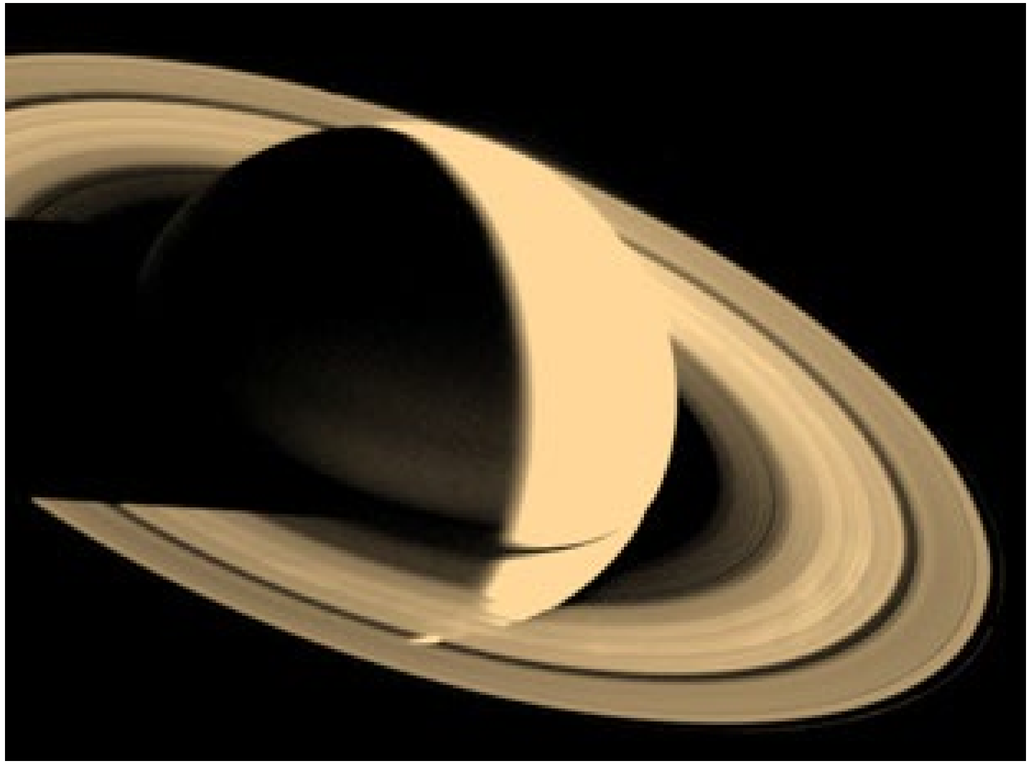
Mathematical Oranges and Deep Space Communication

Marcy Robertson

University of Melbourne

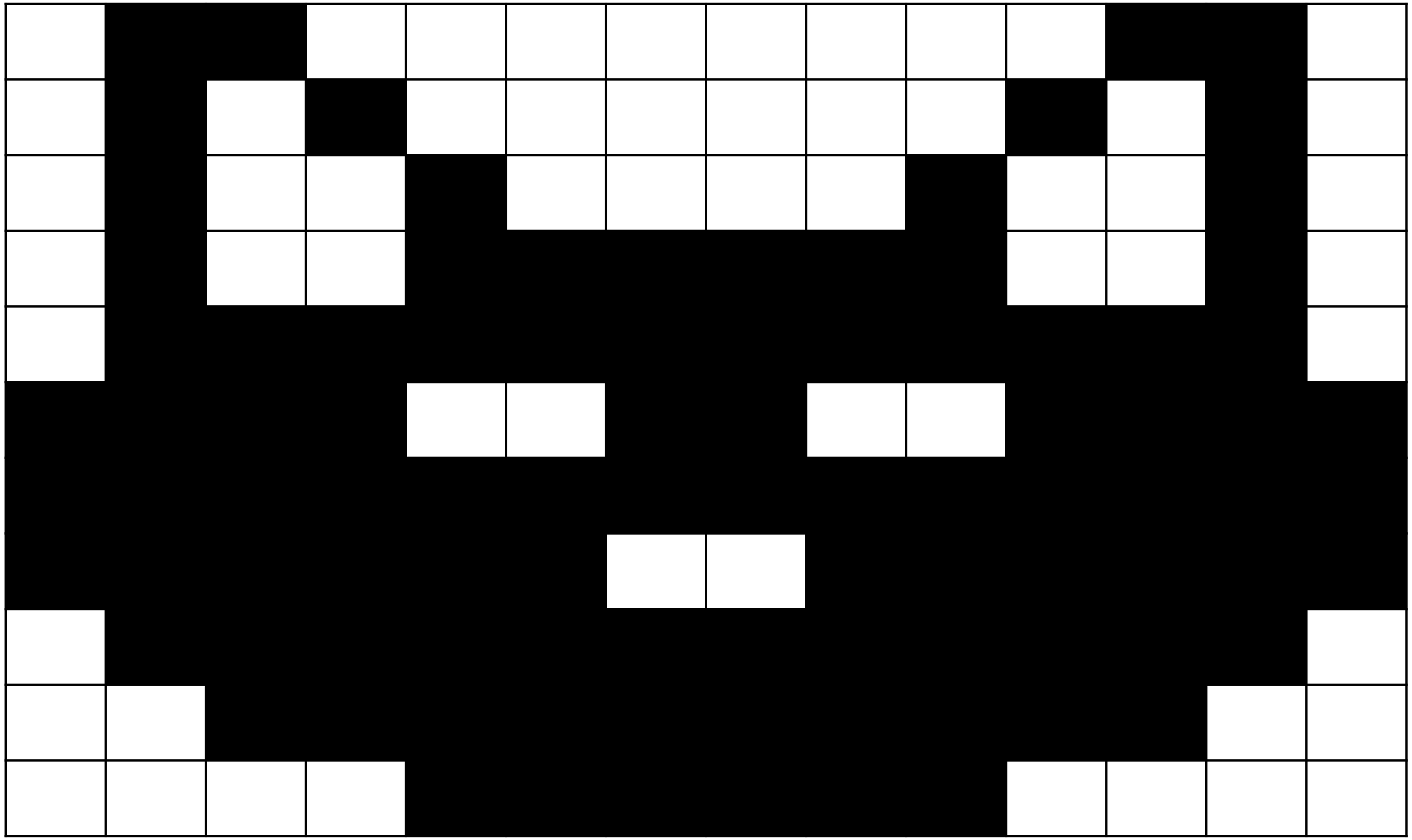
How do we get pictures from space?





How do I encode a digital picture?

- We can represent an image in binary – as strings of 1’s and 0’s
- The code word for “white” is 0.
- The code word for “black” is 1.



How do I encode a digital picture?

- We can represent an image in binary – as strings of 1’s and 0’s
- 0 is the code word for “white”
- 1 is the code word for “black”

0	1	1	0	0	0	0	0	0	0	0	1	1	0
0	1	0	1	0	0	0	0	0	0	1	0	1	0
0	1	0	0	1	0	0	0	0	1	0	0	1	0
0	1	0	0	1	1	1	1	1	1	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	0	0	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	0	0	0	0

How do we account for errors?

0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	0
0	1	0	0	1	0	0	0	0	1	0	0	1	0
0	1	0	0	1	1	1	1	1	1	0	0	1	0
0	1	1	0	1	1	1	1	1	1	1	1	1	0
1	1	1	1	0	0	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	0	0	0	0

Source → Transmit → Noise → Receiver → Received Signal

In between when our signal is transmitted and received it can encounter "noise" --meaning some 0's might get flipped to 1's and some 1's might get flipped to 0's.

Error Detecting Code

We can add “bits” to our code words: we make the code words more complicated so that errors can more easily be detected

Code Word for ”White" = 00

Code Word for ”Black" = 11

00

01

10

11

Error Correcting Code:

We can add “bits” to our code words: If the code words are more complicated, we cannot only **detect errors** but **correct those errors** which are “far enough away” from other code words.

"White" = 000
"Black" = 111

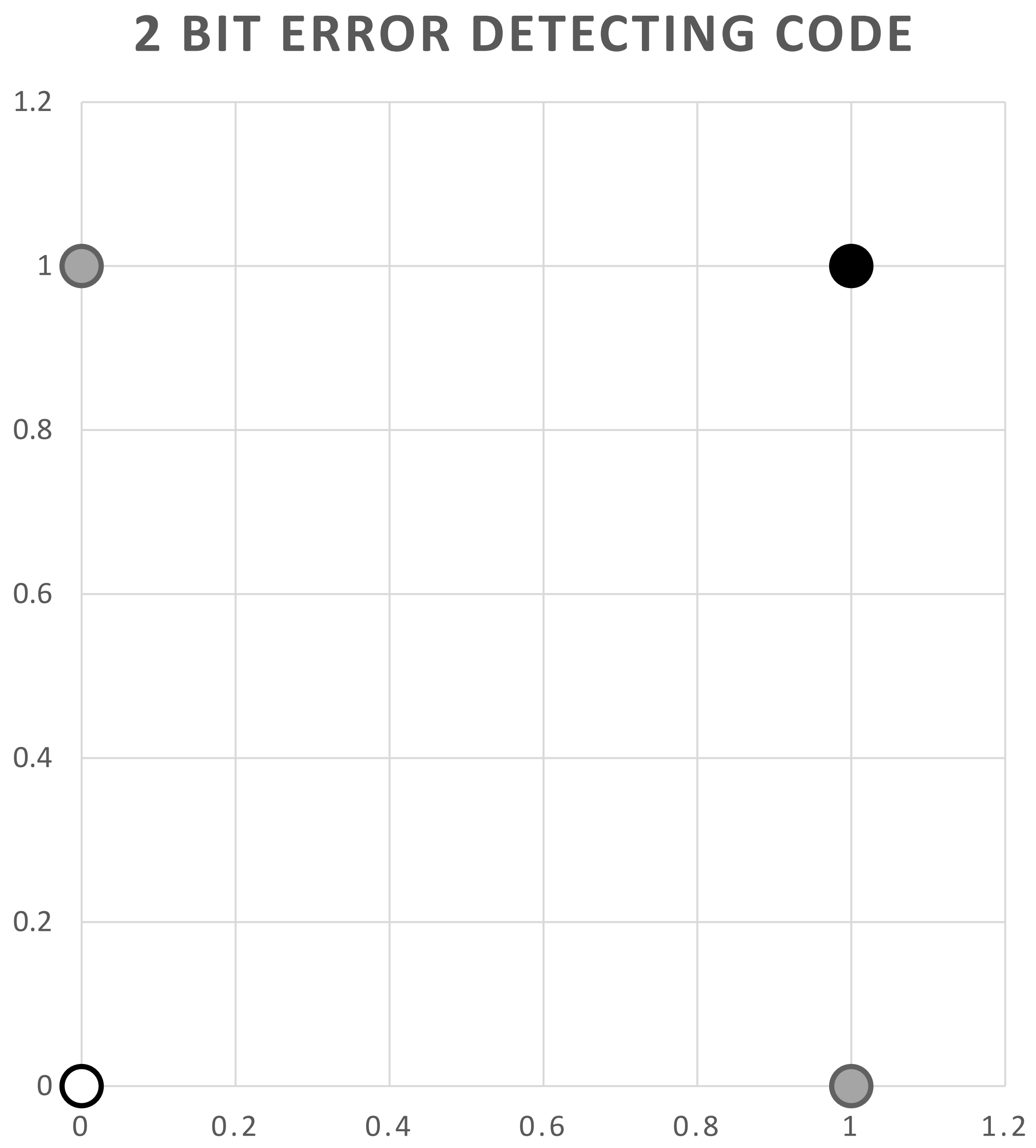
000	010	100	001	011	101	110	111

Distance between good and bad code

- Distance between two points (a,b) and (c,d) in the plane is :

$$\mathrm{d} = \sqrt{(c-a)^2 + (d-b)^2}$$

- Distance between code words is the # of bits which flipped.

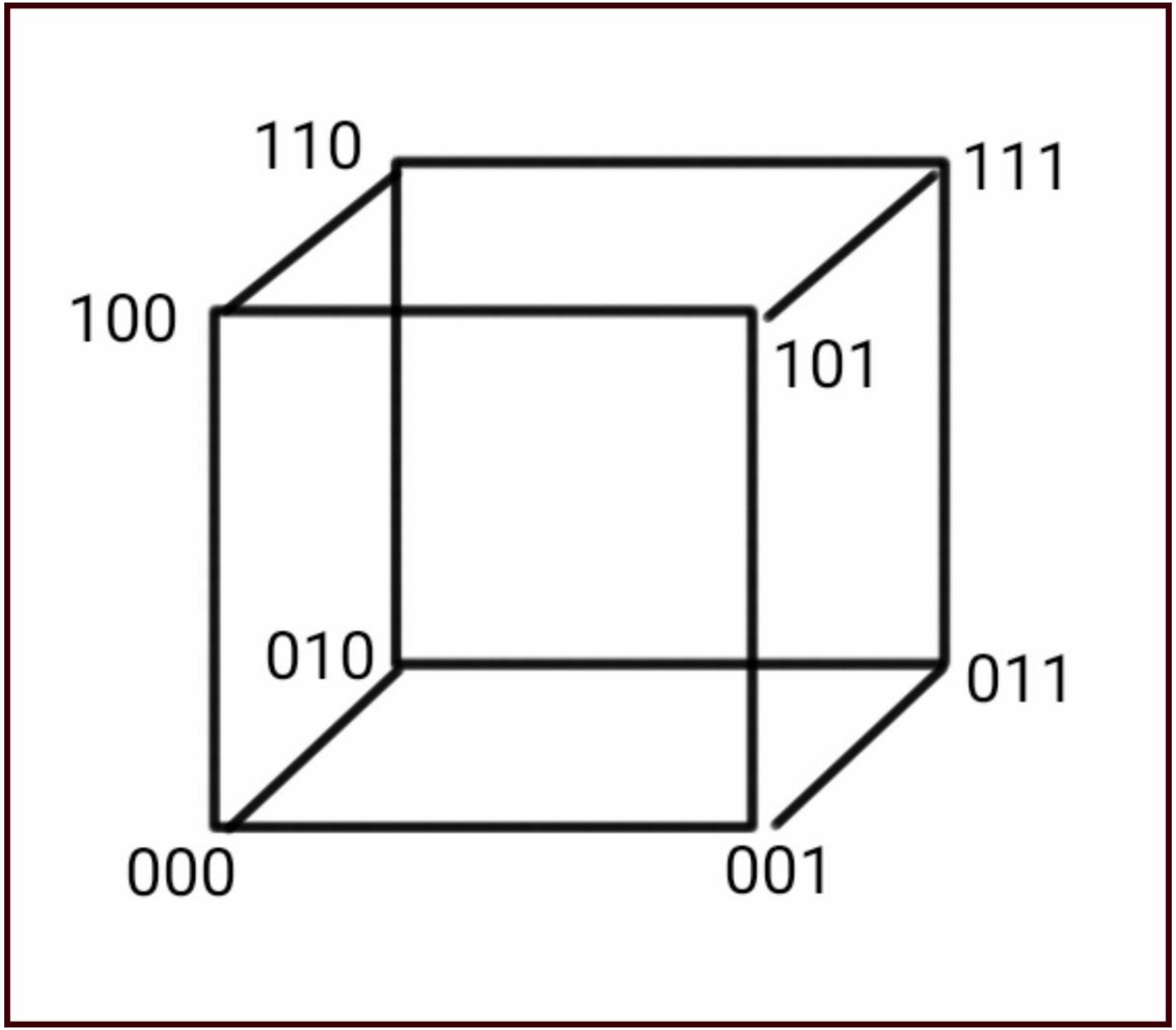


Hamming Distance:= How many bits were flipped?

The Hamming distance between the code words 000 = Black and 111= White is 3.

A code is **k-error detecting** if the minimum distance between two code words is at least k+1

A code is **k-error correcting** if the minimum distance between two code words is at least 2k+1



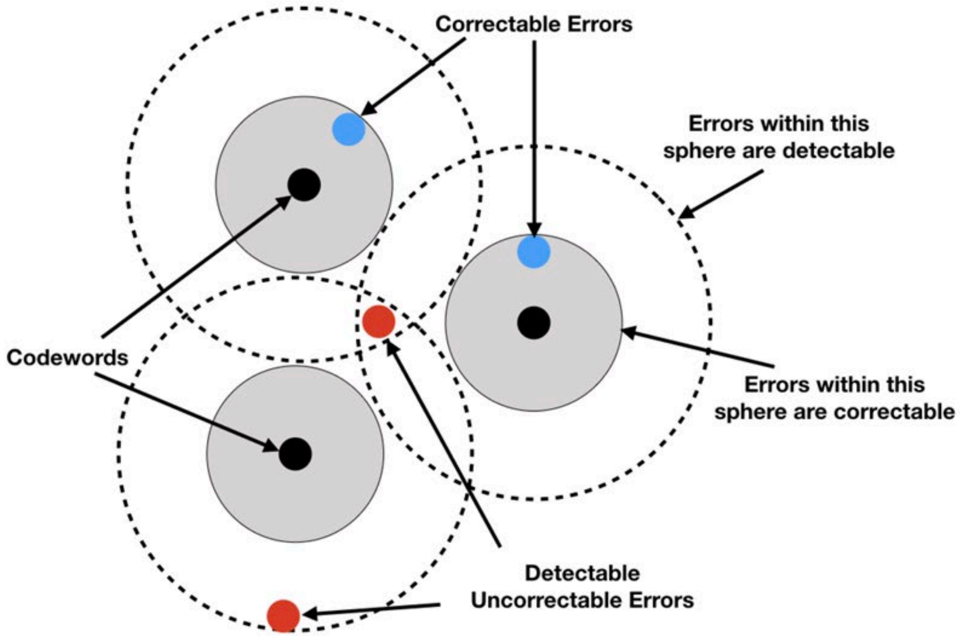
Hamming Spheres

A **sphere** is a collection of points all equally distant from a chosen center.

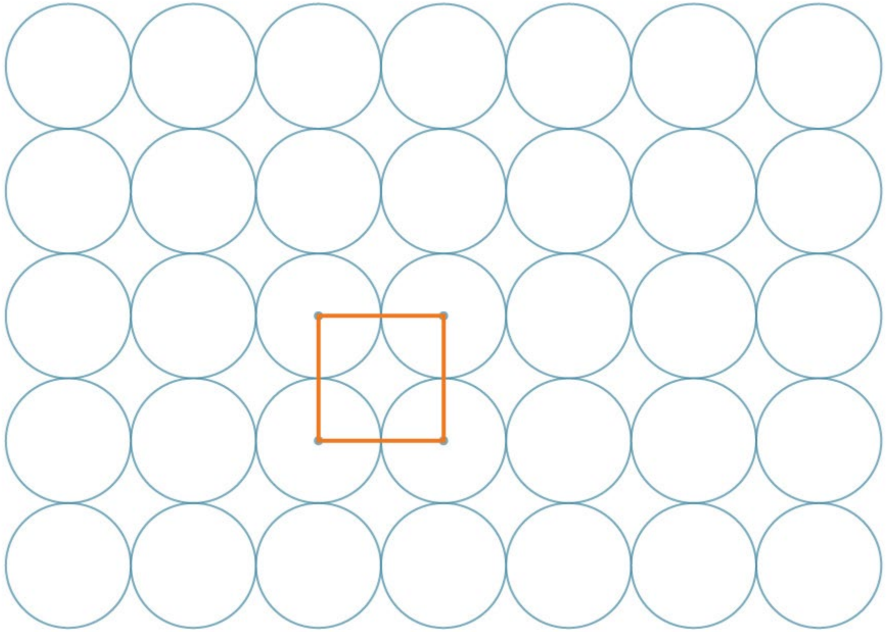
In symbols an n-dimensional sphere of radius r is:

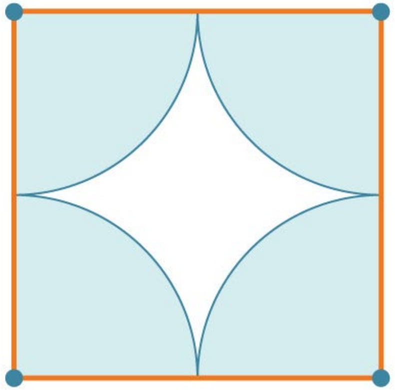
$$S^n(x) = \{y: d(x,y) \le r\}$$

The **hamming sphere** of radius r around a code word x is the collection of points which differ from x by at most r-flipped bits.



Sphere Packing: Fill space with non-overlapping spheres



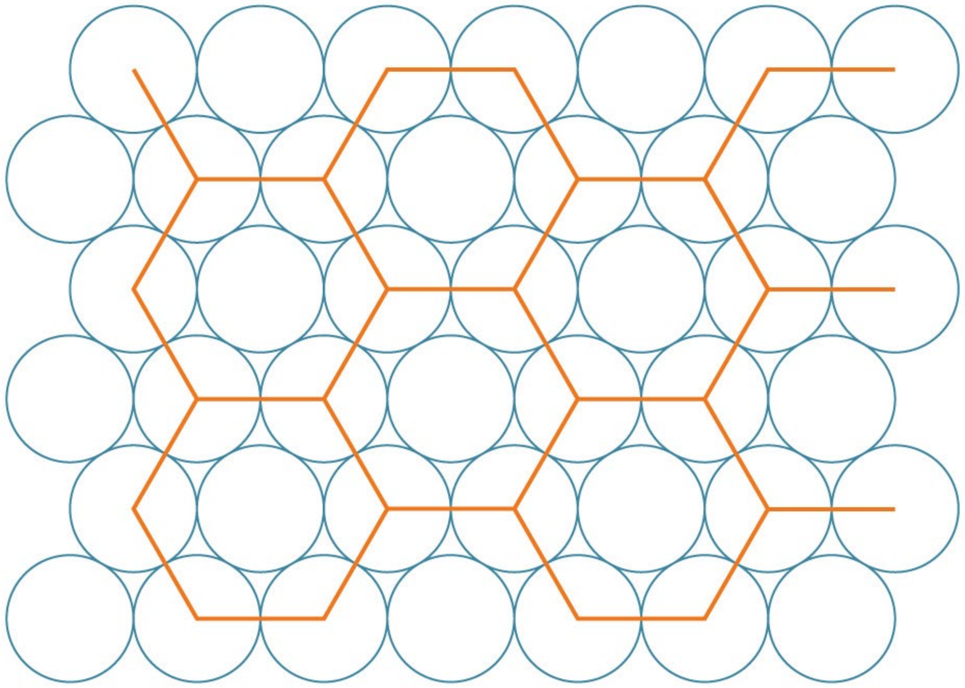


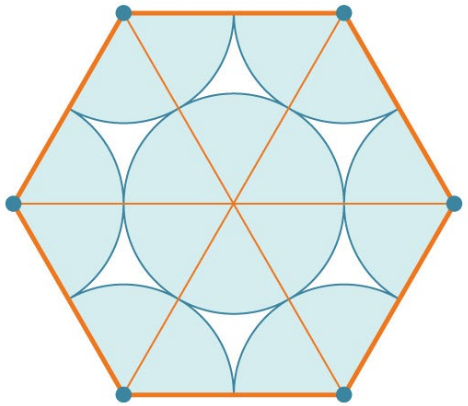
$$\text{Density of the packing} = \frac{\textit{Area of the square covered by circles}}{\textit{Area of the square}}$$

$$= \frac{4\left(\frac{1}{4}\pi r^2\right)}{(2r)^2} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4} \approx .785$$

Pictures from Quanta Magazine: “The math of social distancing is a lesson in Geometry”

Sphere Packing: Fill space with non-overlapping spheres





$$\text{Density of the packing} = \frac{\textit{Portion of the hexagon covered by circles}}{\textit{Area of the hexagon}}$$

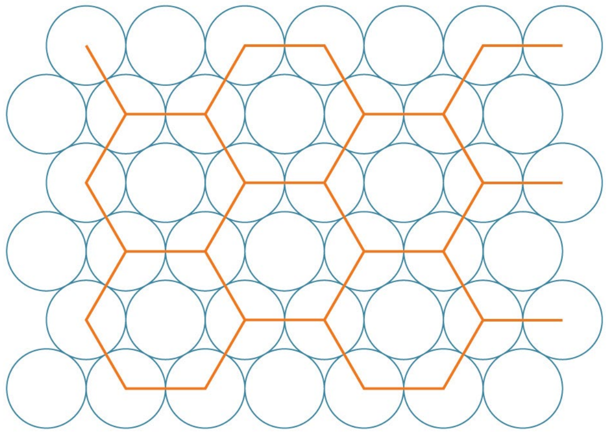
$$= \frac{3(\pi r^2)}{\frac{3(2r)^2\sqrt{3}}{4}} = \frac{3\pi}{6\sqrt{3}} = \frac{\pi}{2\sqrt{3}} \approx .907$$

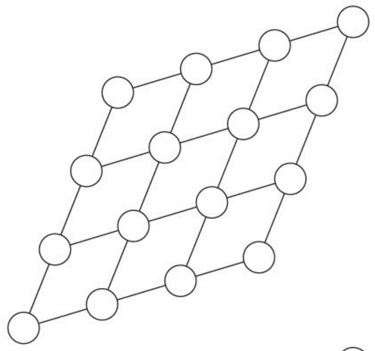
Lattice Packings

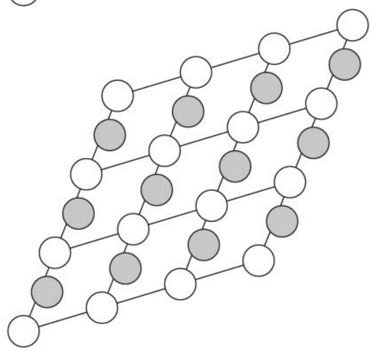
- A lattice is collection of points in space with a well-defined relation between them. Formally:
- a lattice in $\mathbb{R}^n$ is a discreet subgroup of rank n.
- If a sphere packing can be associated to a translationally invariant lattice – we can compute its density

Density of lattice packing =

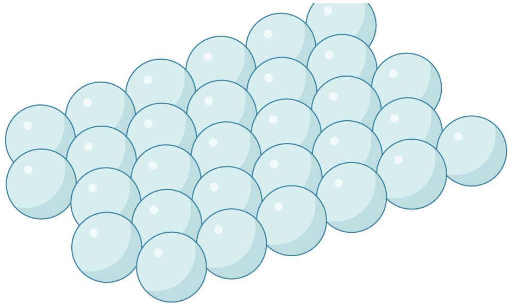
$$\frac{Portion\ of\ the\ polytope\ covered\ by\ circles}{volume\ of\ the\ polytope}$$

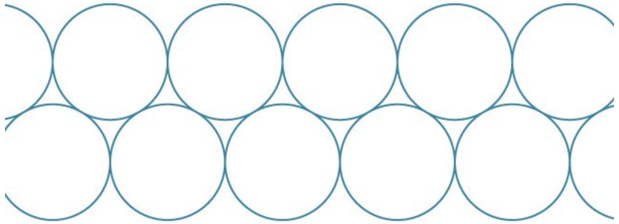


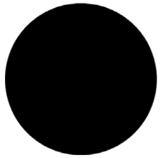




Top Image from Quanta Magazine: “The math of social distancing is a lesson in geometry”.







Dimension 3

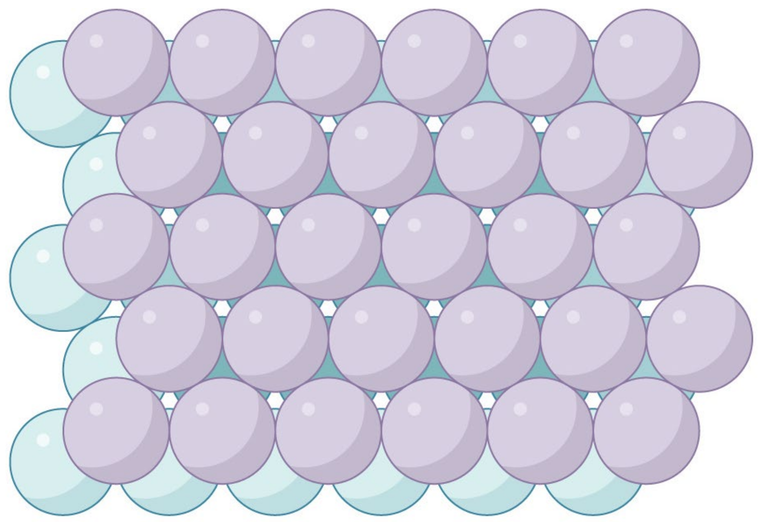
Our first layer should be arranged so that the 2D “shadow” is our hexagonal arrangement.

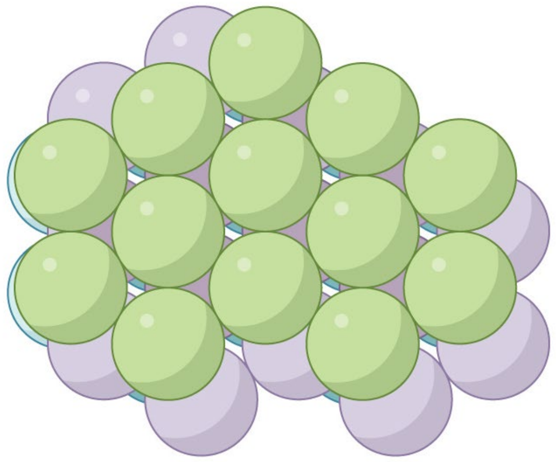
For the second layer: we nestle the spheres in the gaps left by the previous layer

Sphere Packing in Dimension 3

For the third layer choices-- if we duplicate the first level the result is a stack with gaps– each of which is shaped like a hexagon.

- This is a **hexagonal close packing** – an arrangement shared by the atoms in metals such as titanium and zinc.
- The hexagonal close packing gives you a density of $\approx 74\%$.





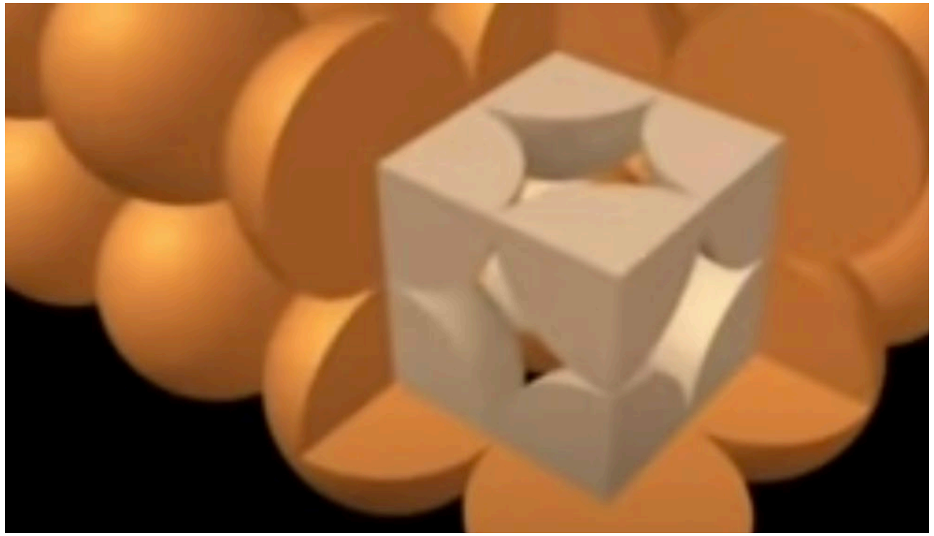
Sphere Packing in Dimension 3

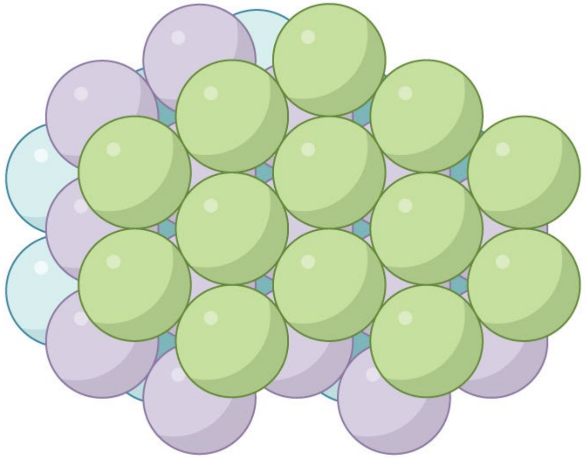
Alternatively, we could try to close the gaps with our third layer.

- This is a **cubic close packing** – an arrangement shared by the atoms in metals such as silver and gold.

- The cubic close packing also gives you a density of $\approx 74\%$.

- Both packings are equally dense, and this is the best you can do! (Proved by Hales in 1998)





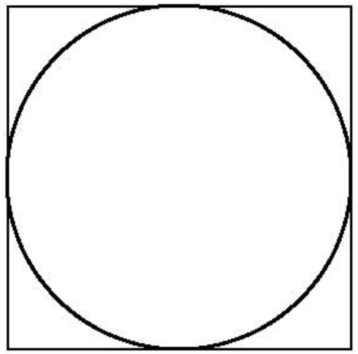
Why is packing in higher dimensions less effective?

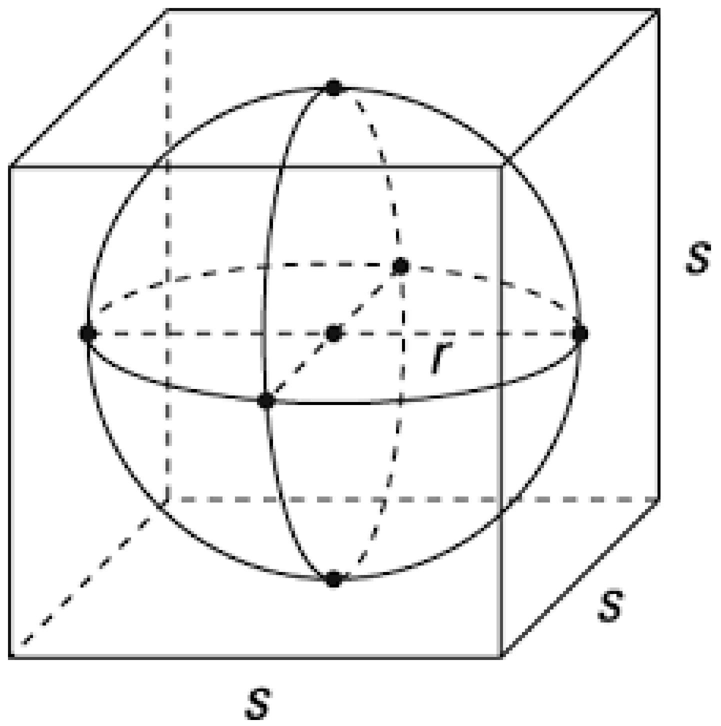
- In 2-dimensions a circle inscribed inside a square covers

$$\frac{Area\ of\ circle}{Area\ of\ square} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4} \approx\ .785$$

- In 3-dimensions a sphere inscribed inside a cube covers

$$\frac{volume\ of\ sphere}{volume\ of\ cube} = \frac{\frac{4}{3}\pi r^3}{(2r)^3} = \frac{\pi}{6} \approx\ .523$$





Higher dimensional spheres get “spikey”

- If you inscribe an n-dimensional sphere inside of an n-dimensional cube:

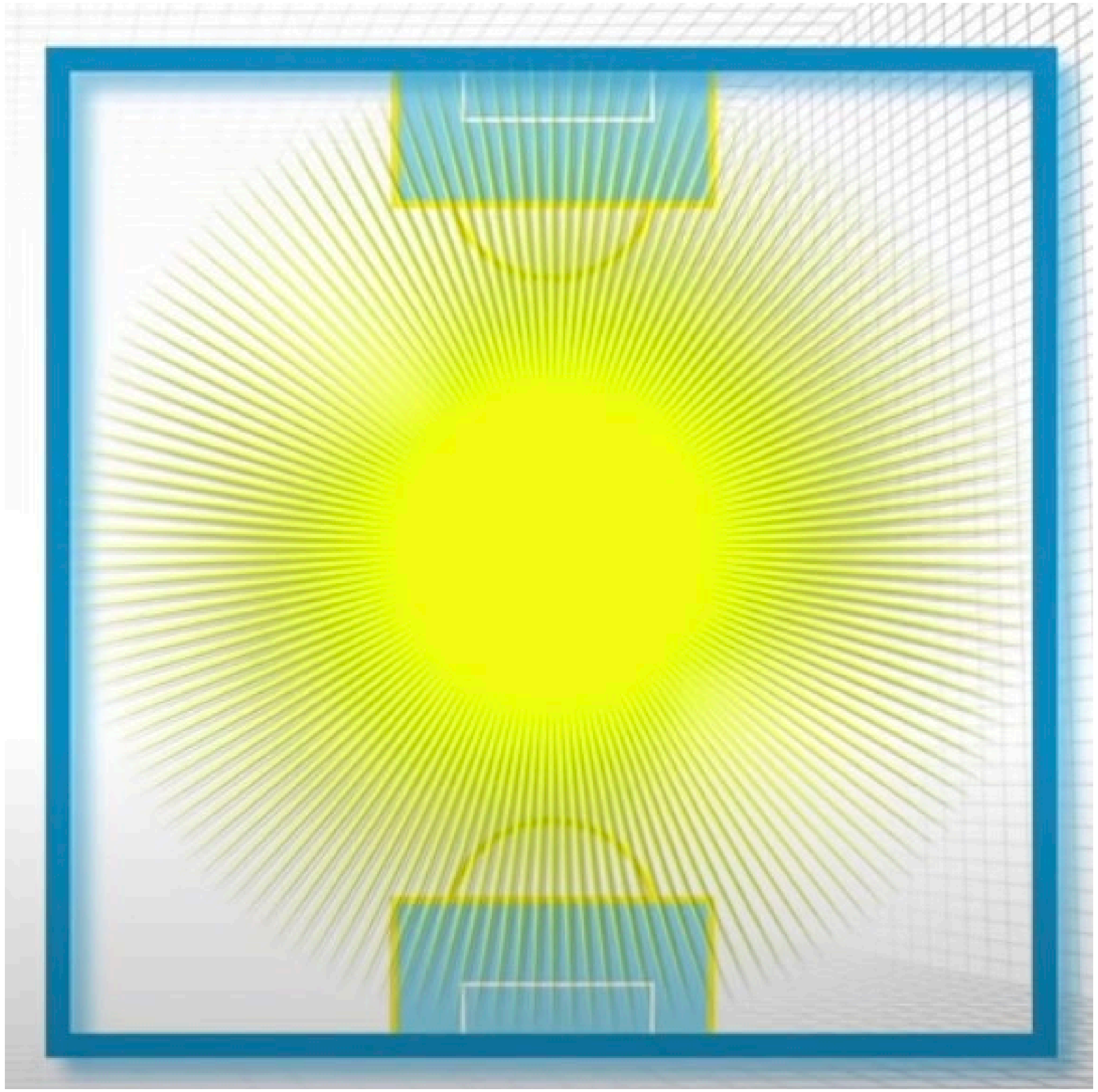
$$\frac{volume\ of\ sphere}{volume\ of\ cube} \to 0\ as\ n\ \to \infty.$$

- In dimension 30, the volume of the sphere is

$$10^{-13} = 0.0000000000001$$

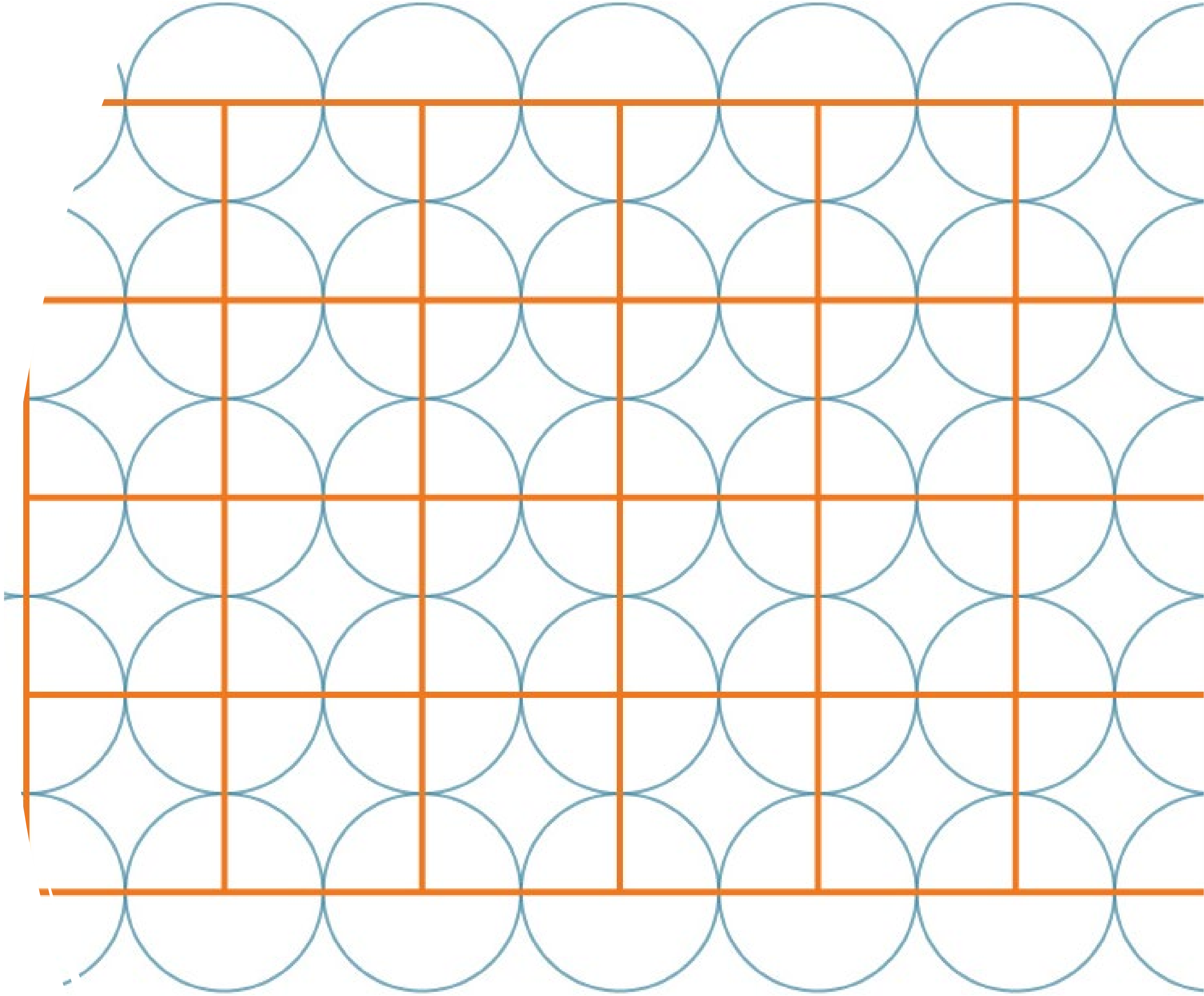
of the volume of the cube.

- This is roughly how large a grain of sand is relative to the the MCG !



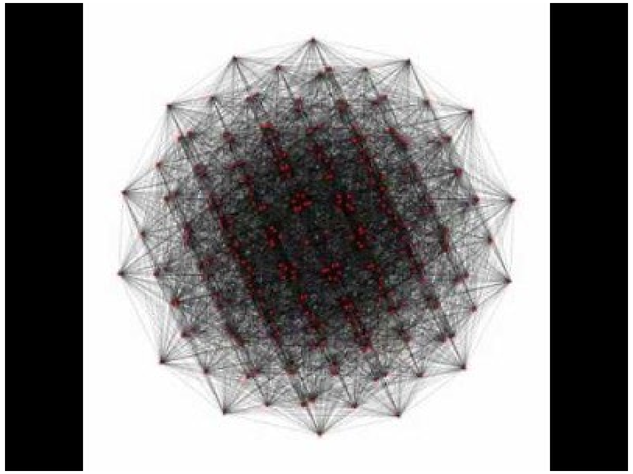
We know almost nothing about sphere packing in higher dimensions!

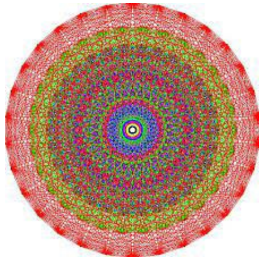
- In dimensions 8 and 24 we have special lattices: The E_8-lattice and the Leech lattice.
- We can build sphere packings around these lattices –just like we did in dimensions 2 and 3!



We know almost nothing about sphere packing in higher dimensions!

- Points in the E_8-lattice are exactly $\sqrt{2}$ apart – we can surround each point by a sphere of radius $\frac{\sqrt{2}}{2}$.
- This sphere packing has a density of $\approx 25.36\%$.
- In her 2016 paper, Maryna Viazovska showed that this was the best possible sphere packing in dimension 8.

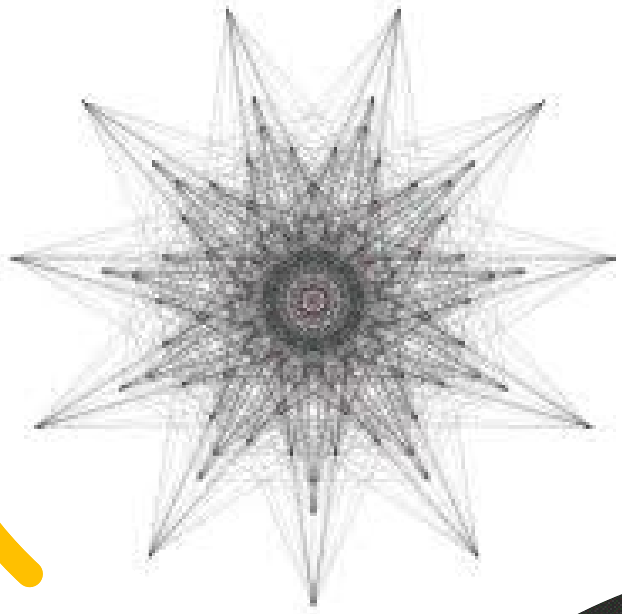


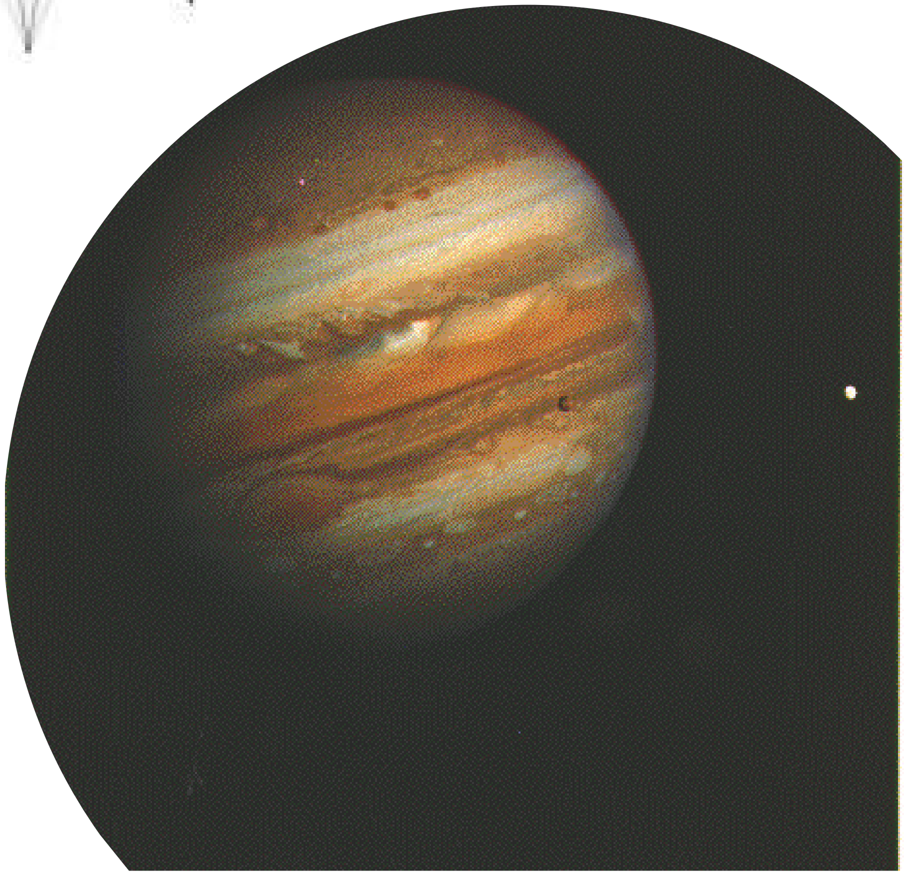


Wait, weren't we talking about codes ?

In 1971, Leech and Sloane built a sphere packing from a 24-dimensional lattice – now called the **Leech lattice**.

- The Leech lattice is very similar to the E_8-lattice, but more elaborate.
- This sphere packing was used to construct the 24-bit error correcting **Golay code**.
- This encodes 12-bit code words in 24-bit sequences.
- It can correct 3-bit errors and detect 7-bit errors.
- The Voyager space craft used the Golay code!





Want to Learn More?

- The article "A conceptual breakthrough in sphere packing" by Henry Cohn in the Notices of the AMS is great for mathematicians wanting to know more about the fine print (and are not experts in modular forms): https://arxiv.org/abs/1611.01685
- This is a great video introducing higher dimensional spheres: https://www.youtube.com/watch?v=ciM6wigZK0w&t=318s
- This is an interesting series on error correcting codes: https://www.youtube.com/watch?v=5sskbSvha9M
- A good video introducing the construction of the Leech lattice: https://www.youtube.com/watch?v=ycpmMnO3-Uk