The Bits Between The Bits

Illustrious 2011

Nicholas Jackson

Easter 2011

Error correcting codes, sphere packings and abstract algebra.

Error correcting codes, sphere packings and abstract algebra.

Further reading

- **T M Thompson**, *From Error-Correcting Codes Through Sphere Packings To Simple Groups*, Carus Mathematical Monographs 21, Mathematical Association of America (1983)
- **J H Conway**, **N J A Sloane**, *Sphere Packings, Lattices and Groups*, third edition, Grundlehren der Mathematischen Wissenschaften 290, Springer (1999)

Data? Rewind Tape

Then five minutes, fingers crossed,
hoping not to witness the terror
of "R: Tape Loading Error"
– M J Hibbett, *Hey Hey 16K*

Then five minutes, fingers crossed,
hoping not to witness the terror
of "R: Tape Loading Error"
– M J Hibbett, *Hey Hey 16K*

Two weekends in a row I came in and found that all my stuff had been dumped and nothing was done. I was really aroused and annoyed because I wanted those answers and two weekends had been lost. And so I said 'Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?'
– Richard W Hamming

Problem

Reliable storage of data on fallible media

The problem

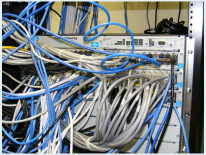
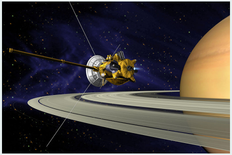
Problem

Reliable storage of data on fallible media




$$\equiv$$

Reliable transmission of data over a noisy channel

Theorem (Noisy Channel Coding Theorem)

1. *For every discrete memoryless channel, the channel capacity*

$$C = \max_{\mathcal{P}_X} I(X;Y)$$

has the property that for any $\epsilon > 0$ *and* $R < C$*, for large enough* N*, there exists a code of length* N *and rate* $\geqslant R$*, and a decoding algorithm, such that the maximal probability of block error is* $< \epsilon$

2. *If a probability of bit error* p_b *is acceptable, rates of up to*

$$R(p_b) = \frac{C}{1 - H_2(p_b)}$$

are achievable.

3. *For any* p_b*, rates greater than* $R(p_b)$ *are not achievable*

Shannon's Theorem

Theorem (Paraphrase)

Information can be communicated over a noisy channel at a nonzero rate with arbitrarily small error probability.

Binary channel: Data transmitted as streams of 1s and 0s.
Most of what we want to store or transmit isn't like this, so encode it using a collection of codewords, such as a character set.

Binary channel: Data transmitted as streams of 1s and 0s. Most of what we want to store or transmit isn't like this, so encode it using a collection of codewords, such as a character set.

ASCII American Standard Code for Information Interchange

Encoding scheme

Binary channel: Data transmitted as streams of 1s and 0s.
Most of what we want to store or transmit isn't like this, so encode it using a collection of codewords, such as a character set.

ASCII	American Standard Code for Information Interchange
EBCDIC	Extended Binary Coded Decimal Interchange Code

Binary channel: Data transmitted as streams of 1s and 0s.
Most of what we want to store or transmit isn't like this, so encode it using a collection of codewords, such as a character set.

ASCII American Standard Code for Information Interchange

EBCDIC Extended Binary Coded Decimal Interchange Code
An alleged character set used on IBM dinosaurs. It exists in at least six mutually incompatible versions, all featuring such delights as non-contiguous letter sequences and the absence of several ASCII punctuation characters fairly important for modern computer languages

Encoding scheme

Binary channel: Data transmitted as streams of 1s and 0s. Most of what we want to store or transmit isn't like this, so encode it using a collection of codewords, such as a character set.

ASCII American Standard Code for Information Interchange

EBCDIC Extended Binary Coded Decimal Interchange Code
An alleged character set used on IBM dinosaurs. It exists in at least six mutually incompatible versions, all featuring such delights as non-contiguous letter sequences and the absence of several ASCII punctuation characters fairly important for modern computer languages... See also fear and loathing.

So. Let's use ASCII...

So. Let's use ASCII...

H	72	01001000
E	69	01000101
L	76	01001100
L	76	01001100
O	79	01001111

So. Let's use ASCII...

H	72	01001000
E	69	01000101
L	76	01001100
L	76	01001100
O	79	01001111

Transmit (or store)

01001000 01000101 01001100 01001100 01001111

So. Let's use ASCII...

H	72	01001000
E	69	01000101
L	76	01001100
L	76	01001100
O	79	01001111

Transmit (or store)

01001000 01000101 01001100 01001100 01001111

Decode at the other end by splitting up into eight-bit chunks and reversing the encoding process.

Transmission error

But suppose something goes wrong in transmission.

$$01001000\ 01000101\ 01001100\ 01001100\ 01001111 = \text{HELLO}$$

But suppose something goes wrong in transmission.

01101000 01000101 00001100 01001100 01001011 $=$ DE?LK

Transmission error

But suppose something goes wrong in transmission.

01101000 01000101 00001100 01001100 01001011 = DE?LK

Question

How do we know that an error has occurred?

Transmission error

But suppose something goes wrong in transmission.

01101000 01000101 00001100 01001100 01001011 = DE?LK

Question

How do we know that an error has occurred?

Answer

Design a clever coding scheme so that we can tell when something's gone wrong.

We can still use ASCII, but we introduce an extra transmission coding/decoding step in the middle.

Transmission error

But suppose something goes wrong in transmission.

01101000 01000101 00001100 01001100 01001011 = DE?LK

Question

How do we know that an error has occurred?

Answer

Design a clever coding scheme so that we can tell when something's gone wrong.

We can still use ASCII, but we introduce an extra transmission coding/decoding step in the middle.

Better answer

Design an even cleverer coding scheme so that we can tell what the message should have been.

Naïve but valid approach

SSeenndd eeaacchh ccooddeewwoorrdd ttwwiiccee

If one letter/codeword in a given pair doesn't agree with the other one, then we know an error has occurred.

Naïve but valid approach

SSeenndd eeaacchh ccooddeewwoorrdd ttwwiiccee

If one letter/codeword in a given pair doesn't agree with the other one, then we know an error has occurred.

Cleverer but still naïve approach

SSSeeennnddd eeeaaaccchhh cccooodddeeewwwooorrrddd ttthhhrrriiiccceee

Assuming we've tweaked transmission rate so that the error probability is small enough, then we can detect and correct single errors.

HELLO $\longrightarrow$ HHHEEELLLLLLOOO $\longrightarrow$ HDHEEELL?LLLKOO

Now use a majority voting algorithm (FPTP!) to correct the error:

$$\begin{array}{ccc} \text{HDH} & \longrightarrow & \text{H} \\ \text{EEE} & \longrightarrow & \text{E} \\ \text{LL?} & \longrightarrow & \text{L} \\ \text{LLL} & \longrightarrow & \text{L} \\ \text{KOO} & \longrightarrow & \text{O} \end{array}$$

This works, but it's not a very efficient way of doing things. We have to transmit three bits of data for every bit of actual information.

$$\text{Rate} = \frac{\text{message bits}}{\text{total bits}}$$

In general we'll talk about (n, r) codes: n total bits, r message bits.

This works, but it's not a very efficient way of doing things. We have to transmit three bits of data for every bit of actual information.

$$\text{Rate} = \frac{\text{message bits}}{\text{total bits}}$$

In general we'll talk about (n, r) codes: n total bits, r message bits. The triple block repetition code has parameters $(3, 1)$, and rate $\frac{1}{3} \approx 0.333$.

This works, but it's not a very efficient way of doing things. We have to transmit three bits of data for every bit of actual information.

$$\text{Rate} = \frac{\text{message bits}}{\text{total bits}}$$

In general we'll talk about (n, r) codes: n total bits, r message bits. The triple block repetition code has parameters $(3, 1)$, and rate $\frac{1}{3} \approx 0.333$.
We expect a certain amount of trade-off for the security of error-correction, but surely we can do better than this?

Better approach (error detection)

Turn 8–bit codewords into 9–bit codewords by adding a parity check bit at the end, so that the total number of 1s is even.

(This is like check digits in credit card numbers and ISBNs.)

Better approach (error detection)

Turn 8–bit codewords into 9–bit codewords by adding a parity check bit at the end, so that the total number of 1s is even.

(This is like check digits in credit card numbers and ISBNs.)

H	72	01001000	010010000
E	69	01000101	010001011
L	76	01001100	010011001
L	76	01001100	010011001
O	79	01001111	010011111

Better approach (error detection)

Turn 8–bit codewords into 9–bit codewords by adding a parity check bit at the end, so that the total number of 1s is even.

(This is like check digits in credit card numbers and ISBNs.)

H	72	01001000	010010000
E	69	01000101	010001011
L	76	01001100	010011001
L	76	01001100	010011001
O	79	01001111	010011111

We can detect single bit errors in any codeword: if the parity is wrong then we know the message has been corrupted during transmission.

The rate of this code is $\frac{8}{9} \approx 0.889$.

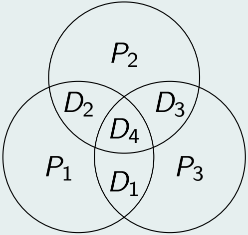
Hamming's $(7,4)$ code $\mathcal{H}_7$

Richard Hamming devised a $(7,4)$ code $\mathcal{H}_7$ with rate $\frac{4}{7} \approx 0.571$. Each codeword has three parity bits and four message bits:

$$P_1\, P_2\, D_1\, P_3\, D_2\, D_3\, D_4$$

and each message bit is checked by at least two of the parity bits:

P_1	checks	D_1	D_2	D_4
P_2	checks	D_2	D_3	D_4
P_3	checks	D_1	D_3	D_4

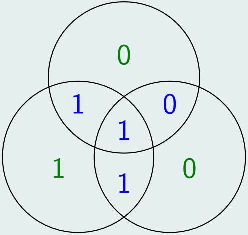


Choose P_1, P_2 and P_3 so that each circle has an even number of 1s.

Hamming's $(7,4)$ code $\mathcal{H}_7$

$\mathcal{H}_7$ can detect and correct a single bit error in any codeword:

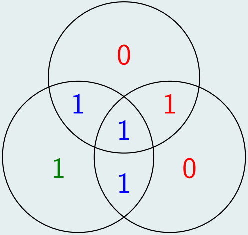
$$1101 \longrightarrow 1010101 \longrightarrow 1010101$$



$$P_1 = 1$$
$$P_2 = 0$$
$$P_3 = 0$$

$\mathcal{H}_7$ can detect and correct a single bit error in any codeword:

$1101 \longrightarrow 1010101 \longrightarrow 1010111$

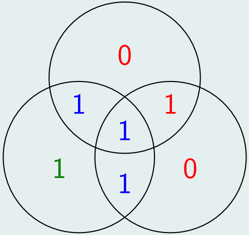


$P_1 = 1$
$P_2 = 0$
$P_3 = 0$

Hamming's $(7,4)$ code $\mathcal{H}_7$

$\mathcal{H}_7$ can detect and correct a single bit error in any codeword:

$1101 \longrightarrow 1010101 \longrightarrow 1010111$



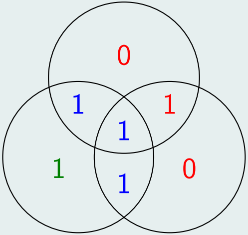
$P_1 = 1$
$P_2 = 0$
$P_3 = 0$

$\mathcal{H}_7$ is one of a family of codes like this.
Use four overlapping spheres to get $\mathcal{H}_{15}$, the Hamming code with parameters $(15, 11)$ and rate $\frac{11}{15} \approx 0.733$.

Hamming's (7, 4) code $\mathcal{H}_7$

$\mathcal{H}_7$ can detect and correct a single bit error in any codeword:

$$1101 \longrightarrow 1010101 \longrightarrow 1010111$$



$P_1 = 1$
$P_2 = 0$
$P_3 = 0$

$\mathcal{H}_7$ is one of a family of codes like this.

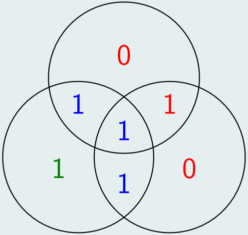
Use four overlapping spheres to get $\mathcal{H}_{15}$, the Hamming code with parameters $(15, 11)$ and rate $\frac{11}{15} \approx 0.733$.

More generally, get a family of $(2^n-1, 2^n-n-1)$ single error-correcting codes. By increasing n we can get a rate arbitrarily close (but not equal) to 1.

Hamming's $(7,4)$ code $\mathcal{H}_7$

$\mathcal{H}_7$ can detect and correct a single bit error in any codeword:

$1101 \longrightarrow 1010101 \longrightarrow 1010111$



$P_1 = 1$
$P_2 = 0$
$P_3 = 0$

$\mathcal{H}_7$ is one of a family of codes like this.

Use four overlapping spheres to get $\mathcal{H}_{15}$, the Hamming code with parameters $(15, 11)$ and rate $\frac{11}{15} \approx 0.733$.

More generally, get a family of $(2^n - 1, 2^n - n - 1)$ single error-correcting codes. By increasing n we can get a rate arbitrarily close (but not equal) to 1.

Practical tradeoff: longer codewords impact on coding/decoding efficiency.

Published as an internal memorandum at Bell Labs, Jul–Sep 1948.

Published as an internal memorandum at Bell Labs, Jul–Sep 1948.
Published externally as
R W Hamming, *Error detecting and error correcting codes*, Bell Systems Tech. J. 29 (1950) 147–160

Published as an internal memorandum at Bell Labs, Jul–Sep 1948.
Published externally as
R W Hamming, *Error detecting and error correcting codes*, Bell Systems Tech. J. 29 (1950) 147–160
Publication delayed due to patent application.

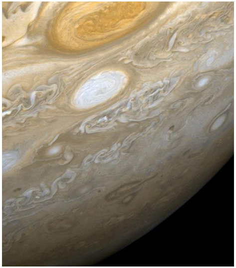
Published as an internal memorandum at Bell Labs, Jul–Sep 1948.
Published externally as
R W Hamming, *Error detecting and error correcting codes*, Bell Systems Tech. J. 29 (1950) 147–160
Publication delayed due to patent application.

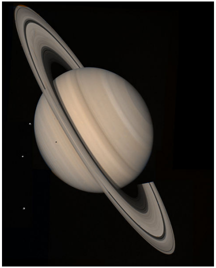
> *I didn't believe that you could patent a bunch of mathematical formulas. I said they couldn't. They said "Watch us." They were right. And since then I have known that I have a very weak understanding of patent laws because, regularly, things that you shouldn't be able to patent – it's outrageous – you can patent.*

1949: Marcel Golay discovers a perfect 3–error-correcting binary code $\mathcal{C}_{23}$ with parameters $(23, 12)$ and rate $\frac{12}{23} \approx 0.522$.

1949: Marcel Golay discovers a perfect 3–error-correcting binary code $\mathcal{C}_{23}$ with parameters $(23, 12)$ and rate $\frac{12}{23} \approx 0.522$.

1979–1981: Voyager 1 and 2 used $\mathcal{C}_{24}$, a modified 24-bit version of this code (with an extra parity bit) to transmit pictures of Jupiter and Saturn.





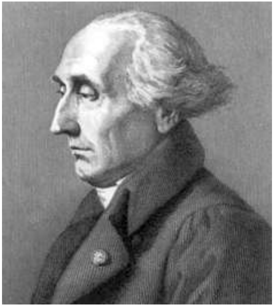
Other codes

- BCH (Bose–Chaudhuri–Hocquenghem) codes: cyclic polynomial codes over finite fields (1959–1960).
- Reed–Solomon codes (1960). Used in CDs, DVDs, DSL, RAID 6, etc.
- Convolutional codes.
- Low-Density Parity Check codes (1960).
- Turbo codes (1993).

Apparently unrelated problem

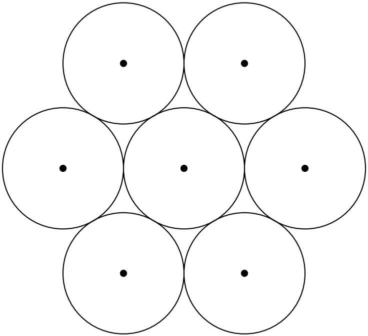
What is the most optimal way of packing together (hyper)spheres in n–dimensional Euclidean space $\mathbb{R}^n$?

Considered by Kepler (1611), Lagrange (1773) and Gauss (1831)



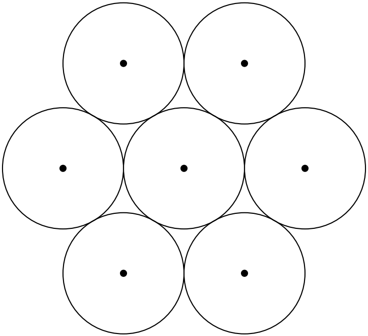
Consider regular or lattice packings of spheres with same radius.

Consider regular or lattice packings of spheres with same radius.



Density Proportion of $\mathbb{R}^n$ occupied by the spheres.

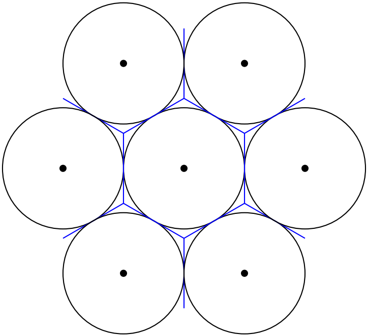
Consider regular or lattice packings of spheres with same radius.



Density Proportion of $\mathbb{R}^n$ occupied by the spheres.

Kissing number Number of adjacent spheres to a given sphere.

Consider regular or lattice packings of spheres with same radius.

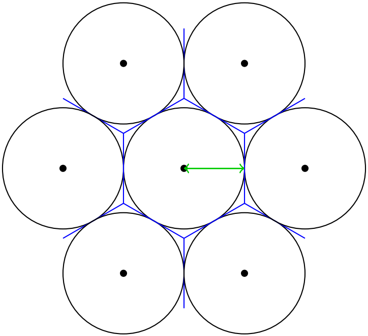


Density Proportion of $\mathbb{R}^n$ occupied by the spheres.

Kissing number Number of adjacent spheres to a given sphere.

Voronoi cell Polygonal/polyhedral/polytopal cell containing the spheres.

Consider regular or lattice packings of spheres with same radius.



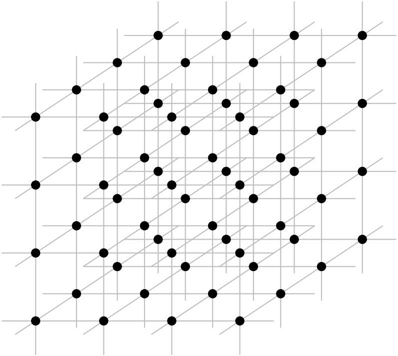
Density Proportion of $\mathbb{R}^n$ occupied by the spheres.

Kissing number Number of adjacent spheres to a given sphere.

Voronoi cell Polygonal/polyhedral/polytopal cell containing the spheres.

Packing radius Half the minimal distance between lattice points.

$\mathbb{Z}^n$: the n–dimensional cubic lattice



Density	$\frac{V_n}{2^n}$
Packing radius	$\frac{1}{2}$
Kissing number	$2n$

$V_n = \frac{\pi^{n/2}}{(n/2)!}$ (volume of n–dimensional ball)

A_n LATTICES

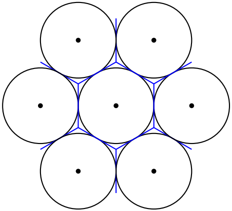
Family of lattices based on the A_n root system.

Density	$\frac{V_n}{\sqrt{2^n(n+1)}}$
Packing radius	$\frac{1}{\sqrt{2}}$
Kissing number	$n(n+1)$

Family of lattices based on the A_n root system.

Density	$\frac{V_n}{\sqrt{2^n(n+1)}}$
Packing radius	$\frac{1}{\sqrt{2}}$
Kissing number	$n(n+1)$

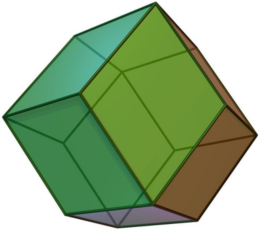
A_2



hexagonal

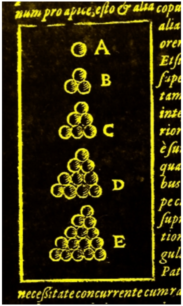
A_3

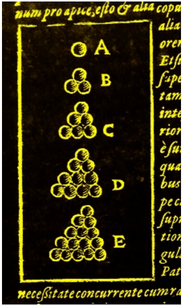
face-centred cubic



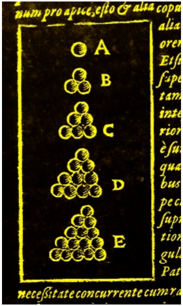
rhombic dodecahedron

- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.

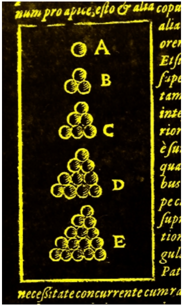




- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.

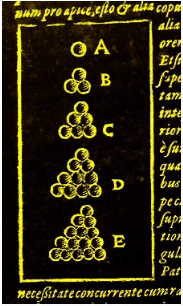


- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.
- 1900: Part of problem 18 on David Hilbert's list of 23 important unsolved problems.



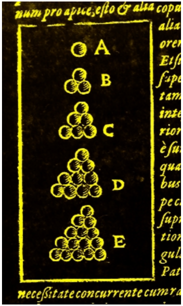
- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.
- 1900: Part of problem 18 on David Hilbert's list of 23 important unsolved problems.
- 1953: László Fejes Tóth proves there are only finitely many irregular lattices to consider.

Kepler's Conjecture

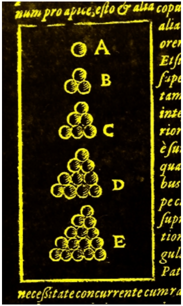


- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.
- 1900: Part of problem 18 on David Hilbert's list of 23 important unsolved problems.
- 1953: László Fejes Tóth proves there are only finitely many irregular lattices to consider.
- 1993: Hsiang publishes possibly incomplete proof.

Kepler's Conjecture

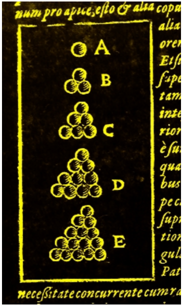


- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.
- 1900: Part of problem 18 on David Hilbert's list of 23 important unsolved problems.
- 1953: László Fejes Tóth proves there are only finitely many irregular lattices to consider.
- 1993: Hsiang publishes possibly incomplete proof.
- 1998: Thomas Hales announces proof.



- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.
- 1900: Part of problem 18 on David Hilbert's list of 23 important unsolved problems.
- 1953: László Fejes Tóth proves there are only finitely many irregular lattices to consider.
- 1993: Hsiang publishes possibly incomplete proof.
- 1998: Thomas Hales announces proof.
- 2003: Referees announce they're "99% certain" that Hales' proof is complete.

Kepler's Conjecture



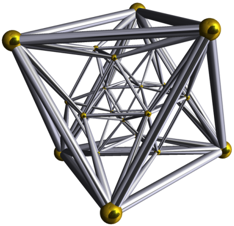
- Kepler (1611): A_3 (face-centred cubic) packing is the densest three-dimensional sphere packing.
- Gauss (1831): it's the densest *regular* packing.
- 1900: Part of problem 18 on David Hilbert's list of 23 important unsolved problems.
- 1953: László Fejes Tóth proves there are only finitely many irregular lattices to consider.
- 1993: Hsiang publishes possibly incomplete proof.
- 1998: Thomas Hales announces proof.
- 2003: Referees announce they're "99% certain" that Hales' proof is complete.
- Greengrocers nonplussed.

D_n: the n–dimensional chessboard lattice.

Points of $\mathbb{Z}^n$ whose coordinates add up to an even number.

Density	$\frac{V_n}{\sqrt{2^{-(n+2)}}}$
Packing radius	$\frac{1}{\sqrt{2}}$
Kissing number	$2n(n-1)$

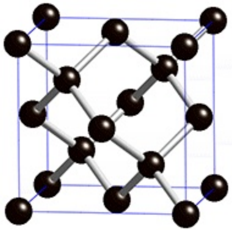
- D_2 is $\mathbb{Z}^2$ (scaled by $\sqrt{2}$ and rotated)
- D_3 is A_3 (face-centred cubic)
- Voronoi cell of D_4 is a 24–cell



D_n^+ LATTICES

D_n^+ is two copies of D_n interleaved.

- D_2^+ is $\mathbb{Z}^2$
- D_3^+ is the molecular structure of diamond



- D_4^+ is $\mathbb{Z}^4$
- D_8^+ is E_8 (one of a finite series with E_6 and E_7)

Dimension	1	2	3	4	5	6	7	8
Density	$\mathbb{Z}$	A_2	A_3	D_4	D_5	E_6	E_7	E_8
Kissing number	$\mathbb{Z}$	A_2	A_3	D_4	D_5	E_6	E_7	E_8
	2	6	12	24	40	72	126	240

Dimension	1	2	3	4	5	6	7	8
Density	$\mathbb{Z}$	A_2	A_3	D_4	D_5	E_6	E_7	E_8
Kissing number	$\mathbb{Z}$	A_2	A_3	D_4	D_5	E_6	E_7	E_8
	2	6	12	24	40	72	126	240

Dimension	12	16	24
Density	K_{12}	Λ_{16}	Λ_{24}
Kissing number	P_{12a}	Λ_{16}	Λ_{24}
	840	4320	196560

Question

What does this have to do with codes?

Question

What does this have to do with codes?

Answer

Good (perfect) codes have an optimal arrangement of codewords in the space of possible codewords: maximise distance between codewords (to allow error correction) *and* number of codewords used.

Relation to codes

Question

What does this have to do with codes?

Answer

Good (perfect) codes have an optimal arrangement of codewords in the space of possible codewords: maximise distance between codewords (to allow error correction) *and* number of codewords used.

Distribute codewords throughout space of n–bit binary strings so that the Hamming spheres don't overlap, but also don't leave many (ideally, any) gaps. Maximise error correction *and* use of codeword space.

QUESTION

What does this have to do with codes?

ANSWER

Good (perfect) codes have an optimal arrangement of codewords in the space of possible codewords: maximise distance between codewords (to allow error correction) *and* number of codewords used.

Distribute codewords throughout space of n–bit binary strings so that the Hamming spheres don't overlap, but also don't leave many (ideally, any) gaps. Maximise error correction *and* use of codeword space.

This is a sphere-packing problem on a 2^n–vertex, n–dimensional hypercube.

Construction A

Choose a linear binary code $\mathcal{C}$ with parameters (n, r).
(A code is linear if the sum, modulo 2, of any two codewords is also a codeword.)

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$.

Choose a linear binary code $\mathcal{C}$ with parameters (n, r).
(A code is linear if the sum, modulo 2, of any two codewords is also a codeword.)

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$.

Geometrically: depict n–bit codewords as vertices of an n–dimensional hypercube, and then glue together lots of copies.

Construction A

Choose a linear binary code $\mathcal{C}$ with parameters (n, r).
(A code is linear if the sum, modulo 2, of any two codewords is also a codeword.)

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$.

Geometrically: depict n–bit codewords as vertices of an n–dimensional hypercube, and then glue together lots of copies.

- The $(n, n-1)$ parity check code gives the D_n lattice.

Construction A

Choose a linear binary code $\mathcal{C}$ with parameters (n, r).
(A code is linear if the sum, modulo 2, of any two codewords is also a codeword.)

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$.

Geometrically: depict n–bit codewords as vertices of an n–dimensional hypercube, and then glue together lots of copies.

- The $(n, n-1)$ parity check code gives the D_n lattice.
- The $(3, 2)$ parity check code gives $D_2 = A_2$, the face-centred cubic lattice.

Choose a linear binary code $\mathcal{C}$ with parameters (n, r).
(A code is linear if the sum, modulo 2, of any two codewords is also a codeword.)

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$.

Geometrically: depict n–bit codewords as vertices of an n–dimensional hypercube, and then glue together lots of copies.

- The $(n, n-1)$ parity check code gives the D_n lattice.
- The $(3, 2)$ parity check code gives $D_2 = A_2$, the face-centred cubic lattice.
- $\mathcal{H}_7$ gives the E_7 lattice.

Choose a linear binary code $\mathcal{C}$ with parameters (n, r).
(A code is linear if the sum, modulo 2, of any two codewords is also a codeword.)

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$.

Geometrically: depict n–bit codewords as vertices of an n–dimensional hypercube, and then glue together lots of copies.

- The $(n, n-1)$ parity check code gives the D_n lattice.
- The $(3, 2)$ parity check code gives $D_2 = A_2$, the face-centred cubic lattice.
- $\mathcal{H}_7$ gives the E_7 lattice.
- $\mathcal{H}_8$ ($\mathcal{H}_7$ with an extra parity bit) gives $E_8 = D_8^+$.

Variation on Construction A:

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$ *and* if the sum $x_1 + \cdots + x_n$ is divisible by 4.

Like Construction A but discard some of the points.

Variation on Construction A:

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$ *and* if the sum $x_1 + \cdots + x_n$ is divisible by 4.

Like Construction A but discard some of the points.

- The $(8, 1)$ repetition code gives the lattice $E_8 = D_8^+$.

Construction B

Variation on Construction A:

A point $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$ is a lattice point if the least significant bits (the 1s columns) of the numbers $x_1, \ldots, x_n$ give a codeword of $\mathcal{C}$ *and* if the sum $x_1 + \cdots + x_n$ is divisible by 4.

Like Construction A but discard some of the points.

- The $(8, 1)$ repetition code gives the lattice $E_8 = D_8^+$.
- $\mathcal{C}_{24}$ gives an interesting 24–dimensional lattice. Slot together two copies of this to get Λ_{24}, the Leech lattice.

The Leech lattice Λ_{24}

- Discovered in 1964 by John Leech (and independently by Ernst Witt in 1940).
- Densest 24–dimensional lattice (density $= \frac{\pi^{12}}{12!} \approx 0.00193$). Densest regular packing; no non-regular packing can be more than 1.65×10^{-30} denser.
- Voronoi cell is a 24–dimensional polytope (hyper-polyhedron) with 16 969 680 faces.
- Related (via vertex algebras and conformal field theory) to string theory.
- Can also be constructed as the product of three copies of E_8. (And in many other ways: qv **J H Conway**, **N J A Sloane**, *Twenty-three constructions for the Leech lattice*, chapter 24 of SPLAG.)

Driven to abstraction

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

The integers

- A set $\mathbb{Z}$ together with addition, a way of combining two elements to get a third (binary operation).

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

The integers

- A set $\mathbb{Z}$ together with addition, a way of combining two elements to get a third (binary operation).
- Associativity: $(a+b)+c = a+(b+c)$

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

The integers

- A set $\mathbb{Z}$ together with addition, a way of combining two elements to get a third (binary operation).
- Associativity: $(a+b)+c = a+(b+c)$
- Commutativity: $a+b = b+a$

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

The integers

- A set $\mathbb{Z}$ together with addition, a way of combining two elements to get a third (binary operation).
- Associativity: $(a+b)+c = a+(b+c)$
- Commutativity: $a+b=b+a$
- Special number 0 (identity) such that $a+0=0+a=a$

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

THE INTEGERS

- A set $\mathbb{Z}$ together with addition, a way of combining two elements to get a third (binary operation).
- Associativity: $(a+b)+c = a+(b+c)$
- Commutativity: $a+b = b+a$
- Special number 0 (identity) such that $a+0 = 0+a = a$
- Inverses: $(-a)+a = 0 = a+(-a)$

Third strand of talk: group theory.
Mathematicians like to generalise and abstract things, so let's do this with the fundamental properties of arithmetic.

The integers

- A set $\mathbb{Z}$ together with addition, a way of combining two elements to get a third (binary operation).
- Associativity: $(a+b)+c = a+(b+c)$
- Commutativity: $a+b = b+a$
- Special number 0 (identity) such that $a+0 = 0+a = a$
- Inverses: $(-a)+a = 0 = a+(-a)$
- Closure: $a+b$ is also an integer

Generalise this to get a group:

A set G and a binary operation $*$ such that:

- $*$ is associative: can ignore parentheses

Generalise this to get a group:

A set G and a binary operation $*$ such that:

- $*$ is associative: can ignore parentheses
- Special identity element e in G such that $e * g = g * e = g$

Generalise this to get a group:

A set G and a binary operation $*$ such that:

- $*$ is associative: can ignore parentheses
- Special identity element e in G such that $e * g = g * e = g$
- Inverses g^{-1} such that $g * g^{-1} = g^{-1} * g = e$

Generalise this to get a group:

A set G and a binary operation $*$ such that:

- $*$ is associative: can ignore parentheses
- Special identity element e in G such that $e * g = g * e = g$
- Inverses g^{-1} such that $g * g^{-1} = g^{-1} * g = e$
- Closure: $g * h$ is in G for all g and h

Generalise this to get a group:

A set G and a binary operation $*$ such that:

- $*$ is associative: can ignore parentheses
- Special identity element e in G such that $e * g = g * e = g$
- Inverses g^{-1} such that $g * g^{-1} = g^{-1} * g = e$
- Closure: $g * h$ is in G for all g and h
- ($*$ is commutative: can ignore order, so $g * h = h * g$)

Examples

Cyclic groups

$\mathbb{Z}_n = \{0, \ldots, n-1\}$ with modulo–n addition.

$+_n$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Klein 4–group

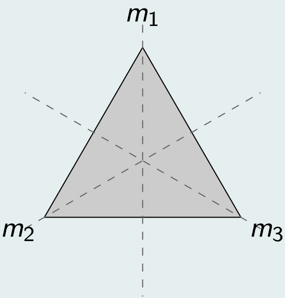
$*$	e	a	b	c
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

See also "Finite Simple Group of Order 2".

Mainly interested in the underlying structure (isomorphism)

Symmetries of geometric objects are a rich source of interesting group structures. Also, groups are a good way of describing symmetry.

Dihedral groups



	e	r_+	r_-	m_1	m_2	m_3
e	e	r_+	r_-	m_1	m_2	m_3
r_+	r_+	r_-	e	m_2	m_3	m_1
r_-	r_-	e	r_+	m_3	m_1	m_2
m_1	m_1	m_3	m_2	e	r_-	r_+
m_2	m_2	m_1	m_3	r_+	e	r_-
m_3	m_3	m_2	m_1	r_-	r_+	e

- Elements are "ways you can flip a triangle round"
- Multiplication operation is "do one after another"
- Nonabelian group (commutativity fails)

Subgroup A smaller group embedded inside a larger one.

	e	r_+	r_-	m_1	m_2	m_3
e	e	r_+	r_-	m_1	m_2	m_3
r_+	r_+	r_-	e	m_2	m_3	m_1
r_-	r_-	e	r_+	m_3	m_1	m_2
m_1	m_1	m_3	m_2	e	r_-	r_+
m_2	m_2	m_1	m_3	r_+	e	r_-
m_3	m_3	m_2	m_1	r_-	r_+	e

Subgroup A smaller group embedded inside a larger one.

	e	r_+	r_-	m_1	m_2	m_3
e	e	r_+	r_-	m_1	m_2	m_3
r_+	r_+	r_-	e	m_2	m_3	m_1
r_-	r_-	e	r_+	m_3	m_1	m_2
m_1	m_1	m_3	m_2	e	r_-	r_+
m_2	m_2	m_1	m_3	r_+	e	r_-
m_3	m_3	m_2	m_1	r_-	r_+	e

Normal subgroup Special sort of subgroup: can decompose larger groups nicely as a product of normal subgroups (qv prime factorisation of integers)

SUBGROUP A smaller group embedded inside a larger one.

	e	r_+	r_-	m_1	m_2	m_3
e	e	r_+	r_-	m_1	m_2	m_3
r_+	r_+	r_-	e	m_2	m_3	m_1
r_-	r_-	e	r_+	m_3	m_1	m_2
m_1	m_1	m_3	m_2	e	r_-	r_+
m_2	m_2	m_1	m_3	r_+	e	r_-
m_3	m_3	m_2	m_1	r_-	r_+	e

NORMAL SUBGROUP Special sort of subgroup: can decompose larger groups nicely as a product of normal subgroups (qv prime factorisation of integers)

SIMPLE GROUP A group with no proper, nontrivial normal subgroups (qv prime numbers)

Classification of finite simple groups

If G is simple, then it is one of the following types:

1. $\mathbb{Z}_p$ where p is prime

Classification of finite simple groups

If G is simple, then it is one of the following types:

1. $\mathbb{Z}_p$ where p is prime
2. A_n where $n \geq 5$

Classification of finite simple groups

If G is simple, then it is one of the following types:

1. $\mathbb{Z}_p$ where p is prime
2. A_n where $n \geq 5$
3. a finite group of Lie type

Classification of finite simple groups

If G is simple, then it is one of the following types:

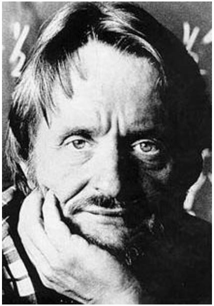
1. $\mathbb{Z}_p$ where p is prime
2. A_n where $n \geq 5$
3. a finite group of Lie type
4. one of 26 others (sporadic groups)

Group	Order	Group	Order	Group	Order
M_{11}	7920	M_{12}	95040	M_{22}	443520
M_{23}	10200960	M_{24}	244823040	J_1	175560
J_2	604800	J_3	50232960	J_4	$\approx 8.68\times10^{19}$
Fi_{22}	$\approx 6.46\times10^{13}$	Fi_{23}	$\approx 4.09\times10^{18}$	Fi_{24}	$\approx 1.26\times10^{24}$
Co_1	$\approx 4.16\times10^{18}$	Co_2	$\approx 4.23\times10^{13}$	Co_3	$\approx 4.96\times10^{11}$
HS	44352000	McL	898128000	He	4030387200
Ru	$\approx 1.46\times10^{11}$	Suz	$\approx 4.48\times10^{11}$	$O'N$	$\approx 4.61\times10^{11}$
HN	$\approx 2.73\times10^{14}$	Ly	$\approx 5.18\times10^{16}$	Th	$\approx 9.07\times10^{16}$
B	$\approx 4.15\times10^{33}$	M	$\approx 8.08\times10^{53}$		

The symmetry group of Λ_{24}

Leech suspected that the symmetry group of his lattice Λ_{24} might contain some interesting simple groups.

1968: The problem came to the attention of John Horton Conway

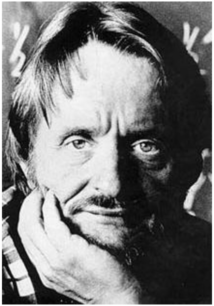


Conway sets aside 6 hours on Wednesday afternoons and 12 hours on Saturdays to solve the problem

The symmetry group of Λ_{24}

Leech suspected that the symmetry group of his lattice Λ_{24} might contain some interesting simple groups.

1968: The problem came to the attention of John Horton Conway



Conway sets aside 6 hours on Wednesday afternoons and 12 hours on Saturdays to solve the problem. . . and finishes just after midnight on the first Saturday, having calculated the structure of the symmetry group Co_0, and found three new sporadic groups Co_1, Co_2 and Co_3.