Carnegie Mellon University

The Curious Case of Sphere Packing in Lean

Federated Logic Conferences (FLoC)

Sidharth Hariharan
based on joint work with Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska, and community contributors

25 July, 2026

sidharthhariharan@cmu.edu

Finding our Footing

The Sphere Packing Problem in Dimension 8

The sphere packing problem is to answer the following question:

What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

The Sphere Packing Problem in Dimension 8

The sphere packing problem is to answer the following question:

> What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4].

The Sphere Packing Problem in Dimension 8

The sphere packing problem is to answer the following question:

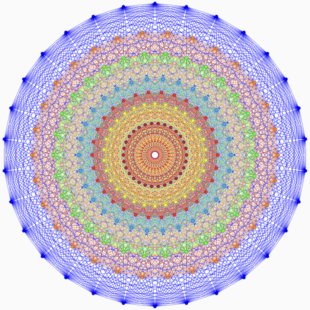
> What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3].

The sphere packing problem is to answer the following question:

> What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

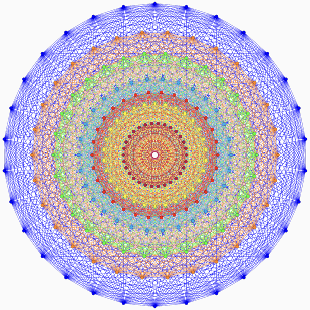
Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3].



The sphere packing problem is to answer the following question:

What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3]. Nodes are vectors and edges connect nearest neighbours.

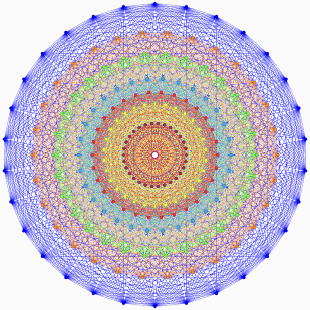


The sphere packing problem is to answer the following question:

What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3]. Nodes are vectors and edges connect nearest neighbours.

This was long suspected, and in 2003, Henry Cohn and Noam Elkies proposed a method that many believed would offer a means to a proof.

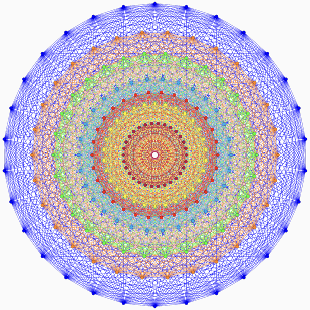


The sphere packing problem is to answer the following question:

> What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3]. Nodes are vectors and edges connect nearest neighbours.

This was long suspected, and in 2003, Henry Cohn and Noam Elkies proposed a method that many believed would offer a means to a proof. They showed that if a certain type of $\mathbb{R}^n \to \mathbb{R}$ function is constructed, then all sphere packings in $\mathbb{R}^n$ are bounded above by a certain quantity that depends on the function.



The Sphere Packing Problem in Dimension 8

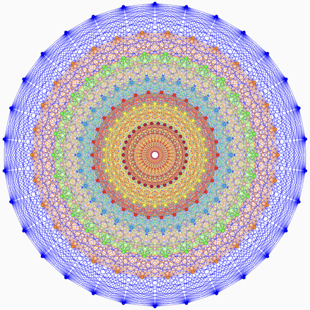
The sphere packing problem is to answer the following question:

What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3]. Nodes are vectors and edges connect nearest neighbours.

This was long suspected, and in 2003, Henry Cohn and Noam Elkies proposed a method that many believed would offer a means to a proof. They showed that if a certain type of $\mathbb{R}^n \to \mathbb{R}$ function is constructed, then all sphere packings in $\mathbb{R}^n$ are bounded above by a certain quantity that depends on the function.

Unfortunately, constructing such functions is extremely difficult.

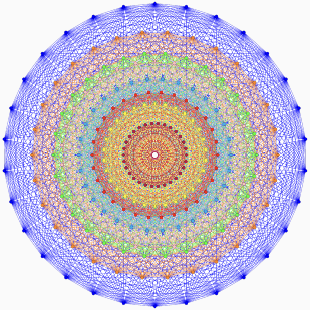


The sphere packing problem is to answer the following question:

> What is the densest arrangement in $\mathbb{R}^n$ of non-overlapping n-spheres of the same radius?

Viazovska famously showed that the densest arrangement in $\mathbb{R}^8$ corresponds to spheres centred at points on the E_8 root lattice [4]. Below, we give an illustration of a projection of the E_8 root system onto its so-called Coxeter plane [3]. Nodes are vectors and edges connect nearest neighbours.

This was long suspected, and in 2003, Henry Cohn and Noam Elkies proposed a method that many believed would offer a means to a proof. They showed that if a certain type of $\mathbb{R}^n \to \mathbb{R}$ function is constructed, then all sphere packings in $\mathbb{R}^n$ are bounded above by a certain quantity that depends on the function.



Unfortunately, constructing such functions is extremely difficult. Viazovska's breakthrough was to construct such a function using the theory of modular forms.

Timeline

Timeline

28 February 2024 — First commit: SH and MV launch project

Timeline

28 February 2024 — First commit: SH and MV launch project

Drafting the blueprint, working on basic definitions

Timeline

28 February 2024	First commit: SH and MV launch project
	Drafting the blueprint, working on basic definitions
Late July 2024	First stable definition of a sphere packing

Timeline

28 February 2024	First commit: SH and MV launch project
	Drafting the blueprint, working on basic definitions
Late July 2024	First stable definition of a sphere packing
	Infrastructure on sphere packings, modular forms, and Fourier analysis
	Blueprint refinements: SL's proof of the modular form inequalities

Timeline

28 February 2024	First commit: SH and MV launch project
	Drafting the blueprint, working on basic definitions
Late July 2024	First stable definition of a sphere packing
	Infrastructure on sphere packings, modular forms, and Fourier analysis
	Blueprint refinements: SL's proof of the modular form inequalities
	Dabbling in metaprogramming
	Groundwork for constructing the magic function

Timeline

28 February 2024	First commit: SH and MV launch project
	Drafting the blueprint, working on basic definitions
Late July 2024	First stable definition of a sphere packing
	Infrastructure on sphere packings, modular forms, and Fourier analysis
	Blueprint refinements: SL's proof of the modular form inequalities
	Dabbling in metaprogramming
	Groundwork for constructing the magic function
13 June 2025	Project opened to the public

Timeline

28 February 2024 — First commit: SH and MV launch project

- *Drafting the blueprint, working on basic definitions*

Late July 2024 — First stable definition of a sphere packing

- *Infrastructure on sphere packings, modular forms, and Fourier analysis*
- *Blueprint refinements: SL's proof of the modular form inequalities*
- *Dabbling in metaprogramming*
- *Groundwork for constructing the magic function*

13 June 2025 — Project opened to the public

- *Growing interest from community contributors*

Timeline

28 February 2024 — First commit: SH and MV launch project

Drafting the blueprint, working on basic definitions

Late July 2024 — First stable definition of a sphere packing

Infrastructure on sphere packings, modular forms, and Fourier analysis

Blueprint refinements: SL's proof of the modular form inequalities

Dabbling in metaprogramming

Groundwork for constructing the magic function

13 June 2025 — Project opened to the public

Growing interest from community contributors

20 October 2025 — First packathon

Timeline

28 February 2024	First commit: SH and MV launch project
	Drafting the blueprint, working on basic definitions
Late July 2024	First stable definition of a sphere packing
	Infrastructure on sphere packings, modular forms, and Fourier analysis
	Blueprint refinements: SL's proof of the modular form inequalities
	Dabbling in metaprogramming
	Groundwork for constructing the magic function
13 June 2025	Project opened to the public
	Growing interest from community contributors
20 October 2025	First packathon
	Interest from AI companies

Timeline

28 February 2024 — First commit: SH and MV launch project

Drafting the blueprint, working on basic definitions

Late July 2024 — First stable definition of a sphere packing

Infrastructure on sphere packings, modular forms, and Fourier analysis

Blueprint refinements: SL's proof of the modular form inequalities

Dabbling in metaprogramming

Groundwork for constructing the magic function

13 June 2025 — Project opened to the public

Growing interest from community contributors

20 October 2025 — First packathon

Interest from AI companies

February 2026 — **Project milestone: formal verification**

Formal Progress

What is a sphere packing, really?

What is a sphere packing, really?

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

What is a sphere packing, really?

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024):

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024):

```
def SpherePacking {P : Set V} (hP : Countable P) {r : ℝ} (hr : r > 0)
  (hPr : ∀ p₁ p₂ : V, p₁ ∈ P → p₂ ∈ P → p₁ ≠ p₂ → Euclidean.dist p₁ p₂ > 2*r) : Set V :=
  {y | ∃ p : V, p ∈ P ∧ y ∈ ball p r}
```

What is a sphere packing, really?

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024): No way to explicitly describe the centres of a packing!

```
def SpherePacking {P : Set V} (hP : Countable P) {r : ℝ} (hr : r > 0)
  (hPr : ∀ p₁ p₂ : V, p₁ ∈ P → p₂ ∈ P → p₁ ≠ p₂ → Euclidean.dist p₁ p₂ > 2*r) : Set V :=
  {y | ∃ p : V, p ∈ P ∧ y ∈ ball p r}
```

What is a sphere packing, really?

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024): No way to explicitly describe the centres of a packing!

```
def SpherePacking {P : Set V} (hP : Countable P) {r : ℝ} (hr : r > 0)
  (hPr : ∀ p₁ p₂ : V, p₁ ∈ P → p₂ ∈ P → p₁ ≠ p₂ → Euclidean.dist p₁ p₂ > 2*r) : Set V :=
  {y | ∃ p : V, p ∈ P ∧ y ∈ ball p r}
```

Second attempt (4 July, 2024):

What is a sphere packing, really?

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024): No way to explicitly describe the centres of a packing!

```
def SpherePacking {P : Set V} (hP : Countable P) {r : ℝ} (hr : r > 0)
  (hPr : ∀ p₁ p₂ : V, p₁ ∈ P → p₂ ∈ P → p₁ ≠ p₂ → Euclidean.dist p₁ p₂ > 2*r) : Set V :=
  {y | ∃ p : V, p ∈ P ∧ y ∈ ball p r}
```

Second attempt (4 July, 2024):

```
def Discrete (X : Set V) : Prop := (Countable X) ∧ (∀ x ∈ X, ∃ ε > 0, ∀ y ∈ X, y ≠ x →
     Euclidean.dist x y > ε)
def Centres (X : Set V) : Prop := (Discrete d X) ∧ (∀ x ∈ X, ∀ y ∈ X, x ≠ y →
     Euclidean.dist x y ≥ 2)
def Packing (X P : Set V) : Prop := (Centres d X) ∧ (P = ⋃ x ∈ X, {y | Euclidean.dist x y <
     1})
def isPacking (P : Set V) : Prop := ∃ X, Packing d X P
def LatticePacking ( P : Set V) : Prop := (Packing d  P) ∧ (is_lattice )
def isLatticePacking (P : Set V) : Prop := ∃ , LatticePacking d  P
```

What is a sphere packing, really?

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024): No way to explicitly describe the centres of a packing!

```
def SpherePacking {P : Set V} (hP : Countable P) {r : ℝ} (hr : r > 0)
  (hPr : ∀ p₁ p₂ : V, p₁ ∈ P → p₂ ∈ P → p₁ ≠ p₂ → Euclidean.dist p₁ p₂ > 2*r) : Set V :=
  {y | ∃ p : V, p ∈ P ∧ y ∈ ball p r}
```

Second attempt (4 July, 2024): A soup of unbundled predicates!

```
def Discrete (X : Set V) : Prop := (Countable X) ∧ (∀ x ∈ X, ∃ ε > 0, ∀ y ∈ X, y ≠ x →
     Euclidean.dist x y > ε)
def Centres (X : Set V) : Prop := (Discrete d X) ∧ (∀ x ∈ X, ∀ y ∈ X, x ≠ y →
     Euclidean.dist x y ≥ 2)
def Packing (X P : Set V) : Prop := (Centres d X) ∧ (P = ⋃ x ∈ X, {y | Euclidean.dist x y <
     1})
def isPacking (P : Set V) : Prop := ∃ X, Packing d X P
def LatticePacking ( P : Set V) : Prop := (Packing d  P) ∧ (is_lattice )
def isLatticePacking (P : Set V) : Prop := ∃ , LatticePacking d  P
```

Here's the definition of a sphere packing from the original paper.

> *Let $X \subseteq \mathbb{R}^d$ be a discrete set of points such that $\|x - y\| \geq 2$ for any distinct $x, y \in X$. Then the union $\mathcal{P} = \bigcup_{x \in X} B_d(x, 1)$ is a sphere packing.*

First attempt (1 March, 2024): No way to explicitly describe the centres of a packing!

```
def SpherePacking {P : Set V} (hP : Countable P) {r : ℝ} (hr : r > 0)
  (hPr : ∀ p₁ p₂ : V, p₁ ∈ P → p₂ ∈ P → p₁ ≠ p₂ → Euclidean.dist p₁ p₂ > 2*r) : Set V :=
  {y | ∃ p : V, p ∈ P ∧ y ∈ ball p r}
```

Second attempt (4 July, 2024): A soup of unbundled predicates! Centres *still* not recoverable from packing!

```
def Discrete (X : Set V) : Prop := (Countable X) ∧ (∀ x ∈ X, ∃ ε > 0, ∀ y ∈ X, y ≠ x →
     Euclidean.dist x y > ε)
def Centres (X : Set V) : Prop := (Discrete d X) ∧ (∀ x ∈ X, ∀ y ∈ X, x ≠ y →
     Euclidean.dist x y ≥ 2)
def Packing (X P : Set V) : Prop := (Centres d X) ∧ (P = ⋃ x ∈ X, {y | Euclidean.dist x y <
     1})
def isPacking (P : Set V) : Prop := ∃ X, Packing d X P
def LatticePacking ( P : Set V) : Prop := (Packing d  P) ∧ (is_lattice )
def isLatticePacking (P : Set V) : Prop := ∃ , LatticePacking d  P
```

What is a sphere packing, really?

Third attempt (15 July, 2024):

What is a sphere packing, really?

Third attempt (15 July, 2024):

```
class SpherePackingCentres (X : Set V) [DiscreteTopology X] where
  nonoverlapping : ∀ x ∈ X, ∀ y ∈ X, x ≠ y → Euclidean.dist x y ≥ 2
class LatticePackingCentres (X : AddSubgroup V) [DiscreteTopology X] [isLattice' X] extends
     SpherePackingCentres d X
class PeriodicPackingCentres (X : Set V) [DiscreteTopology X] [SpherePackingCentres d X] { :
     AddSubgroup V} [DiscreteTopology ] [isLattice' ] where
  periodic : ∀ x ∈ X, ∀ y ∈ , x + y ∈ X
```

What is a sphere packing, really?

Third attempt (15 July, 2024): How do you refer to a packing?

```
class SpherePackingCentres (X : Set V) [DiscreteTopology X] where
  nonoverlapping : ∀ x ∈ X, ∀ y ∈ X, x ≠ y → Euclidean.dist x y ≥ 2
class LatticePackingCentres (X : AddSubgroup V) [DiscreteTopology X] [isLattice' X] extends
     SpherePackingCentres d X
class PeriodicPackingCentres (X : Set V) [DiscreteTopology X] [SpherePackingCentres d X] { :
     AddSubgroup V} [DiscreteTopology ] [isLattice' ] where
  periodic : ∀ x ∈ X, ∀ y ∈ , x + y ∈ X
```

What is a sphere packing, really?

Third attempt (15 July, 2024): How do you refer to a packing?

```
class SpherePackingCentres (X : Set V) [DiscreteTopology X] where
  nonoverlapping : ∀ x ∈ X, ∀ y ∈ X, x ≠ y → Euclidean.dist x y ≥ 2
class LatticePackingCentres (X : AddSubgroup V) [DiscreteTopology X] [isLattice' X] extends
    SpherePackingCentres d X
class PeriodicPackingCentres (X : Set V) [DiscreteTopology X] [SpherePackingCentres d X] { :
    AddSubgroup V} [DiscreteTopology ] [isLattice' ] where
  periodic : ∀ x ∈ X, ∀ y ∈ , x + y ∈ X
```

Fourth attempt (30 July, 2024):

What is a sphere packing, really?

Third attempt (15 July, 2024): How do you refer to a packing?

```
class SpherePackingCentres (X : Set V) [DiscreteTopology X] where
  nonoverlapping : ∀ x ∈ X, ∀ y ∈ X, x ≠ y → Euclidean.dist x y ≥ 2
class LatticePackingCentres (X : AddSubgroup V) [DiscreteTopology X] [isLattice' X] extends
     SpherePackingCentres d X
class PeriodicPackingCentres (X : Set V) [DiscreteTopology X] [SpherePackingCentres d X] { :
     AddSubgroup V} [DiscreteTopology ] [isLattice' ] where
  periodic : ∀ x ∈ X, ∀ y ∈ , x + y ∈ X
```

Fourth attempt (30 July, 2024):

```
structure SpherePacking (d : ℕ) where
  centers : Set (EuclideanSpace ℝ (Fin d))
  radius : ℝ
  radius_pos : 0 < radius
  centers_dist : Pairwise (radius ≤ dist · · : centers → centers → Prop)
structure PeriodicSpherePacking (d : ℕ) extends SpherePacking d where
   : AddSubgroup (EuclideanSpace ℝ (Fin d))
  _discrete : DiscreteTopology  := by infer_instance
  _lattice : IsZlattice ℝ  := by infer_instance
  _action : ∀ ⦃x y⦄, x ∈  → y ∈ centers → x + y ∈ centers
```

What is a sphere packing, really?

Third attempt (15 July, 2024): How do you refer to a packing?

```
class SpherePackingCentres (X : Set V) [DiscreteTopology X] where
  nonoverlapping : ∀ x ∈ X, ∀ y ∈ X, x ≠ y → Euclidean.dist x y ≥ 2
class LatticePackingCentres (X : AddSubgroup V) [DiscreteTopology X] [isLattice' X] extends
     SpherePackingCentres d X
class PeriodicPackingCentres (X : Set V) [DiscreteTopology X] [SpherePackingCentres d X] { :
     AddSubgroup V} [DiscreteTopology ] [isLattice' ] where
  periodic : ∀ x ∈ X, ∀ y ∈ , x + y ∈ X
```

Fourth attempt (30 July, 2024): Our first stable definition!

```
structure SpherePacking (d : ℕ) where
  centers : Set (EuclideanSpace ℝ (Fin d))
  radius : ℝ
  radius_pos : 0 < radius
  centers_dist : Pairwise (radius ≤ dist · · : centers → centers → Prop)
structure PeriodicSpherePacking (d : ℕ) extends SpherePacking d where
   : AddSubgroup (EuclideanSpace ℝ (Fin d))
  _discrete : DiscreteTopology  := by infer_instance
  _lattice : IsZlattice ℝ  := by infer_instance
  _action : ∀ ⦃x y⦄, x ∈  → y ∈ centers → x + y ∈ centers
```

What is a sphere packing, really?

Third attempt (15 July, 2024): How do you refer to a packing?

```
class SpherePackingCentres (X : Set V) [DiscreteTopology X] where
  nonoverlapping : ∀ x ∈ X, ∀ y ∈ X, x ≠ y → Euclidean.dist x y ≥ 2
class LatticePackingCentres (X : AddSubgroup V) [DiscreteTopology X] [isLattice' X] extends
    SpherePackingCentres d X
class PeriodicPackingCentres (X : Set V) [DiscreteTopology X] [SpherePackingCentres d X] { :
    AddSubgroup V} [DiscreteTopology ] [isLattice' ] where
  periodic : ∀ x ∈ X, ∀ y ∈ , x + y ∈ X
```

Fourth attempt (30 July, 2024): Our first stable definition! Thank you Gareth Ma!

```
structure SpherePacking (d : ℕ) where
  centers : Set (EuclideanSpace ℝ (Fin d))
  radius : ℝ
  radius_pos : 0 < radius
  centers_dist : Pairwise (radius ≤ dist · · : centers → centers → Prop)
structure PeriodicSpherePacking (d : ℕ) extends SpherePacking d where
   : AddSubgroup (EuclideanSpace ℝ (Fin d))
  _discrete : DiscreteTopology  := by infer_instance
  _lattice : IsZlattice ℝ  := by infer_instance
  _action : ∀ ⦃x y⦄, x ∈  → y ∈ centers → x + y ∈ centers
```

Sphere Packing Theory (SH, GM, BM)

A Very Incomplete Summary

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

A Very Incomplete Summary

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

Modular Forms (CB, SL)

A Very Incomplete Summary

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

Modular Forms (CB, SL)

- Fixing the definition of E_2
- q-expansions of the Eisenstein Series
- H-functions (4th powers of Jacobi Θ-functions) as modular forms
- Definition/API for Δ, relation to η
- Dimension formula for level one

A Very Incomplete Summary

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

Modular Forms (CB, SL)

- Fixing the definition of E_2
- q-expansions of the Eisenstein Series
- H-functions (4th powers of Jacobi Θ-functions) as modular forms
- Definition/API for Δ, relation to η
- Dimension formula for level one

Construction of the Magic Function (SH, BM)

A Very Incomplete Summary

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

Modular Forms (CB, SL)

- Fixing the definition of E_2
- q-expansions of the Eisenstein Series
- H-functions (4th powers of Jacobi Θ-functions) as modular forms
- Definition/API for Δ, relation to η
- Dimension formula for level one

Construction of the Magic Function (SH, BM)

- Definitions and API for a, b, g, their integrands and parametrisations
- Proving `PolyFourierCoeffBound`, a tool to analyse asymptotic behaviour
- Systematically bounding the integrands of a
- Proving slash action relations for ψ_I, ψ_S, ψ_T
- Cauchy-Goursat for unbounded rectangles

A Very Incomplete Summary

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

Modular Forms (CB, SL)

- Fixing the definition of E_2
- q-expansions of the Eisenstein Series
- H-functions (4th powers of Jacobi Θ-functions) as modular forms
- Definition/API for Δ, relation to η
- Dimension formula for level one

Construction of the Magic Function (SH, BM)

- Definitions and API for a, b, g, their integrands and parametrisations
- Proving `PolyFourierCoeffBound`, a tool to analyse asymptotic behaviour
- Systematically bounding the integrands of a
- Proving slash action relations for ψ_I, ψ_S, ψ_T
- Cauchy-Goursat for unbounded rectangles

Quasimodular Forms and Inequalities (SL)

Sphere Packing Theory (SH, GM, BM)

- Definitions of sphere packings, periodic packings, density, sphere packing constant, periodic sphere packing constant
- Density formula for periodic packings
- The theory of the E_8 lattice and the associated sphere packing
- Progress on Cohn-Elkies [2, Theorem 3.1]

Modular Forms (CB, SL)

- Fixing the definition of E_2
- q-expansions of the Eisenstein Series
- H-functions (4th powers of Jacobi Θ-functions) as modular forms
- Definition/API for Δ, relation to η
- Dimension formula for level one

Construction of the Magic Function (SH, BM)

- Definitions and API for a, b, g, their integrands and parametrisations
- Proving `PolyFourierCoeffBound`, a tool to analyse asymptotic behaviour
- Systematically bounding the integrands of a
- Proving slash action relations for ψ_I, ψ_S, ψ_T
- Cauchy-Goursat for unbounded rectangles

Quasimodular Forms and Inequalities (SL)

- Serre derivatives: definitions and API, equivariance and invariance
- Serre derivatives of E_2, E_4, E_6
- Defining F and G, restriction to the imaginary axis, positivity of restrictions
- Modular linear differential equations

Metaprogramming

Metaprogramming

A `norm_num` extension for $\mathbb{C}$

Metaprogramming

A **`norm_num`** extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH

A `norm_num` extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a `conv`-tactic, `norm_numI`

A `norm_num` extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a `conv`-tactic, `norm_numI`
- Simplifies expressions in $\mathbb{C}$ by splitting to real and imaginary parts, running `norm_num` on each part, and recombining them

A `norm_num` extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a `conv`-tactic, `norm_numI`
- Simplifies expressions in $\mathbb{C}$ by splitting to real and imaginary parts, running `norm_num` on each part, and recombining them

```
example : (1 + 3.5 + I) * (1 + I) = 7 / 2 +
    11 / 2 * I := by norm_num1
example : (3 + 4 * I)⁻¹ * (3 + 4 * I) = 1 :=
    by norm_num1
example : 1 + I ≠ 0 := by norm_num1
example : 1 + I ≠ 1 + 2 * I := by norm_num1
example : im ((1 + 3 * I)⁻¹) = -0.3 := by
    norm_num1
example : 10 * re ((1 + 3 * I)⁻¹) = 1 := by
    norm_num1
example : (conj (3 + 4 * I) : ℂ) * (3 + 4 *
    I) = 25 := by norm_num1
```

Metaprogramming

A `norm_num` extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a `conv`-tactic, `norm_numI`
- Simplifies expressions in $\mathbb{C}$ by splitting to real and imaginary parts, running `norm_num` on each part, and recombining them

```
example : (1 + 3.5 + I) * (1 + I) = 7 / 2 +
    11 / 2 * I := by norm_num1
example : (3 + 4 * I)⁻¹ * (3 + 4 * I) = 1 :=
    by norm_num1
example : 1 + I ≠ 0 := by norm_num1
example : 1 + I ≠ 1 + 2 * I := by norm_num1
example : im ((1 + 3 * I)⁻¹) = -0.3 := by
    norm_num1
example : 10 * re ((1 + 3 * I)⁻¹) = 1 := by
    norm_num1
example : (conj (3 + 4 * I) : ℂ) * (3 + 4 *
    I) = 25 := by norm_num1
```

Proving `Tendsto` given atomic limits

Metaprogramming

A norm_num extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a conv-tactic, norm_numI
- Simplifies expressions in $\mathbb{C}$ by splitting to real and imaginary parts, running norm_num on each part, and recombining them

```
example : (1 + 3.5 + I) * (1 + I) = 7 / 2 +
    11 / 2 * I := by norm_num1
example : (3 + 4 * I)⁻¹ * (3 + 4 * I) = 1 :=
    by norm_num1
example : 1 + I ≠ 0 := by norm_num1
example : 1 + I ≠ 1 + 2 * I := by norm_num1
example : im ((1 + 3 * I)⁻¹) = -0.3 := by
    norm_num1
example : 10 * re ((1 + 3 * I)⁻¹) = 1 := by
    norm_num1
example : (conj (3 + 4 * I) : ℂ) * (3 + 4 *
    I) = 25 := by norm_num1
```

Proving Tendsto given atomic limits

- Developed mainly by Cameron Freer with inputs from Seewoo Lee and Bhavik Mehta

Metaprogramming

A `norm_num` extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a `conv`-tactic, `norm_numI`
- Simplifies expressions in $\mathbb{C}$ by splitting to real and imaginary parts, running `norm_num` on each part, and recombining them

```
example : (1 + 3.5 + I) * (1 + I) = 7 / 2 +
    11 / 2 * I := by norm_num1
example : (3 + 4 * I)⁻¹ * (3 + 4 * I) = 1 :=
    by norm_num1
example : 1 + I ≠ 0 := by norm_num1
example : 1 + I ≠ 1 + 2 * I := by norm_num1
example : im ((1 + 3 * I)⁻¹) = -0.3 := by
    norm_num1
example : 10 * re ((1 + 3 * I)⁻¹) = 1 := by
    norm_num1
example : (conj (3 + 4 * I) : ℂ) * (3 + 4 *
    I) = 25 := by norm_num1
```

Proving `Tendsto` given atomic limits

- Developed mainly by Cameron Freer with inputs from Seewoo Lee and Bhavik Mehta
- Proves `Tendsto (fun z => expr(f₁ z, ..., fₙ z)) l (nhds c)` where `Tendsto fᵢ l (nhds aᵢ)` are known and the expression is continuous

A norm_num extension for $\mathbb{C}$

- Developed mainly by Edison Xie with inputs from Heather Macbeth, Bhavik Mehta, SH
- Initially a `conv`-tactic, `norm_numI`
- Simplifies expressions in $\mathbb{C}$ by splitting to real and imaginary parts, running `norm_num` on each part, and recombining them

```
example : (1 + 3.5 + I) * (1 + I) = 7 / 2 +
    11 / 2 * I := by norm_num1
example : (3 + 4 * I)⁻¹ * (3 + 4 * I) = 1 :=
    by norm_num1
example : 1 + I ≠ 0 := by norm_num1
example : 1 + I ≠ 1 + 2 * I := by norm_num1
example : im ((1 + 3 * I)⁻¹) = -0.3 := by
    norm_num1
example : 10 * re ((1 + 3 * I)⁻¹) = 1 := by
    norm_num1
example : (conj (3 + 4 * I) : ℂ) * (3 + 4 *
    I) = 25 := by norm_num1
```

Proving Tendsto given atomic limits

- Developed mainly by Cameron Freer with inputs from Seewoo Lee and Bhavik Mehta
- Proves `Tendsto (fun z => expr(f₁ z, ..., fₙ z)) l (nhds c)` where `Tendsto fᵢ l (nhds aᵢ)` are known and the expression is continuous

```
-- Two atoms, polynomial
example (h₁ : Tendsto f atTop (nhds 0)) (h₂ :
    Tendsto g atTop (nhds 1)) :
    Tendsto (fun z => f z ^ 2 + f z * g z + g
    z ^ 2) atTop (nhds 1) := by tendsto_cont

-- Unused hypotheses don't interfere
example (h₁ : Tendsto f atTop (nhds 0)) (h₂ :
    Tendsto g atTop (nhds 1))
    (unrelated : Tendsto f atBot (nhds 5)) :
    Tendsto (fun z => f z * g z) atTop (nhds
    0) := by tendsto_cont
```

Formal Verification

Formal Verification

Conversation 19 | Commits 18 | Checks 0 | Files changed 287 | +52,187 -15,457

augustepoiroux commented last month • edited Collaborator

Summary

This PR completes the Lean formalization of the theorem that the optimal sphere packing in `ℝ⁸` is the `E₈` lattice packing.
The branch contains the final theorem together with all remaining dependencies, and is checked by the Lean kernel (sorry-free).

Context

The code in this branch was generated by **Gauss** (Math Inc's autoformalization agent) starting from the existing Sphere Packing blueprint and Lean codebase, and then packaged into this PR for review.

Notable changes

- Completes the remaining formalization steps needed for `E₈` optimality in `ℝ⁸`
- Project-wide cleanup / proof golf (including existing files)
- Fixes a sign inconsistency in Prop. 7 (`b(t)`), with the downstream definition of `g` in Thm. 4 updated accordingly
- Migrates to the new module system
- Bumps Lean to `v4.28.0`

Review notes

- We made a best effort to avoid overwriting parts of the codebase that changed independently on `main`. As such, some results may be duplicated.
- ~~The following blueprint-linked Lean statements were commented out in this branch: `ModularForm.dimension_level_one`, `dim_gen_cong_levels`, D_qexp_tsum_pnat, serre_D_two_H₃`~~ -> they have been added back in `38876bc` #diff-ebf1c68f284079a6364607bb7113a4b517ffb358424f42f02a128c225340f25d

🎉 10 🚀 3

Reviewers
- dwrensha
- b-mehta
- seewoo5
- thefundamentaltheor3m
- CBirkbeck
- pitmonticone
- viazovska

At least 1 approving review is required to merge this pull request.

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Finding our Footing, Part 2

Is the project finished?

Is the project finished?

It’s worth drawing the following distinction.

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Is the project finished?

It’s worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

Is the project finished?

It’s worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

To me, 'formalisation' means strictly more: it involves organising it into an extensible, reusable artefact that communicates insight.

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

To me, 'formalisation' means strictly more: it involves organising it into an extensible, reusable artefact that communicates insight.

The best example is `Mathlib`: things are highly canonicalised, with new improvements every day.

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

To me, 'formalisation' means strictly more: it involves organising it into an extensible, reusable artefact that communicates insight.

The best example is `Mathlib`: things are highly canonicalised, with new improvements every day.

What does the sphere packing project truly seek to do?

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

To me, 'formalisation' means strictly more: it involves organising it into an extensible, reusable artefact that communicates insight.

The best example is `Mathlib`: things are highly canonicalised, with new improvements every day.

What does the sphere packing project truly seek to do?

We're much more interested in formalisation than formal verification.

Is the project finished?

It's worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

To me, 'formalisation' means strictly more: it involves organising it into an extensible, reusable artefact that communicates insight.

The best example is `Mathlib`: things are highly canonicalised, with new improvements every day.

What does the sphere packing project truly seek to do?

We're much more interested in formalisation than formal verification.

Is it finished?

Is the project finished?

It’s worth drawing the following distinction.

Formal Verification

Formal verification is the checking of a formal proof by a trusted verifier.

When a proof is formally verified, we can be (fairly) confident it is correct.

Formalisation

To me, ‘formalisation’ means strictly more: it involves organising it into an extensible, reusable artefact that communicates insight.

The best example is `Mathlib`: things are highly canonicalised, with new improvements every day.

What does the sphere packing project truly seek to do?

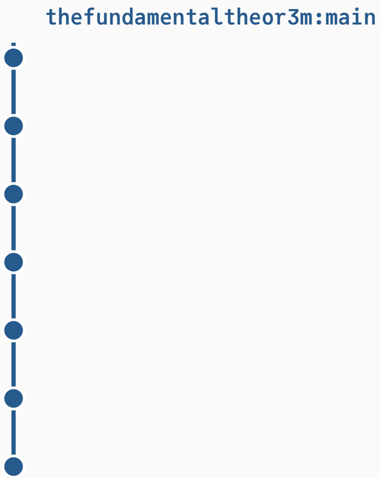
We’re much more interested in formalisation than formal verification.

Is it finished?

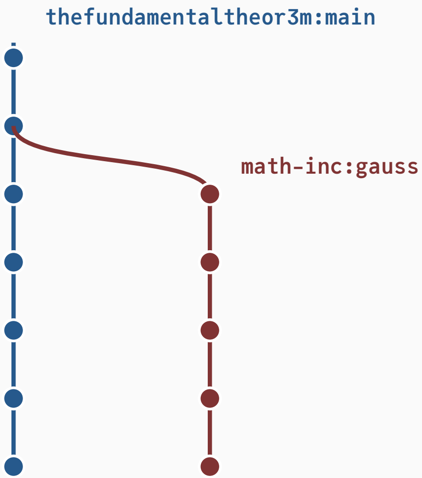
NO!

`main` as the Source of Truth

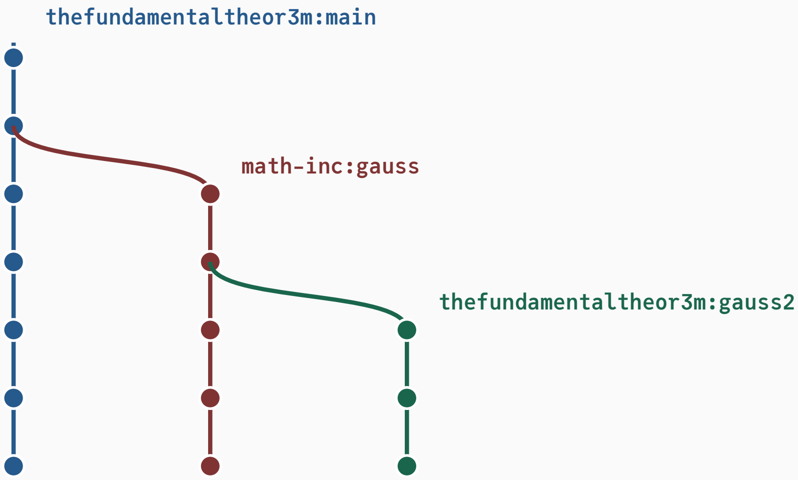
`main` as the Source of Truth



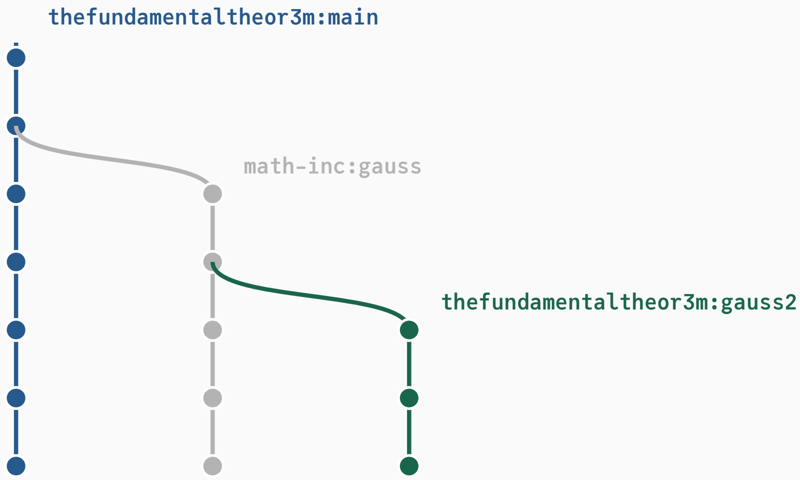
`main` as the Source of Truth



`main` as the Source of Truth

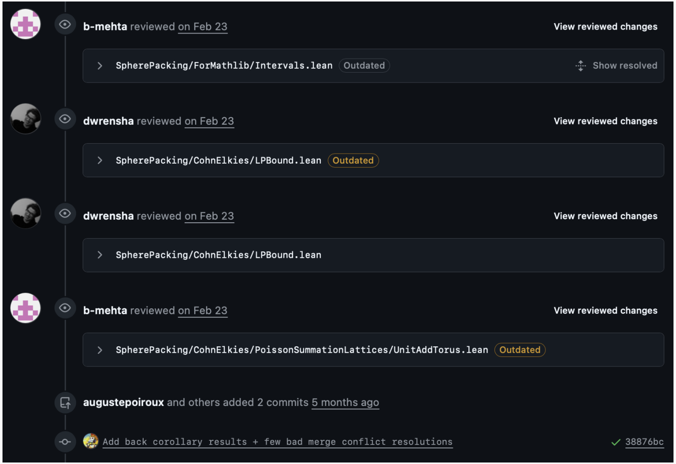


`main` as the Source of Truth



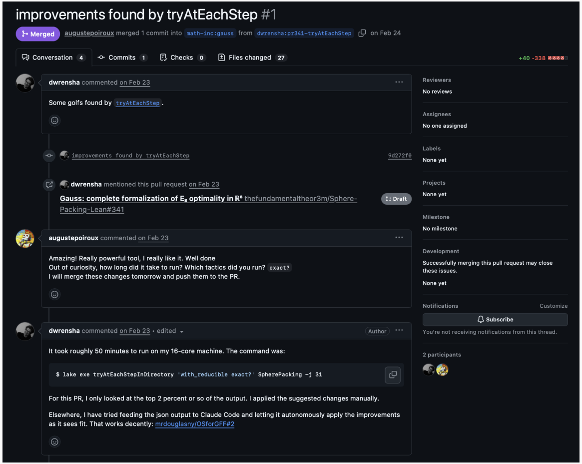
Initial Reviews

Initial Reviews



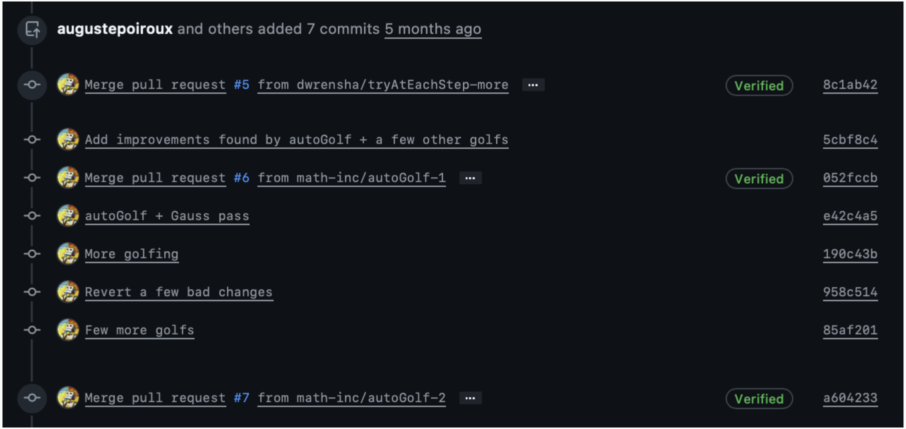
David Renshaw's `TryAtEachStep`

David Renshaw’s `TryAtEachStep`

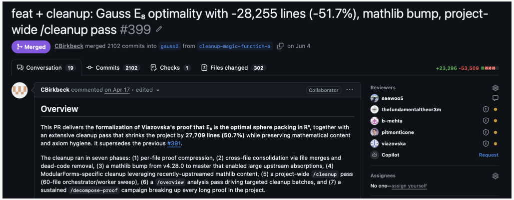


Math, Inc's `AutoGolf`

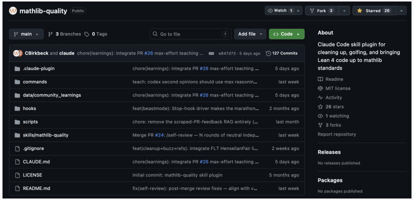
Math, Inc's AutoGolf



The Many Claudes of Christopher Birkbeck

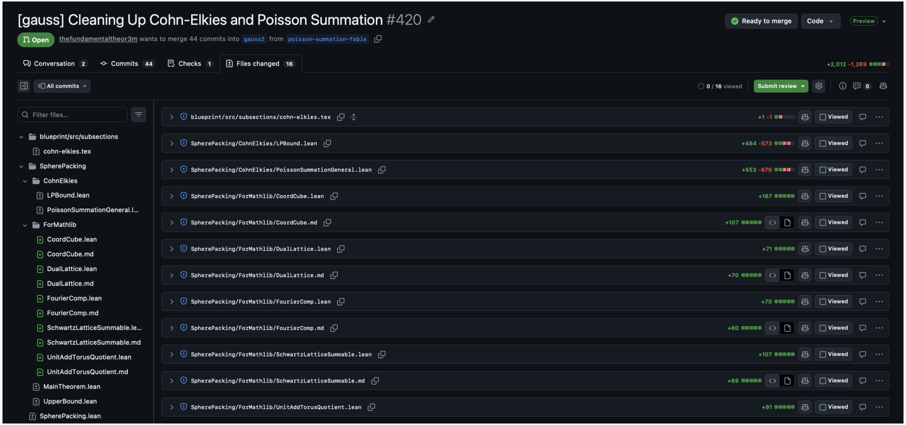


(a) The pull requests



(b) The `mathlib-quality` repository

Targeted Claudeing



Is it really working?

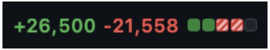
Is it really working?

At some point, the effectiveness of AI-powered cleanup begins to plateau.

Is it really working?

At some point, the effectiveness of AI-powered cleanup begins to plateau.

The diff is currently better:



Is it really working?

At some point, the effectiveness of AI-powered cleanup begins to plateau.

The diff is currently better:

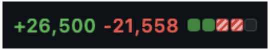
+26,500 -21,558

But lines of code isn't everything. The code quality still needs improvement.

Is it really working?

At some point, the effectiveness of AI-powered cleanup begins to plateau.

The diff is currently better:



But lines of code isn't everything. The code quality still needs improvement.

In the meantime, progress on `main` has become uncomfortably slow: 45 PRs merged between 23 February 2026 and today as opposed to 150 merged between 1 September 2025 and 31 January 2026.

Looking Ahead

The Virtues of Formalisation

In “Mathematics and the Formal Turn” [1], Avigad effectively presents 18 positives of the advent of formal methods.

The Virtues of Formalisation

In “Mathematics and the Formal Turn” [1], Avigad effectively presents 18 positives of the advent of formal methods.

The first four were both my greatest motivators: **verifying theorems**, **correcting mistakes** (in my understanding), **gaining insight**, and **building libraries**.

The Virtues of Formalisation

In “Mathematics and the Formal Turn” [1], Avigad effectively presents 18 positives of the advent of formal methods.

The first four were both my greatest motivators: **verifying theorems**, **correcting mistakes** (in my understanding), **gaining insight**, and **building libraries**.

The fifth (**searching for definitions and tools**) and fourteenth (**improving access**) greatly benefitted me in my journey.

The Virtues of Formalisation

In "Mathematics and the Formal Turn" [1], Avigad effectively presents 18 positives of the advent of formal methods.

The first four were both my greatest motivators: **verifying theorems**, **correcting mistakes** (in my understanding), **gaining insight**, and **building libraries**.

The fifth (**searching for definitions and tools**) and fourteenth (**improving access**) greatly benefitted me in my journey.

The eighth (**engineering concepts**), ninth (**communicating**), and tenth (**collaborating**) have been key outcomes.

What does this mean for formalisation and our project?

What does this mean for formalisation and our project?

The virtues tell us that formalisation is a human endeavour.

What does this mean for formalisation and our project?

The virtues tell us that formalisation is a human endeavour.

A good formalisation needs human architects.

What does this mean for formalisation and our project?

The virtues tell us that formalisation is a human endeavour.

A good formalisation needs human architects.

We need to continue to be architects of sphere packing.

What does this mean for formalisation and our project?

The virtues tell us that formalisation is a human endeavour.

A good formalisation needs human architects.

We need to continue to be architects of sphere packing.

Maybe we need to go back to `main` and continue doing things by hand.

Acknowledgements

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)

 and

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)
   and 
- Jeremy Avigad, Kevin Buzzard, Simon DeDeo, Leo de Moura, Emily Riehl, the Mathlib maintainers

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)

and

- Jeremy Avigad, Kevin Buzzard, Simon DeDeo, Leo de Moura, Emily Riehl, the Mathlib maintainers
- Hoskinson Center for Formal Mathematics

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)
 and
- Jeremy Avigad, Kevin Buzzard, Simon DeDeo, Leo de Moura, Emily Riehl, the Mathlib maintainers
- Hoskinson Center for Formal Mathematics
- Institute for Computer-Aided Reasoning in Mathematics

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)
   and
- Jeremy Avigad, Kevin Buzzard, Simon DeDeo, Leo de Moura, Emily Riehl, the Mathlib maintainers
- Hoskinson Center for Formal Mathematics
- Institute for Computer-Aided Reasoning in Mathematics
- G-Research

Acknowledgements

- Chris Birkbeck, Seewoo Lee, Gareth Ma, Bhavik Mehta, Maryna Viazovska
- Matthew Cushman, Cameron Freer, Auguste Poiroux, Aayush Rajasekaran, David Renshaw, Tito Sacchi
- David Loeffler, Pietro Monticone, Utensil Song
- All our contributors! (Human and non-human alike)
   and
- Jeremy Avigad, Kevin Buzzard, Simon DeDeo, Leo de Moura, Emily Riehl, the Mathlib maintainers
- Hoskinson Center for Formal Mathematics
- Institute for Computer-Aided Reasoning in Mathematics
- G-Research

Thank you very much!

Questions?

[1] J. Avigad.
Mathematics and the formal turn.
arXiv preprint arXiv:2311.00007, 2023.

[2] H. Cohn and N. Elkies.
New Upper Bounds on Sphere Packings I.
Annals of Mathematics, 157(2):689–714, 2003.

[3] T. F. Görbe.
Exceptionally Beautiful Symmetries.
`https://tamasgorbe.com/symmetry`.

[4] M. S. Viazovska.
The sphere packing problem in dimension 8.
Annals of Mathematics, 185(3):991–1015, 2017.