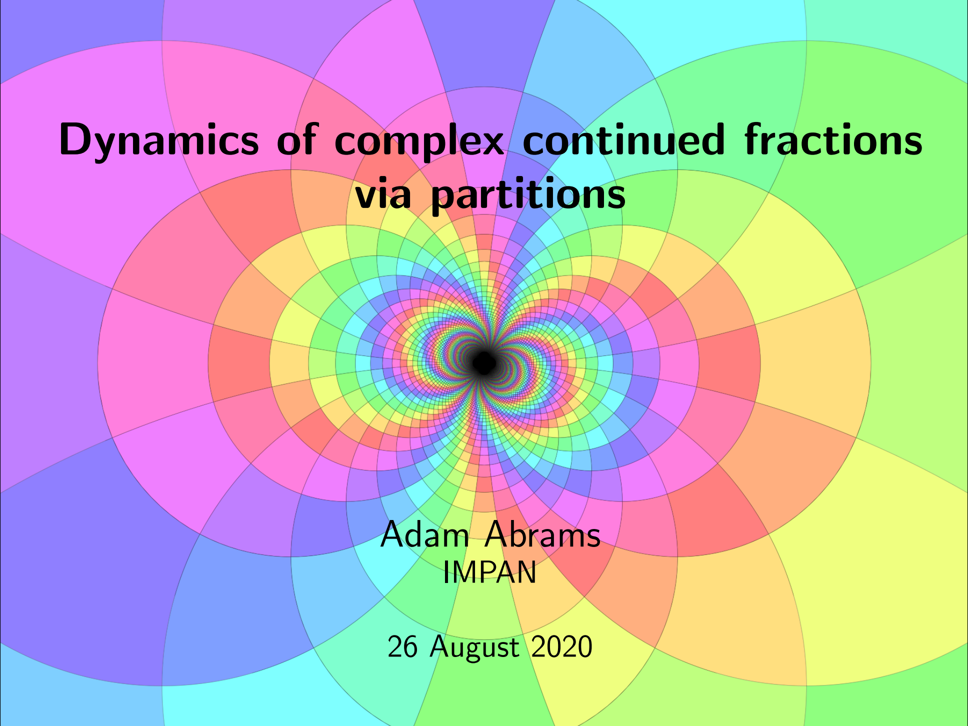


Dynamics of complex continued fractions via partitions

Adam Abrams
IMPAN

26 August 2020

Dynamics of complex continued fractions via partitions

Adam Abrams
IMPAN

26 August 2020

On geometric complexity of Julia sets II

Continued fractions

- Minus continued fractions are of the form

$$a_0 - \cfrac{1}{a_1 - \cfrac{1}{a_2 - \cfrac{1}{\ddots}}}$$

  with
  - $a_i \in \mathbb{Z}$ for Real CF.
  - $a_i \in \mathbb{Z}[\mathbf{i}] = \mathbb{Z} + \mathbb{Z}\mathbf{i}$ for Complex CF.

- There are multiple algorithms to generate a digit sequence $\{a_i\}$ for a given $x \in \mathbb{R}$ or $z \in \mathbb{C}$.

References/timeline

1887 A. Hurwitz. "Über die entwicklung complexer grössen in kettenbruc̈he." *Acta Mathematica*.

1902 J. Hurwitz. "Über die reduction der binären quadratischen formen mit complexen coefficienten und variabeln." *Acta Mathematica*.

1985 Tanaka. "A complex continued fraction transformation and its ergodic properties." *Tokyo Journal of Mathematics*.

2013 Dani, Nogueira. "Continued fractions for complex numbers and values of binary quadratic forms." *Trans. American Math. Society*.

2019 Ei, Ito, Nakada, Natsui. "On the construction of the natural extension of the Hurwitz complex continued fraction map." *Monatshefte für Mathematik*.

2020 Abrams. "Finite partitions for several complex continued fraction algorithms." *Experimental Mathematics*.

Choice functions

A choice function[1] is a function $\lfloor\cdot\rceil : \mathbb{C} \to \mathbb{Z}[\mathbf{i}]$ such that z and $\lfloor z\rceil$ are at most 1 apart.

[1] Dani and Nogueira, 2013.

Choice functions

A choice function[1] is a function $\lfloor \cdot \rceil : \mathbb{C} \to \mathbb{Z}[\mathbf{i}]$ such that z and $\lfloor z \rceil$ are at most 1 apart.

- *Non*-example: floor $\lfloor z \rfloor = \lfloor \operatorname{Re} z \rfloor + \lfloor \operatorname{Im} z \rfloor \mathbf{i}$ does not work since $\lfloor 0.9 + 0.9\mathbf{i} \rfloor = 0$ is distance 1.273 from $0.9 + 0.9\mathbf{i}$.

[1] Dani and Nogueira, 2013.

Choice functions

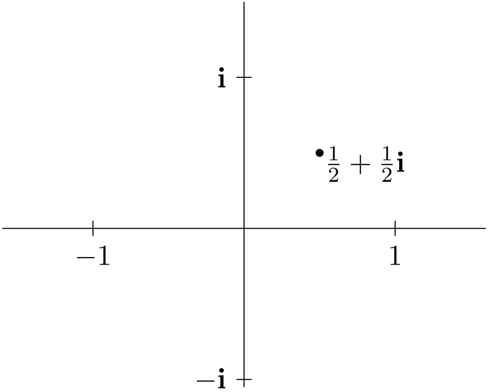
A choice function[1] is a function $\lfloor\cdot\rceil : \mathbb{C} \to \mathbb{Z}[\mathbf{i}]$ such that z and $\lfloor z\rceil$ are at most 1 apart.

- Simplest example: choose the closest Gaussian integer to z. This is called the "nearest integer" or "Hurwitz"[2] algorithm.

Choice functions

A choice function[1] is a function $\lfloor \cdot \rceil : \mathbb{C} \to \mathbb{Z}[\mathbf{i}]$ such that z and $\lfloor z \rceil$ are at most 1 apart.

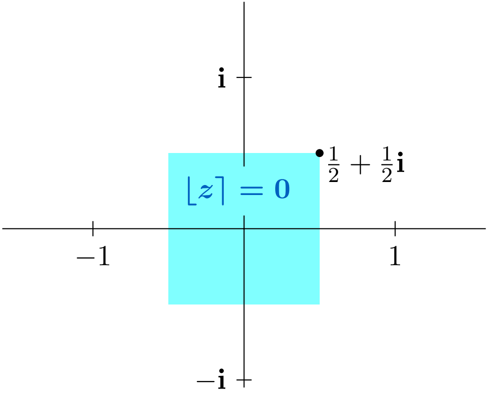
- Simplest example: choose the closest Gaussian integer to z. This is called the "nearest integer" or "Hurwitz"[2] algorithm.



Choice functions

A choice function[1] is a function $\lfloor\cdot\rceil : \mathbb{C} \to \mathbb{Z}[\mathbf{i}]$ such that z and $\lfloor z\rceil$ are at most 1 apart.

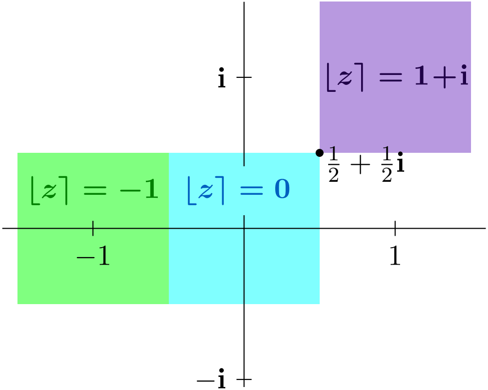
- Simplest example: choose the closest Gaussian integer to z. This is called the "nearest integer" or "Hurwitz"[2] algorithm.



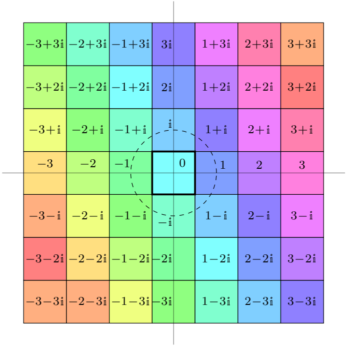
Choice functions

A choice function[1] is a function $\lfloor\cdot\rceil : \mathbb{C} \to \mathbb{Z}[\mathbf{i}]$ such that z and $\lfloor z\rceil$ are at most 1 apart.

- Simplest example: choose the closest Gaussian integer to z. This is called the "nearest integer" or "Hurwitz"[2] algorithm.



Examples

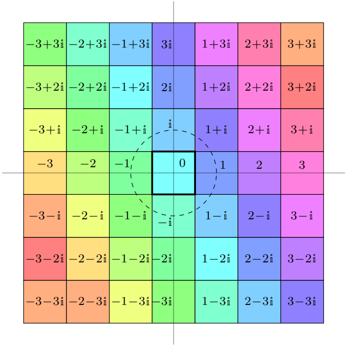


Nearest integer or Hurwitz algorithm[2]

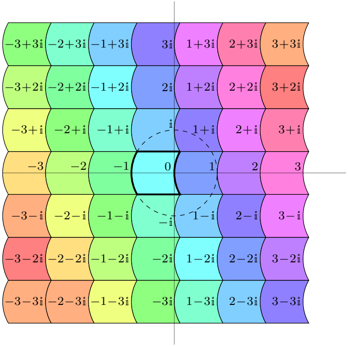
[1] Dani and Nogueira, 2013.

[2] Adolf Hurwitz, 1887.

Examples



Nearest integer or Hurwitz algorithm[2]

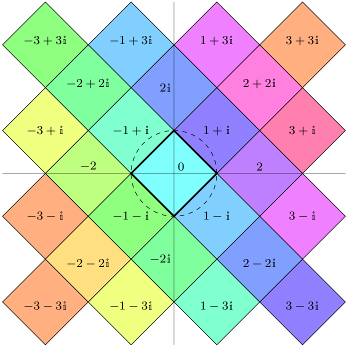


Shifted Hurwitz[1] algorithm

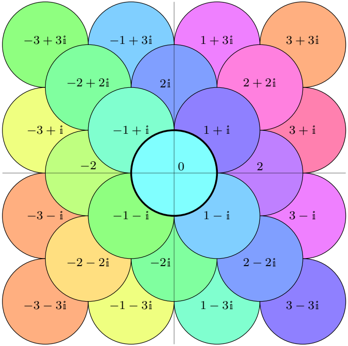
[1] Dani and Nogueira, 2013.

[2] Adolf Hurwitz, 1887.

Examples



Nearest even algorithm[2]



Disk algorithm[3]

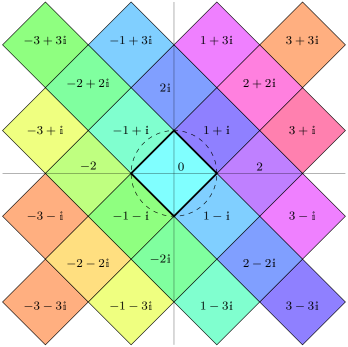
- For both, digits are in $\{\, x + y\mathbf{i} \in \mathbb{Z}[\mathbf{i}] \; : \; x + y \text{ even} \,\}$.

---

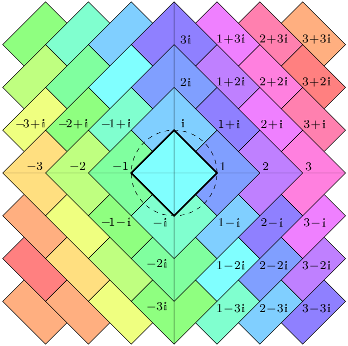
[2] Julius Hurwitz, 1902.

[3] Shigeru Tanaka, 1985.

Examples



Nearest even algorithm[3]



Diamond algorithm

Gauss maps

The fundamental set for an algorithm is

$$K = \overline{\{\, z - \lfloor z \rceil \;:\; z \in \mathbb{C} \,\}}.$$

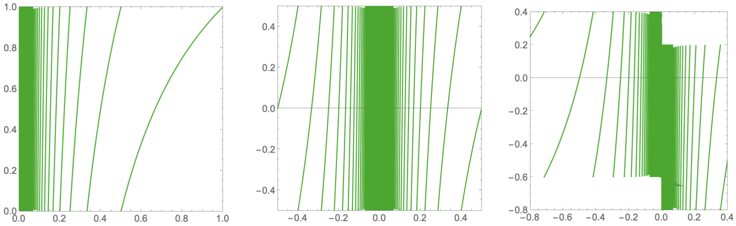
- Example: For the nearest integer algorithm, K is the unit square centered at the origin.
- Note $K \subseteq \overline{B(0,1)}$ by the definition of a choice function.

The Gauss map $g : K \to K$ is given by $g(0) = 0$ and

$$g(z) = \frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil.$$

Gauss maps

Real-valued Gauss maps:



$$\frac{-1}{x} - \left\lfloor \frac{-1}{x} \right\rfloor \qquad \frac{-1}{x} - \left\lfloor \frac{-1}{x} \right\rceil_{\text{Hurwitz}} \qquad \frac{-1}{x} - \left\lfloor \frac{-1}{x} \right\rceil_{\text{Zagier}}$$

$$K = [0,1] \qquad K = [\tfrac{-1}{2}, \tfrac{1}{2}] \qquad K = [\tfrac{-4}{5}, \tfrac{2}{5}]$$

Gauss maps

Theorem (Dani-Nogueira for + c.f.)

Let $\lfloor \cdot \rceil$ be such that $K \subset B(0,1)$. For any $z \in \mathbb{C} \setminus \mathbb{Q}[\mathbf{i}]$, set

$$a_0 = \lfloor z \rceil \quad \text{and} \quad a_n = \left\lfloor \frac{-1}{g^{n-1}(z-a_0)} \right\rceil \ \forall\, n \geq 1.$$

Then the value of

$$a_0 - \cfrac{1}{a_1 - \cfrac{1}{a_2 - \cfrac{1}{\ddots - \frac{1}{a_n}}}}$$

approaches z as $n \to \infty$.

Gauss maps

Theorem (Dani-Nogueira for + c.f.)

Let $\lfloor\cdot\rceil$ be such that $K \subset B(0,1)$. For any $z \in \mathbb{C} \setminus \mathbb{Q}[\mathbf{i}]$, set

$$a_0 = \lfloor z \rceil \quad \text{and} \quad a_n = \left\lfloor \frac{-1}{g^{n-1}(z-a_0)} \right\rceil \ \forall\ n \geq 1.$$

Then the value of

$$a_0 - \cfrac{1}{a_1 - \cfrac{1}{a_2 - \cfrac{1}{\ddots - \frac{1}{a_n}}}}$$

approaches z as $n \to \infty$.

Sometimes it is easier to use the map $w \mapsto \frac{-1}{w - \lfloor w \rceil}$ instead of $g : z \mapsto \frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil$.

Gauss maps

Theorem (Dani-Nogueira for + c.f.)

Let $\lfloor \cdot \rceil$ be such that $K \subset B(0,1)$. For any $z \in \mathbb{C} \setminus \mathbb{Q}[\mathbf{i}]$, set

$$w_0 = z, \qquad a_n = \lfloor w_n \rceil, \qquad w_{n+1} = \frac{-1}{w_n - a_n}.$$

Then the value of

$$a_0 - \cfrac{1}{a_1 - \cfrac{1}{a_2 - \cfrac{1}{\ddots - \cfrac{1}{a_n}}}}$$

approaches z as $n \to \infty$.

Sometimes it is easier to use the map $w \mapsto \frac{-1}{w - \lfloor w \rceil}$ instead of $g : z \mapsto \frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil$.

Gauss maps

The natural extension of the Gauss map

$$g(z) = \frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil$$

is the function

$$F(u,v) = \left(\frac{-1}{u} - a, \frac{-1}{v} - a \right) \quad \text{where } a = \left\lfloor \frac{-1}{v} \right\rceil$$

or, after change of variables $(z,w) = (v, -1/u)$,

$$G(z,w) = \left(\frac{-1}{z} - a, \frac{-1}{w-a} \right) \quad \text{where } a = \left\lfloor \frac{-1}{z} \right\rceil.$$

The real-valued version of G has an attracting set with a "finite rectangular structure."

Gauss maps

The natural extension of the Gauss map

$$g(z) = \frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil$$

is the function

$$F(u, v) = \left(\frac{-1}{u} - a, \frac{-1}{v} - a \right) \quad \text{where } a = \left\lfloor \frac{-1}{v} \right\rceil$$

or, after change of variables $(z, w) = (v, -1/u)$,

$$G(z, w) = \left(\frac{-1}{z} - a, \frac{-1}{w - a} \right) \quad \text{where } a = \left\lfloor \frac{-1}{z} \right\rceil .$$

The real-valued version of G has an attracting set with a "finite rectangular structure."

Modular and Fuchsian results[4]

Real (a,b)-continued fractions:

- Family of algorithms with parameters $a, b \in \mathbb{R}$.
- Natural extension is $G(x,y) = (\frac{-1}{x} - n, \frac{-1}{y-n})$ where n is the "(a,b)-generalized integer part" of x.

Co-compact Fuchsian setting:

- Instead of $\mathrm{PSL}(2,\mathbb{Z})$ we use $\Gamma = \langle T_1, \dots, T_m \rangle$.
- Natural extension is $F(x,y) = (T_i(x), T_i(y))$ where i depends only on x.

[4] Katok and Ugarcovici, 2010 and 2017.

Modular and Fuchsian results[4]

Real (a, b)-continued fractions:

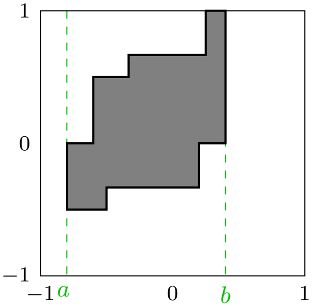
- Family of algorithms with parameters $a, b \in \mathbb{R}$.
- Natural extension is $G(x, y) = (\frac{-1}{x} - n, \frac{-1}{y-n})$ where n is the "(a, b)-generalized integer part" of x.

Co-compact Fuchsian setting:

- Instead of $\mathrm{PSL}(2, \mathbb{Z})$ we use $\Gamma = \langle T_1, \ldots, T_m \rangle$.
- Natural extension is $F(x, y) = (T_i(x), T_i(y))$ where i depends only on x.

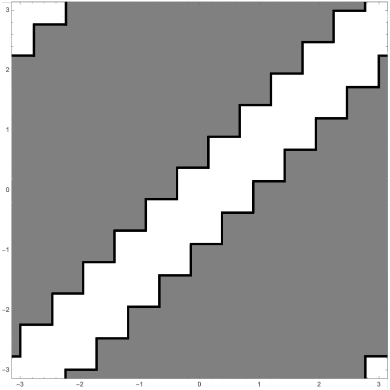
[4] Katok and Ugarcovici, 2010 and 2017.

Modular and Fuchsian results[4]



Real C.F.

$\Omega_{a,b} \subset \mathbb{R} \times \mathbb{R}$



Fuchsian

$\Omega_{\bar{A}} \subset S^1 \times S^1$

[4] Katok and Ugarcovici, 2010 and 2017.

Philosophy

For classical ($X = \mathbb{R}$) and Fuchsian (with $X = S^1$), we have

1. One-variable $f : X \to X$ is not injective.

2. Two-variable $F : X \times X \to X \times X$ of the form
$$F(x, y) = \big(\rho_x(x), \rho_x(y)\big)$$
is not injective on $X \times X$, but

3. restricting F to its global attractor $\Omega \subset X \times X$ does give an a. e. bijective function.

4. The set $\Omega \subset X \times X$ has finite rectangular structure.

5. This is a result of the cycle property of $f : X \to X$.

The goal is to recreate this for complex c. f.

Philosophy

For classical ($X = \mathbb{R}$) and Fuchsian (with $X = S^1$), we have

1. One-variable $f : X \to X$ is not injective.

2. Two-variable $F : X \times X \to X \times X$ of the form
$$F(x, y) = \big(\rho_x(x), \rho_x(y)\big)$$
is not injective on $X \times X$, but

3. restricting F to its global attractor $\Omega \subset X \times X$ does give an a. e. bijective function.

4. The set $\Omega \subset X \times X$ has finite rectangular structure.

5. This is a result of the cycle property of $f : X \to X$.

The goal is to recreate this for complex c. f.

Philosophy

For classical ($X = \mathbb{R}$) and Fuchsian (with $X = S^1$), we have

1. One-variable $f : X \to X$ is not injective.
2. Two-variable $F : X \times X \to X \times X$ of the form
$$F(x, y) = \big(\rho_x(x), \rho_x(y)\big)$$
is not injective on $X \times X$, but
3. restricting F to its global attractor $\Omega \subset X \times X$ does give an a. e. bijective function.
4. The set $\Omega \subset X \times X$ has finite rectangular structure.
5. This is a result of the cycle property of $f : X \to X$.

The goal is to recreate this for complex c. f.

Finite product structure

What is a good analogue in $\bar{\mathbb{C}} \times \bar{\mathbb{C}}$ for the 2-real-dimensional finite rectangular structure?

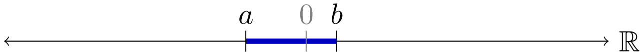
Definition

A set $\Omega \subset \bar{\mathbb{C}} \times \bar{\mathbb{C}}$ has finite product structure if there exist $N \in \mathbb{N}$ and sets $K_1, \ldots, K_N \subset \bar{\mathbb{C}}$ and $L_1, \ldots, L_N \subset \bar{\mathbb{C}}$ each connected on the Reimann sphere such that

$$\Omega = \bigcup_{i=1}^{N} K_i \times L_i.$$

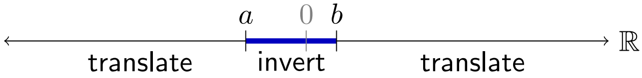
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



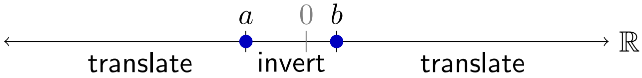
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f:\mathbb{R}\to\mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



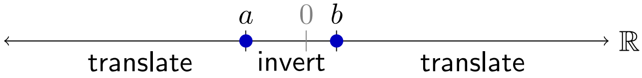
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f:\mathbb{R}\to\mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



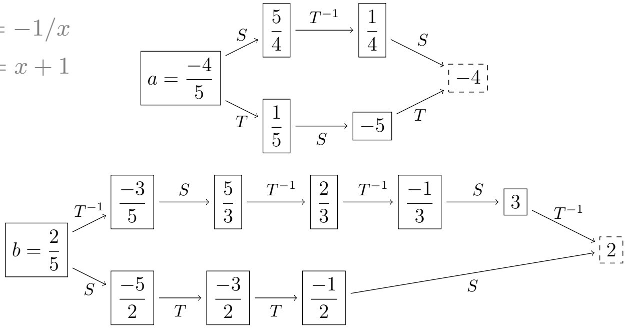
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f:\mathbb{R}\to\mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



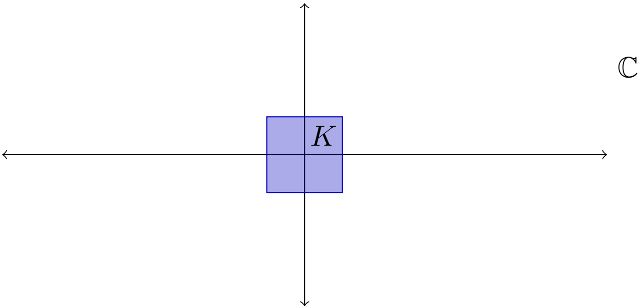
$S(x) = -1/x$

$T(x) = x+1$



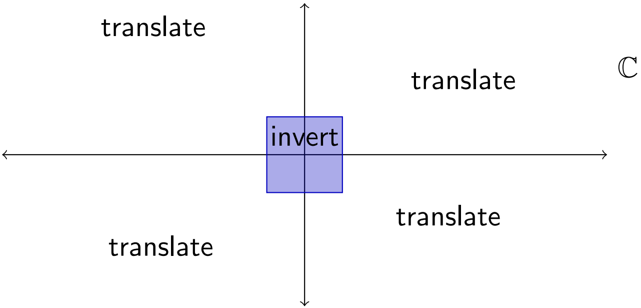
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



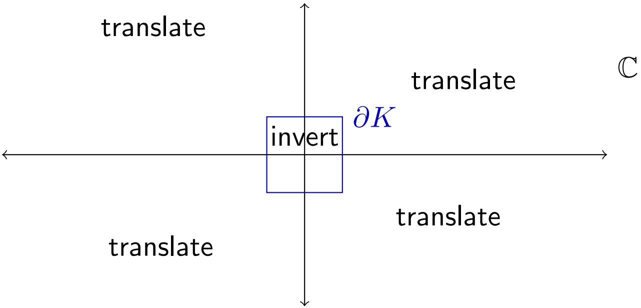
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



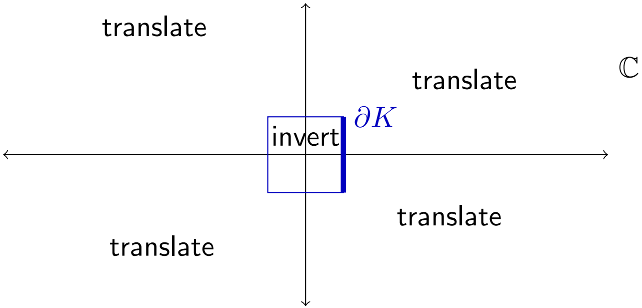
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



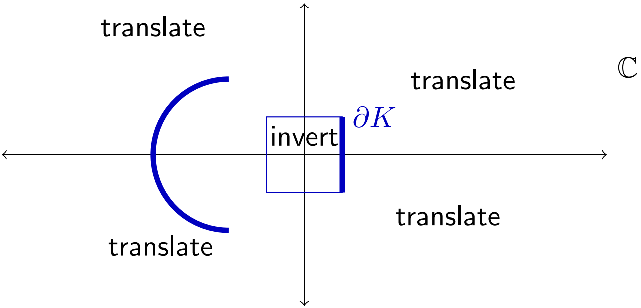
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



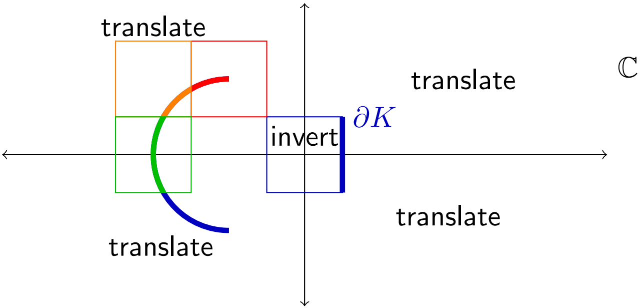
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



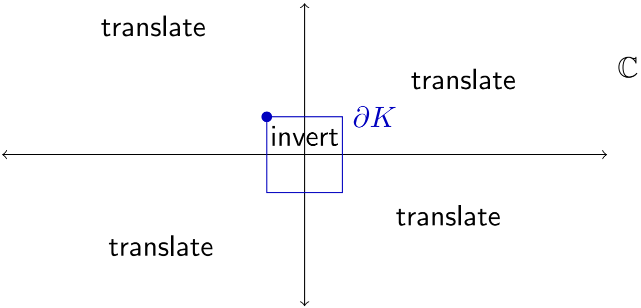
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



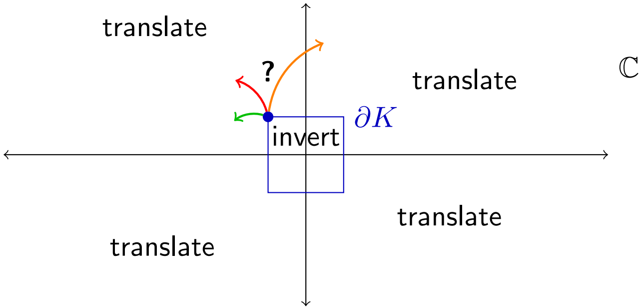
Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f : \mathbb{R} \to \mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



Cycle property

Cycle property: the orbits of $\frac{-1}{a}$ and $a+1$ under $f:\mathbb{R}\to\mathbb{R}$ intersect, and the orbits of $\frac{-1}{b}$ and $b-1$ also intersect.



Finite building property

The following replaces the cycle property:

Definitions

- Let $\mathcal{C}$ be a collection of sets. A set is called buildable from $\mathcal{C}$ if it is equal to some union of elements of $\mathcal{C}$.
- A continued fraction algorithm with Gauss map $g : K \to K$ satisfies the finite building property if there exists a finite partition $\mathcal{P} = \{K_1, \ldots, K_N\}$ with $N > 1$ such that each $g(K_i)$ is buildable from $\mathcal{P}$.

"Partition" here means the elements of $\mathcal{P}$ have disjoint interiors.

Finite building property

We want each $g(K_i)$ to be some $\bigcup_j K_j$.

Define the maps

$$S(z) := -1/z$$
$$T^a(z) := z + a \qquad \text{for any } a \in \mathbb{Z}[\mathbf{i}]$$

Finite building property

We want each $g(K_i)$ to be some $\bigcup_j K_j$.

Define the maps

$$S(z) := -1/z$$
$$T^a(z) := z + a \qquad \text{for any } a \in \mathbb{Z}[\mathbf{i}]$$

- For an individual point,

$$g(z) = T^{-a}Sz \quad \text{where} \quad a = \lfloor Sz \rceil,$$

but for $X \subset \mathbb{C}$, $g(X)$ might NOT be of the form $T^{-a}SX$ because $\lfloor \cdot \rceil$ will not generally be constant on SX.

Finite building property

We want each $g(K_i)$ to be some $\bigcup_j K_j$.

Define the maps

$$S(z) := -1/z$$
$$T^a(z) := z + a \qquad \text{for any } a \in \mathbb{Z}[\mathbf{i}]$$

- For an individual point,
  $$g(z) = T^{-a} S z \quad \text{where} \quad a = \lfloor Sz \rceil,$$
  but for $X \subset \mathbb{C}$, $g(X)$ might NOT be of the form $T^{-a}SX$ because $\lfloor \cdot \rceil$ will not generally be constant on SX.
- In practice, we need to decompose K_i further into sets for which g acts the same.

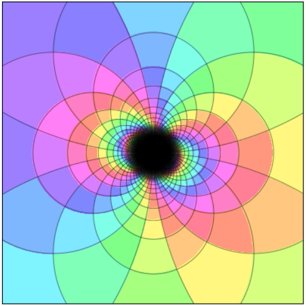
Cells

Some new notation will be useful.

- For $a \in \mathbb{Z}[\mathbf{i}]$, we have the cell

$$\langle a \rangle := \{\, z \in K \;:\; \lfloor -1/z \rceil = a \,\}\,.$$

Note that $g\big|_{\langle a \rangle}$ is exactly $T^{-a}S$.



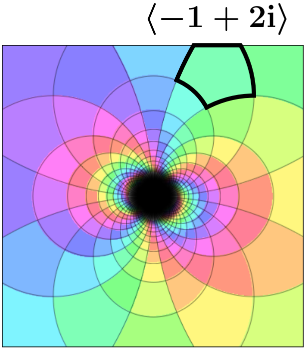
Cells

Some new notation will be useful.

- For $a \in \mathbb{Z}[\mathbf{i}]$, we have the cell

$$\langle a \rangle := \{\, z \in K \,:\, \lfloor -1/z \rceil = a \,\}.$$

Note that $g\big|_{\langle a \rangle}$ is exactly $T^{-a}S$.



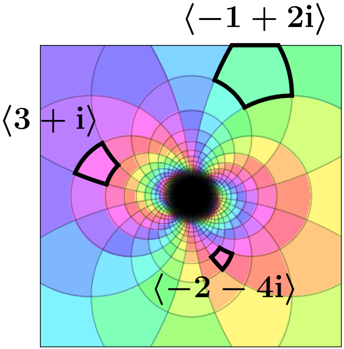
Cells

Some new notation will be useful.

- For $a \in \mathbb{Z}[\mathbf{i}]$, we have the cell

$$\langle a \rangle := \{\, z \in K \ : \ \lfloor -1/z \rceil = a \,\}.$$

Note that $g\big|_{\langle a \rangle}$ is exactly $T^{-a}S$.



Cells

Some new notation will be useful.

- For $a \in \mathbb{Z}[\mathbf{i}]$, we have the cell

$$\langle a \rangle := \{\, z \in K \;:\; \lfloor -1/z \rceil = a \,\}\,.$$

Note that $g\big|_{\langle a \rangle}$ is exactly $T^{-a}S$.

Cells

Some new notation will be useful.

- For $a \in \mathbb{Z}[\mathbf{i}]$, we have the cell

  $$\langle a \rangle := \{ z \in K \, : \, \lfloor -1/z \rceil = a \} .$$

  Note that $g\big|_{\langle a \rangle}$ is exactly $T^{-a}S$.

- Denote $K_{i,a} := K_i \cap \langle a \rangle$.
  Then we have

  $$K_i = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} K_{i,a}$$

  and

  $$g(K_i) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} T^{-a} S \, K_{i,a} .$$

Cells

Some new notation will be useful.

- For $a \in \mathbb{Z}[\mathbf{i}]$, we have the cell

$$\langle a \rangle := \{\, z \in K \ :\ \lfloor -1/z \rceil = a \,\}.$$

Note that $g\big|_{\langle a \rangle}$ is exactly $T^{-a}S$.

- Denote $K_{i,a} := K_i \cap \langle a \rangle$.

Some $K_{i,a}$ will be empty.

Then we have

$$K_i = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} K_{i,a}$$

and

$$g(K_i) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} T^{-a} S K_{i,a}.$$

Cells

Lemma (A.)

If all $g(K_{i,a})$ are buildable from $\{K_1, \ldots, K_N\}$, then the algorithm satisfies the finite building property.

Cells

Lemma (A.)

If all $g(K_{i,a})$ are buildable from $\{K_1, \ldots, K_N\}$, then the algorithm satisfies the finite building property.

Proof. Let $J(i,a) \subset \{1, \ldots, N\}$ be such that $g(K_{i,a}) = \bigcup_{j \in J(i,a)} K_j$.

Then

$$g(\ K_i\ ) = g\left(\bigcup_{a \in \mathbb{Z}[\mathbf{i}]} K_{i,a}\right) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} g(K_{i,a})$$

$$= \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} \left(\bigcup_{j \in J(i,a)} K_j\right) = \bigcup_{j \in J(i)} K_j$$

where $J(i) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} J(i,a)$. Thus all $g(K_i)$ are buildable from $\mathcal{P}$. $\square$

Cells

Lemma (A.)

If all $g(K_{i,a})$ are buildable from $\{K_1, \dots, K_N\}$, then the algorithm satisfies the finite building property.

Proof. Let $J(i,a) \subset \{1, \dots, N\}$ be such that $g(K_{i,a}) = \bigcup_{j \in J(i,a)} K_j$.

Then

$$g(K_i) = g\left(\bigcup_{a \in \mathbb{Z}[\mathbf{i}]} K_{i,a}\right) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} g(K_{i,a})$$

$$= \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} \left(\bigcup_{j \in J(i,a)} K_j \right) = \bigcup_{j \in J(i)} K_j$$

where $J(i) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} J(i,a)$. Thus all $g(K_i)$ are buildable from $\mathcal{P}$. □

Cells

Lemma (A.)

If all $g(K_{i,a})$ are buildable from $\{K_1, \ldots, K_N\}$, then the algorithm satisfies the finite building property.

Proof. Let $J(i,a) \subset \{1, \ldots, N\}$ be such that $g(K_{i,a}) = \bigcup_{j \in J(i,a)} K_j$.

Then

$$g(\ K_i\ ) = g\left(\bigcup_{a \in \mathbb{Z}[\mathbf{i}]} K_{i,a} \right) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} g(K_{i,a})$$

$$= \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} \left(\bigcup_{j \in J(i,a)} K_j \right) = \bigcup_{j \in J(i)} K_j$$

where $J(i) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} J(i,a)$. Thus all $g(K_i)$ are buildable from $\mathcal{P}$. □

Cells

Lemma (A.)

If all $g(K_{i,a})$ are buildable from $\{K_1, \ldots, K_N\}$, then the algorithm satisfies the finite building property.

Proof. Let $J(i,a) \subset \{1, \ldots, N\}$ be such that $g(K_{i,a}) = \bigcup_{j \in J(i,a)} K_j$.

Then

$$g(K_i) = g\left(\bigcup_{a \in \mathbb{Z}[\mathbf{i}]} K_{i,a}\right) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} g(K_{i,a})$$

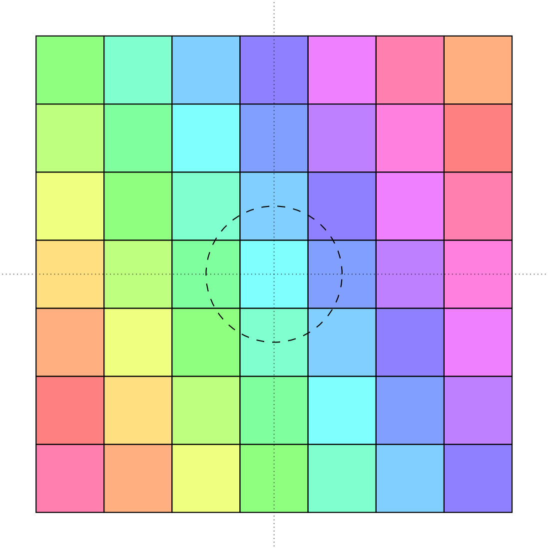
$$= \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} \left(\bigcup_{j \in J(i,a)} K_j \right) = \bigcup_{j \in J(i)} K_j$$

where $J(i) = \bigcup_{a \in \mathbb{Z}[\mathbf{i}]} J(i,a)$. Thus all $g(K_i)$ are buildable from $\mathcal{P}$. □

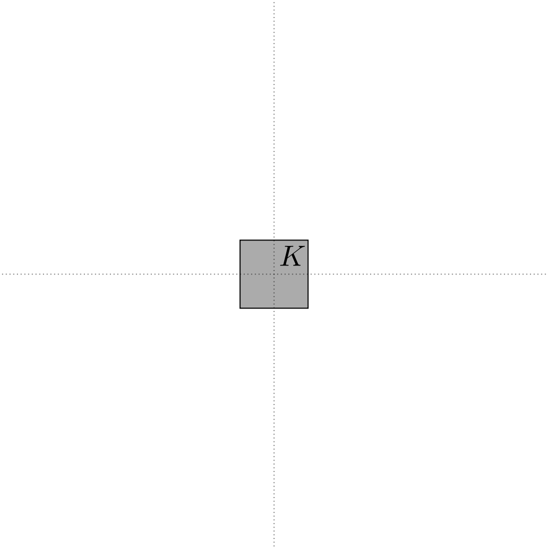
Partitions

- We actually have three partitions of K now.
  1. $\{\langle a\rangle\}$ indexed by $a \in \lfloor\mathbb{C}\rceil \subset \mathbb{Z}[\mathbf{i}]$.
  2. $\{K_i\}$ indexed by $i \in \{1,\ldots,N\}$.
  3. $\{K_{i,a}\}$ indexed by (i,a) for which $K_i \cap \langle a\rangle \neq \emptyset$.
- The partition into $\langle a\rangle$ is based on the algorithm only.
- The partition $\mathcal{P} = \{K_1,\ldots,K_N\}$ is finite.
  How do we construct these sets?

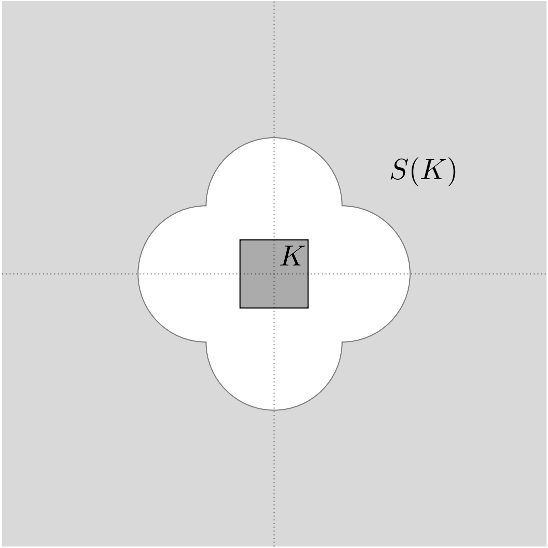
Hurwitz



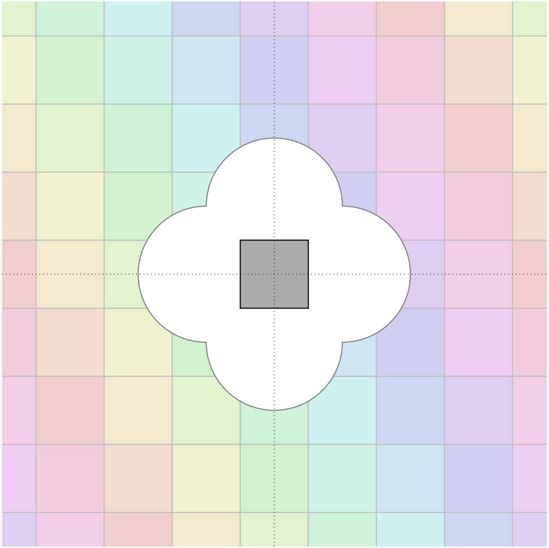
Hurwitz



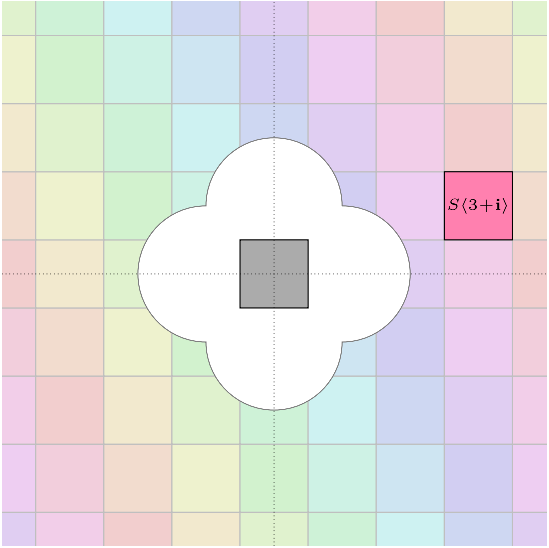
Hurwitz



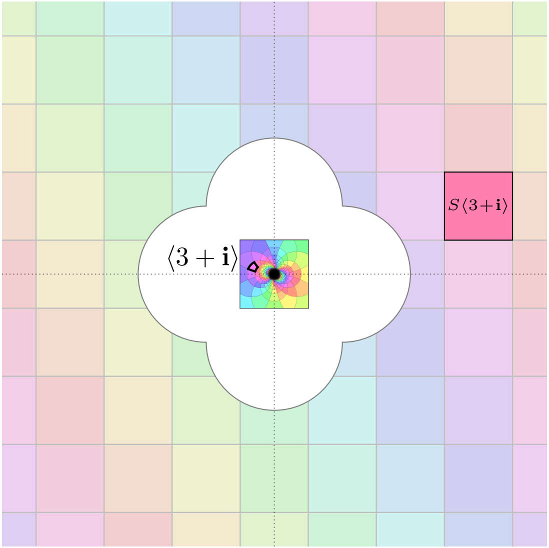
Hurwitz



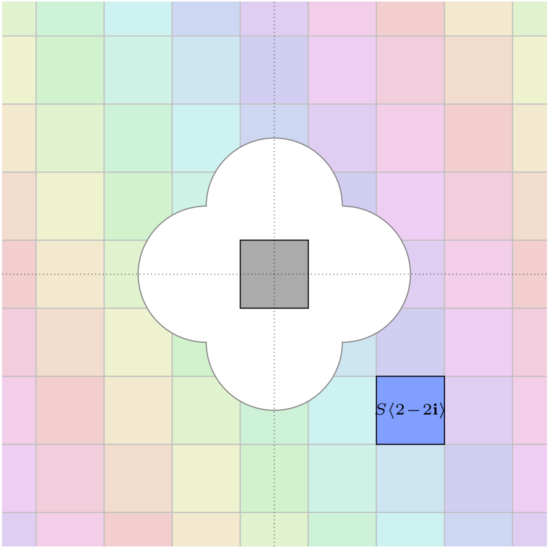
Hurwitz



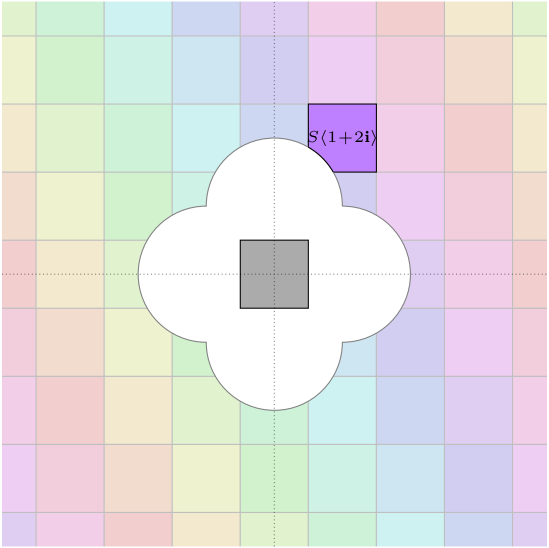
Hurwitz



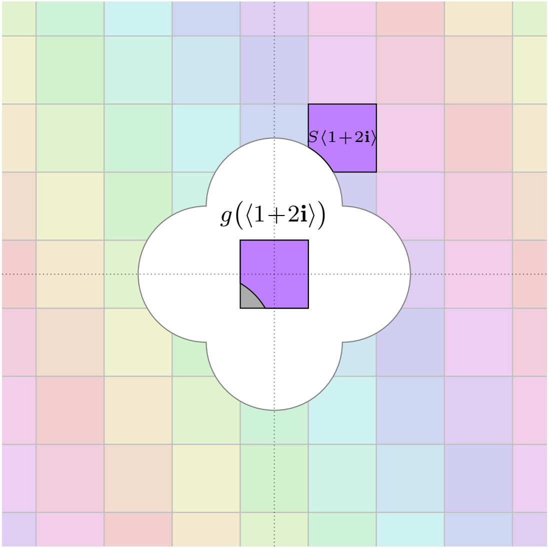
Hurwitz



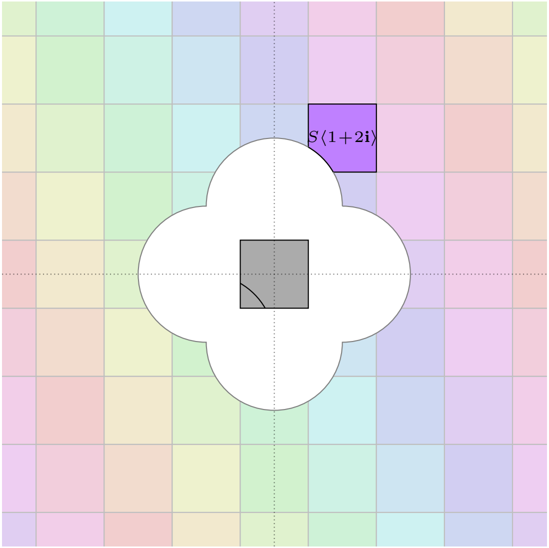
Hurwitz



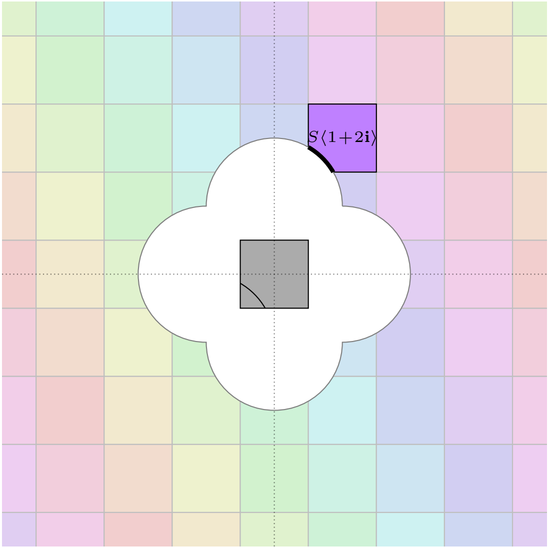
Hurwitz


$S\langle 1+2\mathbf{i}\rangle$
$g\big(\langle 1+2\mathbf{i}\rangle\big)$

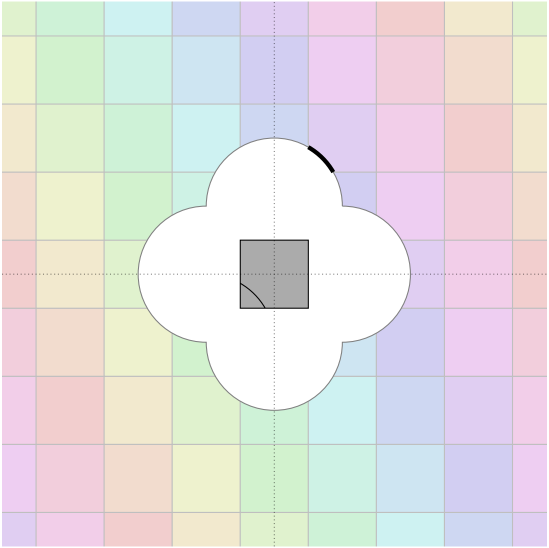
Hurwitz



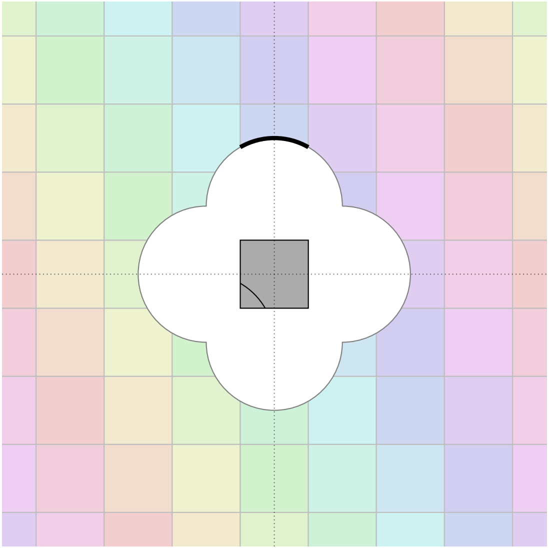
Hurwitz



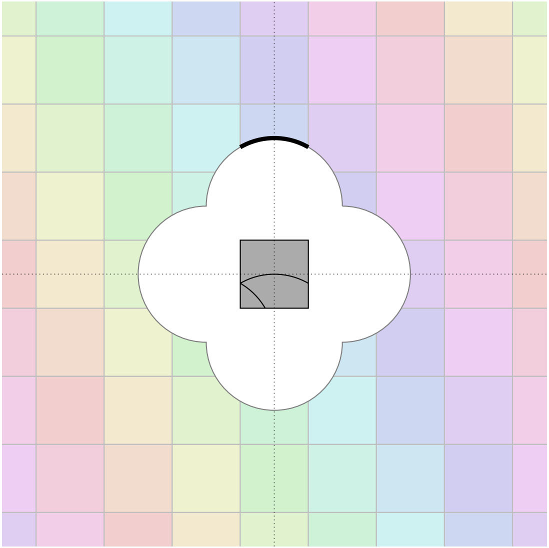
Hurwitz



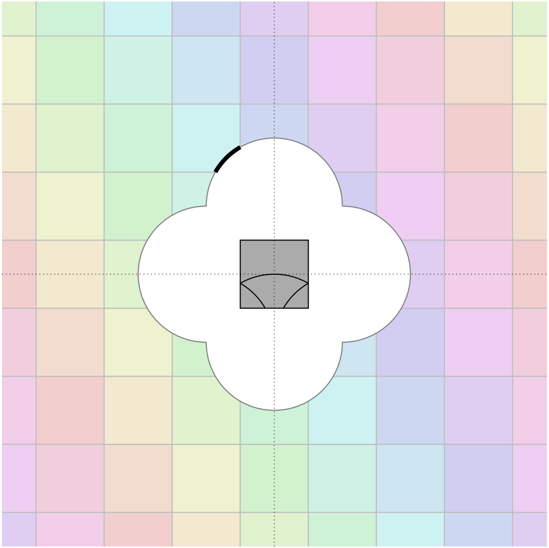
Hurwitz



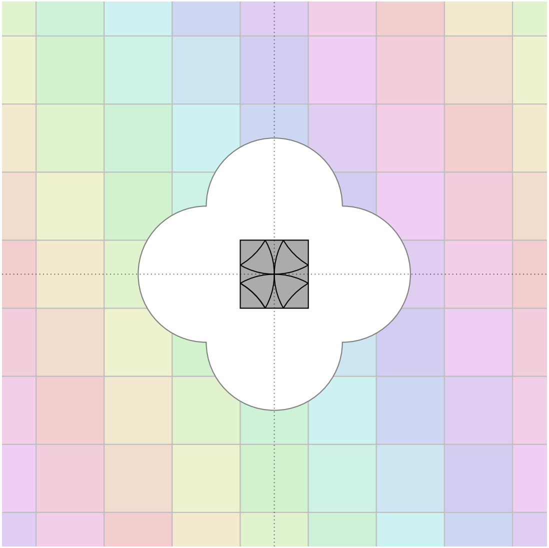
Hurwitz



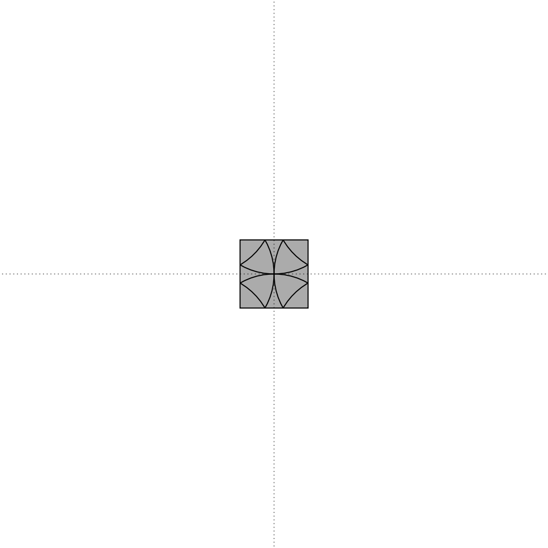
Hurwitz



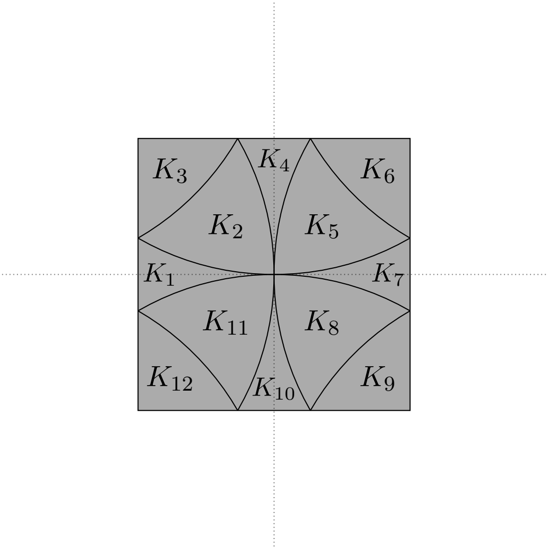
Hurwitz



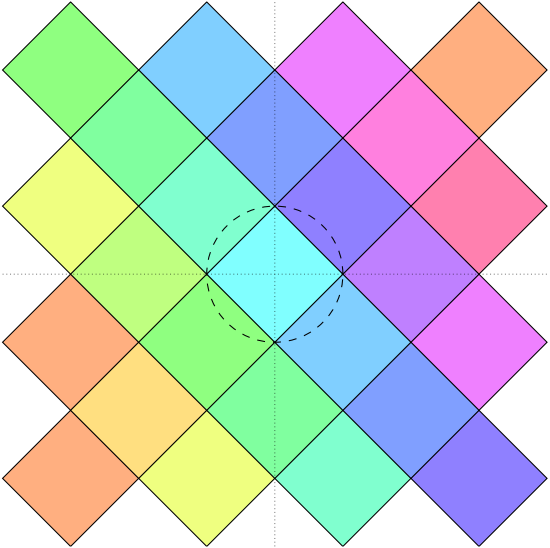
Hurwitz



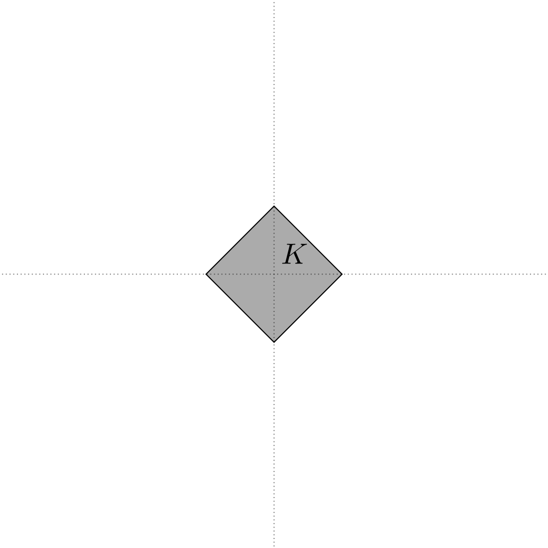
Hurwitz



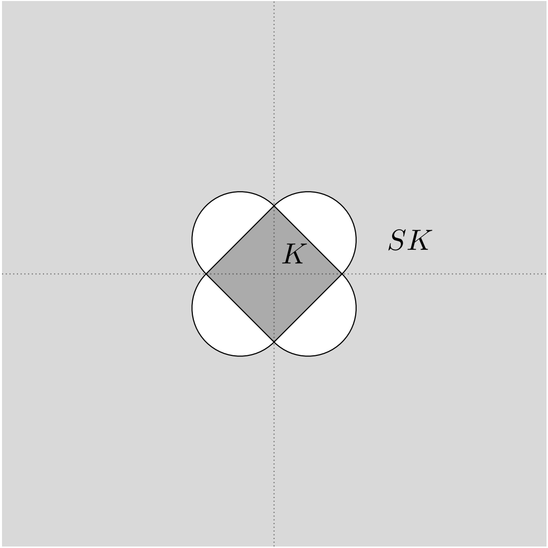
Nearest even



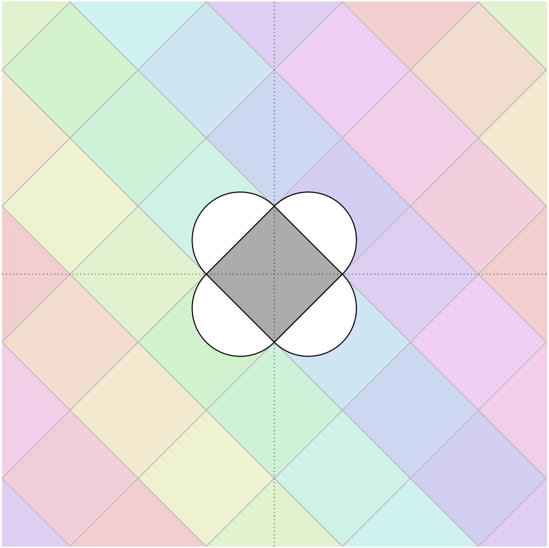
Nearest even



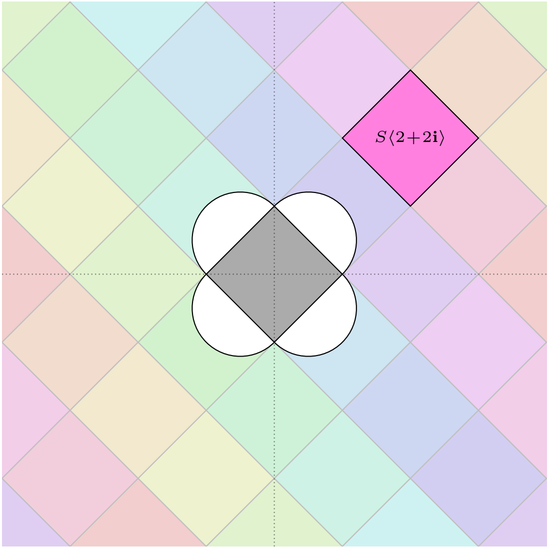
Nearest even



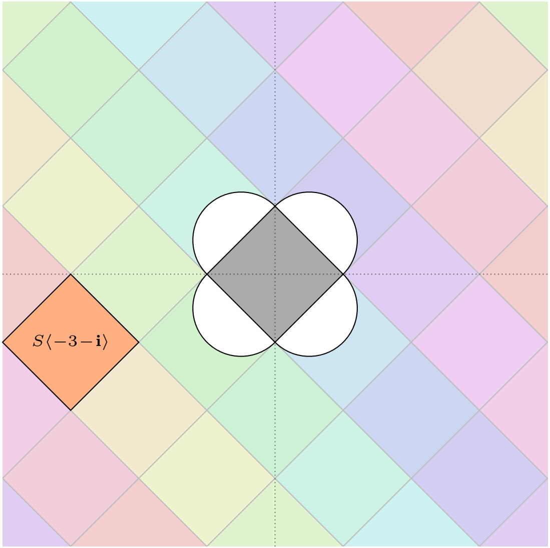
Nearest even



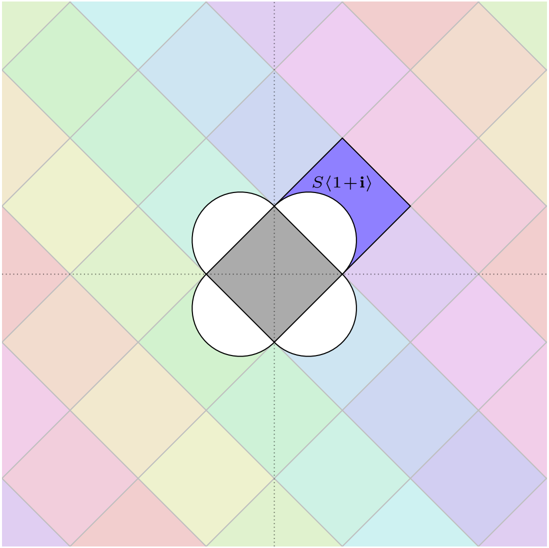
Nearest even



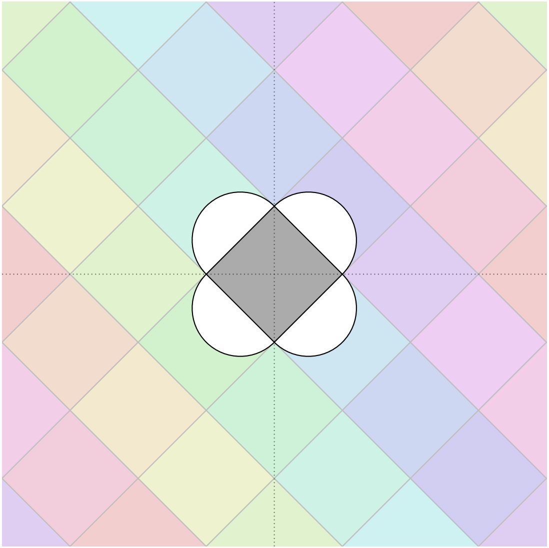
Nearest even



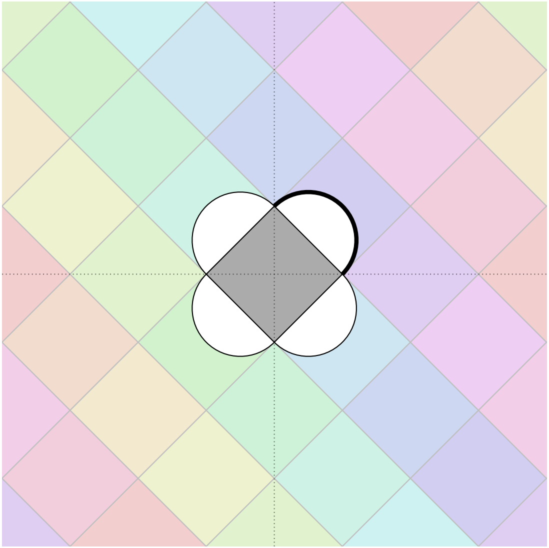
Nearest even



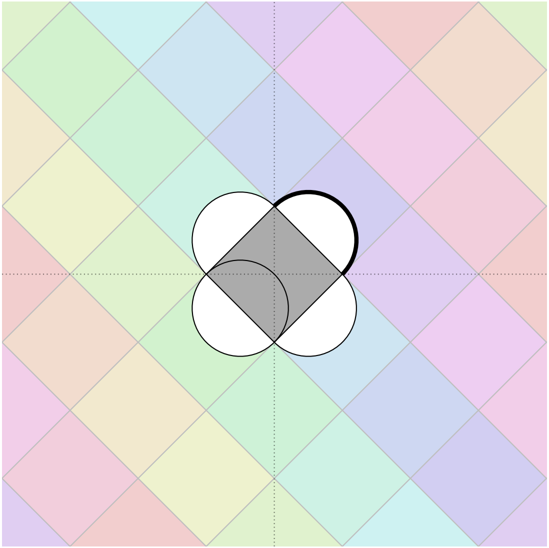
Nearest even



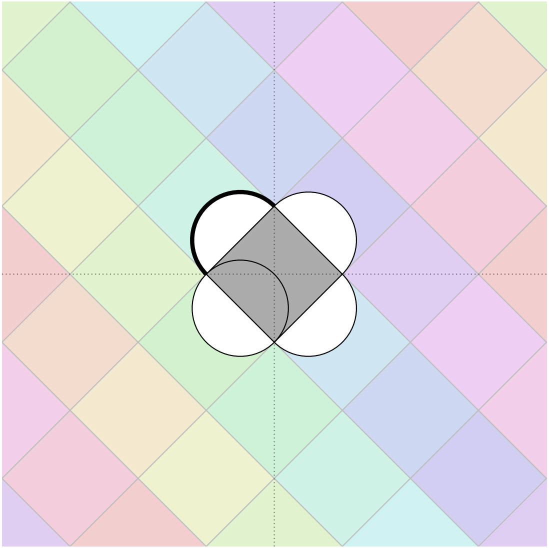
Nearest even



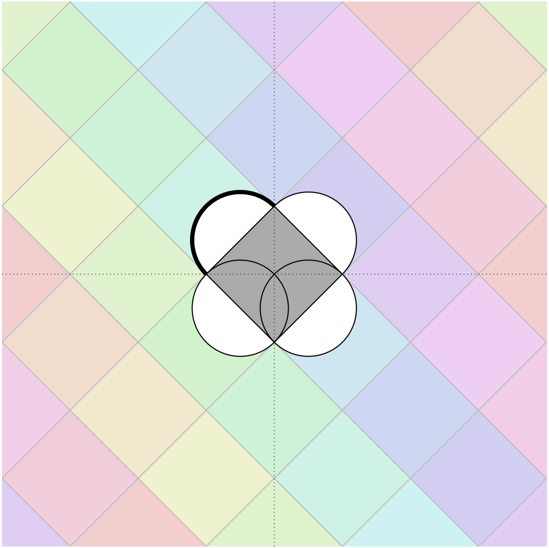
Nearest even



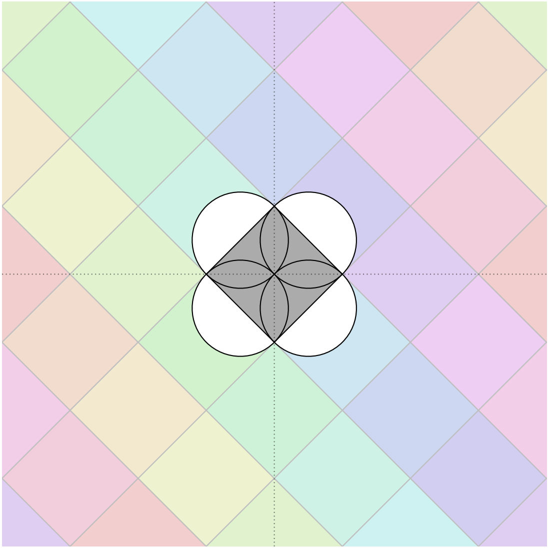
Nearest even



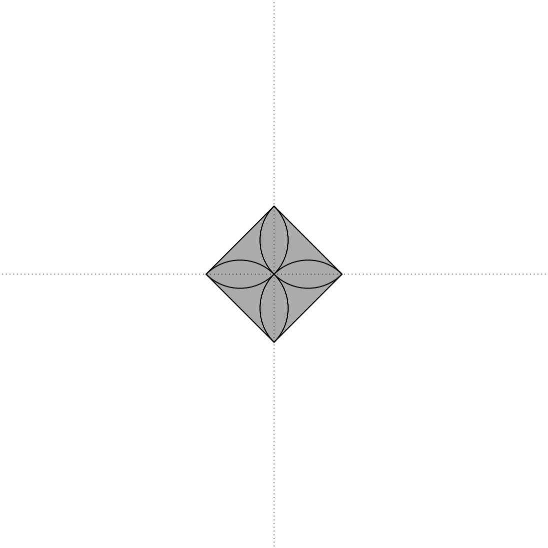
Nearest even



Nearest even



Nearest even



Progress

The partitions $\mathcal{P} = \{K_1, \ldots, K_N\}$ just shown for the nearest even and nearest integer algorithms are constructed to satisfy

$$g(\langle a \rangle) = T^{-a} S \langle a \rangle = \bigcup_{j \in J(a)} K_j \qquad \text{for some } J(a).$$

but the finite building property requires that that all $g(K_i)$ are buildable from $\mathcal{P}$, not that all $g(\langle a \rangle)$ are buildable.

Progress

The partitions $\mathcal{P} = \{K_1, \ldots, K_N\}$ just shown for the nearest even and nearest integer algorithms are constructed to satisfy

$$g(\langle a \rangle) = T^{-a} S \langle a \rangle = \bigcup_{j \in J(a)} K_j \qquad \text{for some } J(a).$$

but the finite building property requires that that all $g(K_i)$ are buildable from $\mathcal{P}$, not that all $g(\langle a \rangle)$ are buildable.

- Recall the lemma: if all $g(K_{i,a})$ are buildable from $\mathcal{P}$, then the finite building property is satisfied.

Sufficient conditions

Lemma (A.)

If each $\langle a \rangle$ *is contained in some* K_i *and each* $S\big(\langle a \rangle\big)$ *can be written as a union of sets of the form* $a + K_j$ *, then the algorithm satisfies the finite building property.*

Sufficient conditions

Lemma (A.)

If each $\langle a \rangle$ is contained in some K_i and each $S\big(\langle a \rangle\big)$ can be written as a union of sets of the form $a + K_j$, then the algorithm satisfies the finite building property.

Proof. Let $J(a)$ be such that $S\langle a \rangle = \bigcup_{j \in J(a)} (a + K_j)$. Then

$$g\big(\langle a \rangle\big) = T^{-a}\ S\langle a \rangle\ = T^{-a} \bigcup_{j \in J(a)} a + K_j$$

$$= \bigcup_{j \in J(a)} T^{-a}(a + K_j) = \bigcup_{j \in J(a)} K_j$$

Each $K_{i,a} = K_i \cap \langle a \rangle$ is either empty or is exactly some $\langle a \rangle$,

so every $g(K_{i,a})$ is buildable from $\mathcal{P}$. □

Sufficient conditions

Lemma (A.)

If each $\langle a \rangle$ is contained in some K_i and each $S(\langle a \rangle)$ can be written as a union of sets of the form $a + K_j$, then the algorithm satisfies the finite building property.

Proof. Let $J(a)$ be such that $S\langle a \rangle = \bigcup_{j \in J(a)} (a + K_j)$. Then

$$g(\langle a \rangle) = T^{-a} S\langle a \rangle = T^{-a} \bigcup_{j \in J(a)} a + K_j$$

$$= \bigcup_{j \in J(a)} T^{-a}(a + K_j) = \bigcup_{j \in J(a)} K_j$$

Each $K_{i,a} = K_i \cap \langle a \rangle$ is either empty or is exactly some $\langle a \rangle$,

so every $g(K_{i,a})$ is buildable from $\mathcal{P}$. □

Sufficient conditions

Lemma (A.)

If each $\langle a \rangle$ is contained in some K_i and each $S(\langle a \rangle)$ can be written as a union of sets of the form $a + K_j$, then the algorithm satisfies the finite building property.

Proof. Let $J(a)$ be such that $S\langle a \rangle = \bigcup_{j \in J(a)} (a + K_j)$. Then

$$g(\langle a \rangle) = T^{-a}\ S\langle a \rangle\ = T^{-a} \bigcup_{j \in J(a)} a + K_j$$

$$= \bigcup_{j \in J(a)} T^{-a}(a + K_j) = \bigcup_{j \in J(a)} K_j$$

Each $K_{i,a} = K_i \cap \langle a \rangle$ is either empty or is exactly some $\langle a \rangle$,

so every $g(K_{i,a})$ is buildable from $\mathcal{P}$. □

Sufficient conditions

Lemma (A.)

If each $\langle a \rangle$ is contained in some K_i and each $S\big(\langle a \rangle\big)$ can be written as a union of sets of the form $a + K_j$, then the algorithm satisfies the finite building property.

Proof. Let $J(a)$ be such that $S\langle a \rangle = \bigcup_{j \in J(a)} (a + K_j)$. Then

$$g\big(\langle a \rangle\big) = T^{-a}\ S\langle a \rangle\ = T^{-a} \bigcup_{j \in J(a)} a + K_j$$

$$= \bigcup_{j \in J(a)} T^{-a}(a + K_j) = \bigcup_{j \in J(a)} K_j$$

Each $K_{i,a} = K_i \cap \langle a \rangle$ is either empty or is exactly some $\langle a \rangle$,

so every $g(K_{i,a})$ is buildable from $\mathcal{P}$. □

Sufficient conditions

Lemma (A.)

If each $\langle a \rangle$ is contained in some K_i and each $S\big(\langle a \rangle\big)$ can be written as a union of sets of the form $a + K_j$, then the algorithm satisfies the finite building property.

Proof. Let $J(a)$ be such that $S\langle a \rangle = \bigcup_{j \in J(a)} (a + K_j)$. Then

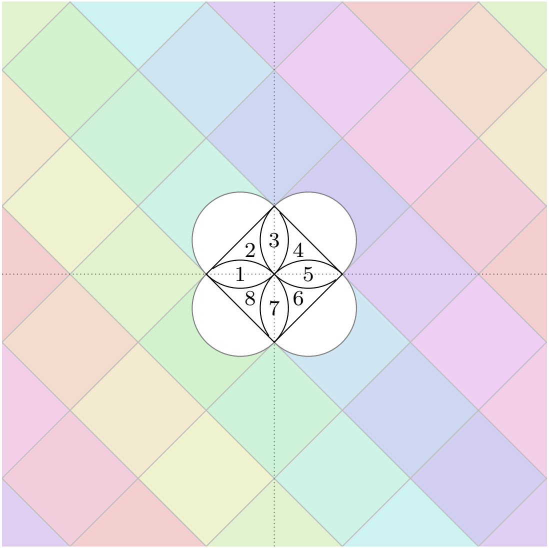
$$g\big(\langle a \rangle\big) = T^{-a}\ S\langle a \rangle\ = T^{-a} \bigcup_{j \in J(a)} a + K_j$$

$$= \bigcup_{j \in J(a)} T^{-a}(a + K_j) = \bigcup_{j \in J(a)} K_j$$

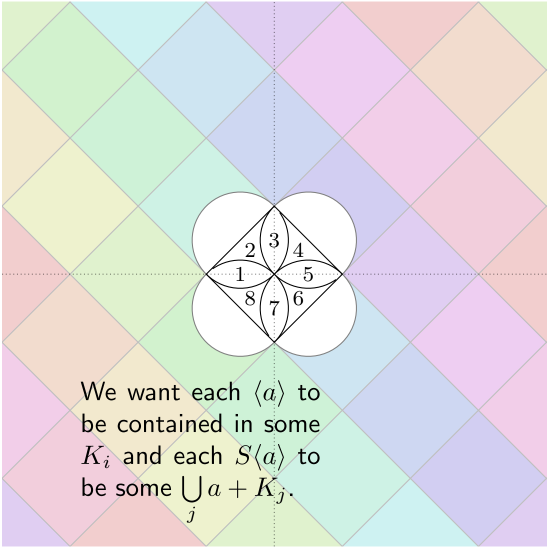
Each $K_{i,a} = K_i \cap \langle a \rangle$ is either empty or is exactly some $\langle a \rangle$,

so every $g(K_{i,a})$ is buildable from $\mathcal{P}$. □

Nearest even

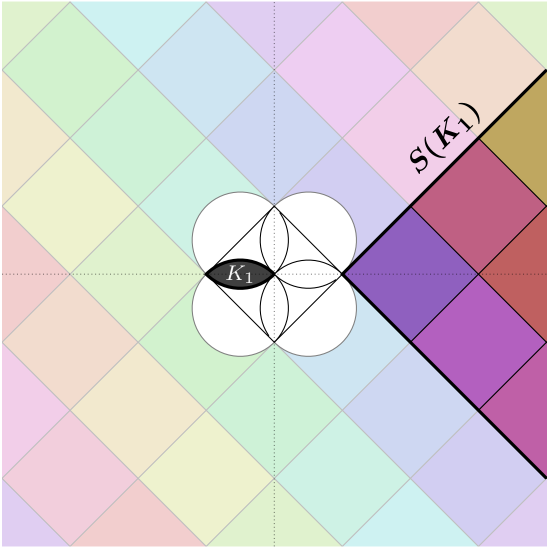


Nearest even

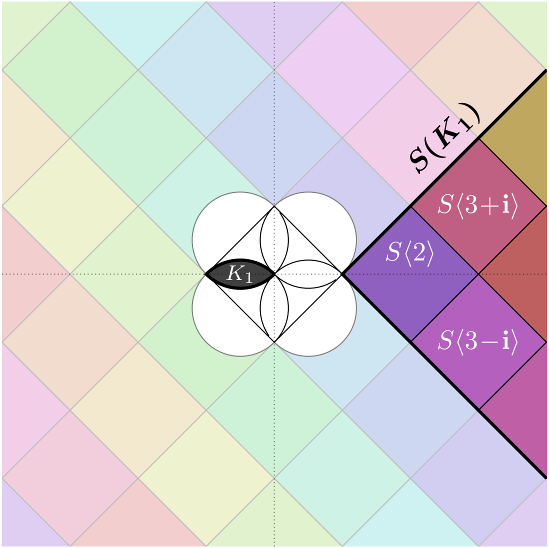


We want each $\langle a \rangle$ to be contained in some K_i and each $S\langle a \rangle$ to be some $\bigcup_j a + K_j$.

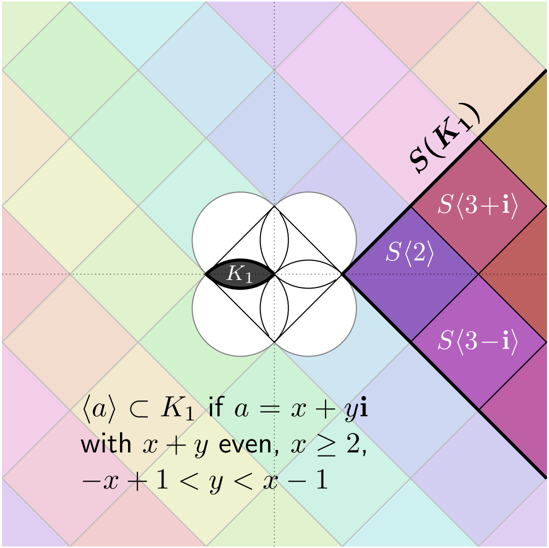
Nearest even



Nearest even

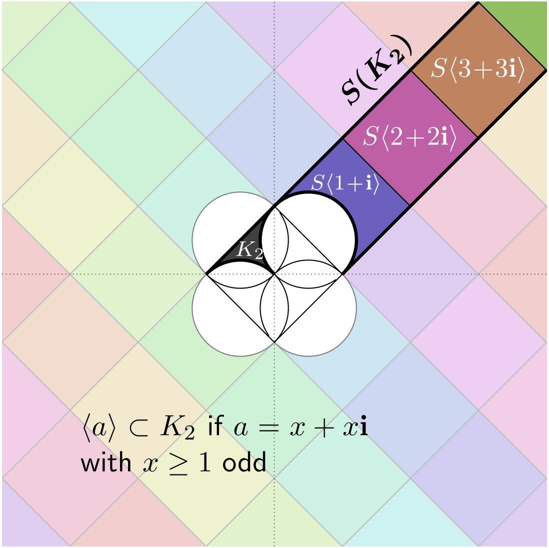


Nearest even



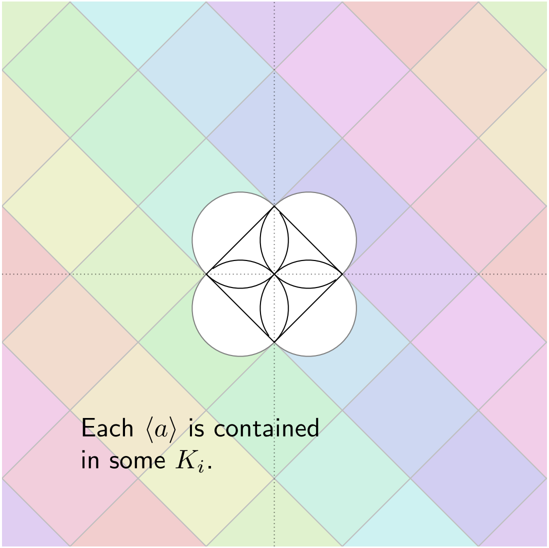
$\langle a\rangle \subset K_1$ if $a = x + y\mathbf{i}$ with $x + y$ even, $x \geq 2$, $-x + 1 < y < x - 1$

Nearest even



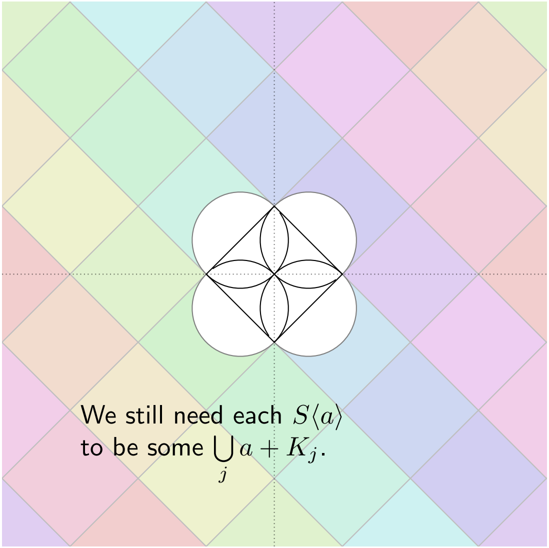
$\langle a\rangle \subset K_2$ if $a = x + x\mathbf{i}$
with $x \geq 1$ odd

Nearest even



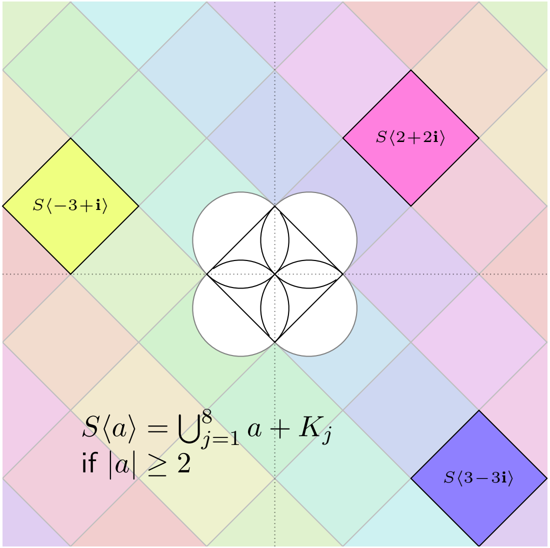
Each $\langle a \rangle$ is contained in some K_i.

Nearest even

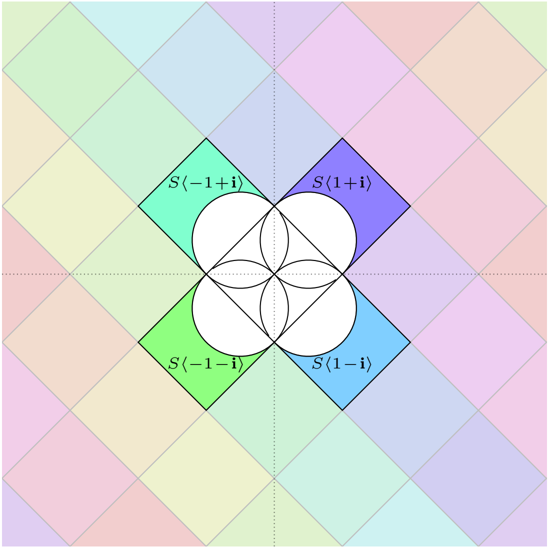


We still need each $S\langle a\rangle$ to be some $\bigcup_j a + K_j$.

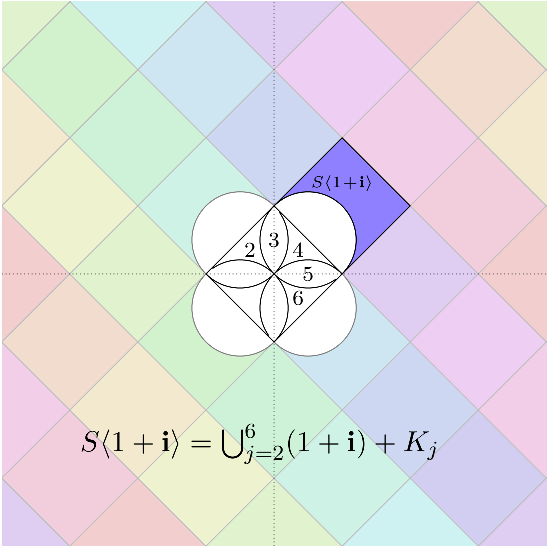
Nearest even



Nearest even



Nearest even



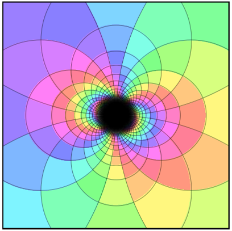
$$S\langle 1+\mathbf{i}\rangle = \bigcup_{j=2}^{6} (1+\mathbf{i}) + K_j$$

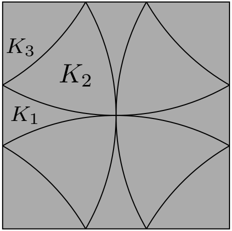
Nearest even

- For the nearest even algorithm we have sets $K_1, \ldots, K_8$ such that
  - each $\langle a \rangle$ is contained in some K_i,
  - each $S\langle a \rangle$ is some $\bigcup_j (a + K_j)$.

- Thus the finite building property is satisfied for the nearest even algorithm.

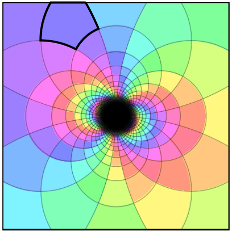
Hurwitz



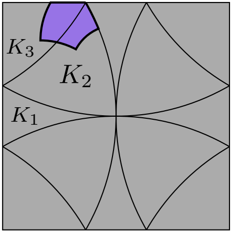


Hurwitz

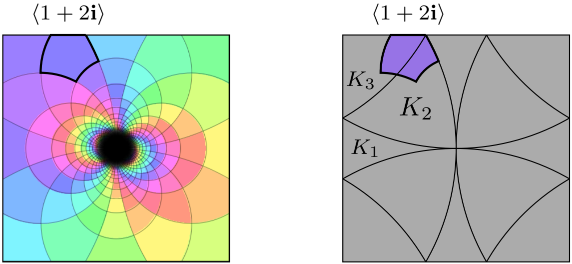
$\langle 1+2\mathbf{i} \rangle$



$\langle 1+2\mathbf{i} \rangle$



Hurwitz

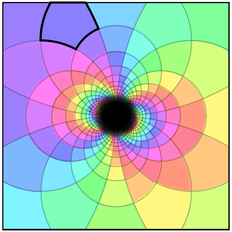


It is not true that each $\langle a \rangle$ is containing in some K_i, so the second lemma cannot be used with this algorithm.

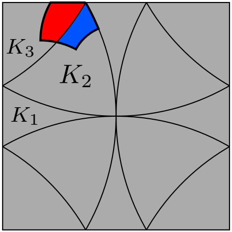
We must use the earlier lemma and show that each $K_{i,a}$ is buildable.

Hurwitz

$\langle 1 + 2\mathbf{i} \rangle$



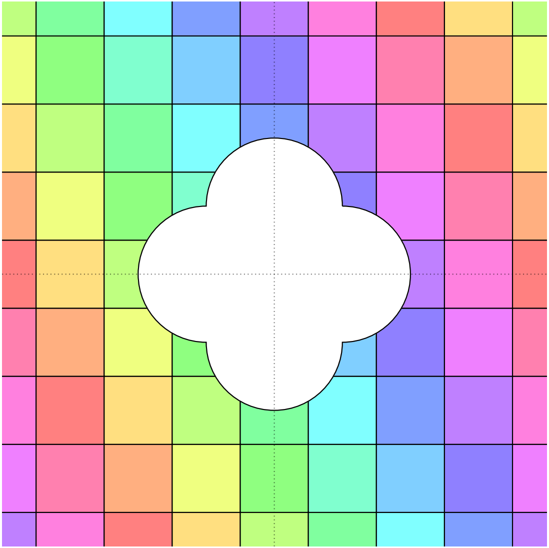
$K_{3,1+2\mathbf{i}}$ $K_{2,1+2\mathbf{i}}$



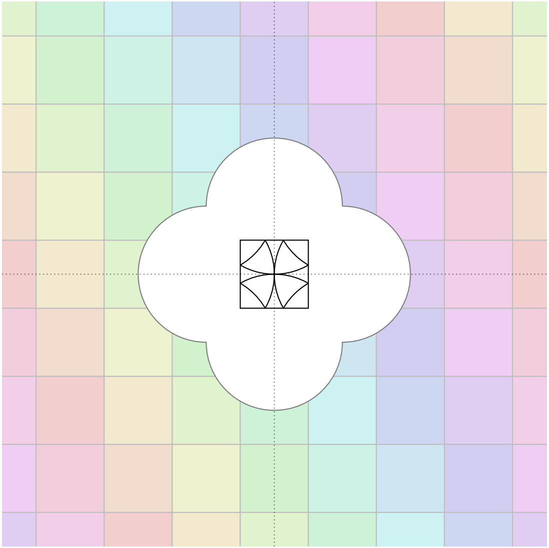
It is not true that each $\langle a \rangle$ is containing in some K_i, so the second lemma cannot be used with this algorithm.

We must use the earlier lemma and show that each $K_{i,a}$ is buildable.

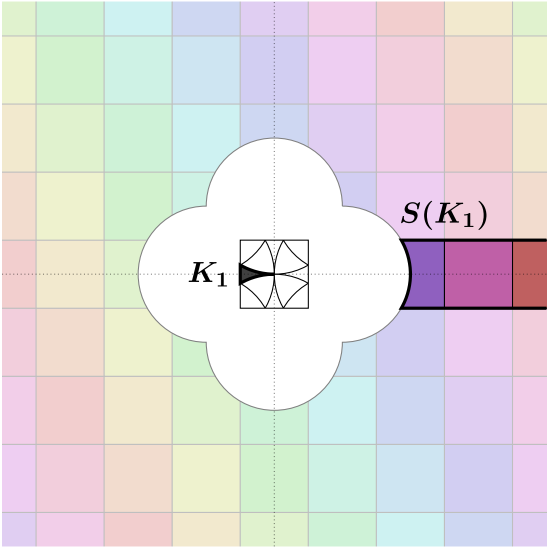
Hurwitz



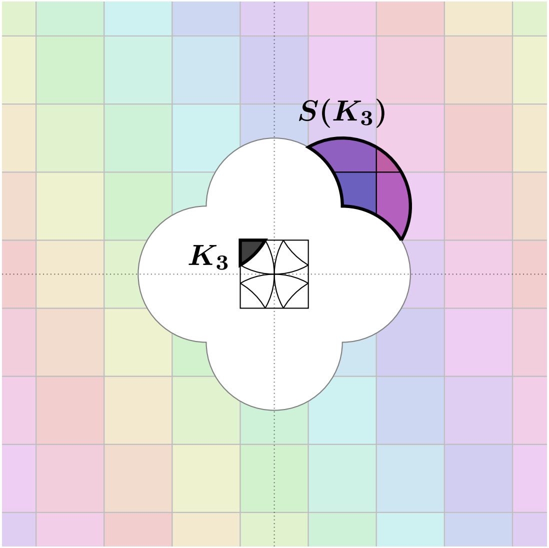
Hurwitz



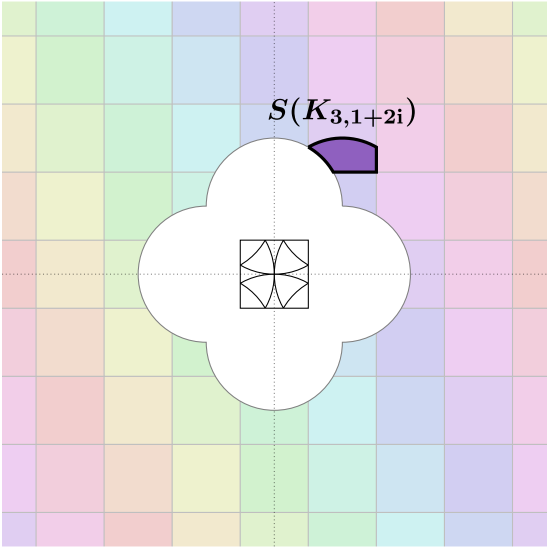
Hurwitz



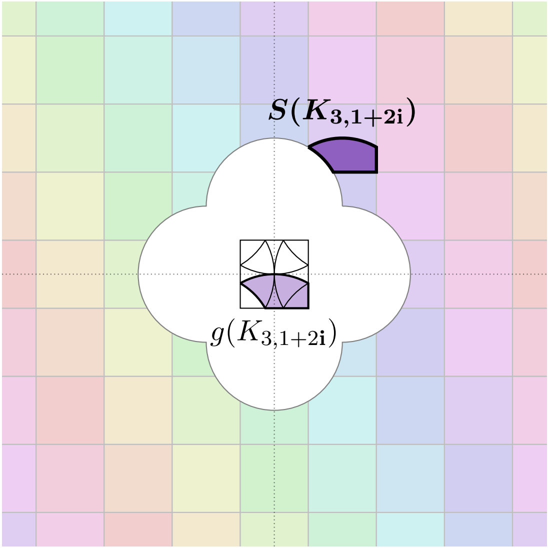
Hurwitz



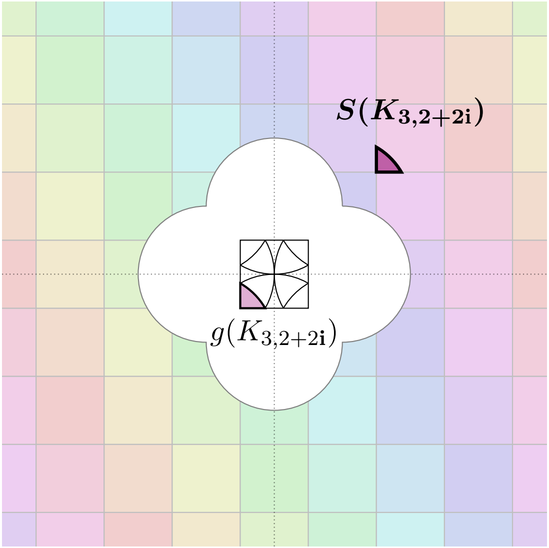
Hurwitz



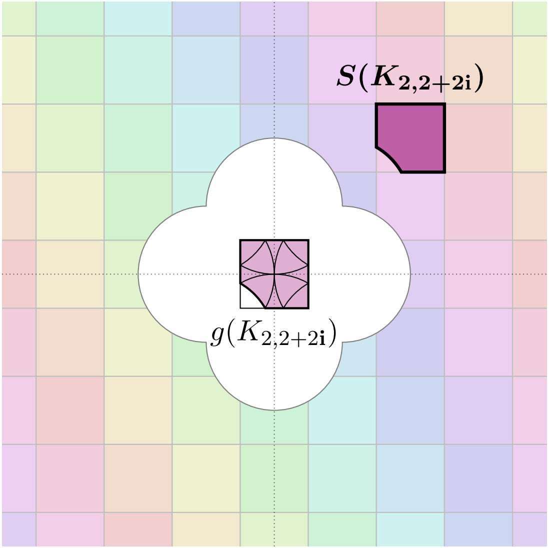
Hurwitz



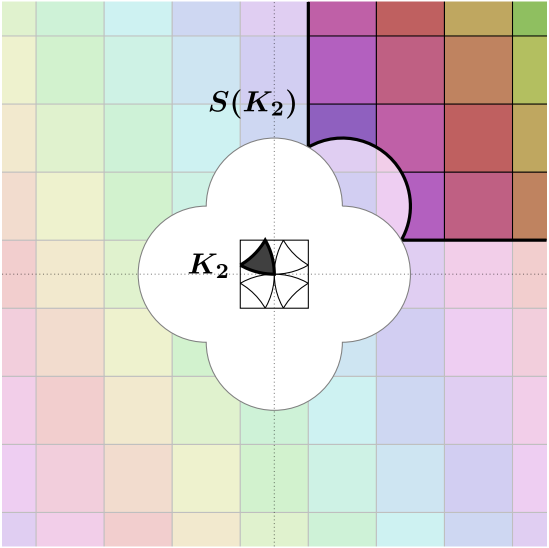
Hurwitz



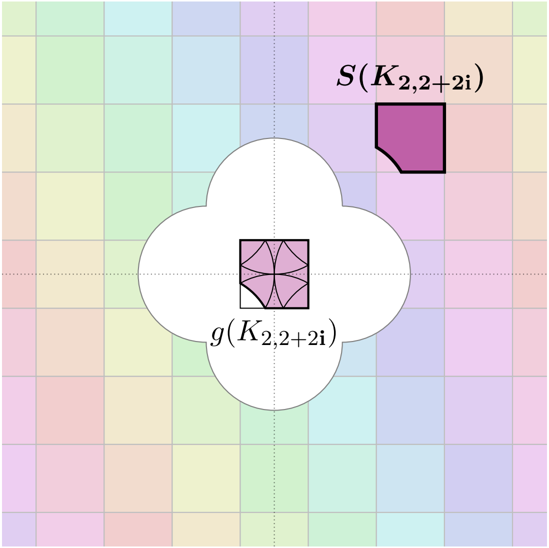
Hurwitz



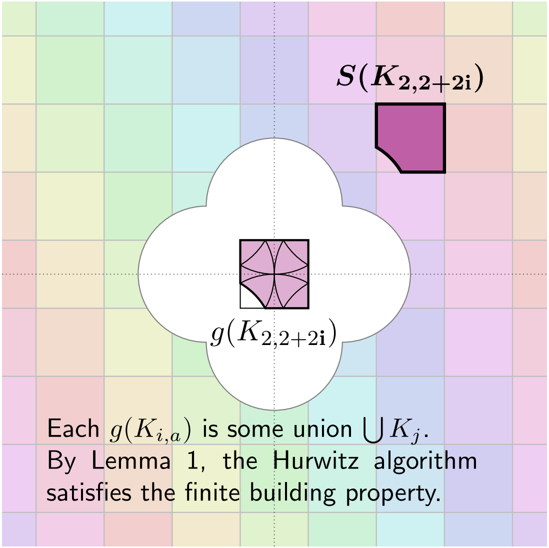
Hurwitz



Hurwitz

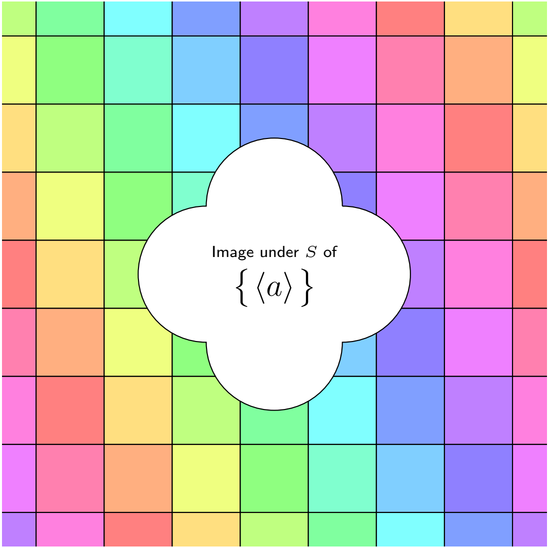


Hurwitz

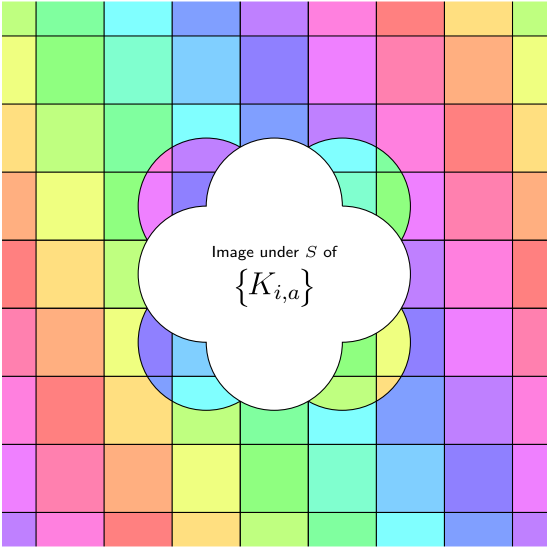


Each $g(K_{i,a})$ is some union $\bigcup K_j$.
By Lemma 1, the Hurwitz algorithm satisfies the finite building property.

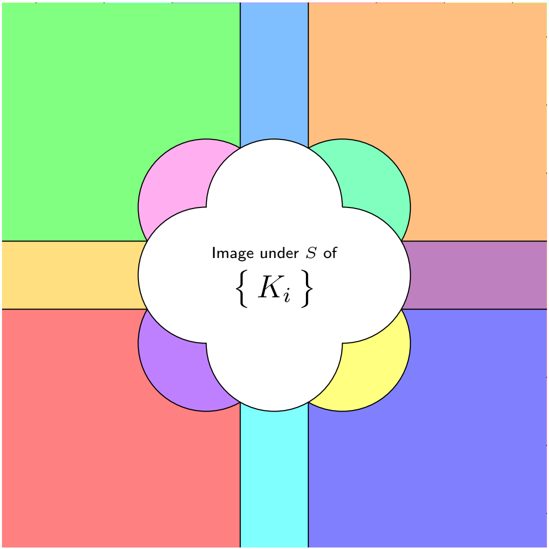
Hurwitz



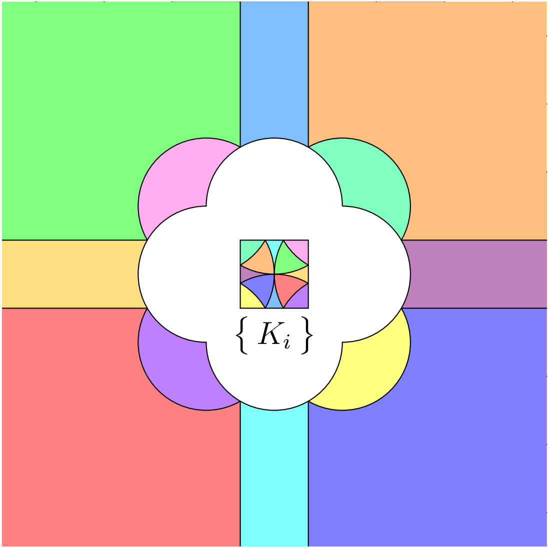
Hurwitz



Hurwitz



Hurwitz



Additional properties

- Some algorithms have additional properties that can help make proofs easier.

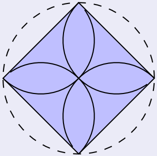
	Z-translates of K tile $\mathbb{C}$	Each $\langle a \rangle$ is contained in some K_i
Nearest integer	Yes, $Z = \mathbb{Z}[\mathbf{i}]$	No
Nearest even	Yes, $Z =$ evens	Yes
Diamond algorithm	No	No
Disk algorithm	No	Yes
Shifted Hurwitz	Yes, $Z = \mathbb{Z}[\mathbf{i}]$	No

Examples

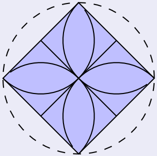
Proposition (A.)

The following algorithms satisfy the finite building property:

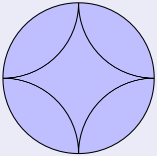
- Nearest even



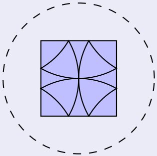
- Diamond



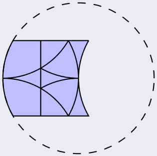
- Disk algorithm



- Nearest integer



- Shifted Hurwitz



Finite product structure

Define $G : \mathbb{C} \times \mathbb{C} \to \mathbb{C} \times \mathbb{C}$ by

$$G(z, w) = \left(\frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil, \frac{-1}{w - \left\lfloor \frac{-1}{z} \right\rceil} \right)$$

$$= (\, T^{-a} S\, z,\ S T^{-a}\, w), \quad a = \lfloor Sz \rceil$$

- One motivation for partitioning K by

  $$\mathcal{P} = \{K_1, K_2, \ldots, K_N\}$$

  is that we want to find a set

  $$\bigcup_{i=1}^{N} K_i \times L_i$$

  that is a bijectivity domain for G.

Finite product structure

Define $G : \mathbb{C} \times \mathbb{C} \to \mathbb{C} \times \mathbb{C}$ by

$$G(z, w) = \left(\frac{-1}{z} - \left\lfloor \frac{-1}{z} \right\rceil, \frac{-1}{w - \left\lfloor \frac{-1}{z} \right\rceil} \right)$$

$$= (T^{-a}Sz, ST^{-a}w), \qquad a = \lfloor Sz \rceil$$

- One motivation for partitioning K by

$$\mathcal{P} = \{K_1, K_2, \dots, K_N\}$$

is that we want to find a set

$$\bigcup_{i=1}^{N} K_i \times L_i$$

that is a bijectivity domain for G.

System of equations

- For each $1 \le i \le N$, define $\mathcal{A}_i \subset \{1, \dots, N\} \times \mathbb{Z}[\mathbf{i}]$ as

$$\begin{aligned} \mathcal{A}_i &= \{ (j,a) \,:\, K_i \subset g(K_{j,a}) \} \\ &= \{ (j,a) \,:\, K_i \subset T^{-a}S(K_{j,a}) \} \\ &= \{ (j,a) \,:\, ST^a(K_i) \subset K_{j,a} \}. \end{aligned}$$

Theorem (A.)

Suppose an algorithm satisfies the finite building property with partition $\{K_1, \dots, K_N\}$, and let $L_1, \dots, L_N$ be arbitrary complex sets. The map G is bijective a.e. on $\bigcup_{i=1}^N K_i \times L_i$ if and only if the following system of equalities holds:

$$S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a} L_j, \qquad 1 \le i \le N.$$

System of equations

- For each $1 \le i \le N$, define $\mathcal{A}_i \subset \{1, \dots, N\} \times \mathbb{Z}[\mathbf{i}]$ as

$$\begin{aligned} \mathcal{A}_i &= \{ (j,a) \,:\, K_i \subset g\big(K_{j,a}\big) \} \\ &= \{ (j,a) \,:\, K_i \subset T^{-a} S\big(K_{j,a}\big) \} \\ &= \{ (j,a) \,:\, ST^a(K_i) \subset K_{j,a} \}. \end{aligned}$$

Theorem (A.)

Suppose an algorithm satisfies the finite building property with partition $\{K_1, \dots, K_N\}$, and let $L_1, \dots, L_N$ be arbitrary complex sets. The map G is bijective a.e. on $\bigcup_{i=1}^N K_i \times L_i$ if and only if the following system of equalities holds:

$$S(L_i) = \bigcup_{(j,a) \in \mathcal{A}_i} T^{-a} L_j, \qquad 1 \le i \le N.$$

System of equations

- For each $1 \leq i \leq N$, define $\mathcal{A}_i \subset \{1, \ldots, N\} \times \mathbb{Z}[\mathbf{i}]$ as

$$\begin{aligned} \mathcal{A}_i &= \{\, (j,a) \;:\; K_i \subset g\big(K_{j,a}\big) \,\} \\ &= \{\, (j,a) \;:\; K_i \subset T^{-a} S\big(K_{j,a}\big) \,\} \\ &= \{\, (j,a) \;:\; ST^a(K_i) \subset K_{j,a} \,\}. \end{aligned}$$

Theorem (A.)

Suppose an algorithm satisfies the finite building property with partition $\{K_1, \ldots, K_N\}$, and let $L_1, \ldots, L_N$ be arbitrary complex sets. The map G is bijective a.e. on $\bigcup_{i=1}^N K_i \times L_i$ if and only if the following system of equalities holds:

$$S(L_i) = \bigcup_{(j,a) \in \mathcal{A}_i} T^{-a} L_j, \qquad 1 \leq i \leq N\,.$$

System of equations

Proof that G bijective on $\bigcup K_i \times L_i$ implies $S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a} L_j$.

$$\begin{aligned}
G\left(\bigcup_{i=1}^{N} K_i \times L_i\right) &= G\left(\bigcup_{\substack{1\le j\le N\\ a\in\mathbb{Z}[\mathbf{i}]}} K_{j,a} \times L_j\right) \\
&= \bigcup_{\substack{1\le j\le N\\ a\in\mathbb{Z}[\mathbf{i}]}} \left(T^{-a} S K_{j,a} \times S T^{-a} L_j\right) \\
&= \bigcup_{\substack{1\le j\le N\\ a\in\mathbb{Z}[\mathbf{i}]}} \left(\left(\bigcup_{\substack{i \text{ s.t.}\\ (j,a)\in\mathcal{A}_i}} K_i\right) \times S T^{-a} L_j\right) \\
&= \bigcup_{\substack{1\le j\le N\\ a\in\mathbb{Z}[\mathbf{i}]}} \bigcup_{\substack{i \text{ s.t.}\\ (j,a)\in\mathcal{A}_i}} \left(K_i \times S T^{-a} L_j\right)
\end{aligned}$$

System of equations

Proof that G bijective on $\bigcup K_i \times L_i$ implies $S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a}L_j$.

$$
\begin{aligned}
G\left(\bigcup_{i=1}^{N} K_i \times L_i\right) &= \cdots \\
&= \bigcup_{\substack{1\le j\le N \\ a\in\mathbb{Z}[\mathbf{i}]}} \; \bigcup_{\substack{i \text{ s.t.} \\ (j,a)\in\mathcal{A}_i}} \left(K_i \times ST^{-a}L_j\right) \\
&= \bigcup_{i=1}^{N} \; \bigcup_{(j,a)\in\mathcal{A}_i} \left(K_i \times ST^{-a}L_j\right) \\
&= \bigcup_{i=1}^{N} \left(K_i \times \left(\bigcup_{(j,a)\in\mathcal{A}_i} ST^{-a}L_j\right)\right)
\end{aligned}
$$

System of equations

Proof that G bijective on $\bigcup K_i \times L_i$ implies $S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a} L_j$.

$$G\left(\bigcup_{i=1}^{N} K_i \times L_i\right) = \bigcup_{i=1}^{N} \left(K_i \times \left(\bigcup_{(j,a)\in\mathcal{A}_i} ST^{-a} L_j\right)\right)$$

System of equations

Proof that G bijective on $\bigcup K_i \times L_i$ implies $S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a}L_j$.

$$G\left(\bigcup_{i=1}^{N} K_i \times L_i\right) = \bigcup_{i=1}^{N}\left(K_i \times \left(\bigcup_{(j,a)\in\mathcal{A}_i} ST^{-a}L_j\right)\right)$$

In order for $G(\Omega)$ to equal Ω, it must be that

$$L_i = \bigcup_{(j,a)\in\mathcal{A}_i} ST^{-a}L_j$$

for $i = 1, \ldots, N$. ☐

System of equations

Proof that G bijective on $\bigcup K_i \times L_i$ implies $S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a}L_j$.

$$G\left(\bigcup_{i=1}^{N} K_i \times L_i\right) = \bigcup_{i=1}^{N}\left(K_i \times \left(\bigcup_{(j,a)\in\mathcal{A}_i} ST^{-a}L_j\right)\right)$$

In order for $G(\Omega)$ to equal Ω, it must be that

$$S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a}L_j$$

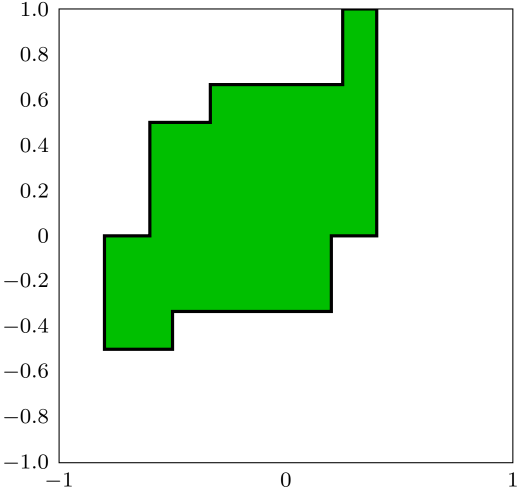
for $i = 1, \ldots, N$. □

Computer simulation

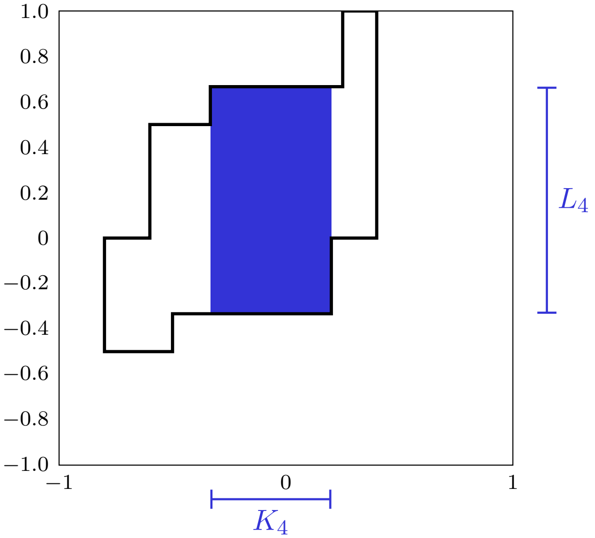
In practice, the system of union equations is not useful for constructing L_i.

- Instead, we can generate a scatter plot of points in L_i using a computer, then verify that a potential collection $L_1, \ldots, L_N$ satisfies the system.

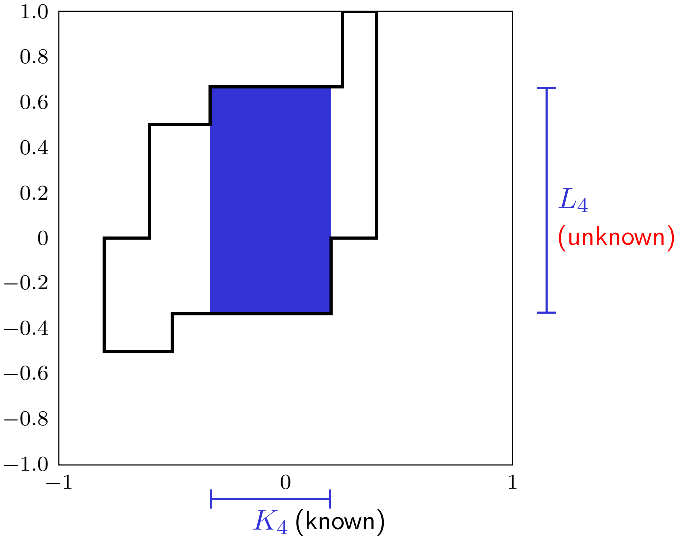
Computer simulation



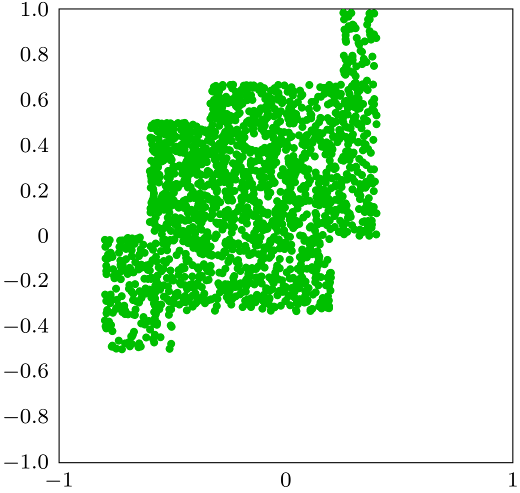
Computer simulation



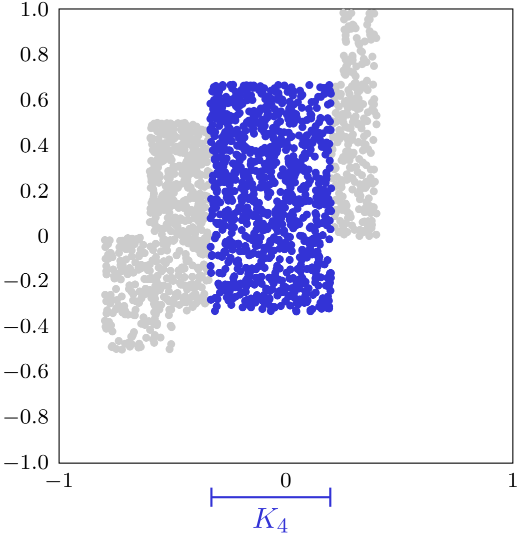
Computer simulation



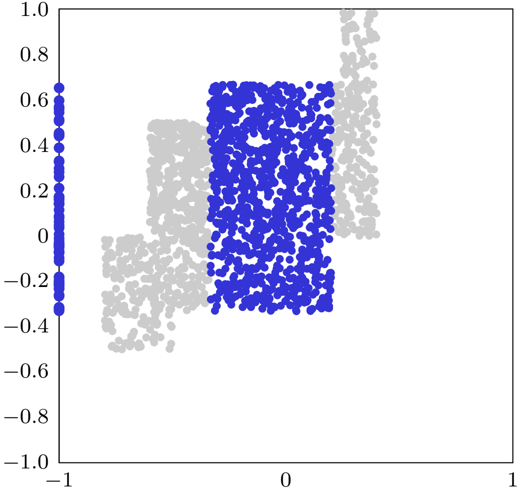
Computer simulation



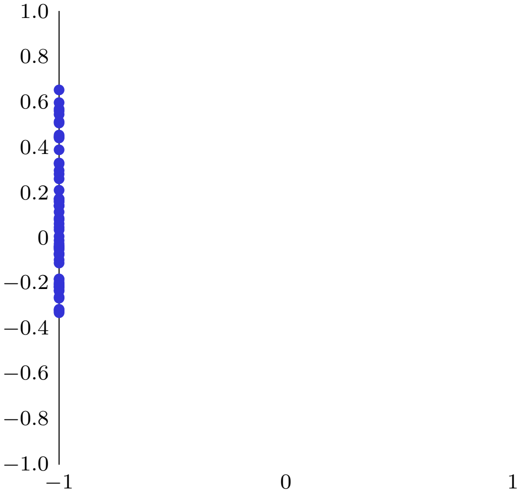
Computer simulation



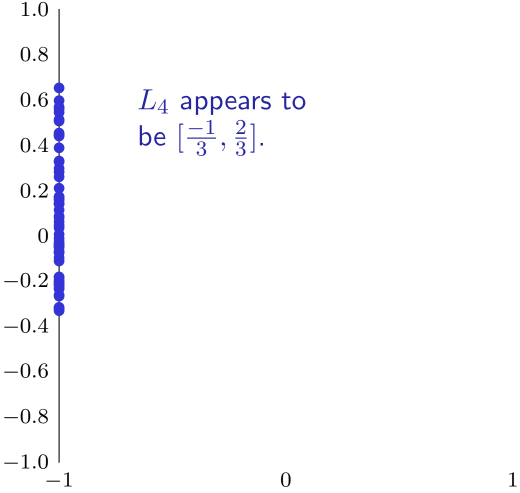
Computer simulation



Computer simulation



Computer simulation



L_4 appears to be $[\frac{-1}{3}, \frac{2}{3}]$.

Computer simulation

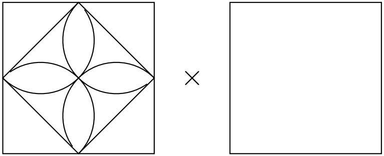
Exact descriptions of L_i are known for

- Nearest even algorithm
- Disk algorithm
- Diamond algorithm

Computer simulation

Exact descriptions of L_i are known for

- Nearest even algorithm

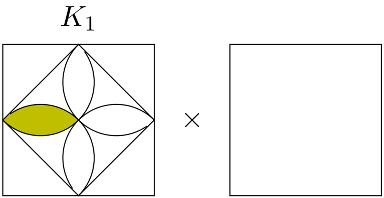


- Disk algorithm
- Diamond algorithm

Computer simulation

Exact descriptions of L_i are known for

- Nearest even algorithm

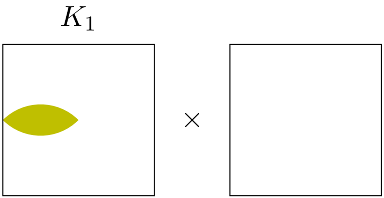


- Disk algorithm
- Diamond algorithm

Computer simulation

Exact descriptions of L_i are known for

- Nearest even algorithm

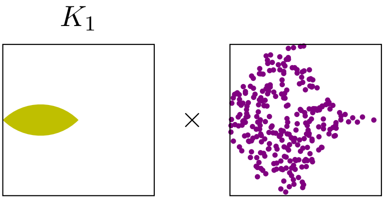


- Disk algorithm
- Diamond algorithm

Computer simulation

Exact descriptions of L_i are known for

- Nearest even algorithm

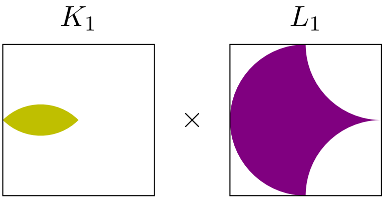


- Disk algorithm
- Diamond algorithm

Computer simulation

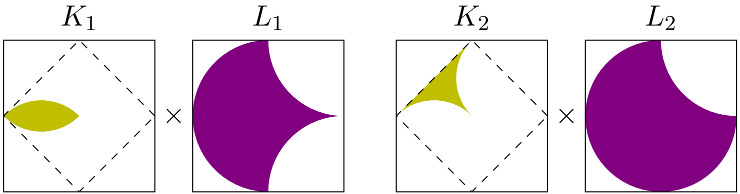
Exact descriptions of L_i are known for

- Nearest even algorithm

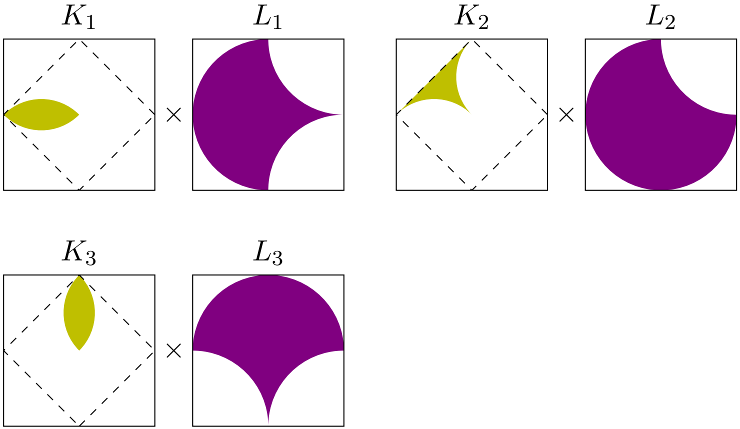


- Disk algorithm
- Diamond algorithm

Nearest even algorithm ($N = 8$)

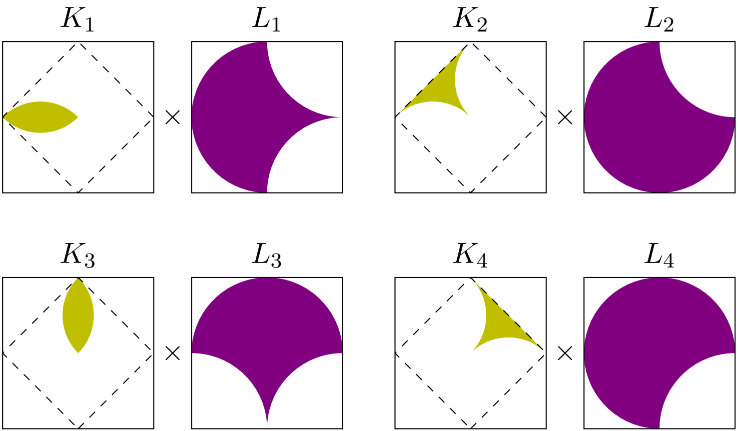


Nearest even algorithm ($N = 8$)



etc.

Nearest even algorithm ($N = 8$)



etc. etc.

System of equations

Recall the theorem that G is bijective a.e. on $\bigcup_{i=1}^{N} K_i \times L_i$ if and only if

$$S(L_i) = \bigcup_{(j,a)\in\mathcal{A}_i} T^{-a} L_j, \qquad 1 \le i \le N,$$

where

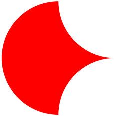
$$\begin{aligned}\mathcal{A}_i &= \{\, (j,a) \,:\, K_i \subset g\big(K_{j,a}\big) \,\} \\ &= \{\, (j,a) \,:\, K_i \subset T^{-a} S\big(K_{j,a}\big) \,\} \\ &= \{\, (j,a) \,:\, ST^{a}(K_i) \subset K_{j,a} \,\}.\end{aligned}$$

- What does this look like visually?

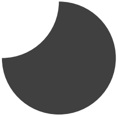
Nearest even algorithm ($N = 8$)

Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a} L_j$

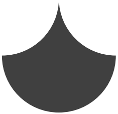
L_1



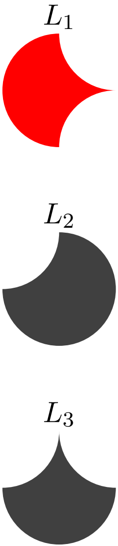
L_2



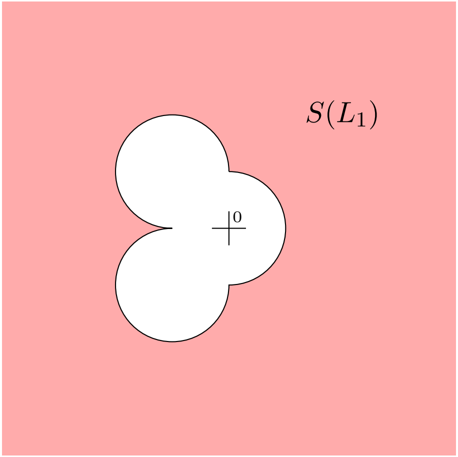
L_3



Nearest even algorithm ($N = 8$)



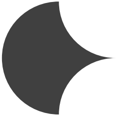
Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a} L_j$



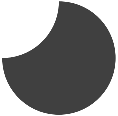
Nearest even algorithm ($N = 8$)

Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a} L_j$

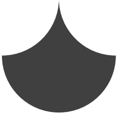
L_1

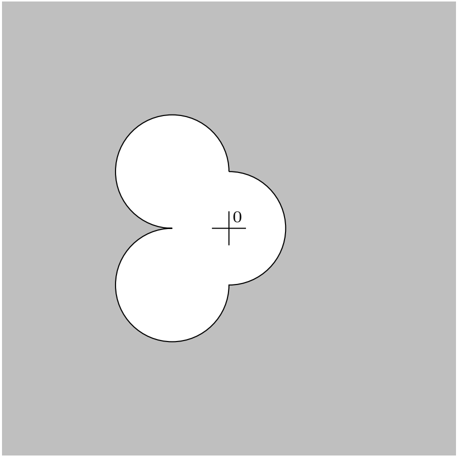


L_2

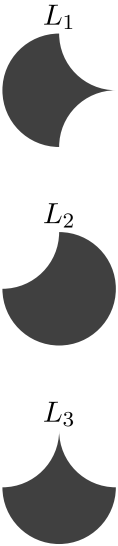


L_3

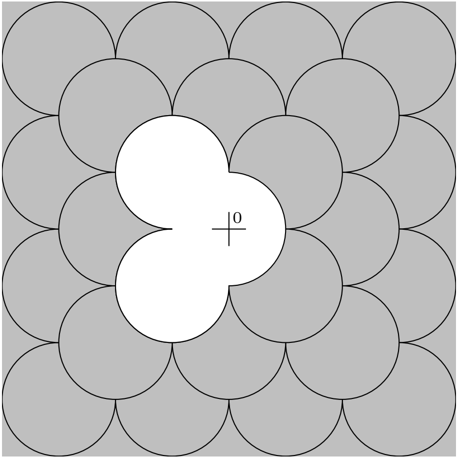




Nearest even algorithm ($N = 8$)

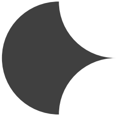


Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a} L_j$

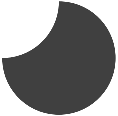


Nearest even algorithm ($N = 8$)

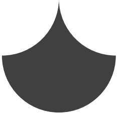
L_1



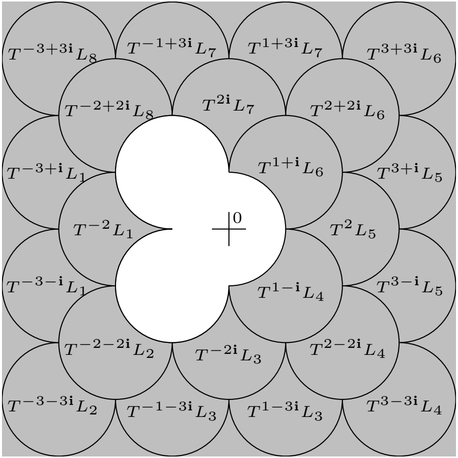
L_2



L_3

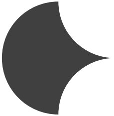


Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a} L_j$

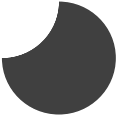


Nearest even algorithm ($N = 8$)

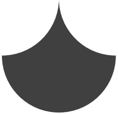
L_1



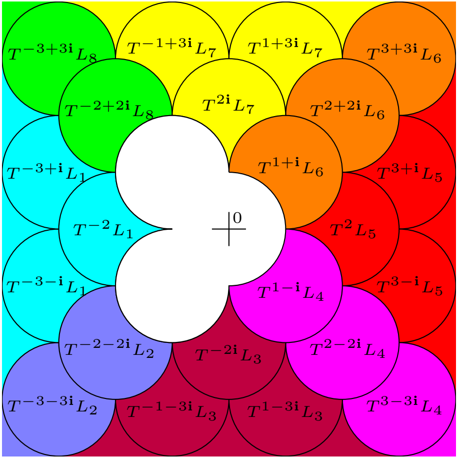
L_2



L_3

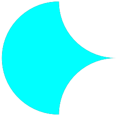


Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a}L_j$

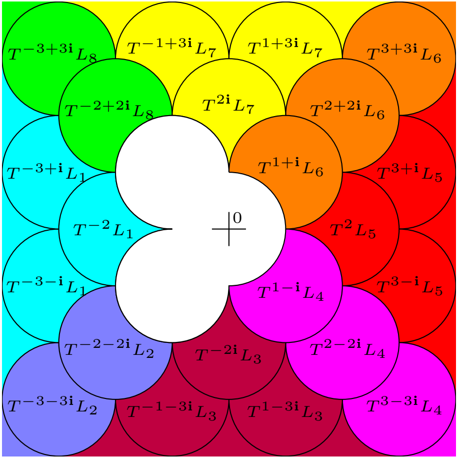


Nearest even algorithm ($N = 8$)

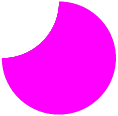
L_1



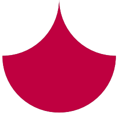
Goal: $S(L_1) = \bigcup_{(j,a)\in\mathcal{A}_1} T^{-a} L_j$



L_2



L_3



Finite product structures

For each algorithm, the set

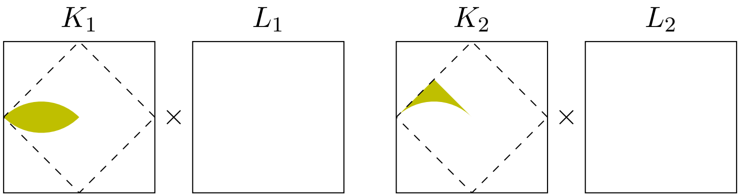
$$\Omega = \bigcup_{i=1}^{N} K_i \times L_i$$

is a bijectivity domain for $G : \mathbb{C} \times \mathbb{C} \to \mathbb{C} \times \mathbb{C}$, and $G\big|_\Omega$ is the natural extension of the Gauss map $g : K \to K$.

As a conclusion, here are images of K_i and L_i for various algorithms.

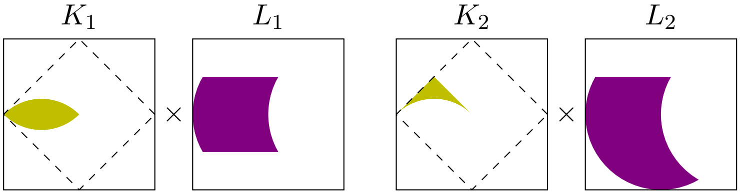
- In many cases only a few sets are shown (not all N) because the rest are rotations or reflections.

Diamond algorithm ($N = 12$)



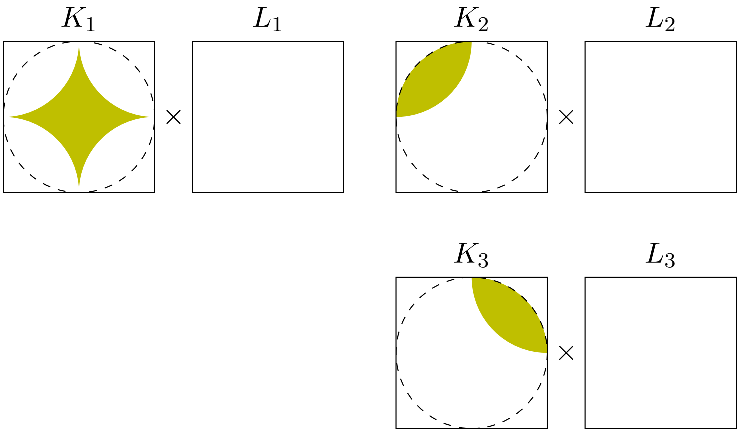
etc. etc.

Diamond algorithm ($N = 12$)



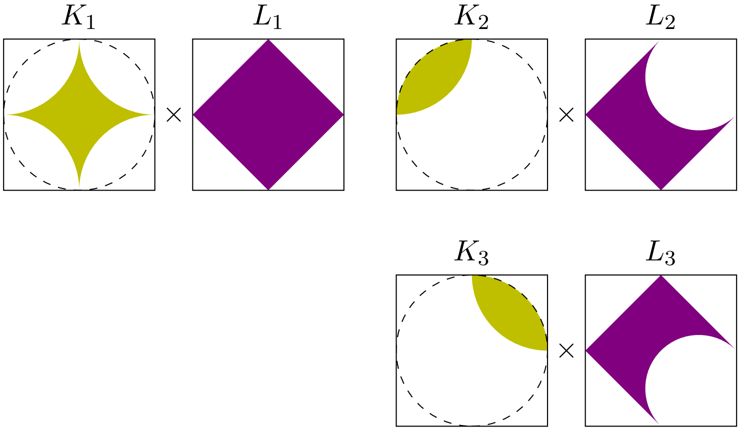
etc. etc.

Disk algorithm ($N = 5$)



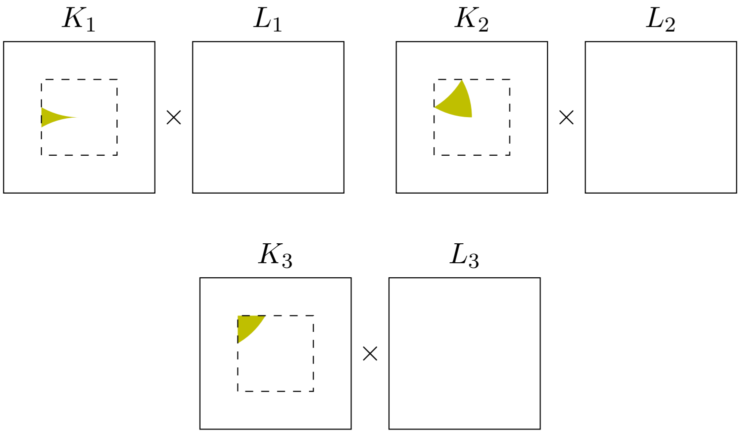
etc.

Disk algorithm ($N = 5$)



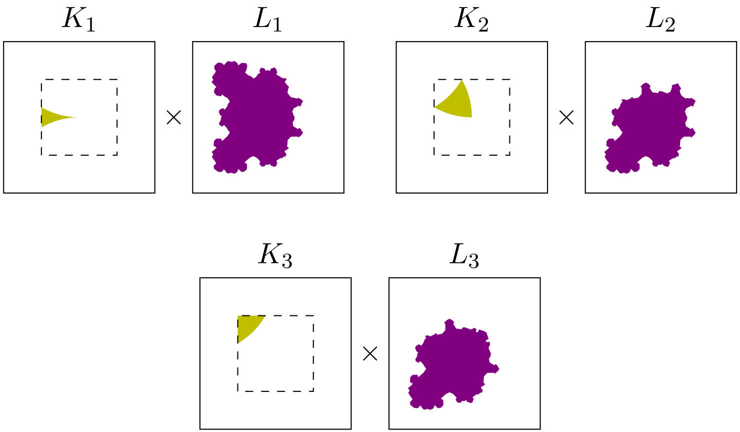
etc.

Hurwitz algorithm ($N = 12$)



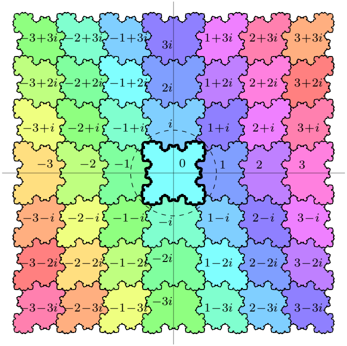
etc.

Hurwitz algorithm ($N = 12$)



etc.

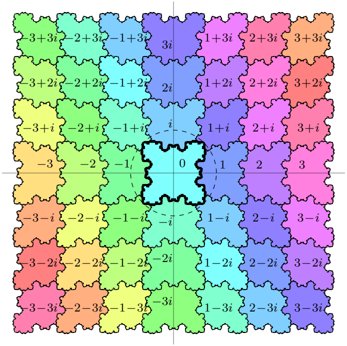
One last choice function



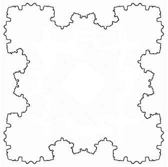
Dual-Hurwitz algorithm[5]

[5] Hiromi Ei, Shunji Ito, Hitoshi Nakada, Rie Natsui, 2019.

One last choice function

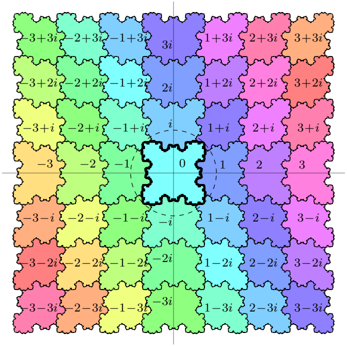


Dual-Hurwitz algorithm[5]

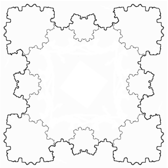


[5] Hiromi Ei, Shunji Ito, Hitoshi Nakada, Rie Natsui, 2019.

One last choice function



Dual-Hurwitz algorithm[5]



$(N = 9)$

[5] Hiromi Ei, Shunji Ito, Hitoshi Nakada, Rie Natsui, 2019.


Thank you!