Scatter, cluster, scatter, model

Nathan Reading
NC State University

Online Algebraic Geometry Seminar
Nottingham University
15 September 2022

Scattering diagrams

Cluster algebras

Combinatorial models

Mutation

Overview

Scattering diagrams arose from mirror symmetry.

Cluster algebras arose in the study of total positivity and found surprising connections to many algebraic and geometric areas.

Cluster scattering diagrams were applied to prove longstanding conjectures about cluster algebras.

More recently, a connection was discovered with scattering amplitudes in physics.

An important output of scattering diagrams are theta functions.

Today's goals:

- Introduce and connect scattering diagrams and cluster algebras.
- Mention the connection to scattering amplitudes.
- Survey the state of complete combinatorial models for cluster algebras and cluster scattering diagrams.

Section 1: Scattering diagrams

Basic setup

Summary:

- skew-symmetric matrix,
- vector space and its dual,
- integer points $\leftrightarrow$ Laurent monomials.

Basic setup

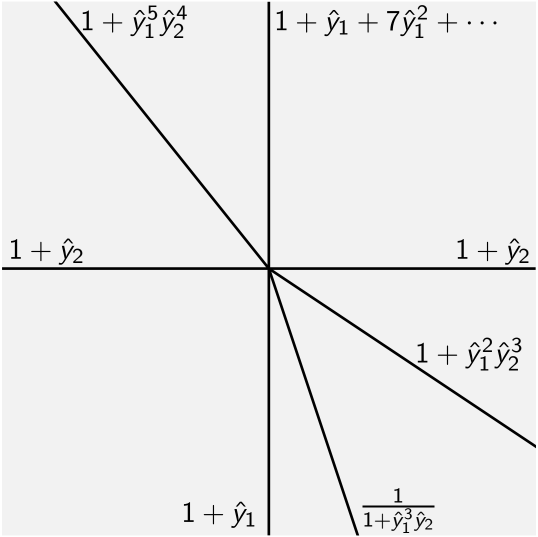
Summary:

- skew-symmetric matrix,
- vector space and its dual,
- integer points ↔ Laurent monomials.

Details:

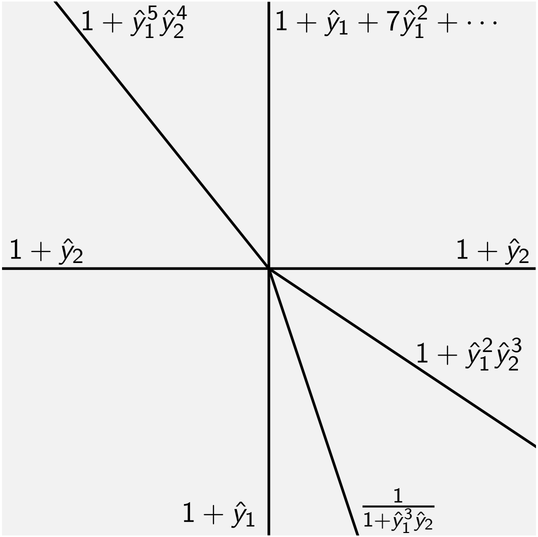
- B is an $n \times n$ skew-symmetric integer "exchange matrix"
- V real vector space, basis $\alpha_1, \ldots, \alpha_n$
- V^* its dual space, basis $\rho_1, \ldots, \rho_n$, dual to α
- $\lambda = \sum_{i=1}^n c_i \rho_i \quad \leftrightarrow \quad x^\lambda = x_1^{c_1} \cdots x_n^{c_n}$
- $\beta = \sum_{i=1}^n d_i \alpha_i \quad \leftrightarrow \quad \hat{y}^\beta = \hat{y}_1^{d_1} \cdots \hat{y}_n^{d_n}$

Scattering diagrams


$1+\hat{y}_1^5\hat{y}_2^4$
$1+\hat{y}_1+7\hat{y}_1^2+\cdots$
$1+\hat{y}_2$
$1+\hat{y}_2$
$1+\hat{y}_1^2\hat{y}_2^3$
$1+\hat{y}_1$
$\frac{1}{1+\hat{y}_1^3\hat{y}_2}$

A scattering diagram is a set of walls. Each wall is a codimension-1 cone in V^*, decorated with a scattering term—a formal power series in the $\hat{y}_i$.

Scattering diagrams


$1+\hat{y}_1^5\hat{y}_2^4$
$1+\hat{y}_1+7\hat{y}_1^2+\cdots$
$1+\hat{y}_2$
$1+\hat{y}_2$
$1+\hat{y}_1^2\hat{y}_2^3$
$1+\hat{y}_1$
$\frac{1}{1+\hat{y}_1^3\hat{y}_2}$

A scattering diagram is a set of walls. Each wall is a codimension-1 cone in V^*, decorated with a scattering term—a formal power series in the $\hat{y}_i$.

Details:

- Each wall is normal to a primitive, positive integer vector β.
- The scattering term is a univariate FPS in $\hat{y}^\beta$ with constant term 1.
- A finiteness condition

Wall-crossing homomorphisms and path-ordered products

Crossing a wall $(\mathfrak{d}, f_{\mathfrak{d}}(\hat{y}^{\beta}))$ acts on polynomials (or FPS):

$$x^{\lambda} \mapsto x^{\lambda} f_{\mathfrak{d}}^{\langle \lambda, \pm\beta \rangle}$$
$$\hat{y}^{\phi} \mapsto \hat{y}^{\phi} f_{\mathfrak{d}}^{\langle \pm\beta, B\phi \rangle}$$

Take "$-$" if crossing *with* β or "$+$" if crossing *against* β.

Path-ordered product $\mathfrak{p}_{\gamma}$: compose these along a path γ.

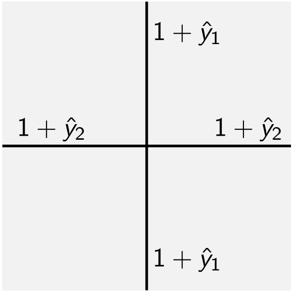
Wall-crossing homomorphisms and path-ordered products

Crossing a wall $(\mathfrak{d}, f_{\mathfrak{d}}(\hat{y}^{\beta}))$ acts on polynomials (or FPS):

$$x^{\lambda} \mapsto x^{\lambda} f_{\mathfrak{d}}^{\langle \lambda, \pm\beta \rangle}$$
$$\hat{y}^{\phi} \mapsto \hat{y}^{\phi} f_{\mathfrak{d}}^{\langle \pm\beta, B\phi \rangle}$$

Take "$-$" if crossing *with* β or "$+$" if crossing *against* β.

Path-ordered product $\mathfrak{p}_{\gamma}$: compose these along a path γ.



Let's try this in an example ($B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$):

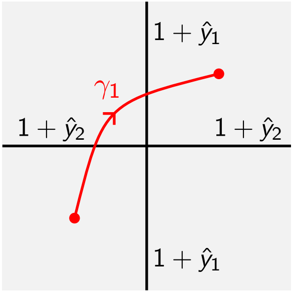
Wall-crossing homomorphisms and path-ordered products

Crossing a wall $(\mathfrak{d}, f_{\mathfrak{d}}(\hat{y}^{\beta}))$ acts on polynomials (or FPS):

$$x^{\lambda} \mapsto x^{\lambda} f_{\mathfrak{d}}^{\langle \lambda, \pm\beta \rangle}$$
$$\hat{y}^{\phi} \mapsto \hat{y}^{\phi} f_{\mathfrak{d}}^{\langle \pm\beta, B\phi \rangle}$$

Take "$-$" if crossing *with* β or "$+$" if crossing *against* β.

Path-ordered product $\mathfrak{p}_{\gamma}$: compose these along a path γ.



Let's try this in an example ($B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$):

$$\mathfrak{p}_{\gamma_1} : x_1^{-1} \mapsto x_1^{-1} \mapsto x_1^{-1}(1+\hat{y}_1)$$

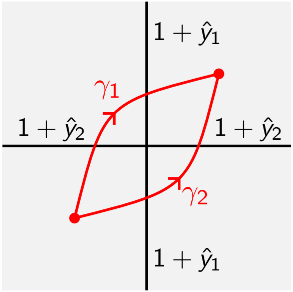
Wall-crossing homomorphisms and path-ordered products

Crossing a wall $(\mathfrak{d}, f_{\mathfrak{d}}(\hat{y}^{\beta}))$ acts on polynomials (or FPS):

$$x^{\lambda} \mapsto x^{\lambda} f_{\mathfrak{d}}^{\langle \lambda, \pm\beta \rangle}$$
$$\hat{y}^{\phi} \mapsto \hat{y}^{\phi} f_{\mathfrak{d}}^{\langle \pm\beta, B\phi \rangle}$$

Take "$-$" if crossing *with* β or "$+$" if crossing *against* β.

Path-ordered product $\mathfrak{p}_{\gamma}$: compose these along a path γ.



Let's try this in an example ($B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$):

$$\mathfrak{p}_{\gamma_1} : x_1^{-1} \mapsto x_1^{-1} \mapsto x_1^{-1}(1 + \hat{y}_1)$$

$$\mathfrak{p}_{\gamma_2} : x_1^{-1} \mapsto x_1^{-1}(1 + \hat{y}_1) \mapsto x_1^{-1}(1 + \hat{y}_1(1 + \hat{y}_2))$$

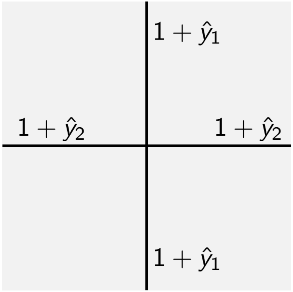
Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Example. Does the diagram below have 2 walls or 4?

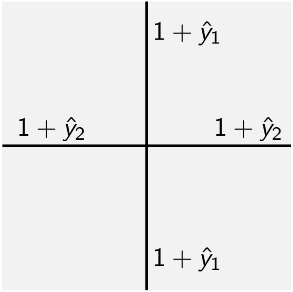


Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Example. Does the diagram below have 2 walls or 4?

A scattering diagram is consistent if path-ordered products depend only on the endpoints of the path.



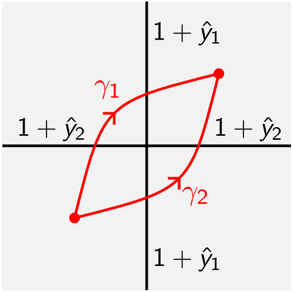
Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Example. Does the diagram below have 2 walls or 4?

A scattering diagram is consistent if path-ordered products depend only on the endpoints of the path.

Example. As we saw, this scattering diagram is not consistent



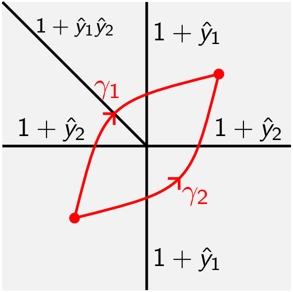
Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Example. Does the diagram below have 2 walls or 4?

A scattering diagram is consistent if path-ordered products depend only on the endpoints of the path.

Example. As we saw, this scattering diagram is not consistent



We can make it consistent by adding one wall.

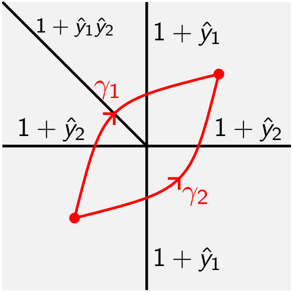
Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Example. Does the diagram below have 2 walls or 4?

A scattering diagram is consistent if path-ordered products depend only on the endpoints of the path.

Example. As we saw, this scattering diagram is not consistent



We can make it consistent by adding one wall.

$$\mathfrak{p}_{\gamma_1} : x_1^{-1} \mapsto x_1^{-1} \mapsto x_1^{-1}(1+\hat{y}_1\hat{y}_2)$$
$$\mapsto x_1^{-1}(1+\hat{y}_1)(1+\hat{y}_1\hat{y}_2(1+\hat{y}_1)^{-1})$$

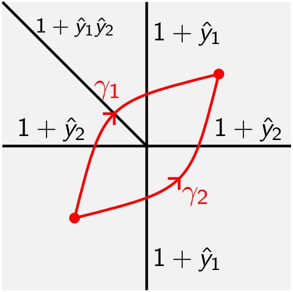
Equivalence and consistency

Two scattering diagrams are equivalent if they give the same path-ordered products.

Example. Does the diagram below have 2 walls or 4?

A scattering diagram is consistent if path-ordered products depend only on the endpoints of the path.

Example. As we saw, this scattering diagram is not consistent



We can make it consistent by adding one wall.

$$\mathfrak{p}_{\gamma_1} : x_1^{-1} \mapsto x_1^{-1} \mapsto x_1^{-1}(1+\hat{y}_1\hat{y}_2)$$
$$\mapsto x_1^{-1}(1+\hat{y}_1)(1+\hat{y}_1\hat{y}_2(1+\hat{y}_1)^{-1})$$

$$\mathfrak{p}_{\gamma_2} : x_1^{-1} \mapsto x_1^{-1}(1+\hat{y}_1) \mapsto x_1^{-1}(1+\hat{y}_1(1+\hat{y}_2))$$

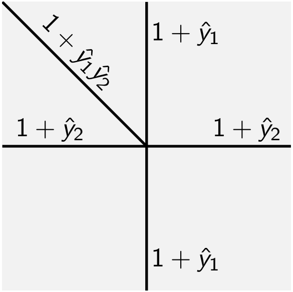
Cluster scattering diagrams

Theorem (Gross, Hacking, Keel, Kontsevich, 2014). Given a skew-symmetric integer matrix B, there is unique (up to equivalence) consistent scattering diagram $\mathrm{Scat}^T(B)$ (the cluster scattering diagram) such that

- $\mathfrak{D}$ contains the walls $(\alpha_i^\perp, 1 + \hat{y}_i)$.
- All other walls are outgoing.

A wall $(\mathfrak{d}, f_\mathfrak{d}(\hat{y}^\beta))$ is outgoing if it does not contain $B\beta$.

Example. The cluster scattering diagram for $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$.



Cluster scattering diagrams

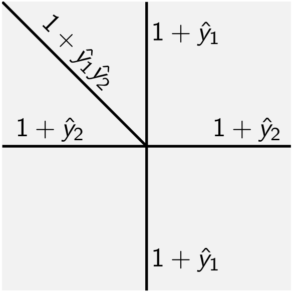
Theorem (Gross, Hacking, Keel, Kontsevich, 2014). Given a skew-symmetric integer matrix B, there is unique (up to equivalence) consistent scattering diagram $\mathrm{Scat}^T(B)$ (the cluster scattering diagram) such that

- $\mathfrak{D}$ contains the walls $(\alpha_i^\perp, 1+\hat{y}_i)$.
- All other walls are outgoing.

A wall $(\mathfrak{d}, f_\mathfrak{d}(\hat{y}^\beta))$ is outgoing if it does not contain $B\beta$.

Example. The cluster scattering diagram for $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$.

The point: The wall we added is outgoing.



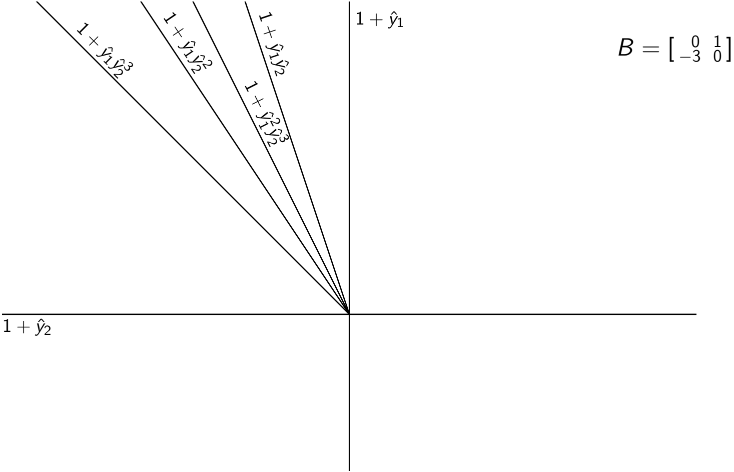
Cluster scattering fans

(Vaguely Stated) Theorem (R., 2017). A consistent scattering diagram cuts space into a complete (not necessarily finite) fan.

Cluster scattering fan: The complete fan cut out by the cluster scattering diagram.

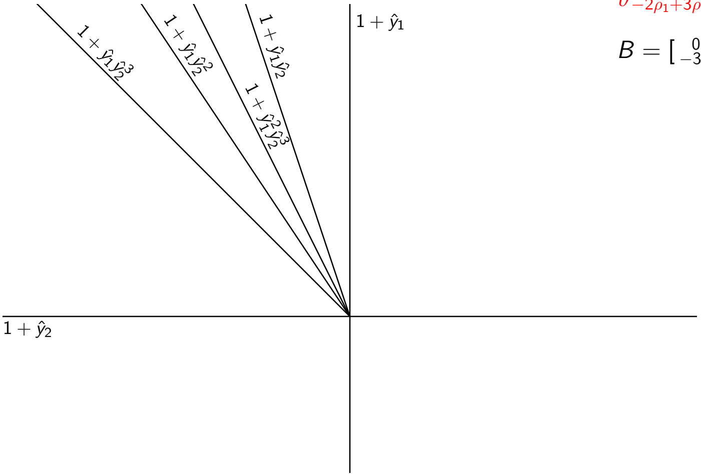
Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.


$1 + \hat{y}_1\hat{y}_2^3$
$1 + \hat{y}_1\hat{y}_2^2$
$1 + \hat{y}_1\hat{y}_2$
$1 + \hat{y}_1^2\hat{y}_2^3$
$1 + \hat{y}_1$
$B = \left[\begin{smallmatrix} 0 & 1 \\ -3 & 0 \end{smallmatrix}\right]$
$1 + \hat{y}_2$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.

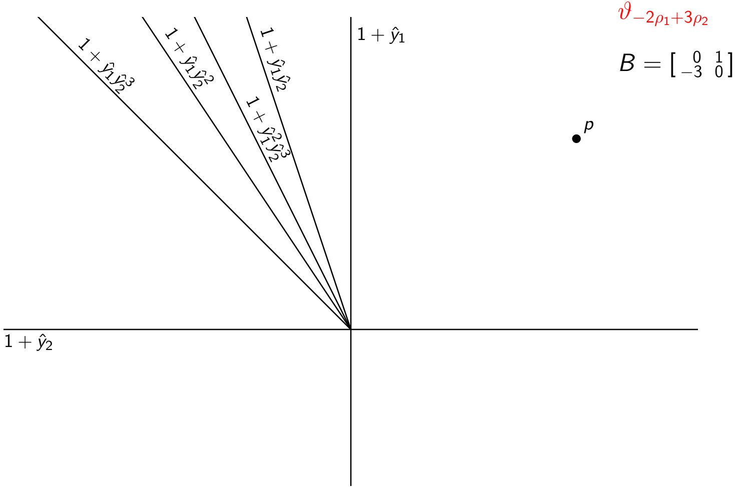


$\vartheta_{-2\rho_1+3\rho_2}$

$B = \left[\begin{smallmatrix} 0 & 1 \\ -3 & 0 \end{smallmatrix}\right]$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.

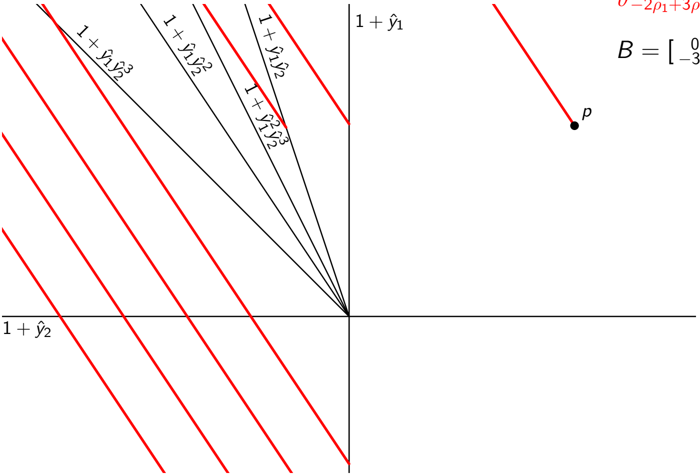


$\vartheta_{-2\rho_1+3\rho_2}$

$B = \left[\begin{smallmatrix} 0 & 1 \\ -3 & 0 \end{smallmatrix}\right]$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.

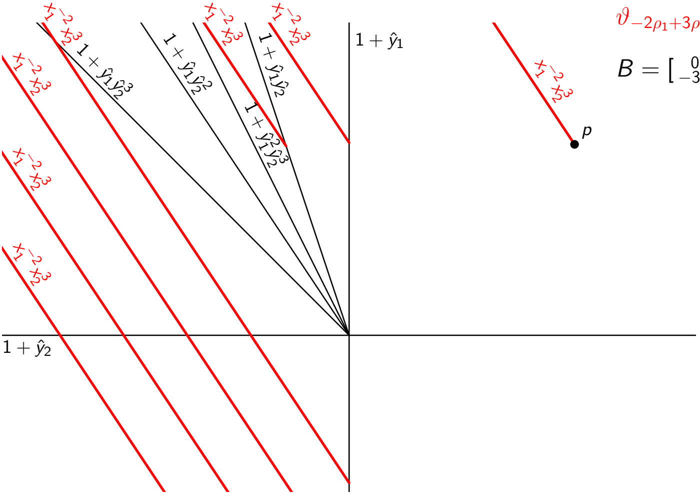


$\vartheta_{-2\rho_1+3\rho_2}$

$B = \left[\begin{smallmatrix} 0 & 1 \\ -3 & 0 \end{smallmatrix}\right]$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.

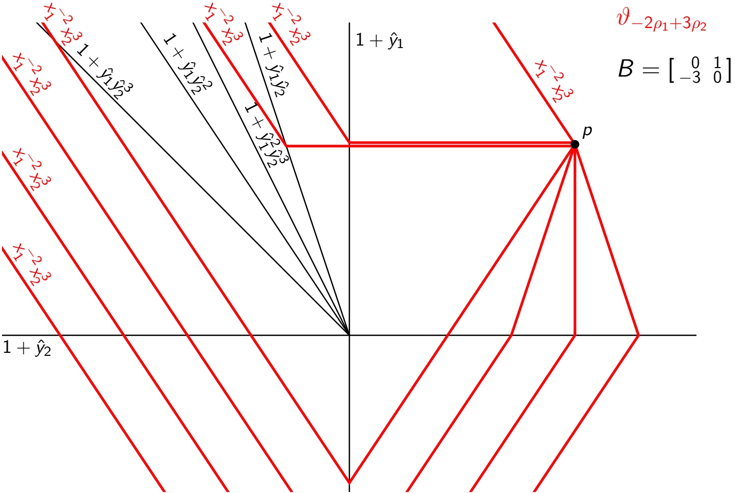


$\vartheta_{-2\rho_1+3\rho_2}$

$B = \left[\begin{smallmatrix} 0 & 1 \\ -3 & 0 \end{smallmatrix}\right]$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.

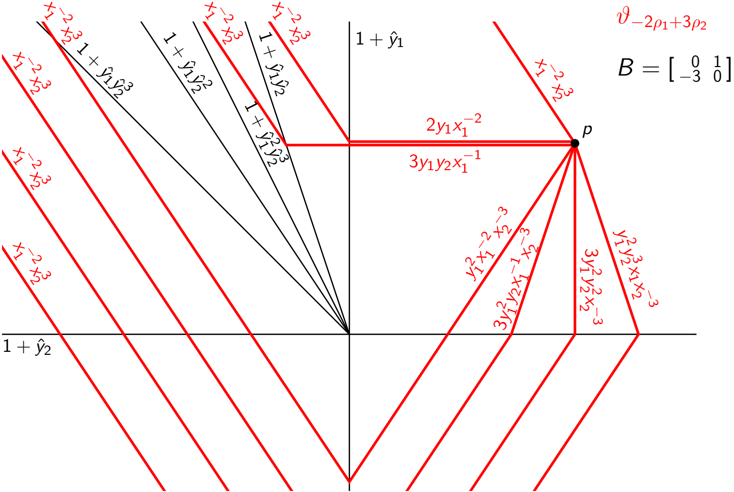


$\vartheta_{-2\rho_1+3\rho_2}$

$B = \left[\begin{smallmatrix} 0 & 1 \\ -3 & 0 \end{smallmatrix}\right]$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.

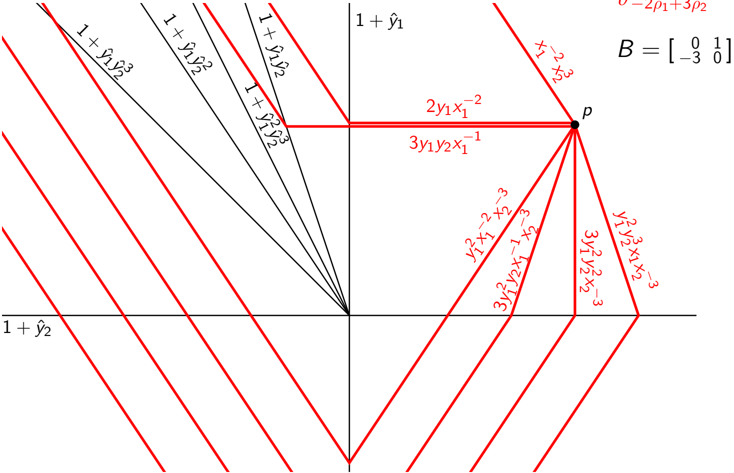


$\vartheta_{-2\rho_1+3\rho_2}$

$B = \begin{bmatrix} 0 & 1 \\ -3 & 0 \end{bmatrix}$

Theta functions

For every rational vector $\lambda \in V^*$, there is theta function ϑ_λ, defined as a sum over monomials labeling broken lines in the cluster scattering diagram.



Recap of Section 1: Scattering diagrams

A scattering diagram is a collection of walls. A wall is $(\mathfrak{d}, f_{\mathfrak{d}}(\hat{y}^{\beta}))$

- $\mathfrak{d}$ is a codimension-1 cone.
- β is a positive integer normal vector.
- $f_{\mathfrak{d}}$ is the scattering term, a formal power series in $\hat{y}^{\beta}$.

Path-ordered product: at each wall crossing, replaces each monomial by itself times a power of the scattering term $f_{\mathfrak{d}}$.

Consistent scattering diagram: path-ordered products depend only on endpoints.

Cluster scattering diagram: Start with coordinate hyperplanes. $\exists!$ way to add "outgoing" walls to get a consistent scattering diagram.

Theta functions: ϑ_{λ} for each rational λ. Sum over "broken lines".

Questions?

Section 2: Cluster algebras

Cluster algebras

Start with an initial seed consisting of initial cluster variables $x_1, \dots x_n$ and a skew-symmetric integer matrix B.

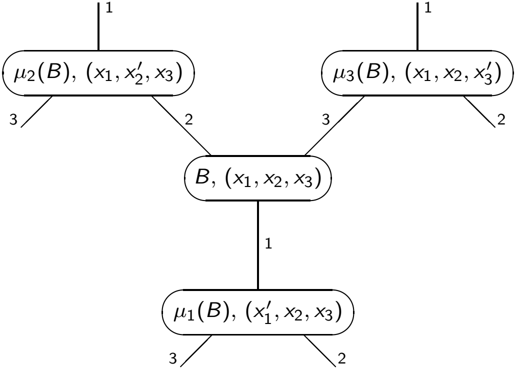
Mutation: an operation that takes a seed and gives a new seed.

- There are n "directions" for mutation.
- Mutation does two things:
  - switches out one cluster variable, replaces it with a new one;
  - changes B (and some extra rows) by matrix mutation.

  The result is a new seed.
- Mutation is involutive.

Cluster algebras (continued)

Do all possible sequences of mutations, and collect all the cluster variables which appear.


1
1
$\mu_2(B)$, (x_1, x_2', x_3)
$\mu_3(B)$, (x_1, x_2, x_3')
3
2
3
2
B, (x_1, x_2, x_3)
1
$\mu_1(B)$, (x_1', x_2, x_3)
3
2

The cluster algebra for the given initial seed is the subalgebra of $\mathcal{F}$ generated by all cluster variables.

Mutation details

Write $[a]_+$ for $\max(a, 0)$.

$B' = \mu_k(B)$ is:

$$b'_{ij} = \begin{cases} -b_{ij} & \text{if } k \in \{i, j\}; \\ b_{ij} + \operatorname{sgn}(b_{kj})[b_{ik}b_{kj}]_+ & \text{otherwise.} \end{cases}$$

We want principal coefficients, meaning we actually work with $\left[\begin{smallmatrix} B \\ I \end{smallmatrix}\right]$ but we only mutate in directions $1, \ldots, n$.

Mutating $(x_1, \ldots, x_n)$ in direction k means keeping x_i for $i \neq k$ and replacing x_k by x'_k according to the exchange relations

$$x_k x'_k = \prod_{i=1}^{n} x_i^{[b_{ik}]_+} y_i^{[b_{(n+i)k}]_+} + \prod_{i=1}^{n} x_i^{[-b_{ik}]_+} y_i^{[-b_{(n+i)k}]_+}.$$

The $y_1, \ldots, y_n$ are certain indeterminates.

Cluster variables example

Exchange matrix: $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, extended to $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

The cluster variables are

$$x_1, \quad x_2, \quad \frac{x_2 + y_1}{x_1}, \quad \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2}, \quad \frac{1 + x_1 y_2}{x_2}$$

Cluster variables example

Exchange matrix: $B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, extended to $\begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$.

The cluster variables are

$$x_1, \quad x_2, \quad \frac{x_2 + y_1}{x_1}, \quad \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2}, \quad \frac{1 + x_1 y_2}{x_2}$$

Make a change of variables:

Set

Cluster variables example

Exchange matrix: $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, extended to $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

The cluster variables are

$$x_1, \quad x_2, \quad \frac{x_2 + y_1}{x_1}, \quad \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2}, \quad \frac{1 + x_1 y_2}{x_2}$$

Make a change of variables:

Set $\hat{y}_1 = y_1 x_2^{-1}$

Cluster variables example

Exchange matrix: $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, extended to $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

The cluster variables are

$$x_1, \quad x_2, \quad \frac{x_2 + y_1}{x_1}, \quad \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2}, \quad \frac{1 + x_1 y_2}{x_2}$$

Make a change of variables:

Set $\hat{y}_1 = y_1 x_2^{-1}$ and $\hat{y}_2 = y_2 x_1$.

Cluster variables example

Exchange matrix: $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, extended to $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

The cluster variables are

$$x_1, \;\; x_2, \;\; \frac{x_2 + y_1}{x_1}, \;\; \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2}, \;\; \frac{1 + x_1 y_2}{x_2}$$

Make a change of variables:

Set $\hat{y}_1 = y_1 x_2^{-1}$ and $\hat{y}_2 = y_2 x_1$. The cluster variables become:

$$x_1, \;\; x_2, \;\; x_1^{-1} x_2 (1 + \hat{y}_1), \;\; x_1^{-1}(1 + \hat{y}_1 + \hat{y}_1 \hat{y}_2), \;\; x_2^{-1}(1 + \hat{y}_2)$$

Cluster variables example

Exchange matrix: $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, extended to $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

The cluster variables are

$$x_1, \;\; x_2, \;\; \frac{x_2 + y_1}{x_1}, \;\; \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2}, \;\; \frac{1 + x_1 y_2}{x_2}$$

Make a change of variables:

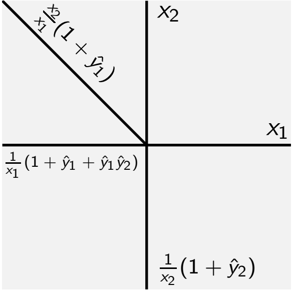
Set $\hat{y}_1 = y_1 x_2^{-1}$ and $\hat{y}_2 = y_2 x_1$. The cluster variables become:

$$x_1, \;\; x_2, \;\; x_1^{-1} x_2 (1 + \hat{y}_1), \;\; x_1^{-1}(1 + \hat{y}_1 + \hat{y}_1 \hat{y}_2), \;\; x_2^{-1}(1 + \hat{y}_2)$$

One of these might look familiar. It was the path-ordered product applied to x^{-1} in the cluster scattering diagram example for $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$.

Cluster algebras and scattering diagrams (Principal coefficients)

Theorem (Fomin-Zelevinsky, 2007). Each cluster variable v is a Laurent monomial $x^{\mathbf{g}(v)}$ times a polynomial in the $\hat{y}_i$. (The vector $\mathbf{g}(v)$ is the g-vector of v.)


$\frac{x_2}{x_1}(1+\hat{y}_1)$
x_2
x_1
$\frac{1}{x_1}(1+\hat{y}_1+\hat{y}_1\hat{y}_2)$
$\frac{1}{x_2}(1+\hat{y}_2)$

Theorem (GHKK, 2014). Part of the cluster scattering diagram cuts out the cluster fan, whose rays are spanned by the $\mathbf{g}$-vectors of cluster variables and whose cones are spanned by the $\mathbf{g}$-vectors of clusters.

Cluster monomial: monomial in the clus. variables in some cluster. $\exists$ cluster monomial for every rational point in the cluster fan.

Theorem (GHKK, 2014). Any cluster monomial v is $\vartheta_{\mathbf{g}(v)}$.

Theorem (morally GHKK, 2014, but R. had fun noticing it, 2017). Let v be a cluster monomial, let γ be a path from "near" $\mathbf{g}(v)$ to the positive cone. Then $v = \mathfrak{p}_\gamma(x^{\mathbf{g}(v)})$ (path-ordered product).

Something like a cluster monomial outside the cluster fan?

From the previous slide (with some new terminology):

- Cluster monomials ↔ Theta functions ϑ_λ for λ in cluster fan.
- Define the return function $\mathrm{ret}_\lambda = \mathfrak{p}_\gamma(x^\lambda)$, for γ a path from "near" λ to the positive cone.

Cluster monomials ↔ ret_λ for λ in the cluster fan.

Something like a cluster monomial outside the cluster fan?

From the previous slide (with some new terminology):

- Cluster monomials ↔ Theta functions ϑ_λ for λ in cluster fan.
- Define the return function $\mathrm{ret}_\lambda = \mathfrak{p}_\gamma(x^\lambda)$, for γ a path from "near" λ to the positive cone.

Cluster monomials ↔ ret_λ for λ in the cluster fan.

ϑ_λ and ret_λ are well-defined even for λ outside the cluster fan.

Something like a cluster monomial outside the cluster fan?

From the previous slide (with some new terminology):

- Cluster monomials ↔ Theta functions ϑ_λ for λ in cluster fan.
- Define the return function $\mathrm{ret}_\lambda = \mathfrak{p}_\gamma(x^\lambda)$, for γ a path from "near" λ to the positive cone.

Cluster monomials ↔ ret_λ for λ in the cluster fan.

ϑ_λ and ret_λ are well-defined even for λ outside the cluster fan.

Theta functions are on everyone's "radar screen" (if they think about cluster scattering diagrams).

Something like a cluster monomial outside the cluster fan?

From the previous slide (with some new terminology):

- Cluster monomials ↔ Theta functions ϑ_λ for λ in cluster fan.
- Define the return function $\text{ret}_\lambda = \mathfrak{p}_\gamma(x^\lambda)$, for γ a path from "near" λ to the positive cone.

Cluster monomials ↔ ret_λ for λ in the cluster fan.

ϑ_λ and ret_λ are well-defined even for λ outside the cluster fan.

Theta functions are on everyone's "radar screen" (if they think about cluster scattering diagrams).

Return functions ret_λ are not on most radar screens.

Are they the same as theta functions?

Are they interesting?

Something like a cluster monomial outside the cluster fan?

From the previous slide (with some new terminology):

- Cluster monomials $\leftrightarrow$ Theta functions ϑ_λ for λ in cluster fan.
- Define the return function $\mathrm{ret}_\lambda = \mathfrak{p}_\gamma(x^\lambda)$, for γ a path from "near" λ to the positive cone.

Cluster monomials $\leftrightarrow$ ret_λ for λ in the cluster fan.

ϑ_λ and ret_λ are well-defined even for λ outside the cluster fan.

Theta functions are on everyone's "radar screen" (if they think about cluster scattering diagrams).

Return functions ret_λ are not on most radar screens.

Are they the same as theta functions? No.

Are they interesting?

Something like a cluster monomial outside the cluster fan?

From the previous slide (with some new terminology):

- Cluster monomials $\leftrightarrow$ Theta functions ϑ_λ for λ in cluster fan.
- Define the return function $\mathrm{ret}_\lambda = \mathfrak{p}_\gamma(x^\lambda)$, for γ a path from "near" λ to the positive cone.

Cluster monomials $\leftrightarrow$ ret_λ for λ in the cluster fan.

ϑ_λ and ret_λ are well-defined even for λ outside the cluster fan.

Theta functions are on everyone's "radar screen" (if they think about cluster scattering diagrams).

Return functions ret_λ are not on most radar screens.

Are they the same as theta functions? No.

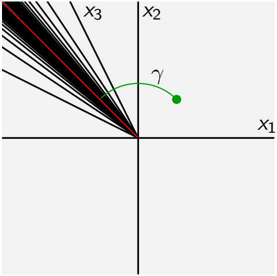
Are they interesting? Yes.

An affine rank-2 example: $B = \left[\begin{smallmatrix} 0 & 2 \\ -2 & 0 \end{smallmatrix}\right]$

The cluster fan is the whole plane except for the ray spanned by $[-1,1]$. Set $\lambda = [-1,1]$. By definition, for some formal power series $\mathcal{N}$,

$$\mathrm{ret}_\lambda = x_1^{-1} x_2 \cdot \mathcal{N}(\hat{y}_1, \hat{y}_2).$$

Number the cluster variables $x_1, x_2, \ldots$ clockwise from the initial cluster.



Each x_i is $x^{\mathbf{g}(x_i)} \cdot F_i$, for some polynomial F_i in the $\hat{y}$.

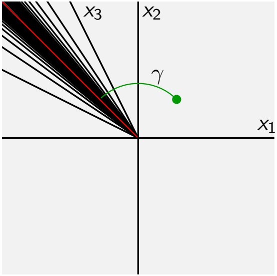
Theorem (R., 2017)

$$\mathcal{N}(\hat{y}_1, \hat{y}_2) = \lim_{i\to\infty} \frac{F_{i+1}}{F_i} = \frac{1 + \hat{y}_1 + \hat{y}_1\hat{y}_2 + \sqrt{(1 + \hat{y}_1 + \hat{y}_1\hat{y}_2)^2 - 4\hat{y}_1\hat{y}_2}}{2}.$$

An affine rank-2 example: $B = \left[\begin{smallmatrix} 0 & 2 \\ -2 & 0 \end{smallmatrix}\right]$

The cluster fan is the whole plane except for the ray spanned by $[-1, 1]$. Set $\lambda = [-1, 1]$. By definition, for some formal power series $\mathcal{N}$,

$$\mathrm{ret}_\lambda = x_1^{-1} x_2 \cdot \mathcal{N}(\hat{y}_1, \hat{y}_2).$$



Number the cluster variables $x_1, x_2, \ldots$ clockwise from the initial cluster.

Each x_i is $x^{\mathbf{g}(x_i)} \cdot F_i$, for some polynomial F_i in the $\hat{y}$.

Theorem (R., 2017) (2nd equality, Çanakçi and Schiffler, '16)

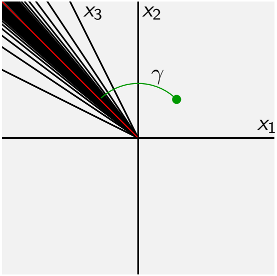
$$\mathcal{N}(\hat{y}_1, \hat{y}_2) = \lim_{i \to \infty} \frac{F_{i+1}}{F_i} = \frac{1 + \hat{y}_1 + \hat{y}_1\hat{y}_2 + \sqrt{(1 + \hat{y}_1 + \hat{y}_1\hat{y}_2)^2 - 4\hat{y}_1\hat{y}_2}}{2}.$$

An affine rank-2 example: $B = \left[\begin{smallmatrix} 0 & 2 \\ -2 & 0 \end{smallmatrix}\right]$

The cluster fan is the whole plane except for the ray spanned by $[-1, 1]$. Set $\lambda = [-1, 1]$. By definition, for some formal power series $\mathcal{N}$,

$$\mathrm{ret}_\lambda = x_1^{-1} x_2 \cdot \mathcal{N}(\hat{y}_1, \hat{y}_2).$$

Number the cluster variables $x_1, x_2, \ldots$ clockwise from the initial cluster.



Each x_i is $x^{\mathbf{g}(x_i)} \cdot F_i$, for some polynomial F_i in the $\hat{y}$.

Theorem (R., 2017)

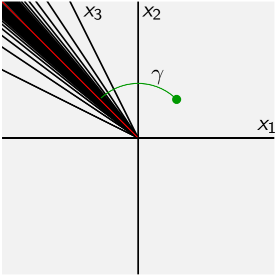
$$\mathcal{N}(\hat{y}_1, \hat{y}_2) = \lim_{i \to \infty} \frac{F_{i+1}}{F_i} = \frac{1 + \hat{y}_1 + \hat{y}_1\hat{y}_2 + \sqrt{(1 + \hat{y}_1 + \hat{y}_1\hat{y}_2)^2 - 4\hat{y}_1\hat{y}_2}}{2}.$$

This is, essentially, the generating function for Narayana numbers.

An affine rank-2 example: $B = \left[\begin{smallmatrix} 0 & 2 \\ -2 & 0 \end{smallmatrix}\right]$

The cluster fan is the whole plane except for the ray spanned by $[-1,1]$. Set $\lambda = [-1,1]$. By definition, for some formal power series $\mathcal{N}$,

$$\mathrm{ret}_\lambda = x_1^{-1} x_2 \cdot \mathcal{N}(\hat{y}_1, \hat{y}_2).$$



Number the cluster variables $x_1, x_2, \ldots$ clockwise from the initial cluster.

Each x_i is $x^{\mathbf{g}(x_i)} \cdot F_i$, for some polynomial F_i in the $\hat{y}$.

Theorem (R., 2017)

$$\mathcal{N}(\hat{y}_1, \hat{y}_2) = \lim_{i \to \infty} \frac{F_{i+1}}{F_i} = \frac{1 + \hat{y}_1 + \hat{y}_1\hat{y}_2 + \sqrt{(1 + \hat{y}_1 + \hat{y}_1\hat{y}_2)^2 - 4\hat{y}_1\hat{y}_2}}{2}.$$

This is, essentially, the generating function for Narayana numbers.

Who knew this was mathematical physics?

Aside: Scattering amplitudes

Empirically measurable probability density in n-space. In principle, computed/approximated as a sum/integral over Feynman diagrams. In practice, computations are forbidingly complex.

Amplitudes can be encoded compactly as sums of tensor products of certain "symbols" (certain functions of $x_1, \ldots, x_n$).

Current hot idea: Can we skip Feynman diagrams and directly find symbols and how symbols combine as tensor products? We need a a machine that makes functions.

In some cases, "symbols" are cluster variables and compatibility describes which symbols appear together in tensor products.

In more complicated cases, there are irrational symbols, related to theta functions and return functions. Compute these? Use the cluster scattering fan to describe which symbols appear together?

Upshot: This is another reason to want combinatorial models.

Recap of Section 2: Cluster algebras

Cluster algebras: A multi-directional recurrence produces "clusters" of cluster variables, which are rational functions in the x_i and y_i.

Cluster monomials: Monomials in the cluster variables in a cluster.

Each cluster monomial v is $x^{\mathbf{g}(v)} \cdot (\text{a polynomial in the } \hat{y}_i)$.
The vector $\mathbf{g}(v)$ is the $\mathbf{g}$-vector of v.

This fits into the cluster scattering diagram picture. Each v is:

- a return function $\mathrm{ret}_{\mathbf{g}(v)}$.
- a theta function $\vartheta_{\mathbf{g}(v)}$.

ret_λ and ϑ_λ make sense even when λ is not the $\mathbf{g}$-vector of a cluster monomial. But computing them takes work.

Questions?

Section 3: Combinatorial models

What we want

Given a skew-symmetric (skew-symmetrizable) B, and some specific B in particular, it would be good to know:

- the cluster variables;
- the scattering diagram;
- the scattering fan;
- the theta functions ϑ_λ; and
- the return functions ret_λ.

What we want

Given a skew-symmetric (skew-symmetrizable) B, and some specific B in particular, it would be good to know:

- the cluster variables;
- the scattering diagram;
- the scattering fan;
- the theta functions ϑ_λ; and
- the return functions ret_λ.

We want combinatorial models for this.

Finite type

Finite type: there are only finitely many cluster variables.

Theorem (Fomin-Zelevinsky 2002).

- The cluster algebra is of finite type if and only if B has associated Cartan matrix A of finite type.

Finite type

Finite type: there are only finitely many cluster variables.

Theorem (Fomin-Zelevinsky 2002).

- The cluster algebra is of finite type if and only if B (is mutation-equivalent to an exchange matrix that) has associated Cartan matrix A of finite type.

Finite type

Finite type: there are only finitely many cluster variables.

Theorem (Fomin-Zelevinsky 2002).

- The cluster algebra is of finite type if and only if B (is mutation-equivalent to an exchange matrix that) has associated Cartan matrix A of finite type.
- Cluster variables are Laurent polynomials. Their denominator vectors are almost positive roots in the root system for A.

Finite type

Finite type: there are only finitely many cluster variables.

Theorem (Fomin-Zelevinsky 2002).

- The cluster algebra is of finite type if and only if B (is mutation-equivalent to an exchange matrix that) has associated Cartan matrix A of finite type.
- Cluster variables are Laurent polynomials. Their denominator vectors are almost positive roots in the root system for A.
- $\{$cluster variables$\} \leftrightarrow \{$almost positive roots$\}$ is a bijection.

Finite type

Finite type: there are only finitely many cluster variables.

Theorem (Fomin-Zelevinsky 2002).

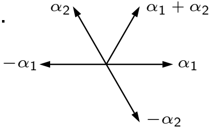
- The cluster algebra is of finite type if and only if B (is mutation-equivalent to an exchange matrix that) has associated Cartan matrix A of finite type.
- Cluster variables are Laurent polynomials. Their denominator vectors are almost positive roots in the root system for A.
- $\{$cluster variables$\} \leftrightarrow \{$almost positive roots$\}$ is a bijection.
- A combinatorial description of "adjacency".

Finite type

Finite type: there are only finitely many cluster variables.

Theorem (Fomin-Zelevinsky 2002).

- The cluster algebra is of finite type if and only if B (is mutation-equivalent to an exchange matrix that) has associated Cartan matrix A of finite type.
- Cluster variables are Laurent polynomials. Their denominator vectors are almost positive roots in the root system for A.
- $\{$cluster variables$\} \leftrightarrow \{$almost positive roots$\}$ is a bijection.
- A combinatorial description of "adjacency".



Example: $B = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, $A = \left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$

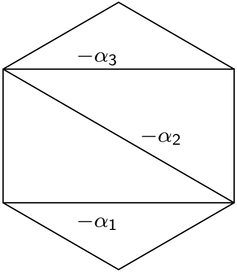
The cluster variables are

$$x_1,\ \ x_2,\ \ \frac{x_2 + y_1}{x_1},\ \ \frac{x_1 y_1 y_2 + x_2 + y_1}{x_1 x_2},\ \ \frac{1 + x_1 y_2}{x_2}$$

Finite type (continued)

Compatibility ("adjacency") of almost positive roots:

- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).



Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$

Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.

Finite type (continued)

Compatibility ("adjacency") of almost positive roots:

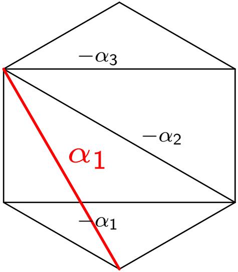
- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).

Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$



Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.

Finite type (continued)

Compatibility ("adjacency") of almost positive roots:

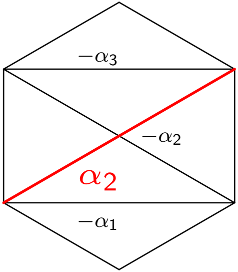
- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).

Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$



Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.

Finite type (continued)

Compatibility ("adjacency") of almost positive roots:

- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).

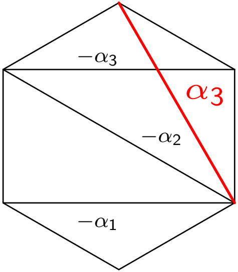
Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$

Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.



Finite type (continued)

Compatibility ("adjacency") of almost positive roots:

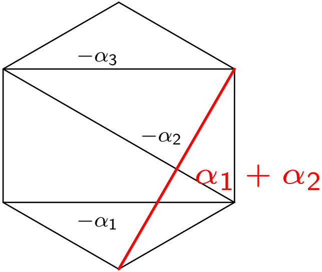
- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).

Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$



Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.

Finite type (continued)

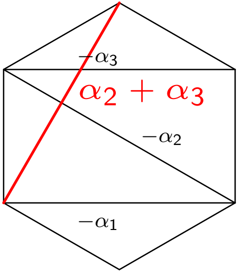
Compatibility ("adjacency") of almost positive roots:

- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).

Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$



Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$

Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.

Finite type (continued)

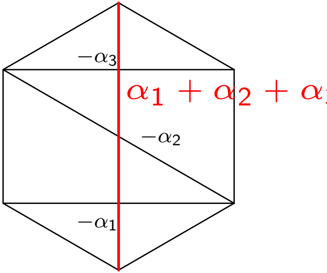
Compatibility ("adjacency") of almost positive roots:

- $-\alpha_i$ compatible with β iff β is a combination of the *other* simple roots.
- Compatibility is invariant under a certain "rotation" of the almost positive roots (a piecewise linear map that is a deformation of a Coxeter element).

Example (and model).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

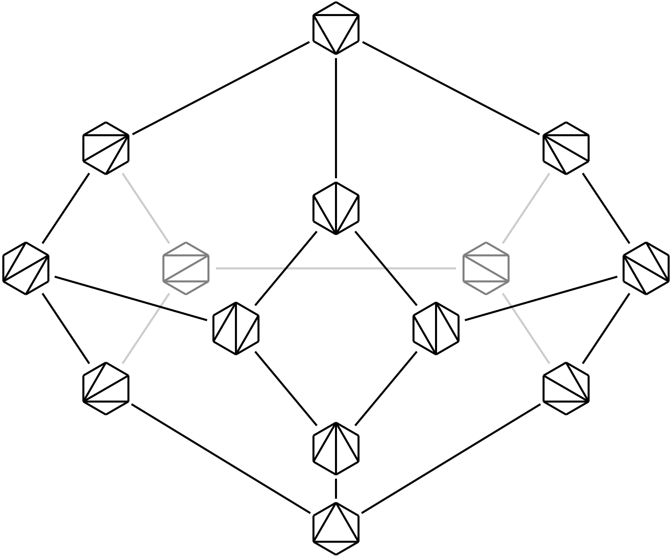


Simples: $\{\alpha_1, \alpha_2, \alpha_3\}$

Other positive roots: $\alpha_1 + \alpha_2$, $\alpha_2 + \alpha_3$, $\alpha_1 + \alpha_2 + \alpha_3$.

Example (continued)   The 3-dimensional associahedron

Clusters ↔ triangulations.   Mutations ↔ diagonal flips.



Another approach to finite type (R., Speyer)

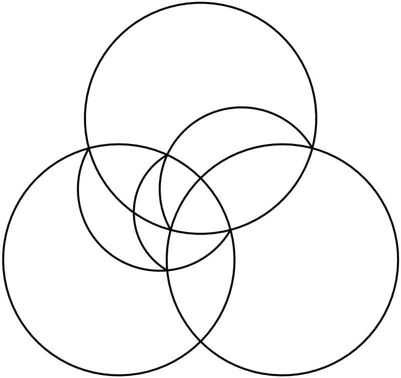
Exchange matrix $B \leftrightarrow$ Cartan matrix A and Coxeter element c.

Coxeter element: product of the simple reflections in some order.

Example: $B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$, $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$, $c = s_1 s_2 s_3$

The cluster fan is the c-Cambrian fan, which is constructed using the combinatorics of reduced words for c-sortable elements in the Weyl group W.

The walls of the c-Cambrian fan are certain pieces (called shards) of the reflecting hyperplanes for W.



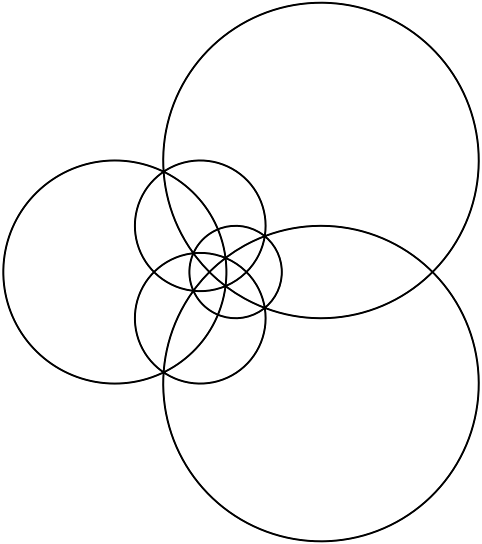
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

$c = s_1 s_2 s_3$



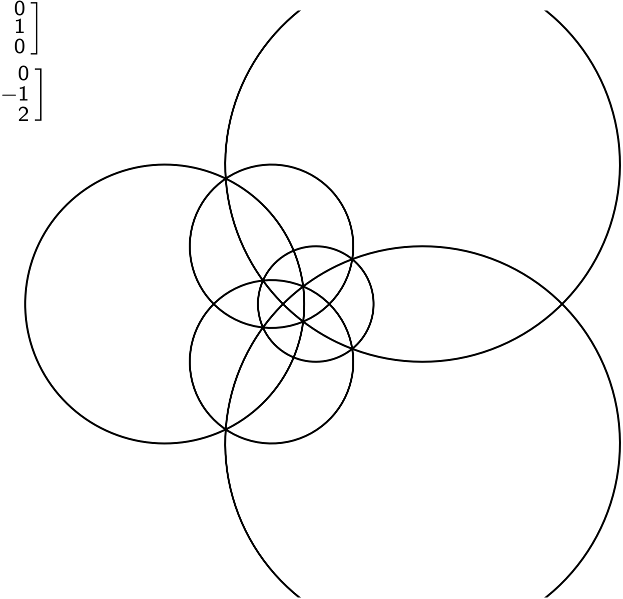
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

$c = s_1 s_2 s_3$



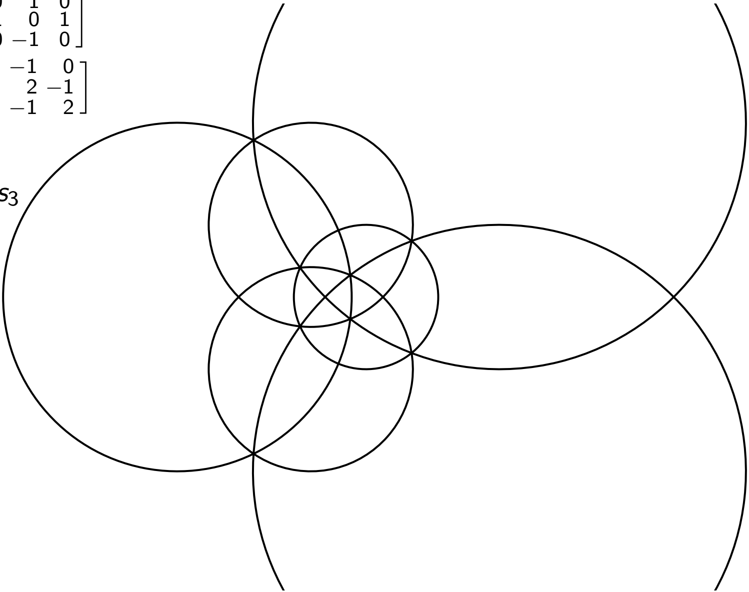
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

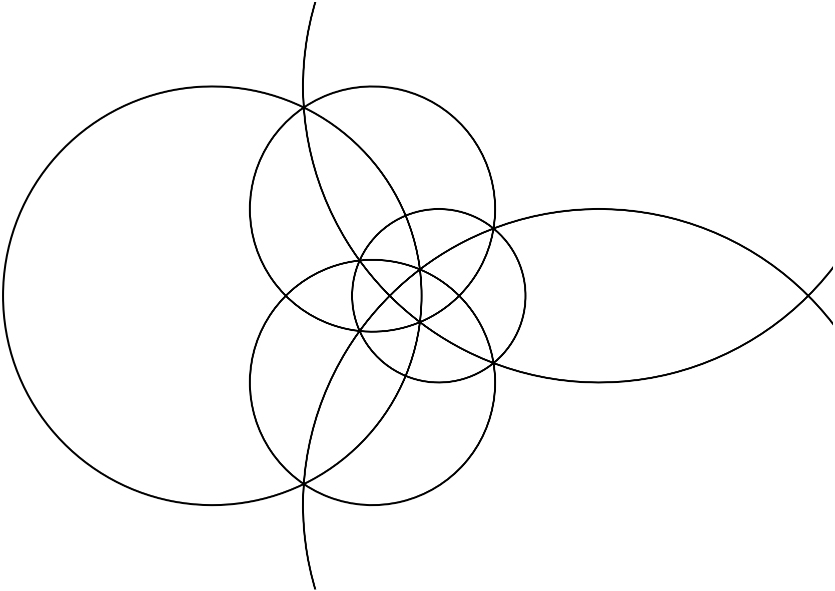
$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

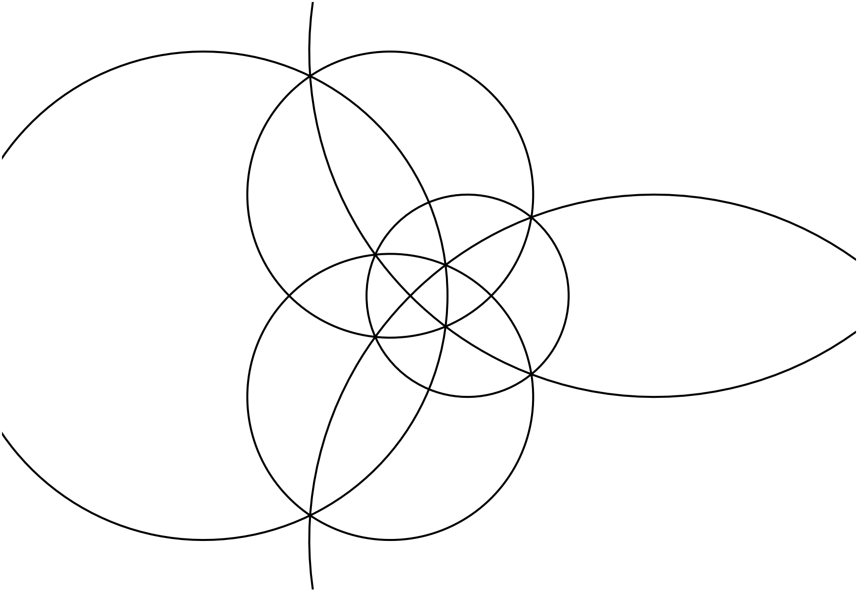
$c = s_1 s_2 s_3$



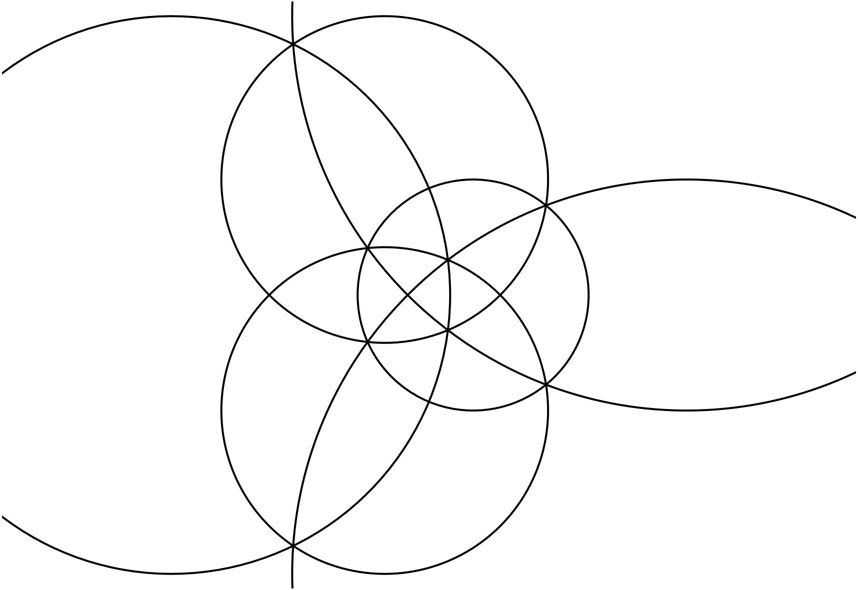
Example: c-Sortable elements, c-Cambrian fans, shards



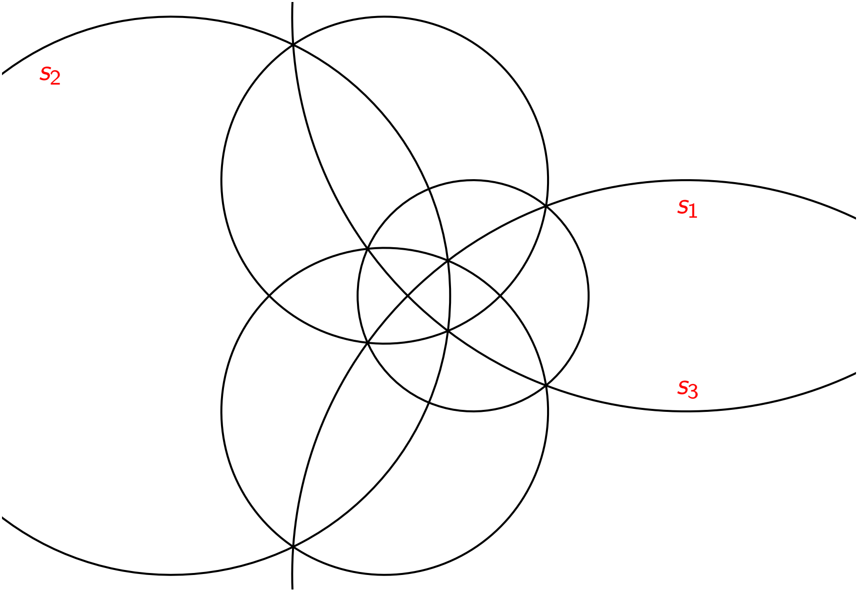
Example: *c*-Sortable elements, *c*-Cambrian fans, shards



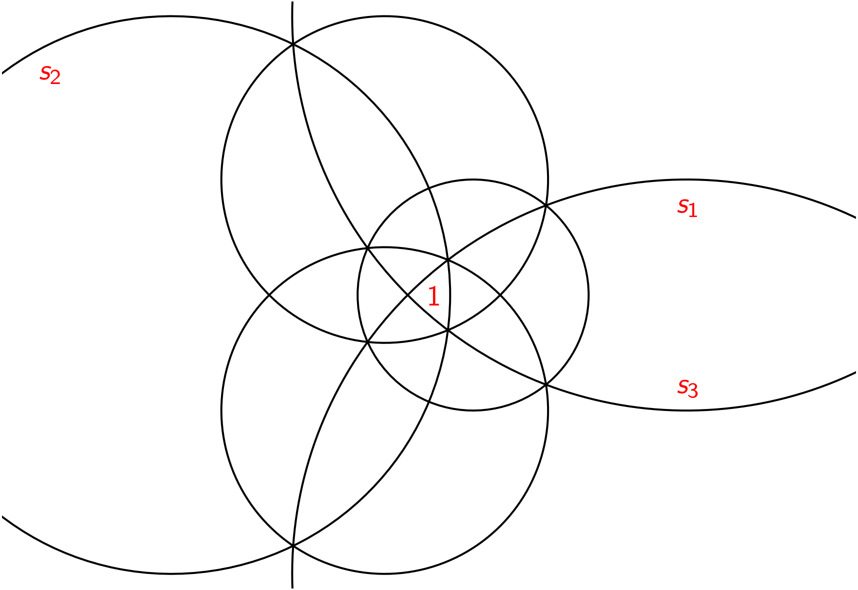
Example: c-Sortable elements, c-Cambrian fans, shards



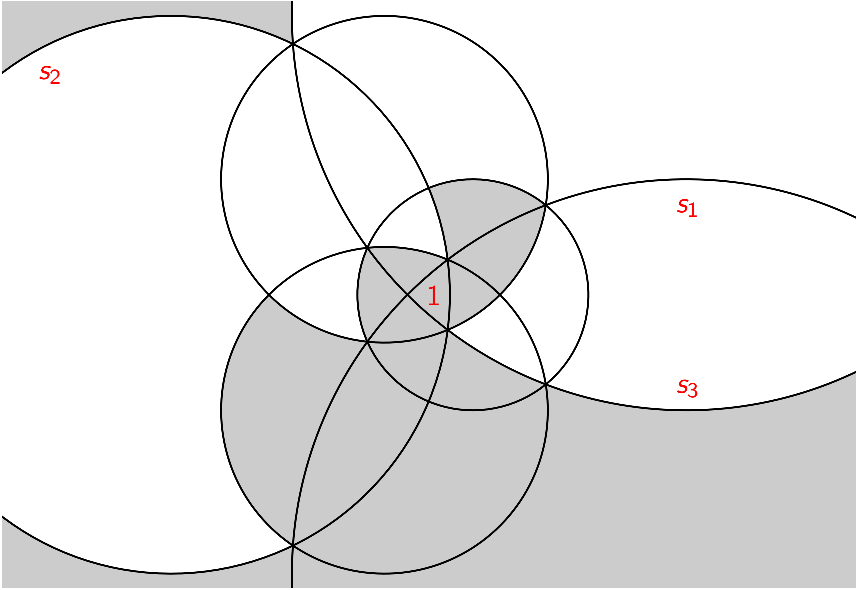
Example: c-Sortable elements, c-Cambrian fans, shards



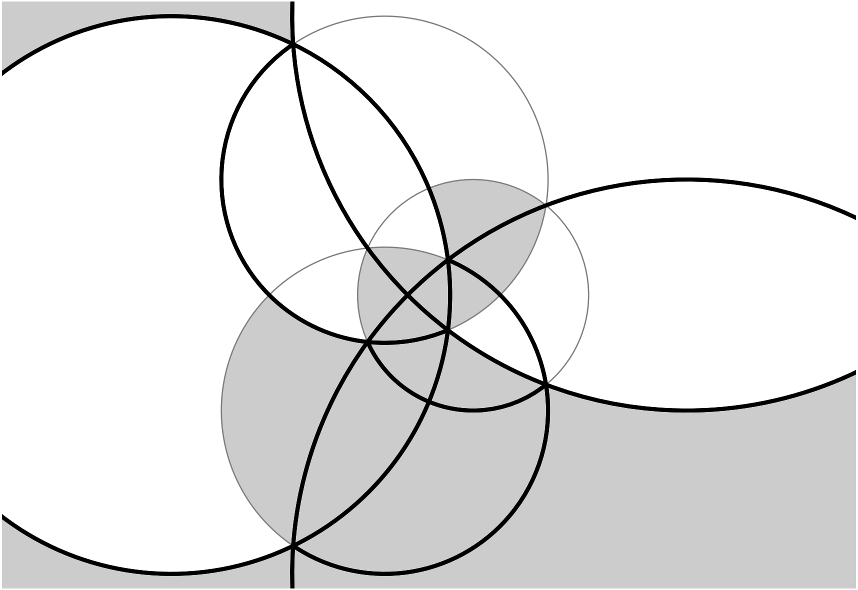
Example: c-Sortable elements, c-Cambrian fans, shards



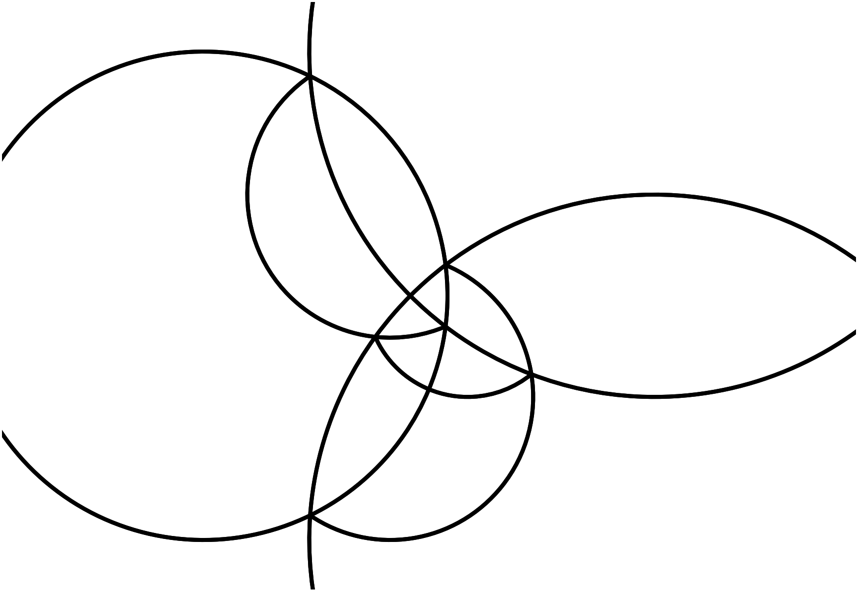
Example: c-Sortable elements, c-Cambrian fans, shards



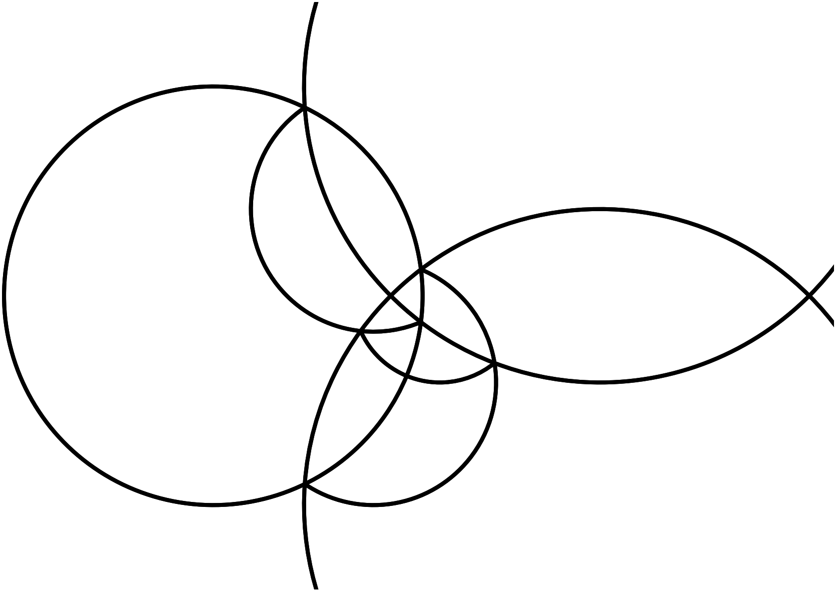
Example: c-Sortable elements, c-Cambrian fans, shards



Example: c-Sortable elements, c-Cambrian fans, shards



Example: c-Sortable elements, c-Cambrian fans, shards



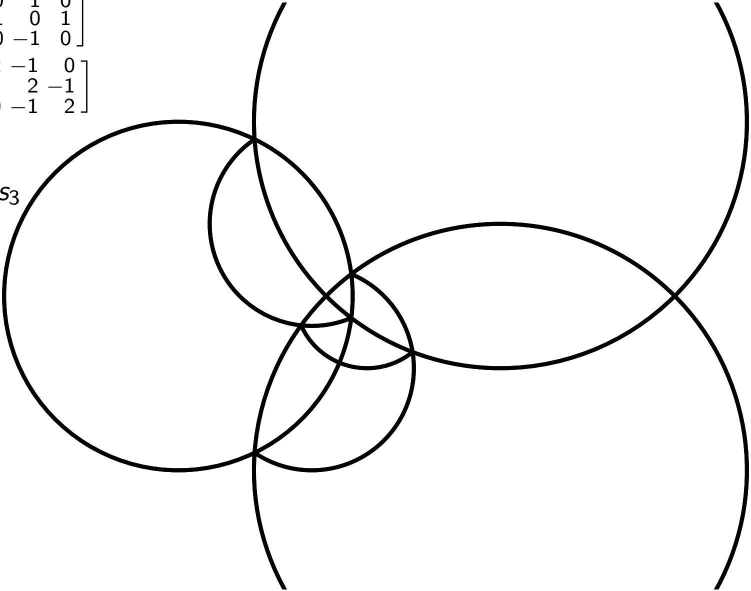
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

$c = s_1 s_2 s_3$



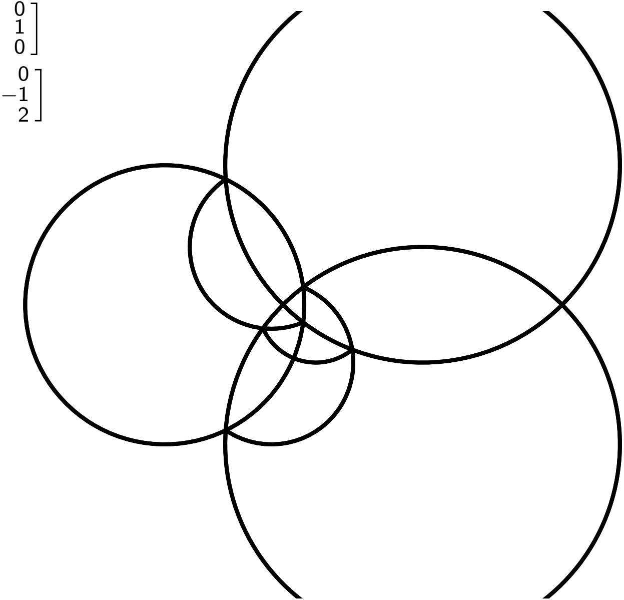
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

$c = s_1 s_2 s_3$



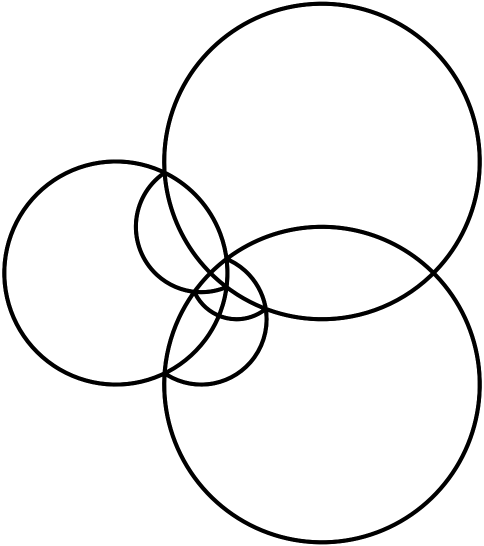
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

$c = s_1 s_2 s_3$



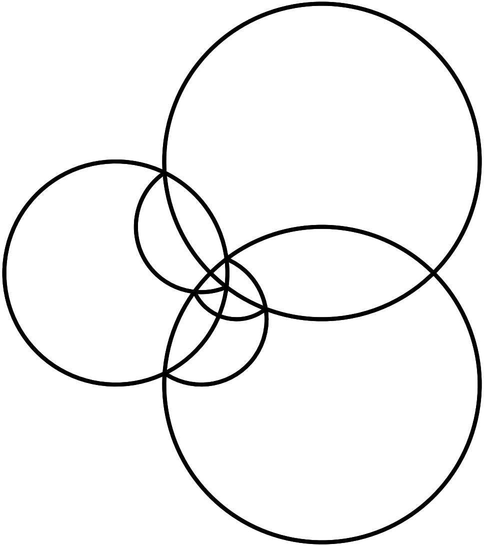
Example: c-Sortable elements, c-Cambrian fans, shards

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$W = S_4$

$c = s_1 s_2 s_3$



Affine type

Two equivalent definitions (for 3×3 and larger):

- B is mutation equivalent to an acyclic exchange matrix whose associated Cartan matrix is of affine type.
- Infinitely many cluster variables, but linear growth.

We can model this completely using the

- Affine almost positive roots model (R.-Stella), and the
- Doubled Cambrian fan construction (R.-Speyer).

In particular, the models

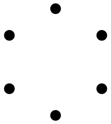
- Build the cluster scattering diagram/fans explicitly using shards (with Stella),
- Construct all theta functions (with Stella, in progress), and
- Construct the return function ret_λ for λ in the direction of the limiting ray (in progress).

Affine type details: Almost positive Schur roots

- Start with a finite root system

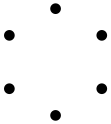
Affine type details: Almost positive Schur roots

- Start with a finite root system (A_2 in this case)



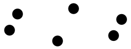
Affine type details: Almost positive Schur roots

- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$)

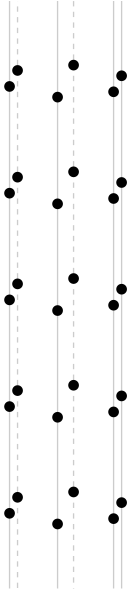


Affine type details: Almost positive Schur roots

- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$)

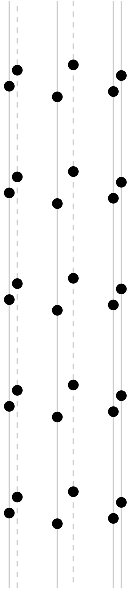


Affine type details: Almost positive Schur roots



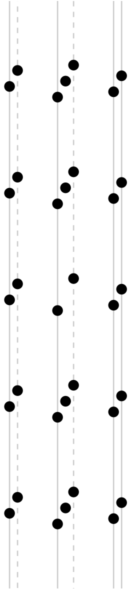
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$)

Affine type details: Almost positive Schur roots



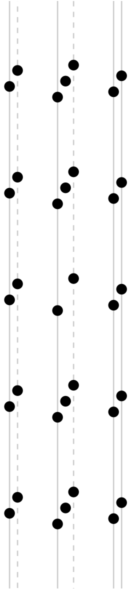
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)

Affine type details: Almost positive Schur roots



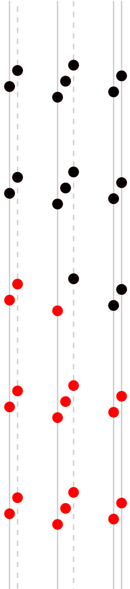
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)

Affine type details: Almost positive Schur roots



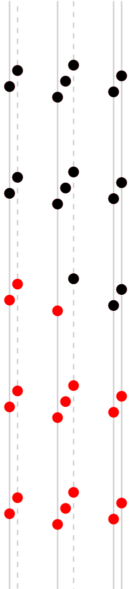
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.

Affine type details: Almost positive Schur roots



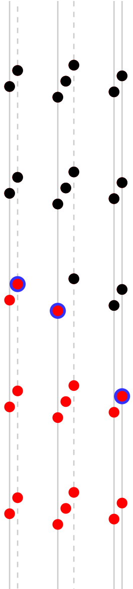
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.

Affine type details: Almost positive Schur roots



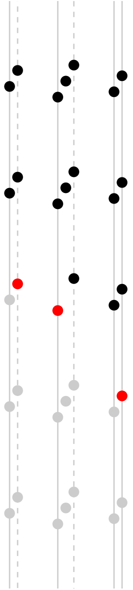
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.

Affine type details: Almost positive Schur roots



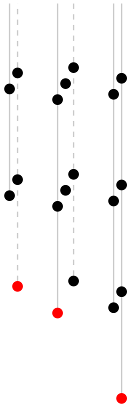
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.

Affine type details: Almost positive Schur roots



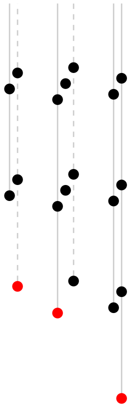
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.

Affine type details: Almost positive Schur roots



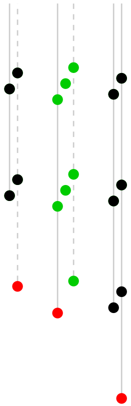
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.

Affine type details: Almost positive Schur roots



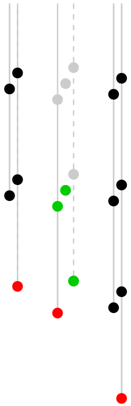
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



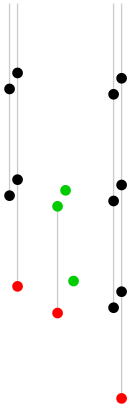
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



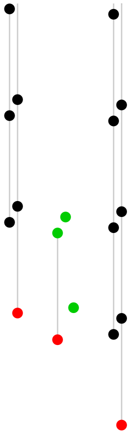
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



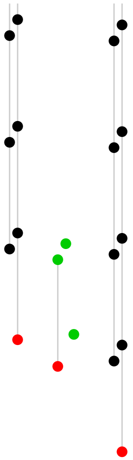
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



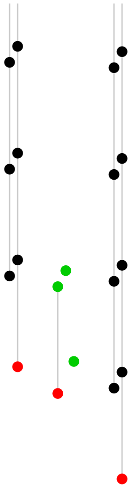
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



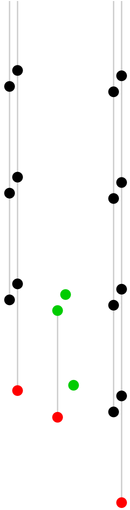
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



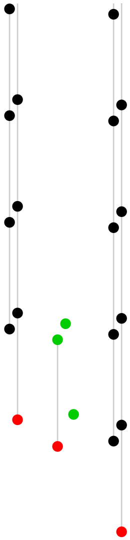
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



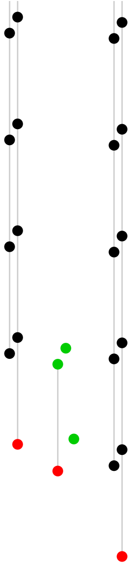
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



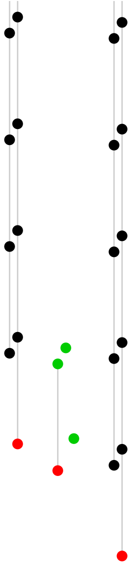
- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Affine type details: Almost positive Schur roots



- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

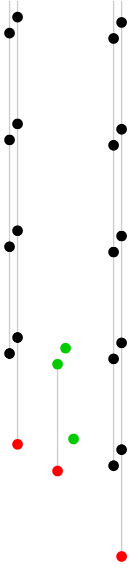
Affine type details: Almost positive Schur roots



- Start with a finite root system (A_2 in this case)
- Construct an affine root system ($\tilde{A}_2$) (including imaginary roots)
- Divide them into positive and negative roots.
- Restrict to the almost positive roots.
- Restrict further to the almost positive Schur roots.

Observation: The real roots limit, from two sides, to the direction of the imaginary root.
(This is true in general affine type.)

A.p. Schur roots and clus. scattering diagrams (R., Stella)



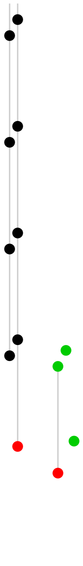
Almost positive Schur roots appear in two ways in cluster scattering diagrams:

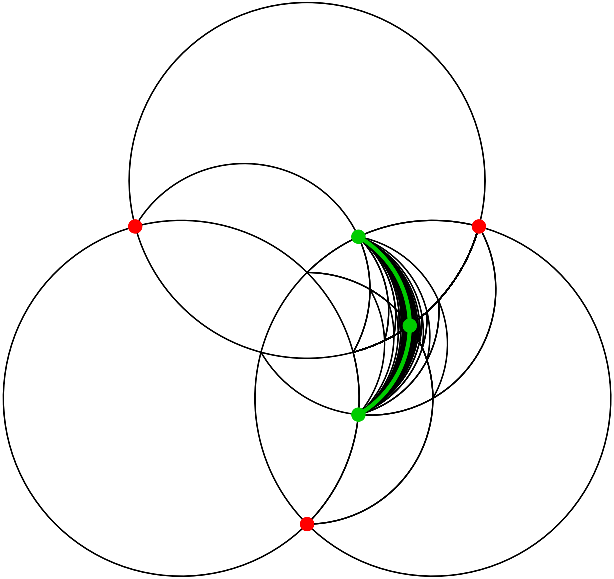
1. They index the rays of the fan.
    - There is a piecewise-linear bijection from real a.p. Schur roots to **g**-vectors of cluster variables.
    - The one imaginary a.p. Schur root corresponds to a "limiting ray" outside the cluster fan.
    - There is a combinatorial description of "compatibility" similar to finite type.
2. {Positive Schur roots}

   $= \{$normal vectors to walls$\}$.

    - Exactly one wall normal to each root.
    - There is one limiting "imaginary wall".
    - All other walls are walls of the cluster fan.

Affine scattering fan example





Affine type details: Doubled Cambrian fans (R., Speyer)

Cambrian fans can be defined even for infinite Coxeter groups. In every case, the Cambrian fan is a subfan of the cluster fan.

In the infinite case, they are not the whole cluster fan.

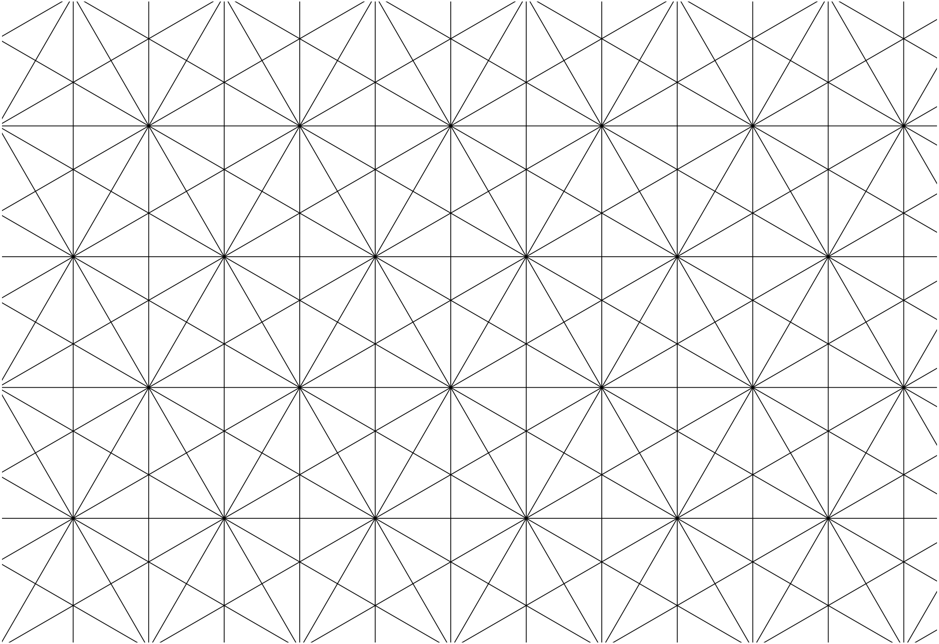
Affine type details: Doubled Cambrian fans (R., Speyer)

Cambrian fans can be defined even for infinite Coxeter groups. In every case, the Cambrian fan is a subfan of the cluster fan.

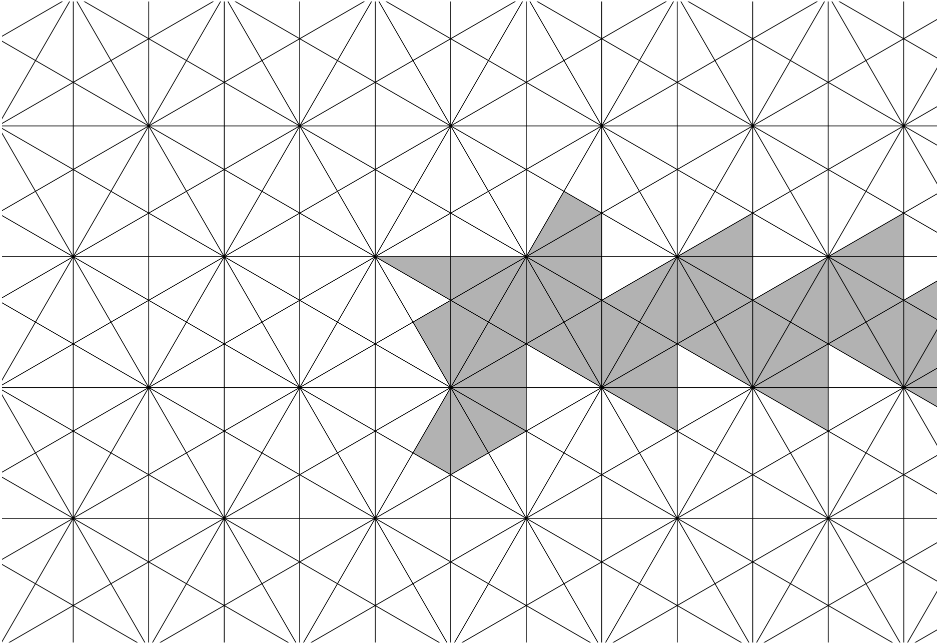
In the infinite case, they are not the whole cluster fan.

In the affine case, the entire cluster fan can be constructed from two overlapping pieces: The c-Cambrian fan and the antipodal opposite of the c^{-1}-Cambrian fan.

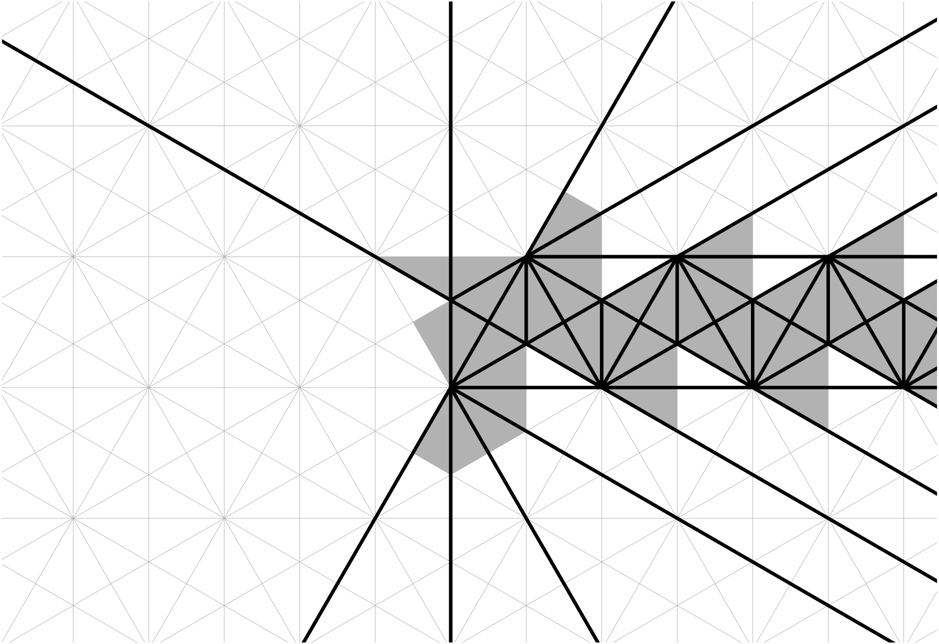
An affine Cambrian fan



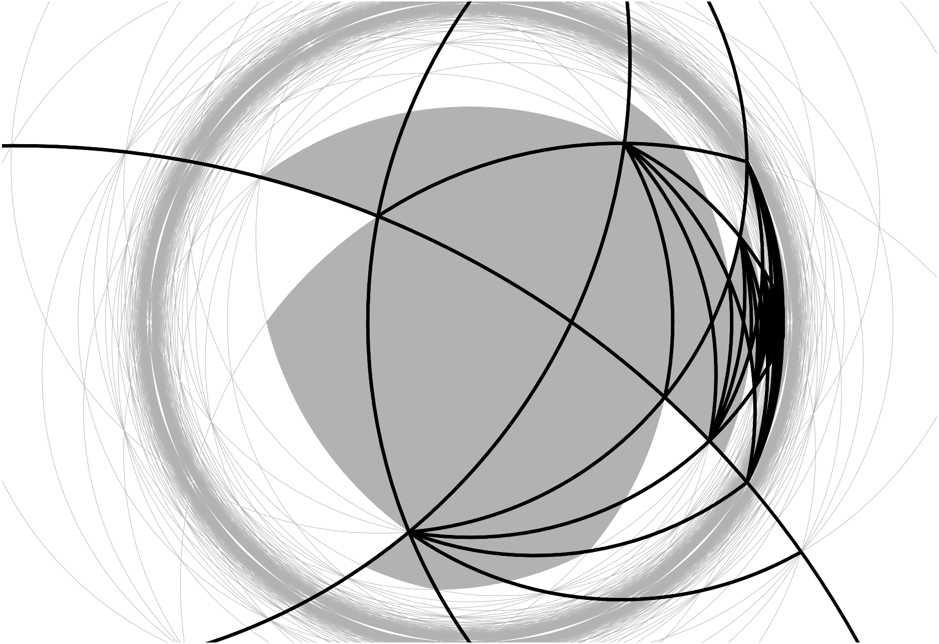
An affine Cambrian fan



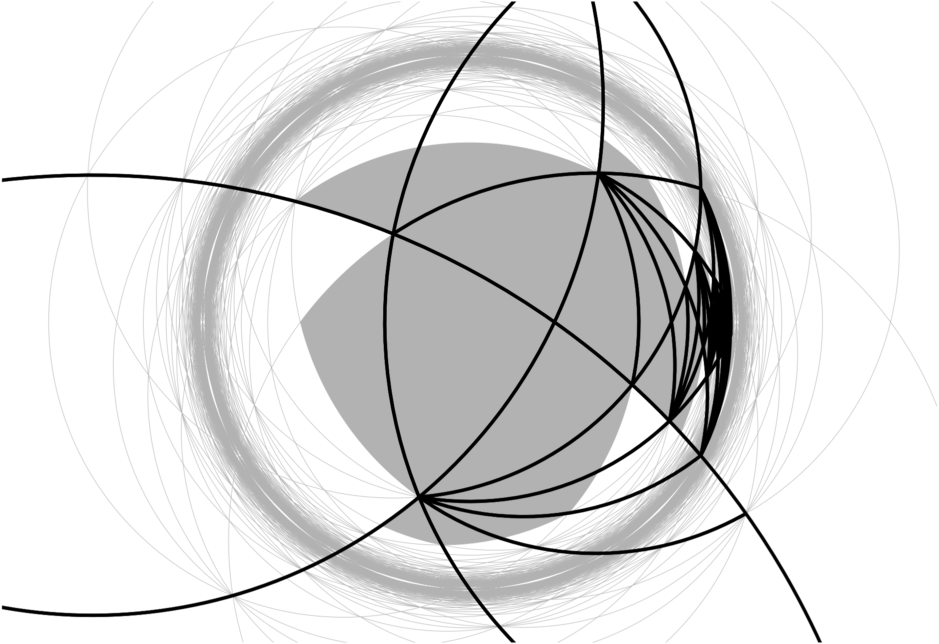
An affine Cambrian fan



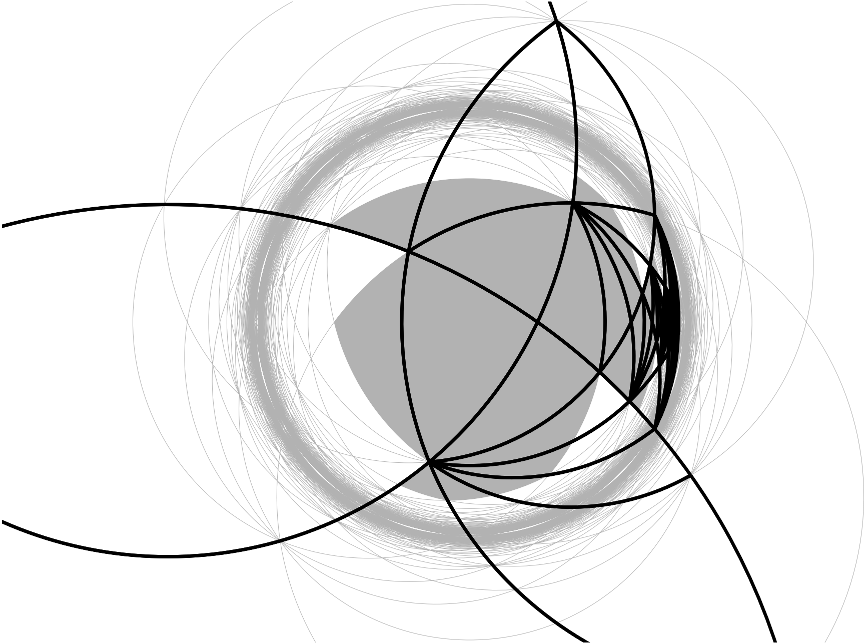
An affine Cambrian fan



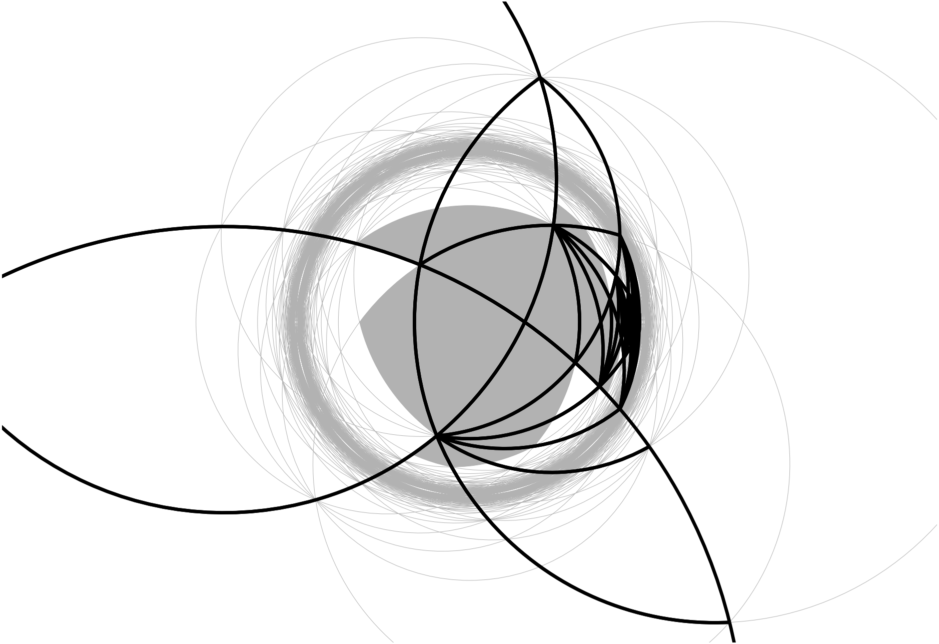
An affine Cambrian fan



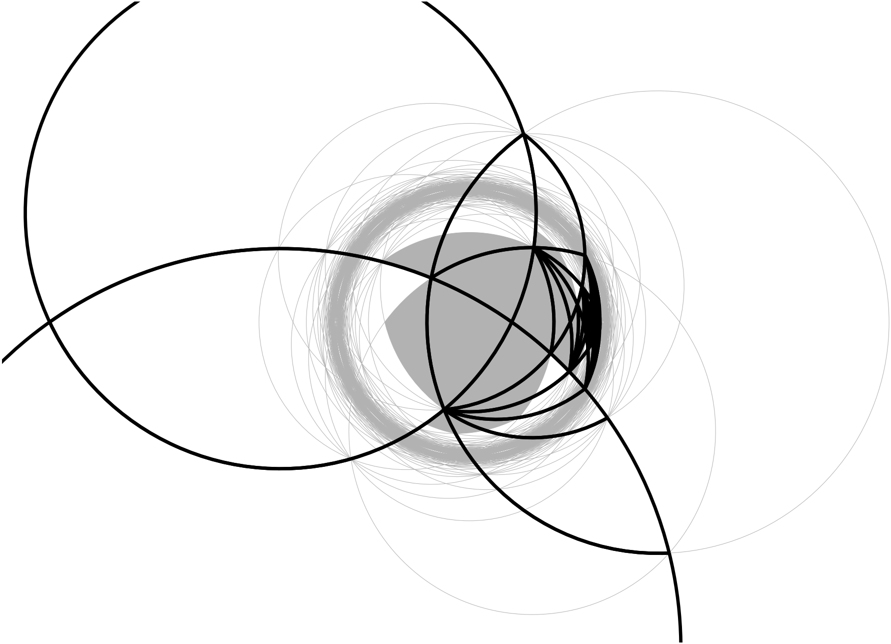
An affine Cambrian fan



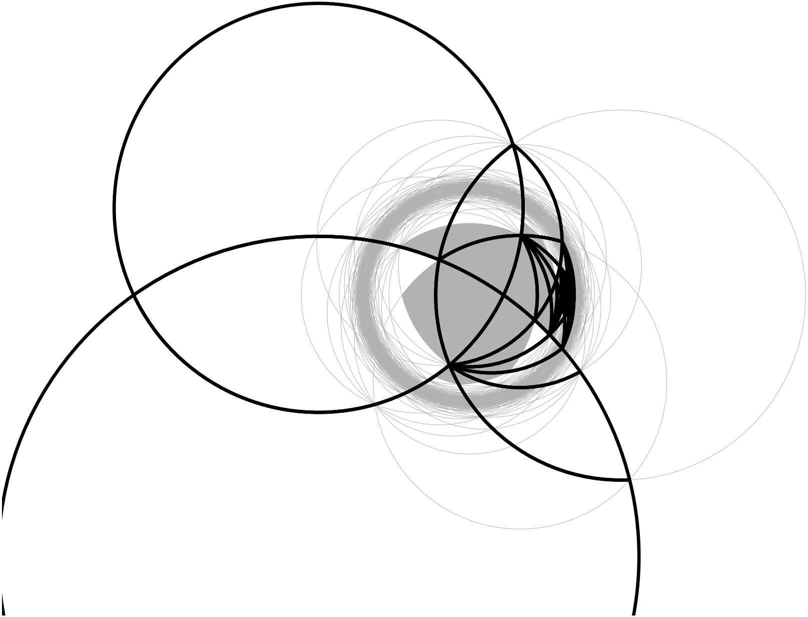
An affine Cambrian fan



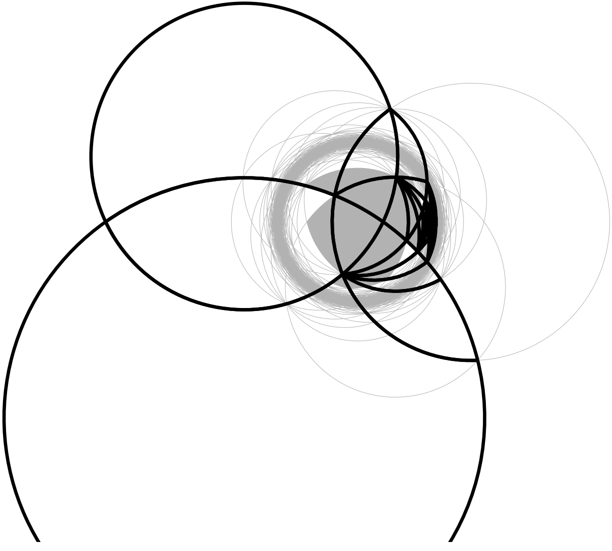
An affine Cambrian fan



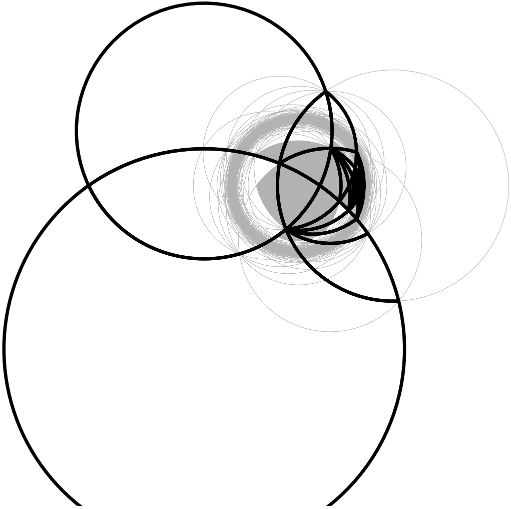
An affine Cambrian fan



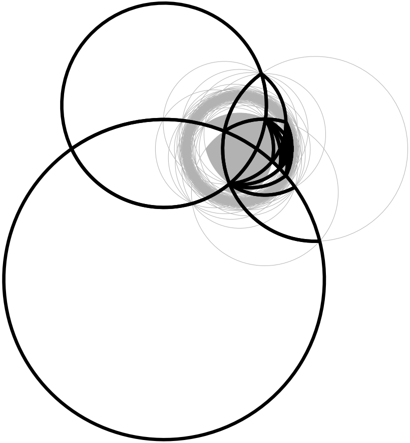
An affine Cambrian fan



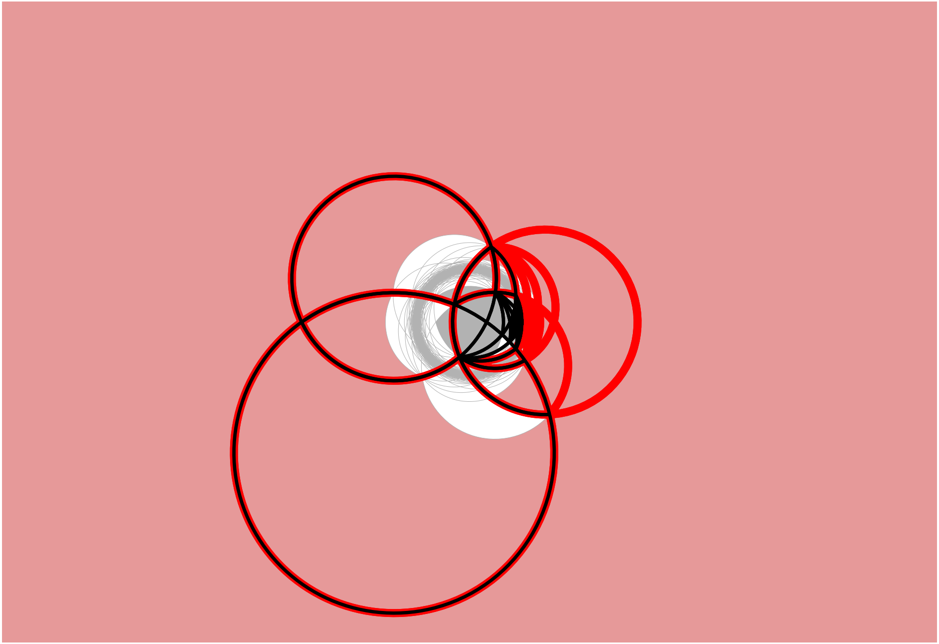
An affine Cambrian fan



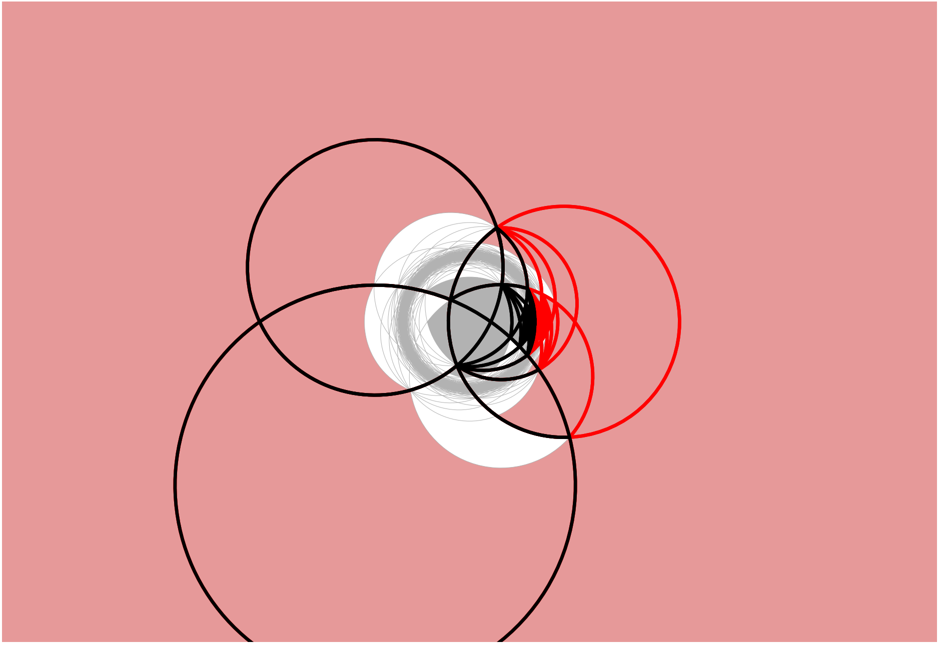
An affine Cambrian fan



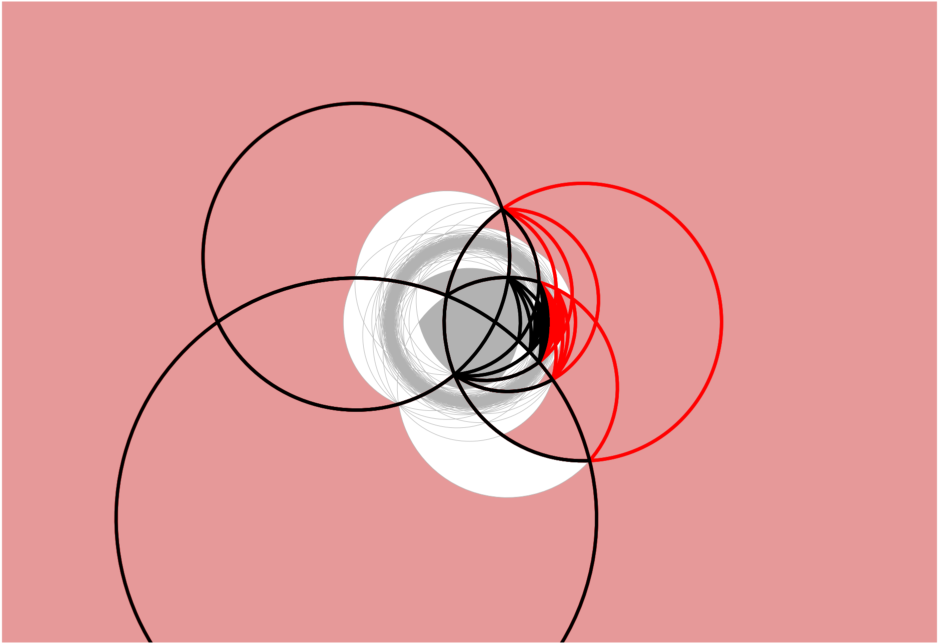
An affine doubled Cambrian fan



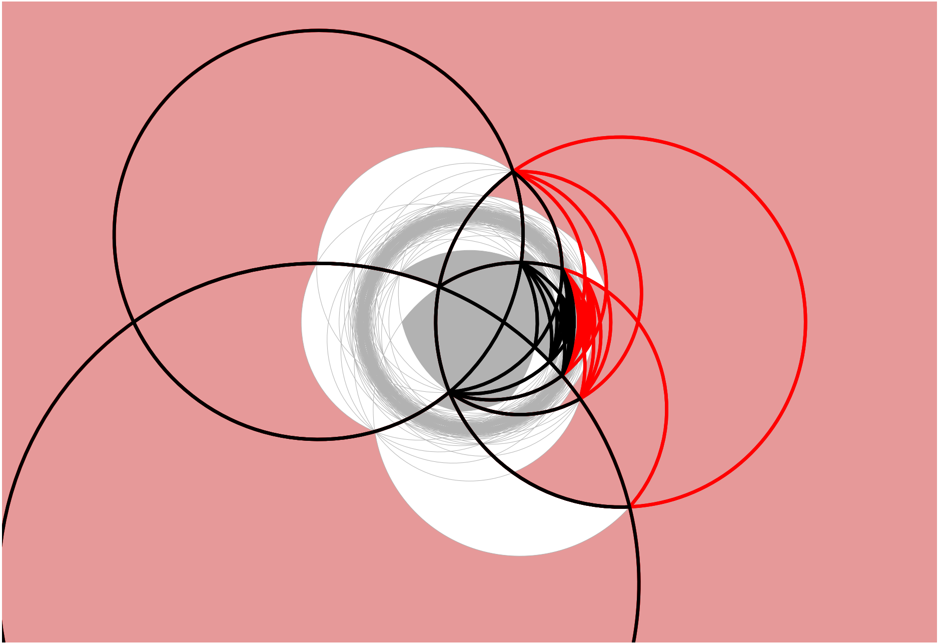
An affine doubled Cambrian fan



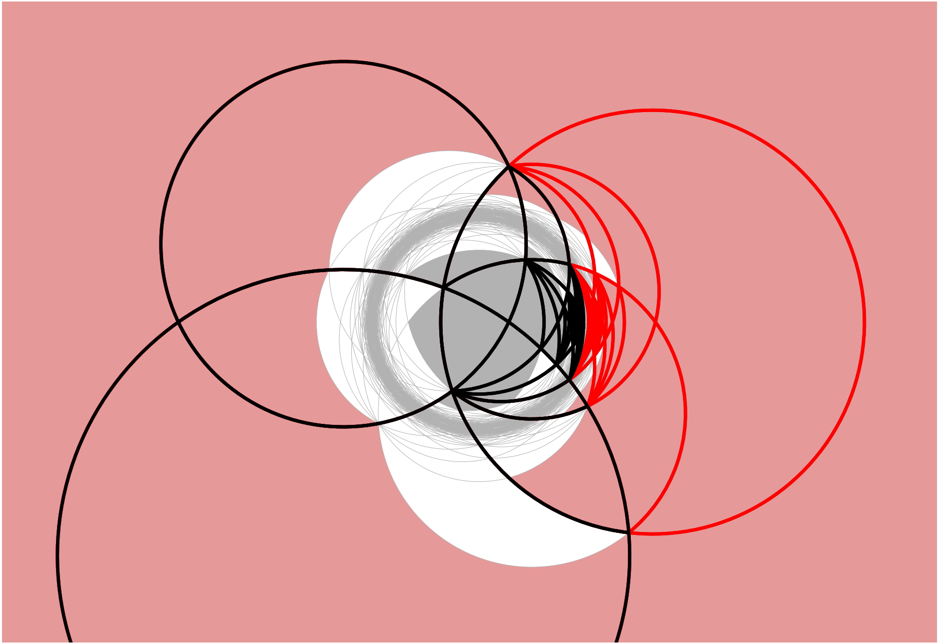
An affine doubled Cambrian fan



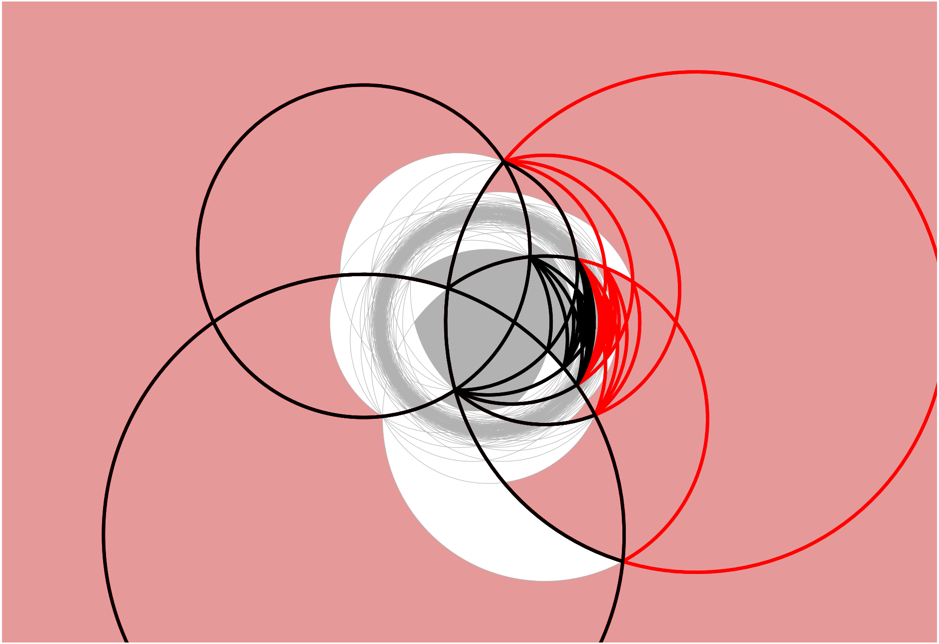
An affine doubled Cambrian fan



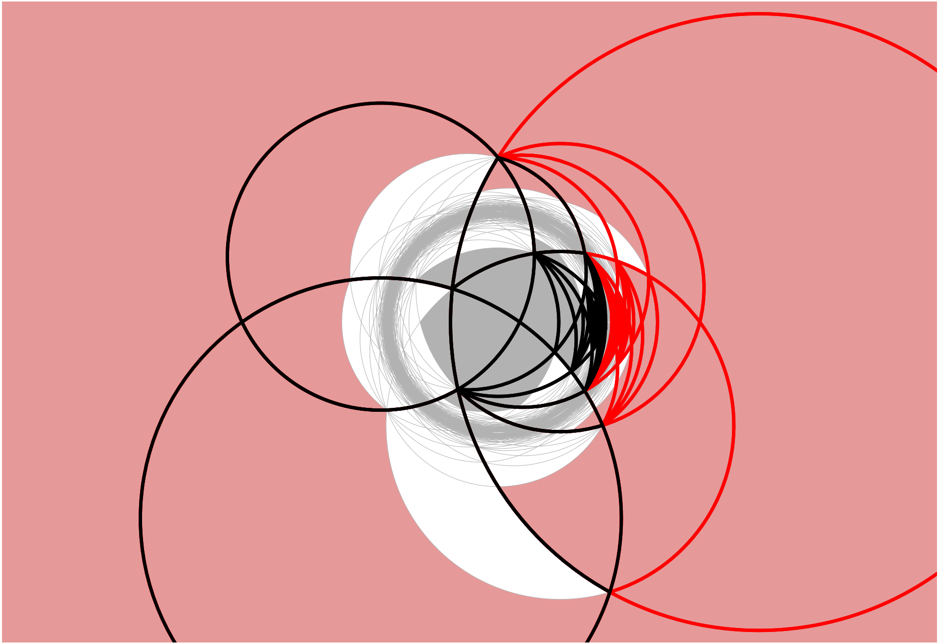
An affine doubled Cambrian fan



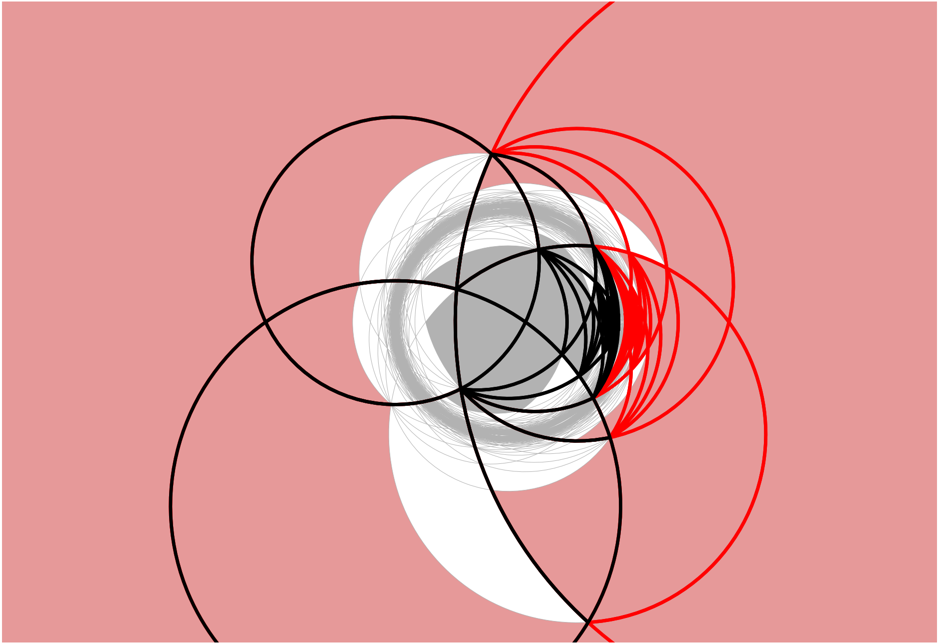
An affine doubled Cambrian fan



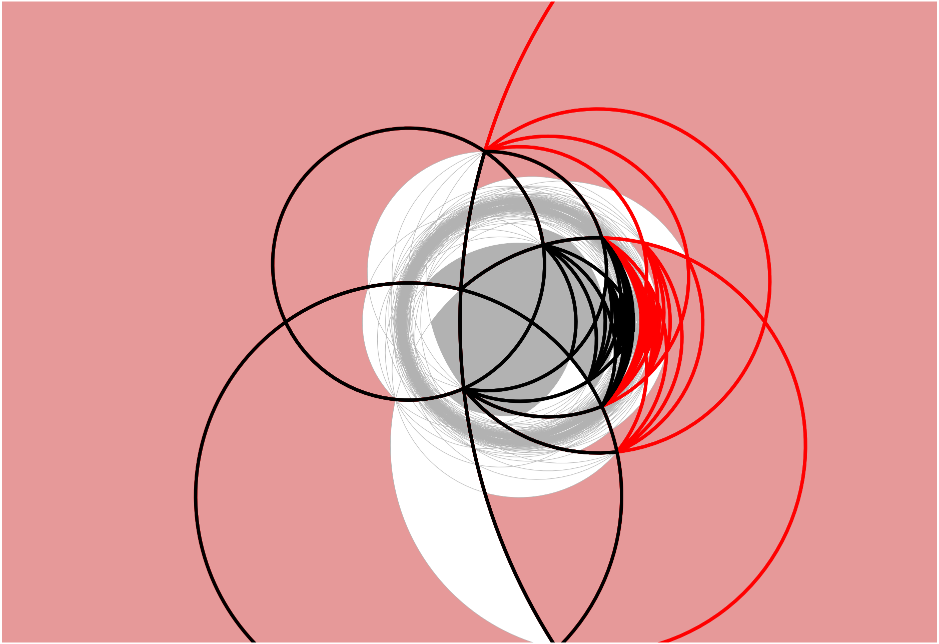
An affine doubled Cambrian fan



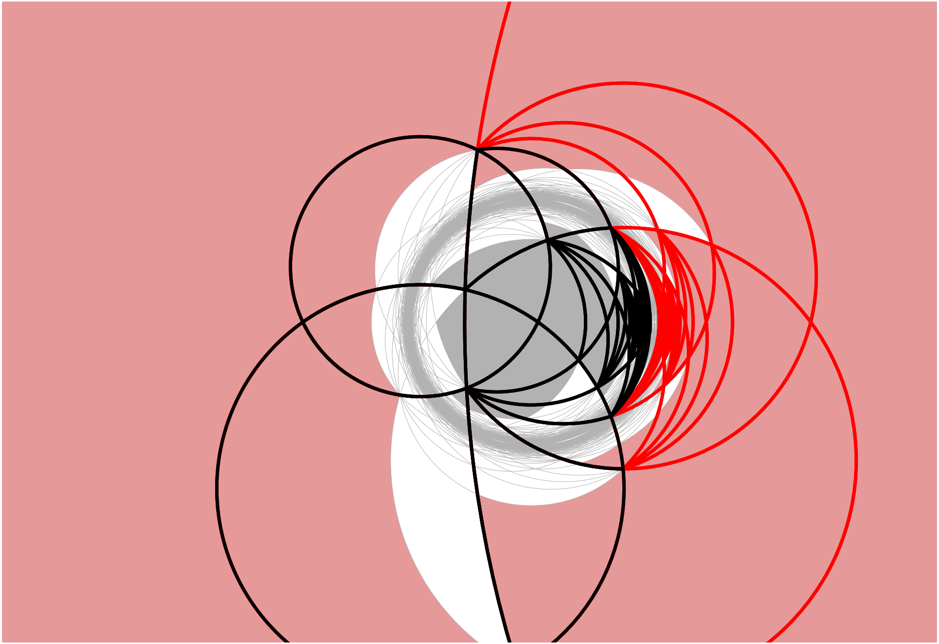
An affine doubled Cambrian fan



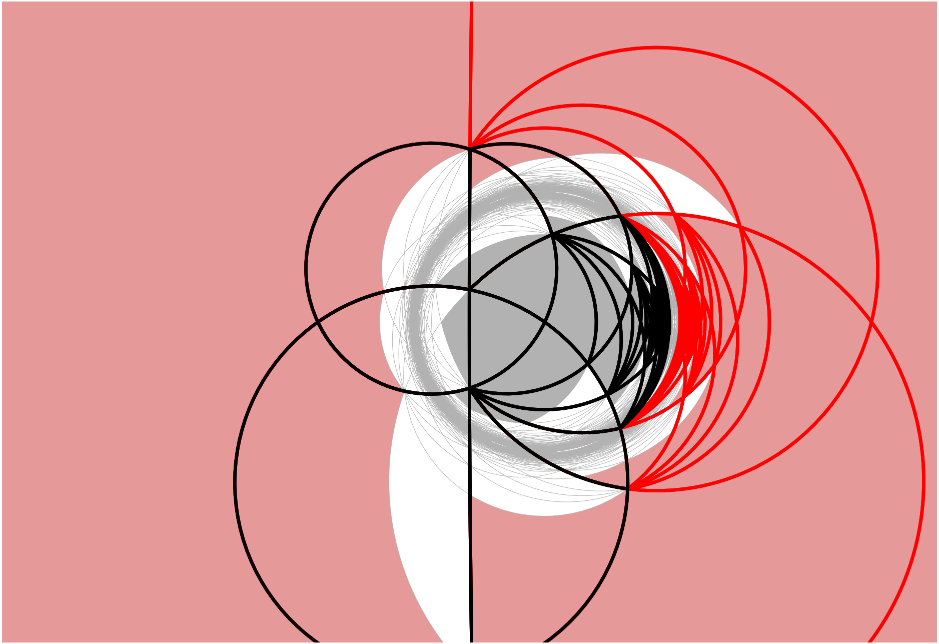
An affine doubled Cambrian fan



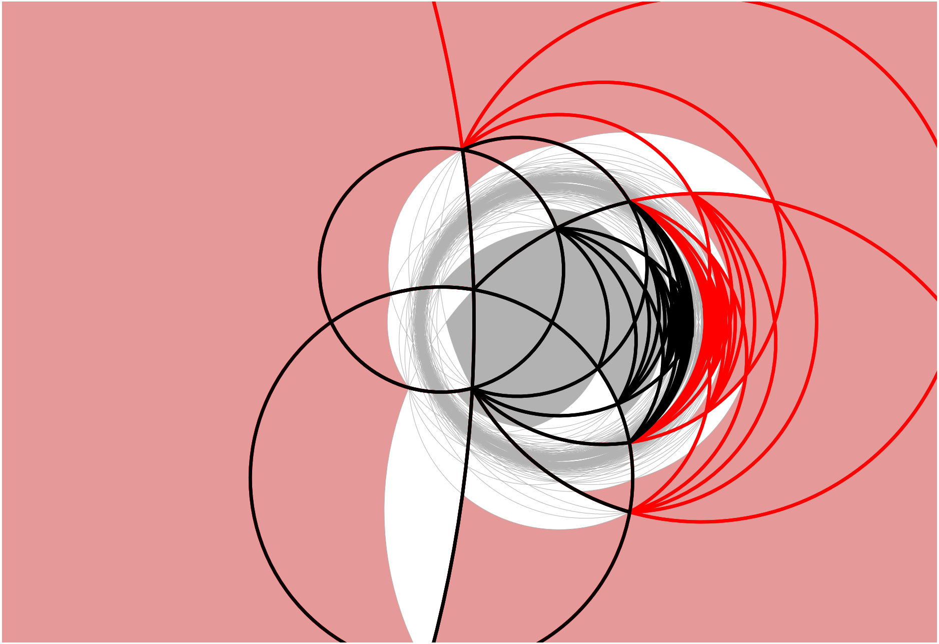
An affine doubled Cambrian fan



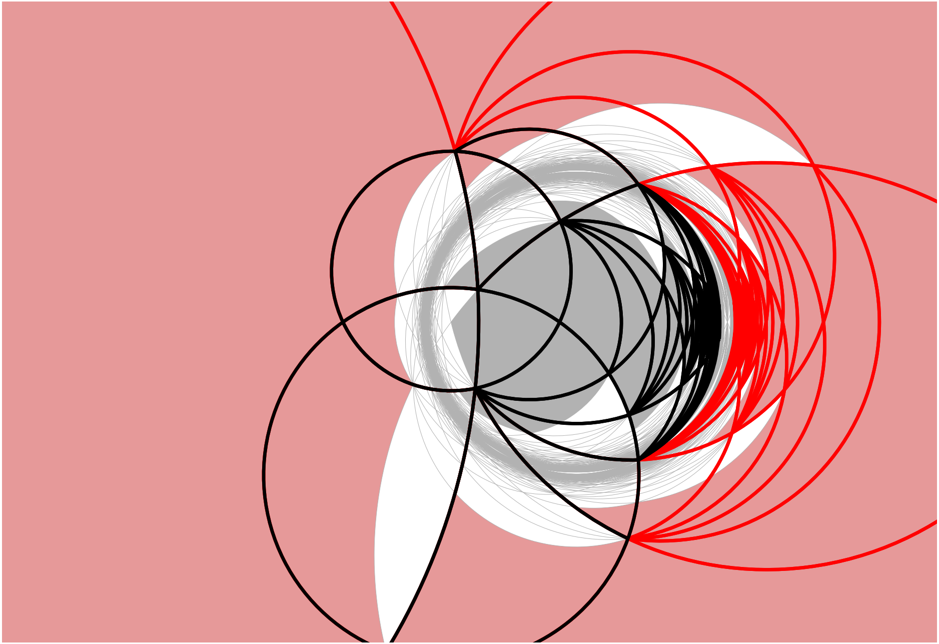
An affine doubled Cambrian fan



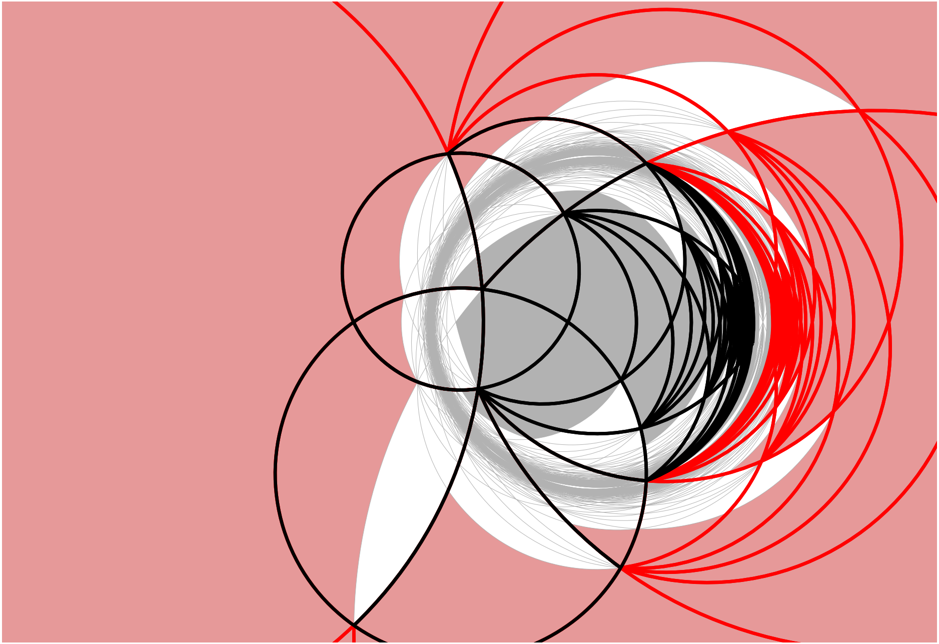
An affine doubled Cambrian fan



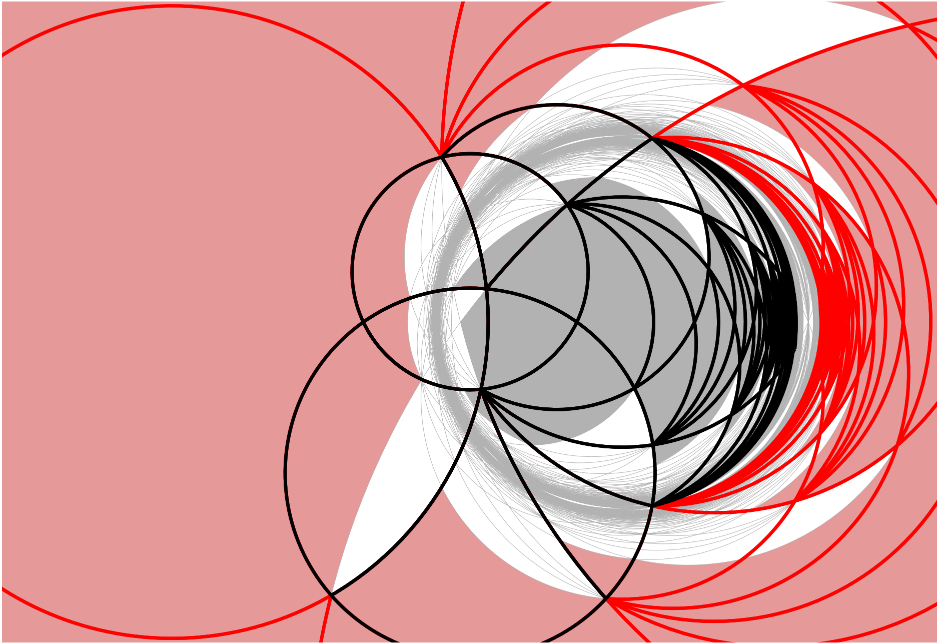
An affine doubled Cambrian fan



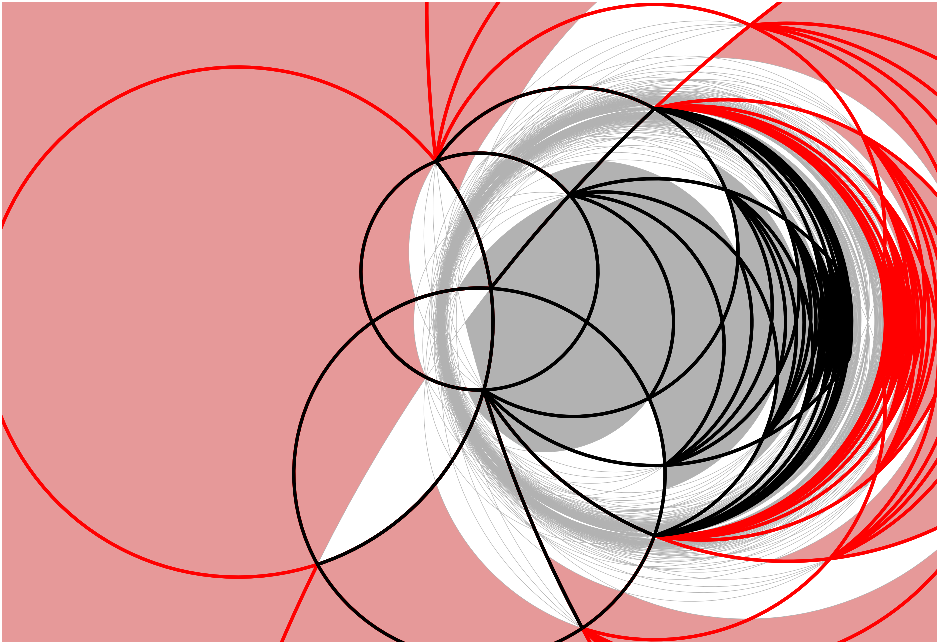
An affine doubled Cambrian fan



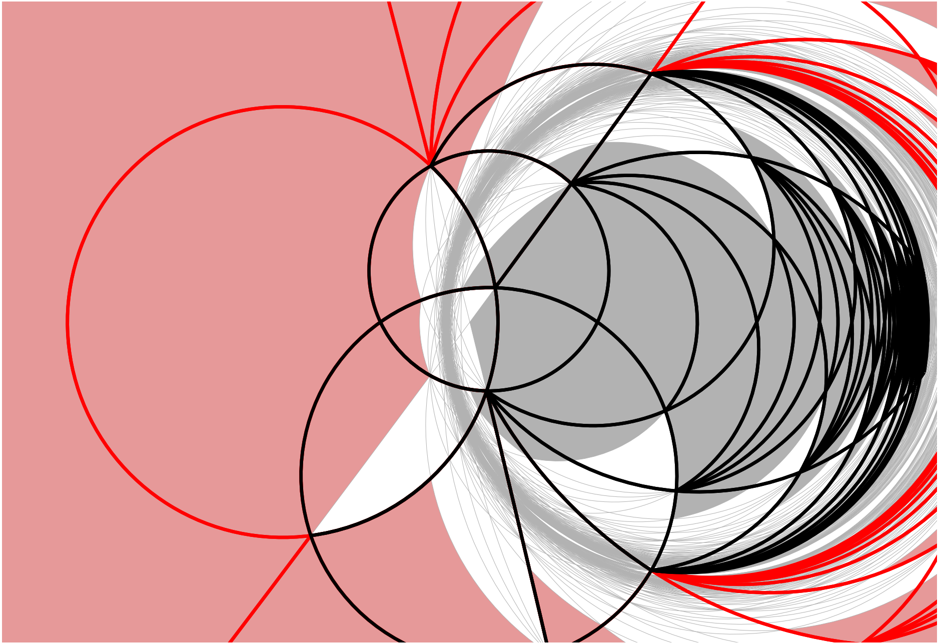
An affine doubled Cambrian fan



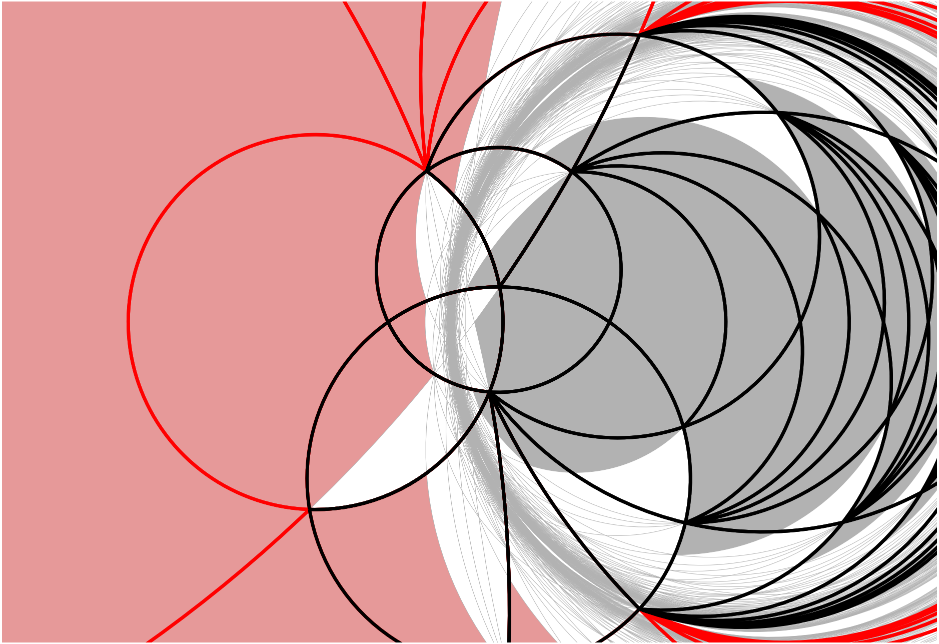
An affine doubled Cambrian fan



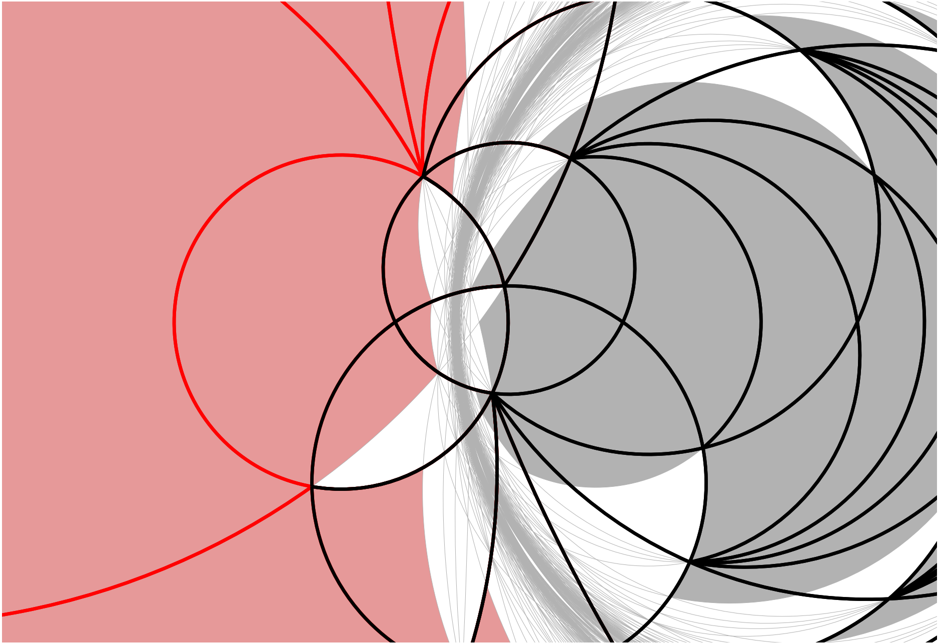
An affine doubled Cambrian fan



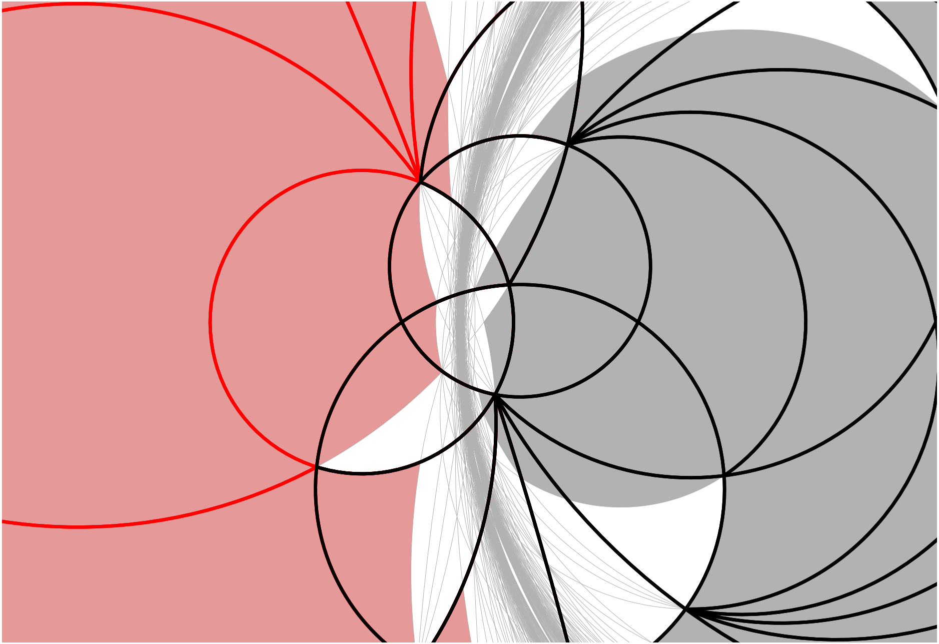
An affine doubled Cambrian fan



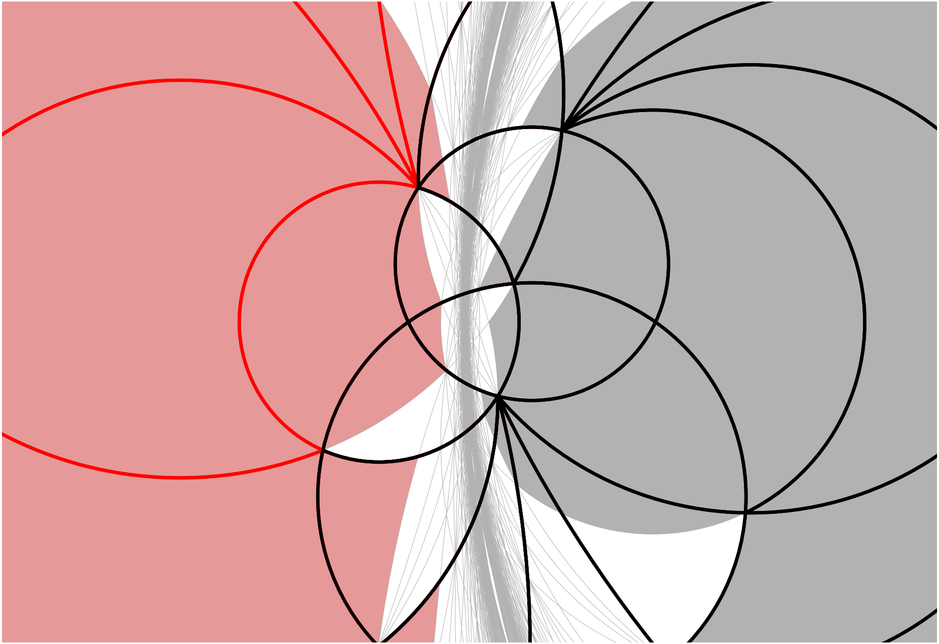
An affine doubled Cambrian fan



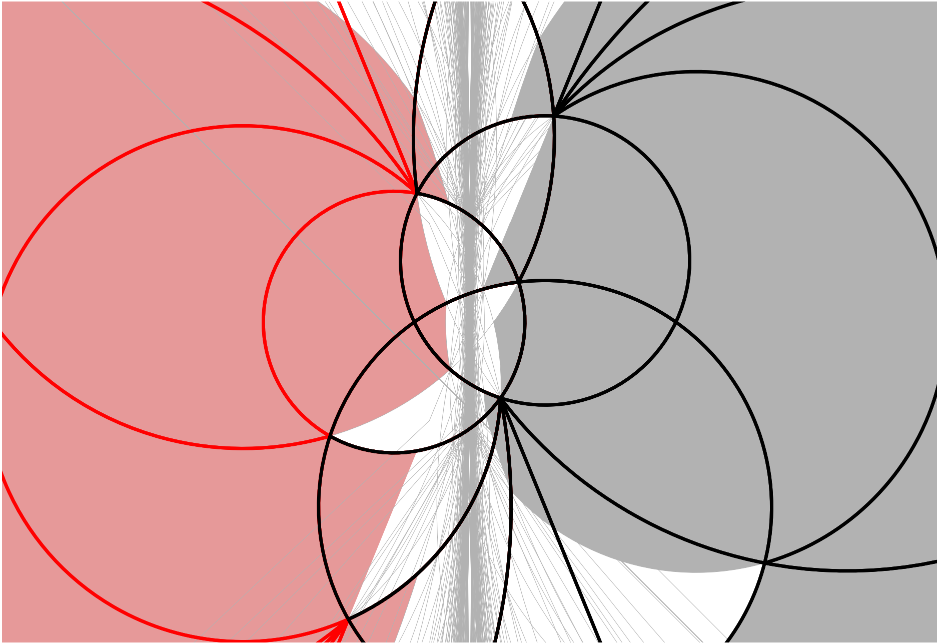
An affine doubled Cambrian fan



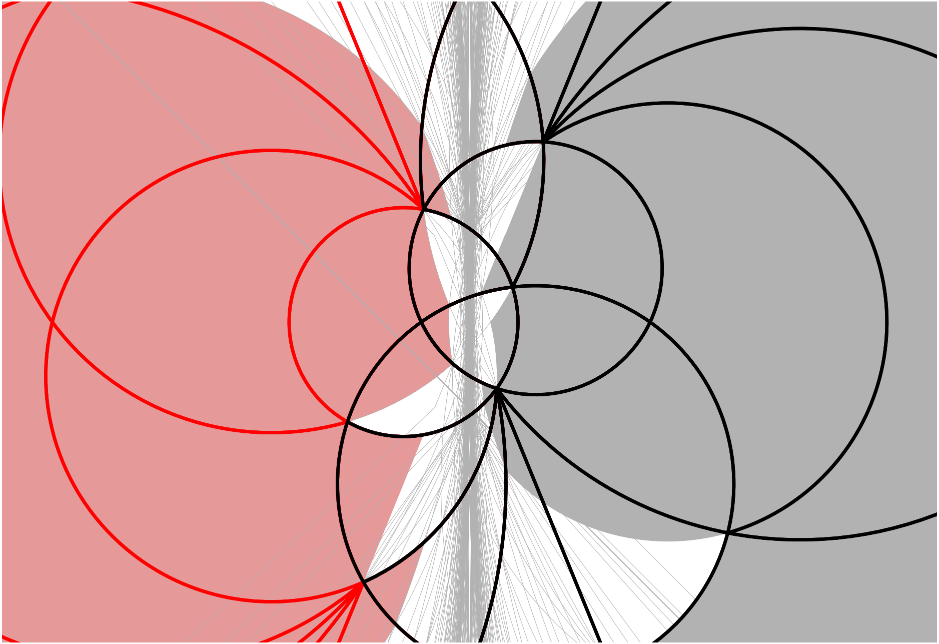
An affine doubled Cambrian fan



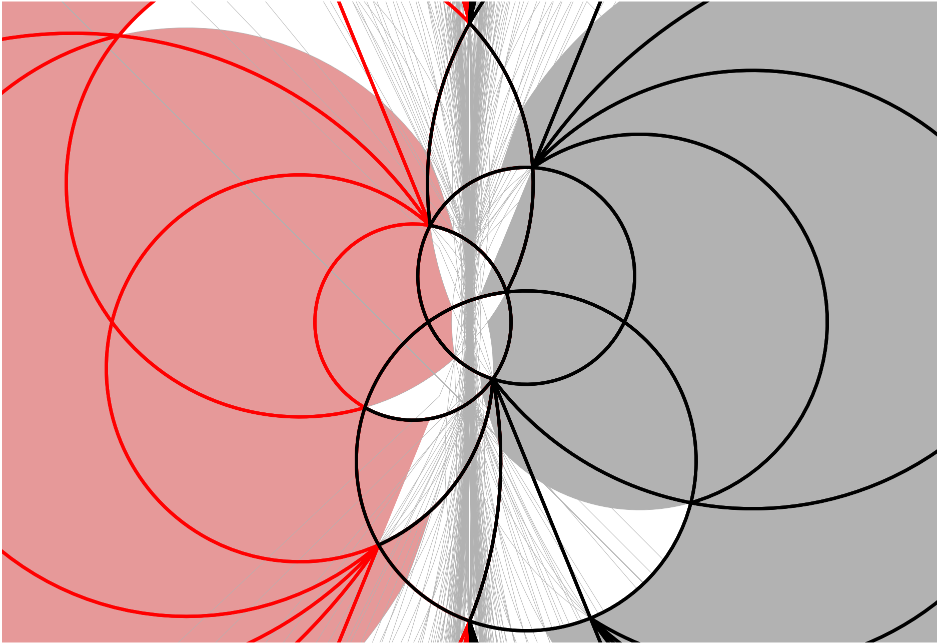
An affine doubled Cambrian fan



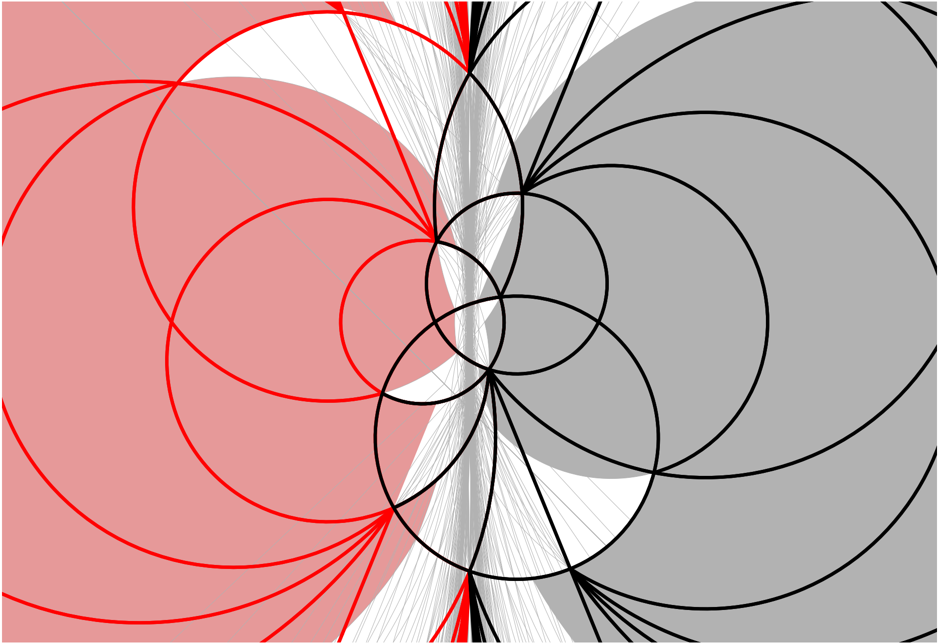
An affine doubled Cambrian fan



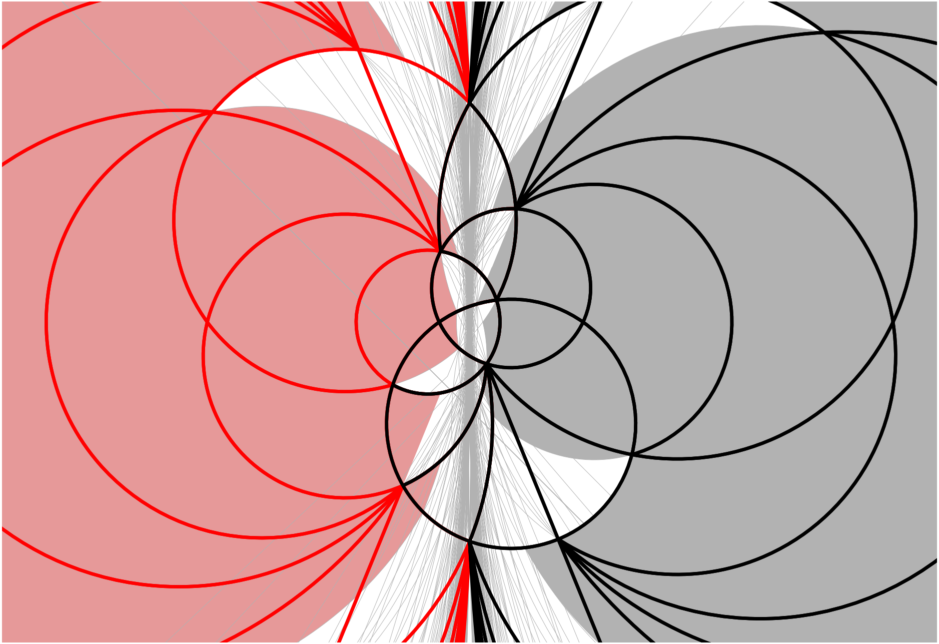
An affine doubled Cambrian fan



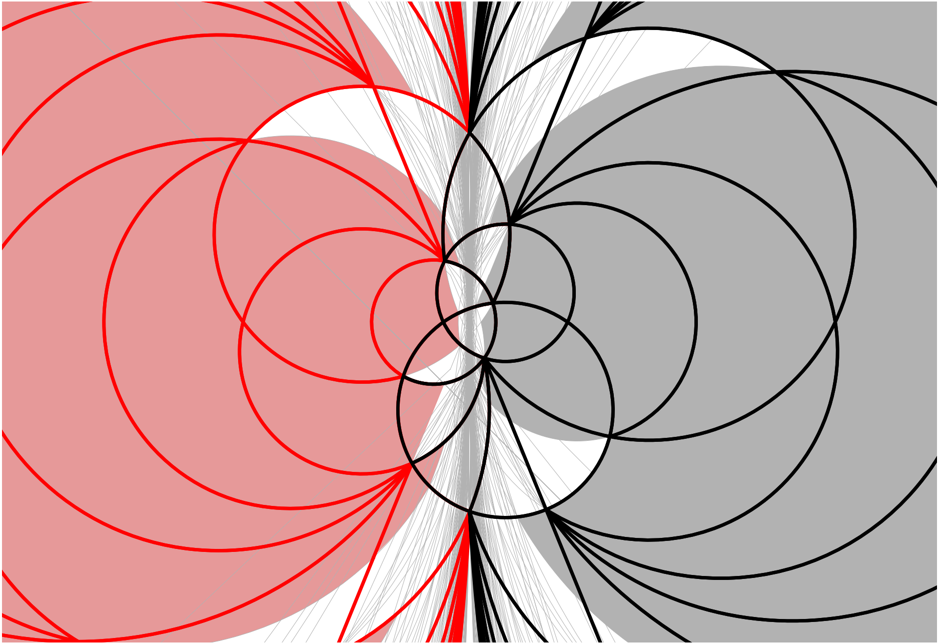
An affine doubled Cambrian fan



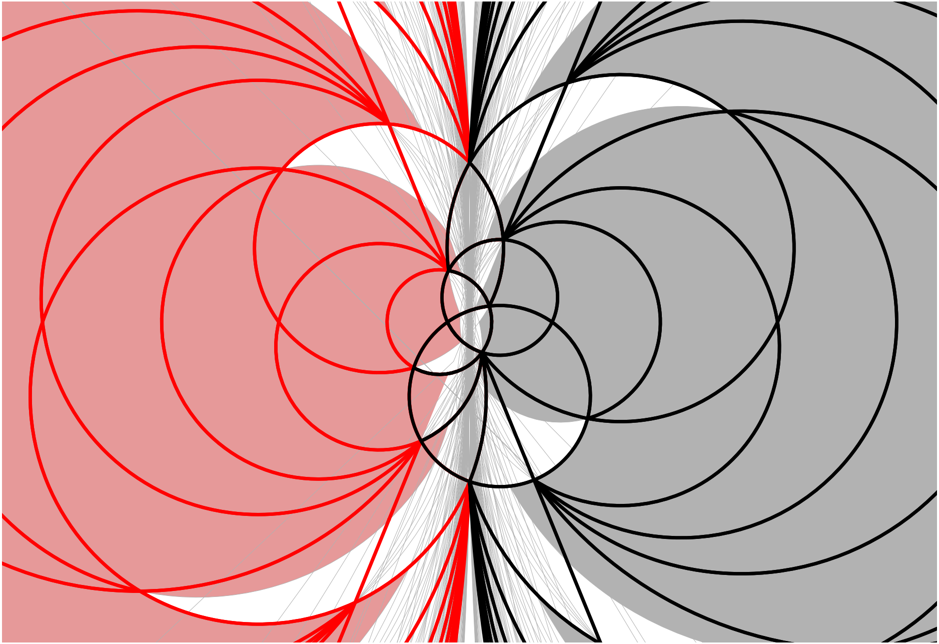
An affine doubled Cambrian fan



An affine doubled Cambrian fan



An affine doubled Cambrian fan



Theta functions and return functions for the limiting ray

Take λ to be the shortest integer vector in the limiting ray. Here are some "Theorems in progress" (some with Stella):

- We give an explicit formula for ϑ_λ as a linear combination of three terms.
  - Two terms are of the form
    $$(\text{monomial in the } y_i\text{'s}) \cdot (\text{cluster variable})$$
  - One term is a product of cluster variables.
- For $k \geq 1$, $\vartheta_{k\lambda}$ is something like a Chebshev polynomial in ϑ_λ.
- $\mathrm{ret}_\lambda = \dfrac{\vartheta_\lambda + \sqrt{\vartheta_\lambda^2 - 4\hat{y}^\delta x^{2\lambda}}}{2}$.
- For a particular sequence $\ldots, x_{-1}, x_0, x_1, \ldots$ of cluster variables and a particular choice of exponents $a_1, \ldots, a_n \in \mathbb{Z}$,

$$\mathrm{ret}_\lambda = \lim_{k\to\infty} \prod_{i=1}^{n} x_{kn+i}^{a_i} = \lim_{k\to-\infty} \prod_{i=1}^{n} x_{kn+i}^{-a_i}.$$

Surfaces

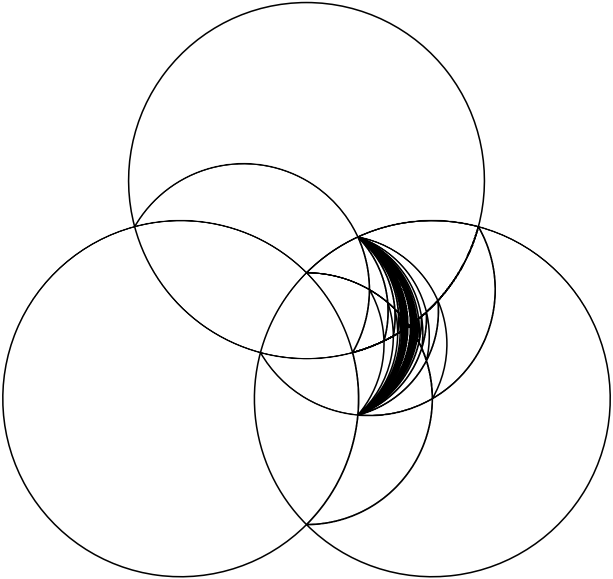
There is no (cluster-algebraically) intrinsic definition here. Rather, there is a model involving triangulated surfaces that covers many, but certainly not all, cluster algebras.

We start with an orientable surface, triangulated using n arcs. The vertices of the triangulation are called marked points.

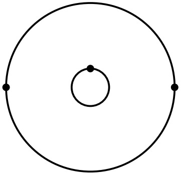
- Cluster variables correspond to non-self-intersecting arcs connecting marked points.
- Other theta functions correspond to non-self-intersecting closed curves.
- “Adjacency” of these functions is non-intersection.

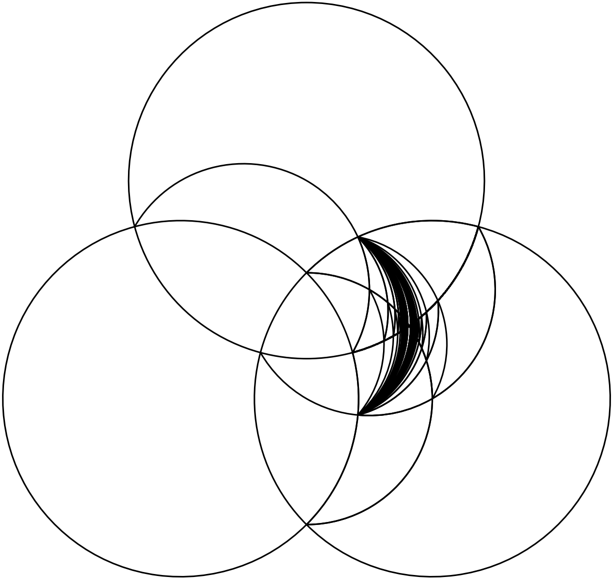
Return functions ret_λ have not been studied here.

Surface example

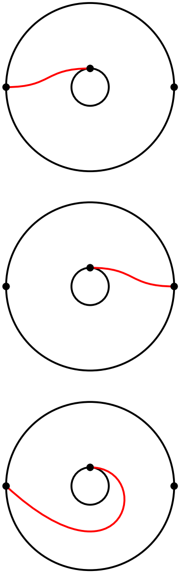


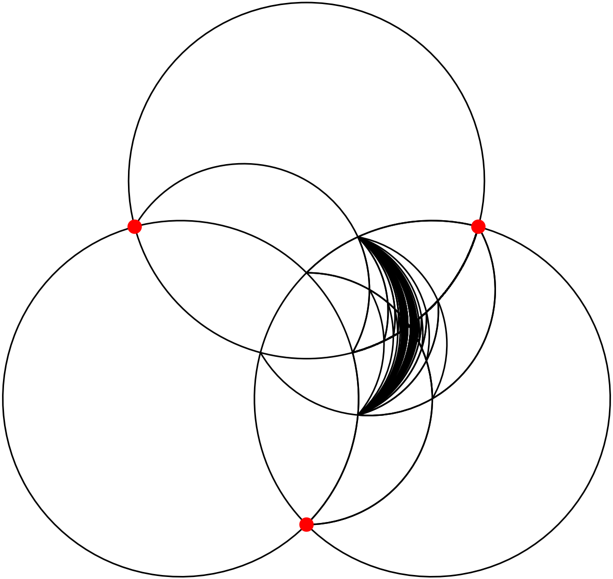
Surface example



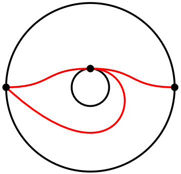


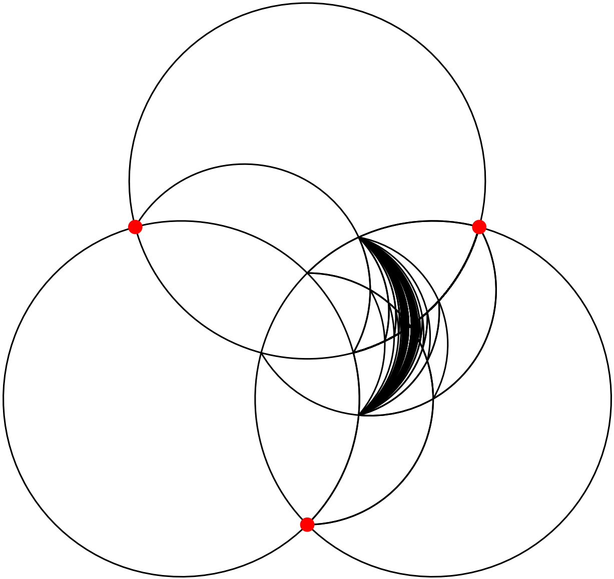
Surface example



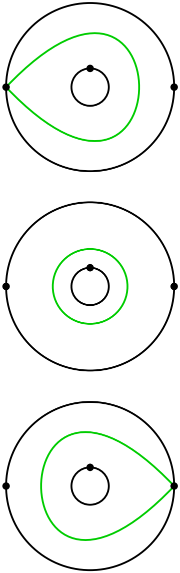


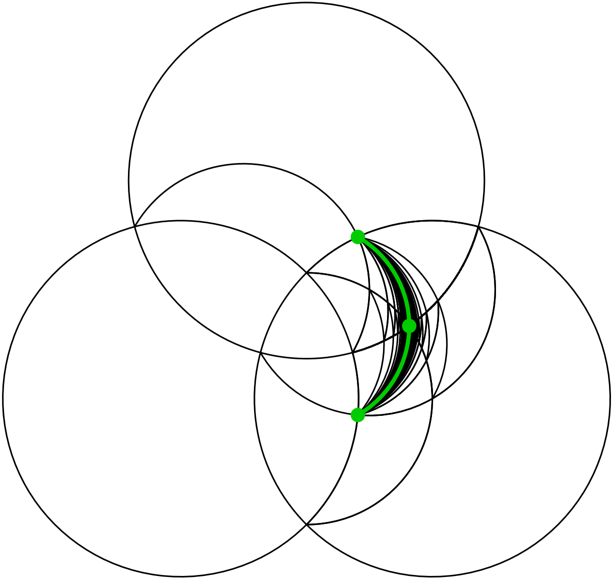
Surface example



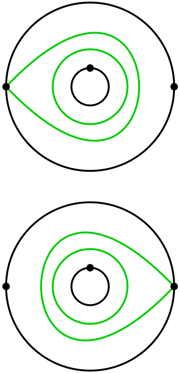


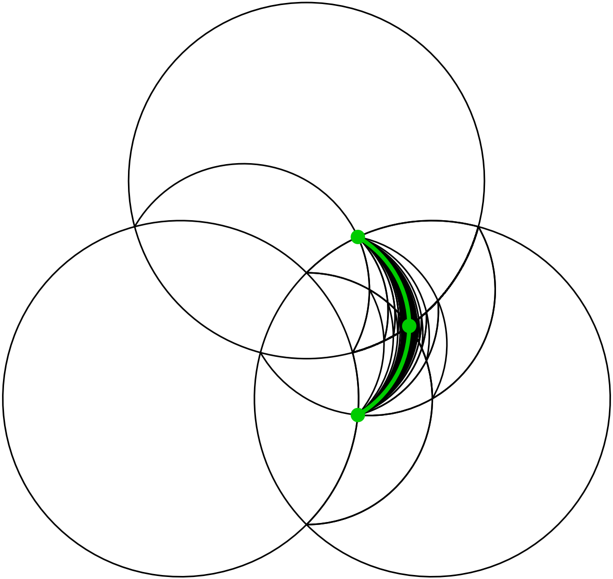
Surface example



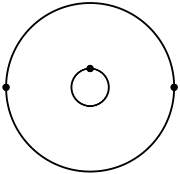


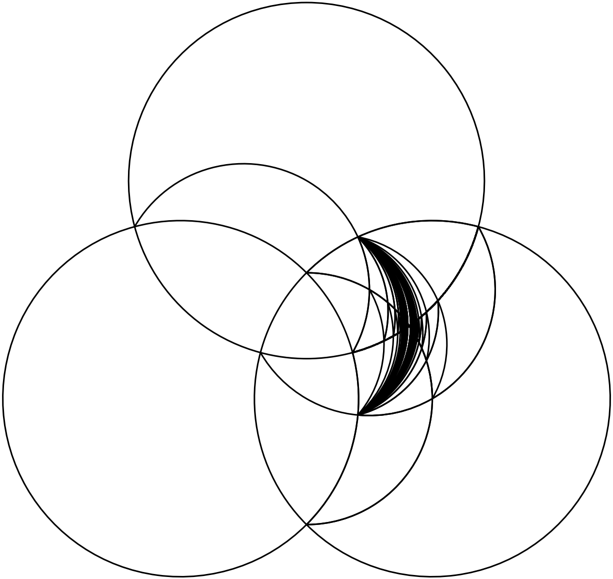
Surface example





Surface example





I should probably mention

Plabic graphs

SL_3 webs

Section 4: Mutation

Mutation and mutation maps

Recall matrix mutation in direction k:

$B' = \mu_k(B)$ is:

$$b'_{ij} = \begin{cases} -b_{ij} & \text{if } k \in \{i,j\}; \\ b_{ij} + \operatorname{sgn}(b_{kj})[b_{ik}b_{kj}]_+ & \text{otherwise.} \end{cases}$$

(Recall also $[a]_+$ means $\max(a,0)$.)

For each B and each sequence $\mathbf{k}$ of indices, there is a mutation map $\eta^B_{\mathbf{k}}$, which takes a vector $\mathbf{a}$, places it below B as a new row, does mutations in indices $\mathbf{k}$, and reads off the new last row.

Mutation symmetry

Key fact: The mutation map $\eta_{\mathbf{k}}^{B^T}$ is an isomorphism from the cluster scattering fan for B to the cluster scattering fan for $\mu_{\mathbf{k}}(B)$. (With a little more work, you can pull along cluster scattering *diagrams*, broken lines, and theta functions.)

A mutation symmetry of B is a sequence $\mathbf{k}$ of indices such that $\mu_{\mathbf{k}}(B) = B$.

Upshot: If $\mathbf{k}$ is a mutation symmetry, then $\eta_{\mathbf{k}}^{B^T}$ is an automorphism of the cluster scattering fan/diagram for B.

In some sense, the Key Fact is due to GHKK, but their mutation definition hides mutation symmetry. (Do a mutation symmetry their way, and you need a linear map to get back to the cluster scattering fan of B.)

Greg Muller was first to mutate scattering diagrams this way. Stella and I wrote down this mutation of theta functions.

Mutation symmetry and structure constants

Stella and I are working to compute theta functions and structure constants for multiplication of theta functions in affine type.

I'll sketch the first way that mutation symmetry helps. (Other uses of mutation symmetry, which we won't talk about, must look more subtly at how broken lines mutate.)

Theorem (R., Stella 2021). Suppose $\mathbf{k}$ is a mutation symmetry and suppose λ and ν are in finite $\eta_{\mathbf{k}}^{B^T}$-orbits. Then $\vartheta_\lambda \cdot \vartheta_\nu$ is a linear combination of theta functions indexed by vectors in finite $\eta_{\mathbf{k}}^{B^T}$-orbits.

The hypotheses of the theorem are beyond what I can introduce today, but include affine type. (For experts: Θ is the whole space.)

Mutation symmetry and structure constants

Stella and I are working to compute theta functions and structure constants for multiplication of theta functions in affine type.

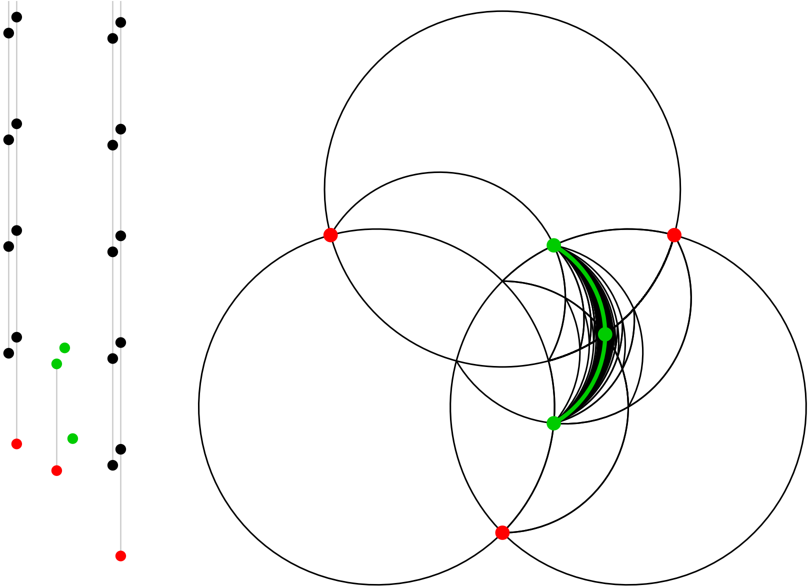
I'll sketch the first way that mutation symmetry helps. (Other uses of mutation symmetry, which we won't talk about, must look more subtly at how broken lines mutate.)

Theorem (R., Stella 2021). Suppose **k** is a mutation symmetry and suppose λ and ν are in finite $\eta_{\mathbf{k}}^{B^T}$-orbits. Then $\vartheta_\lambda \cdot \vartheta_\nu$ is a linear combination of theta functions indexed by vectors in finite $\eta_{\mathbf{k}}^{B^T}$-orbits.

The hypotheses of the theorem are beyond what I can introduce today, but include affine type. (For experts: Θ is the whole space.)

Let's look at an example.

Affine scattering fan example (again)



Mutation symmetry and structure constants (continued)

Theorem (R., Stella 2021). If λ and ν are in finite $\eta_{\mathbf{k}}^{B^T}$-orbits, then $\vartheta_\lambda \cdot \vartheta_\nu$ is a combination of theta functions indexed by vectors in finite $\eta_{\mathbf{k}}^{B^T}$-orbits.

Proof idea. Hypotheses imply that $\vartheta_\lambda \cdot \vartheta_\nu$ has a unique expression as a finite linear combination of theta functions:

$$\vartheta_\lambda \cdot \vartheta_\nu = \sum_{\tau \in T} c_\tau \vartheta_\tau.$$

A detailed description of mutation of theta functions + the fact that λ is in a finite $\eta_{\mathbf{k}}^{B^T}$-orbit implies that $\left(\eta_{\mathbf{k}}^{B^T}\right)^\ell(\vartheta_\lambda)$ is ϑ_λ times a monomial in the "coefficients", for some ℓ. Same for ν.

So: Applying the right power of $\eta_{\mathbf{k}}^{B^T}$ fixes $\vartheta_\lambda \cdot \vartheta_\nu$ (up to a coefficient). Therefore it fixes every theta-function that appears in $\sum_{\tau \in T} c_\tau \vartheta_\tau$. □Proof idea

Thank you for listening.

Scattering fans (R. 2020).

A combinatorial approach to scattering diagrams (R. 2020).

An affine almost positive roots model (R., Stella 2020).

Cluster scattering diagrams of acyclic affine type (R, Stella 2022).

Cambrian frameworks for cluster algebras of affine type (R., Speyer 2018).

...and more to come soon (some with Stella).