Journal of Software for Algebra and Geometry

Quivers and moduli of their thin sincere representations in Macaulay2

Mary Barker and Patricio Gallardo

vol 15 2025

JSAG **15** (2025), 93–103
https://doi.org/10.2140/jsag.2025.15.93

The Journal of Software for
Algebra and Geometry

Quivers and moduli of their thin sincere representations in Macaulay2

MARY BARKER AND PATRICIO GALLARDO

ABSTRACT: We introduce the Macaulay2 package *ThinSincereQuivers* for studying acyclic quivers, the moduli of their thin sincere representations, and the reflexive flow polytopes associated to them. We provide some background on the topic and illustrate how the package recovers examples from the literature.

1. INTRODUCTION. Our work, the package *ThinSincereQuivers* for Macaulay2, provides computational tools at the crossroads of toric geometry, graph theory, and moduli spaces. The main objects are acyclic quivers (finite directed graphs) and representations that associate a one-dimensional vector space to each vertex, that is a dimension vector equal to all 1s. These representations are called thin sincere ones and their moduli spaces are projective toric varieties associated to the well-studied flow polytopes; see [1; 10; 12; 13]. Additional motivation is given by Craw and Smith [9] who showed that every toric variety is the fine moduli space for stable thin representations of an appropriate quiver with relations. In a different context, thin sincere quiver representations have been used for theoretical foundations of certain neural networks [4].

Our package has many potential applications. It can be used to construct families of reflexive polytopes which play a central role in mirror symmetry [3]. It explicitly describes the changes among the possible flow polytopes constructed from a given quiver [13], and it also provides toric compactifications of moduli spaces such as $\mathrm{M}_{0,n}$ [7]. Throughout our work, we highlight these applications by revisiting examples from the aforementioned references

The main functionality is encapsulated in the `ToricQuiver` data type. Quivers are equipped with either an integer number for each of the arrows, also called an integer flow, or an integer weight for the vertices. The type `ToricQuiver` can be constructed by either an incidence matrix and a vector of weights or a function that produces an acyclic quiver from an arbitrary graph by utilizing the Package *Graphs* [8] (Section 2.1). Given a quiver, the user can study its subquivers and their stability (Section 2.2), the space of regular flows and the chamber structure within the space of weights (Section 3.1), the flow polytope associated to a given weight (Section 4.1), and the moduli spaces of thin sincere representations (Section 4.2). We also include a function for detecting if a quiver is tight (an important technical assumption) and implement an algorithm for tightening it.

MSC2020: primary 05-04, 16G20; secondary 05E14.
Keywords: toric quiver, moduli, sincere thin representation.
ThinSincereQuivers version 0.1

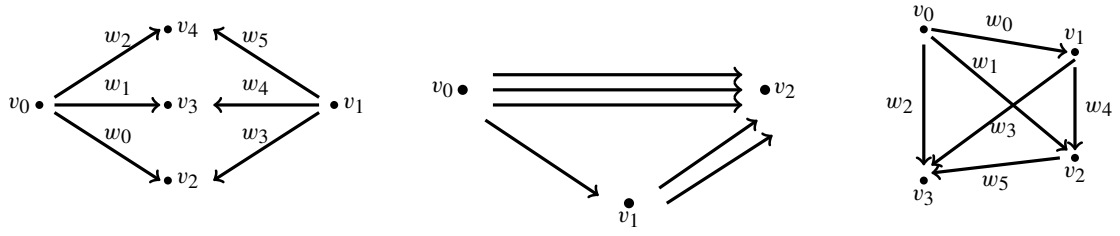

Figure 1. Examples of acyclic quivers.

2. DESCRIPTION. Let Q represent an acyclic quiver, also referred to as an acyclic finite directed graph. We label the vertices as Q_0 and the arrows as Q_1, as illustrated in Figure 1. For each arrow, we associate an integer via the function $\boldsymbol{w} : Q_1 \to \mathbb{Z}$. This function is known as the integral flow of Q. An integral flow is described as regular if it is nonnegative, meaning $\boldsymbol{w}(a) \geq 0$ for every $a \in Q_1$.

A function $\theta : Q_0 \to \mathbb{Z}$ is termed an integral weight when it meets the condition $\sum_{i \in Q_0} \theta(i) = 0$. All such integral weights form the set $\mathrm{Wt}(Q)$, which is a subset of $\mathbb{R}^{Q_0}$. There exists a surjective map, $\mathrm{inc}(\boldsymbol{w}) : \mathbb{Z}^{Q_1} \to \mathrm{Wt}(Q)$, linking an integral flow to its corresponding weight. If a user doesn't specify the flow $\boldsymbol{w}$, the program defaults to $\mathbf{1} := (1, \ldots, 1)$. The integral weight corresponding to this default flow is called the canonical weight, represented as $\delta_Q := \mathrm{inc}(\mathbf{1})$.

2.1. ***Constructing quivers.*** To define a quiver, we can enter the edges as pairs $\{a, b\}$ where a and b are vertices and the arrow takes the orientation from a to b. The user can determine a particular flow, a random flow, or use the default flow of $\mathbf{1} = (1, \ldots, 1)$ by leaving it undeclared.

The main data type in this package is a type of hash table called `ToricQuiver`. It contains the following information: a list Q_0 of integers representing the vertices, an ordered list Q_1 of pairs of integers, representing the edges of the quiver, and the incidence matrix representation of the underlying directed graph. This is the $|Q_0| \times |Q_1|$ matrix consisting of entries

$$a_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ is the head of the } j\text{-th edge,} \\ -1 & \text{if vertex } i \text{ is the tail of the } j\text{-th edge,} \\ 0 & \text{otherwise.} \end{cases}$$

In addition to these, the `ToricQuiver` data type contains a list of integers representing the flow $\boldsymbol{w}$ and the weight $\mathrm{inc}(\boldsymbol{w})$.

Example 2.1. We generate the quiver Q associated to a bipartite graph (see Figure 1, left) with a random flow $\boldsymbol{w}$ as follows:

```
i1 : needsPackage("ThinSincereQuivers")
o1 = ThinSincereQuivers
o1 : Package

i2 : Q0 = {{0,2},{0,3},{0,4},{1,2},{1,3},{1,4}}
```

```
o2 = {{0, 2}, {0, 3}, {0, 4}, {1, 2}, {1, 3}, {1, 4}}
o2 : List

i3 : BipariteQuiver23 = toricQuiver(Q0, Flow => "Random")
o3 = ToricQuiver{flow => {8, 1, 3, 4, 6, 0}                              }
                 IncidenceMatrix => | -1 -1 -1 0  0  0  |
                                    | 0  0  0  -1 -1 -1 |
                                    | 1  0  0  1  0  0  |
                                    | 0  1  0  0  1  0  |
                                    | 0  0  1  0  0  1  |
                 Q0 => {0, 1, 2, 3, 4}
                 Q1 => {{0, 2}, {0, 3}, {0, 4}, {1, 2}, {1, 3}, {1, 4}}
                 synonym => toric quiver
                 weights => {-12, -10, 12, 7, 3}
o3 : ToricQuiver
```

We change the flow of the quiver `BipartiteQuiver23` from its random value to another one `w`, and then obtain information about the quiver.

```
i4 : w = {1,1,1,1,1,1};
i5 : BPw = toricQuiver(BipartiteQuiver23, w)
o5 = ToricQuiver{flow => {1, 1, 1, 1, 1, 1}                              }
                 IncidenceMatrix => | -1 -1 -1 0  0  0  |
                                    | 0  0  0  -1 -1 -1 |
                                    | 1  0  0  1  0  0  |
                                    | 0  1  0  0  1  0  |
                                    | 0  0  1  0  0  1  |
                 Q0 => {0, 1, 2, 3, 4}
                 Q1 => {{0, 2}, {0, 3}, {0, 4}, {1, 2}, {1, 3}, {1, 4}}
                 synonym => toric quiver
                 weights => {-3, -3, 2, 2, 2}
o5 : ToricQuiver

i6 : isAcyclic(BPw)
o6 = true

i7 : getWeights(BPw)
o7 = {-3, -3, 2, 2, 2}
o7 : List

i8 : incInverse(getWeights(BPw),BPw)
o8 = {-1, 2, 2, 3, 0, 0}
o8 : List
```

Our package includes several built-in quiver families such as `bipartiteQuiver(r,n)`. Additionally, our package can generate quivers based on graphs by leveraging the *Graphs* Package [8]. As an illustration, to create a quiver associated with the complete graph of four vertices (refer to [13, Example 2]), one would utilize the following method:

```
i9 : needsPackage("Graphs");
i10 : CG = toricQuiver completeGraph 4
```

```
o10 = ToricQuiver{flow => {1, 1, 1, 1, 1, 1}                              }
                  IncidenceMatrix => | -1 -1 -1 0  0  0  |
                                     | 1  0  0  -1 -1 0  |
                                     | 0  1  0  1  0  -1 |
                                     | 0  0  1  0  1  1  |
                  Q0 => {0, 1, 2, 3}
                  Q1 => {{0, 1}, {0, 2}, {0, 3}, {1, 2}, {1, 3}, {2, 3}}
                  synonym => toric quiver
                  weights => {-3, -1, 1, 3}
o10 : ToricQuiver
```

2.2. ***Subquivers and their stability.*** A subquiver P of Q is defined as a quiver P such that $P_0 \subset Q_0$, $P_1 \subset Q_1$ and the flow function $P_1 \to \mathbb{Z}$ is obtained by restricting the one from Q. The user has two options for working with subquivers. First, they can interpret the subquiver $P \subset Q$ as one with the same vertices $P_0 = Q_0$ but with a nonzero flow in the arrows $P_1 \subsetneq Q_1$. Since the arrows in Q_1 are labeled, there is a subset $I \subset \{0, \ldots, |Q_1| - 1\}$ associated to P_1. The subquiver P can be constructed with the command `Q^I` (see Figure 2). The command `subquivers(Q)` lists all of the subquivers of Q with the format $Q^\wedge I$. The second option is to describe P without any reference to Q. This is done by the command `Q_I`. Finally, the list of arrows defining all spanning trees can be obtained with the command `allSpanningTrees(Q)`.

Given a weight $\theta \in \mathrm{Wt}(Q)$ and a subquiver $P \subset Q$, we can associate the concepts of θ-stable, θ-semistable, and θ-unstable to P. They depend on a certain subset of vertices defined as follows: $V \subset Q_0$ is called P-successor closed if there is no arrow in P_1 leaving V. That is, for all $a \in P_1$ with $a^- \in V$, we also have $a^+ \in V$. We can check if subset $V \subset Q_0$ is P-successor closed by using

```
i11 : Q = bipartiteQuiver(2, 3);
i12 : VA = {0,3};
i13 : VB = {1,3};
i14 : P = Q_{0,1,2,4}
o14 = ToricQuiver{flow => {1, 1, 1, 1}                  }
                  IncidenceMatrix => | -1 -1 -1 0  |
                                     | 0  0  0  -1 |
                                     | 1  0  0  0  |
                                     | 0  1  0  1  |
                                     | 0  0  1  0  |
                  Q0 => {0, 1, 2, 3, 4}
                  Q1 => {{0, 2}, {0, 3}, {0, 4}, {1, 3}}
                  synonym => toric quiver
                  weights => {-3, -1, 1, 2, 1}
o14 : ToricQuiver

i15 : isClosedUnderArrows(VA,P)
o15 = false

i16 : isClosedUnderArrows(VB,P)
o16 = true
```

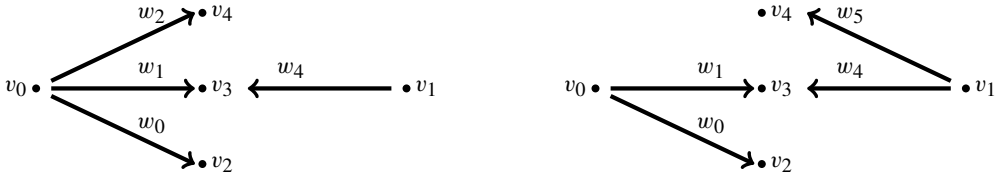

Figure 2. Subquivers of the bipartite quiver as defined in Example 2.1. They are labeled by the subsets $I = \{0, 1, 2, 4\}$ and $\{0, 1, 4, 5\}$

A key concept in the theory of quiver representations is that of θ-(semi)-stability. Our package enables users to determine whether a given subquiver P of Q is either θ-stable or θ-semistable by using

```
i17 : Q = toricQuiver(bipartiteQuiver(2,3));
i18 : isStable(P,Q)
o18 = false

i19 : isSemistable(P,Q)
o19 = false
```

A quiver is θ-*unstable* if it is not θ-semistable. If $P \subset Q$ is a θ-unstable quiver, then any quiver $R \subset P$ is also θ-unstable. Therefore, we can define maximal θ-unstable subquivers as those that are not contained properly in any other θ-unstable quiver. Given a quiver Q with weight θ, we can assume any maximal unstable quiver P satisfies $P_0 = Q_0$ and $P_1 \subsetneq Q_1$. Since P is determined by a subset $I \subset Q_1$. We enumerate all maximal θ-unstable quivers by listing such subsets.

```
i20 : maximalUnstableSubquivers bipartiteQuiver(2,3)
o20 = HashTable{NonSingletons => {{0, 1, 2, 3}, {0, 1, 2, 4}, {0, 1, 2, 5},
      {0, 3, 4, 5}, {1, 3, 4, 5}, {2, 3, 4, 5}}}
o20 : HashTable
```

Similarly, if $P \subset Q$ is not θ-stable, then any quiver $R \subset P$ is also not θ-stable. Therefore, we can consider maximal not θ stable θ-subquivers. The command in this case is

```
i21 : maximalNonstableSubquivers bipartiteQuiver(2,3)
o21 = HashTable{NonSingletons => {{0, 1, 2, 3}, {0, 1, 2, 4}, {0, 1, 2, 5},
      {0, 3, 4, 5}, {1, 3, 4, 5}, {2, 3, 4, 5}}}
o21 : HashTable
```

The output of above commands is in the form of a hash table. We remark that the list of unstable and not-stable quivers depends on the weight θ. We illustrate such a difference for the quiver `bipartiteQuiver(2,3)` but an alternative weight equal to $\{-5, -1, 2, 2, 2\}$.

```
i22 : w = incInverse({-5,-1,2,2,2},bipartiteQuiver(2,3))
o22 = {1, 2, 2, 1, 0, 0}
o22 : List

i23 : Q = toricQuiver(bipartiteQuiver(2,3),w);
i24 : maximalUnstableSubquivers(Q)
o24 = HashTable{NonSingletons => {{0, 1, 3, 4, 5}, {0, 2, 3, 4, 5},
      {1, 2, 3, 4, 5}}}
o24 : HashTable
```

We now turn to an important technical condition. A weight θ is tight, if for every arrow $\alpha \in Q_1$ the subquiver P with $P_0 = Q_0$ and $P_1 = Q_1 \setminus \{\alpha\}$ is θ-stable.

```
i25 : Q = toricQuiver(completeGraph(4),{1,-2,3,0,0,0});
i26 : isTight Q
o26 = false
```

The function "makeTight", returns a tight quiver such that the associated flow polytope (to be defined next section) does not change. The tightening process is outlined in [2]; see also [10].

```
i27 : Qb = makeTight({-2,1,-2,3},toricQuiver(completeGraph(4)))
o27 = ToricQuiver{flow => {-2, 1, 1, 1}                 }
                  IncidenceMatrix => | -1 -1 -1 -1 |
                                     | 1  1  1  1  |
                  Q0 => {0, 1}
                  Q1 => {{0, 1}, {0, 1}, {0, 1}, {0, 1}}
                  synonym => toric quiver
                  weights => {-1, 1}
o27 : ToricQuiver

i28 : isTight Qb
o28 = true
```

3. REGULAR FLOWS AND CONE OF WEIGHTS. We next turn to other combinatorial structures associated to our quivers. The main reference is [13]. For a given weight $\theta \in \mathrm{Wt}(Q)$, there is an integral polytope

$$\Delta(\theta) = \{\boldsymbol{w} \in \mathbb{R}^{Q_1} \mid \mathrm{inc}(\boldsymbol{w}) = \theta\} \cap \mathbb{R}^{Q_1}_{\geq 0}$$

which is called a flow polytope. In general, given a weight $\theta \in \mathrm{Wt}(Q)$ the polytope $\Delta(\theta)$ can be empty. However, there exists a cone of weights $C(Q) \subset \mathbb{R}^{Q_0}$ such that $\Delta(\theta)$ is not empty if and only if $\theta \in C(Q)$. The polytope has the expected dimension, $|Q_1| - |Q_0| + 1$, if and only if θ is in the interior of $C(Q)$. The vertices of $C(Q)$ are constructed with the so called primitive arrows from the quiver; see [13, Proposition 4.7]. The user can recover the list of primitive arrows, the cone $C(Q)$, and test the membership of θ in $C(Q)$ with the code

```
i29 : needsPackage("Polyhedra");
i30 : Q = toricQuiver completeGraph 4;
i31 : A = primitiveArrows Q;
i32 : CQ = coneFromVData(quiverIncidenceMatrix(Q_A));
i33 : CanonicalWeight = transpose matrix {{-3,-1,2,2}};
               4        1
o33 : Matrix ZZ  <--- ZZ

i34 : inInterior(CanonicalWeight,CQ)
o34 = true
```

We can interpret $\Delta(\theta)$ as parametrizing all possible regular flows with input equal to θ. Each point at the interior of $\Delta(\theta)$ parametrizes a flow with entries that are strictly positive. The vertices of $\Delta(\theta)$

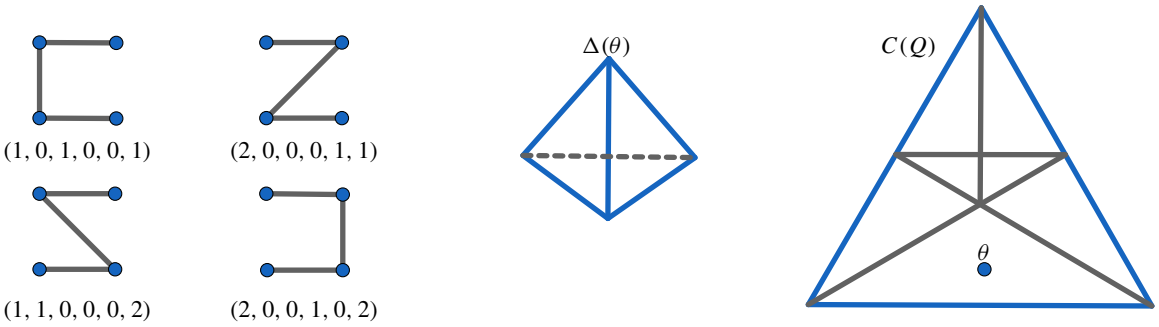

Figure 3. Cone of weights $C(Q)$ (right), polytope $\Delta(\theta)$ associated to the weight $\theta = (-2, 1, -1, 2)$ (center) and θ-stable trees parametrized by the vertices of the polytope $\Delta(\theta)$ (left). Here, the quiver Q is constructed from the complete graph with four vertices.

parametrize regular flows with input θ whose support is a spanning tree; see Figure 3. The user can recover all such spanning trees $T_i \subset Q$ by

```
i35 : tht = {-2,1,-1,2};
i36 : stableTrees(tht,Q)
o36 = {{0, 4, 5}, {0, 3, 5}, {0, 2, 5}, {0, 1, 5}}
o36 : List
```

Each element of the output is a tuple $\{b_1, \ldots, b_m\}$, where b_k denotes the arrows of Q that define the θ-stable tree. In particular, we recover examples such as [13, Figure 10]. Finally, the regular flows supported on above stable trees can be found by using

```
i37 : for i in stableTrees(tht,Q) do print incInverse(tht, Q^i)
{2, 0, 0, 0, 1, 1}
{2, 0, 0, 1, 0, 2}
{1, 0, 1, 0, 0, 1}
{1, 1, 0, 0, 0, 2}
```

3.1. ***Walls and chambers on the space of weights.*** The cone $C(Q)$ has a wall-chamber decomposition induced by the following equivalence relation: $\theta \sim \theta'$ if for every subquiver $P \subset Q$, P is θ-(semi)stable implies P is θ'-(semi)stable and vice versa. In particular, $\theta \sim t\theta$ for every $t > 0$ and if $\theta \sim \theta'$ then $\Delta(\theta)$ is isomorphic to $\Delta(\theta')$ up to affine isomorphism. Our package allows the user to study this structure within $C(Q)$ and recover it for given cases. We remark this problem with a different perspective and implementation has been solved in [5, Section 4]. For us, the starting point to describe the walls of this decomposition is the following result due to Hille.

Lemma 3.1 [13, Section 2.2]. *Each wall W in $C(Q)$ is contained in a hyperplane of the form*

$$W(Q_0^+) = \left\{ \theta \;\middle|\; \sum_{i \in Q_0^+} \theta(i) = 0 \right\},$$

where $Q = Q_0^+ \sqcup Q_0^-$ and the full subquivers Q^+ and Q^- with vertices Q_0^+ and Q_0^- are connected.

We denote any wall $W(Q_o^+)$ by the subset of vertices Q_0^+ used for defining it. The type of the wall $W(Q_0^+)$ which is defined as (t^+, t^-), where t^+ is the number of arrows starting Q_0^+ and ending in Q_0^-, and t^- is the number of arrows starting Q^- and ending in Q^+. One can therefore describe a wall uniquely by the pair: Q_0^+, (t^+, t^-).

```
i38 : L = potentialWalls toricQuiver completeGraph 4;
i39 : L#0
o39 = Wall{Qplus => {1, 2, 3}}
           WallType => (0, 3)
o39 : Wall
```

We remark that although every wall is contained in a hyperplane $W(Q_0)^+$, there are contiguous chambers in which the hyperplane is not a "real wall", that there exist two points x and y in $C(Q)$ such that $x \sim y$ and yet x and y are in different half-spaces defined by the hyperplane W. A necessary condition for $\theta \sim \theta'$ is that $\Delta(\theta) \cong \Delta(\theta')$. We rely on the Package *Lattice Polytopes* [15] for deciding whenever two polytopes are isomorphic (the required hypothesis that $\Delta(\theta)$ is smooth is satisfied for θ within the interior of a chamber).

```
i40 : Q = toricQuiver completeGraph 4;
i41 : samePolytope({-3,2,-1,2},{-2,1,-2,3},Q)
o41 = true

i42 : samePolytope({2,-1,1,-2},{3,-1,-1,-1},Q)
o42 = true
```

We remark that if the weights are not in $C(Q)$ the flow polytope is empty.

Another perspective on describing the chamber decomposition is also given by [13]. Indeed, the spanning trees of Q induce chambers within $C(Q)$ which are defined as

$$C_T = \mathrm{inc}(D_P), \quad D_T = \{\boldsymbol{w} \in \mathbb{R}^{Q_1} \mid \boldsymbol{w}(a) = 0 \text{ for all } \alpha \notin T_1,\ \boldsymbol{w}(a) \geq 0 \text{ for all } \alpha \in T_1\}.$$

A chamber C_T maybe the whole $C(Q)$, so they are not minimal with respect to the equivalence relation defined at the beginning of this section. However, the intersections $\cap C_{T_i}$ which are of maximal dimension define a chamber system which either refines or equals the chamber decomposition of $C(Q)$; see [13, Lemma 4.4]. The relevant commands to produce the refined chamber decomposition are

```
i43 : CQ = coneSystem(CG);
i44 : rts = referenceThetas CQ
o44 = {{-1, -1, -1, 3}, {-2, 1, -1, 2}, {-3, 1, 1, 1}, {-2, -2, 1, 3},
      {-3, -1, 2, 2}, {-1, -3, 2, 2}, {-2, -2, 3, 1}}
o44 : List
```

This results recovers the chamber system for the quiver associated to the complete graph of four vertices, see [13, Figure 7]. A similar computation recovers the 18 chambers associated to the bipartite graph with (2, 3) vertices; see [11, Section 9].

4. APPLICATIONS. We illustrate a few of the possible uses of our packages.

4.1. ***To polytopes.*** The polytopes $\Delta(\theta)$ are a priori contained in $\mathbb{R}^{Q_1}$ and the user can recover them formatted as a list of vertices with

```
i45 : flowPolytopeVertices({-3,-3,2,2,2},bipartiteQuiver(2,3),Format=>"FullBasis")
o45 = {{2, 0, 1, 0, 2, 1}, {2, 1, 0, 0, 1, 2}, {0, 2, 1, 2, 0, 1},
      {1, 2, 0, 1, 0, 2}, {0, 1, 2, 2, 1, 0}, {1, 0, 2, 1, 2, 0}}
o45 : List
```

However, a better representation can be obtained as follows. The polytope $\Delta(\theta)$ is contained in the fiber $\mathrm{inc}^{-1}(\theta)$. Therefore, we translate them to $\mathrm{Cir}(Q) = \mathrm{inc}^{-1}(0)$. This linear subspace is more natural because $\dim \Delta(\theta) = \dim(\mathrm{Cir}(Q)_{\mathbb{R}})$ for θ in the interior of $C(Q)$. Moreover, there is a basis of the lattice $\mathrm{Cir}(Q)$ for each spanning tree T of Q. For example, given the quiver `bipartiteQuiver(2,3)` the following command computes a basis of the two-dimensional vector space $\mathrm{Cir}_{\mathbb{R}}(Q) \subset \mathbb{R}^6$ using the spanning tree `{0, 1, 4, 5}`.

```
i46 : basisForFlowPolytope({0,1,4,5},bipartiteQuiver(2,3))
o46 = | 0  1  |
      | 1  -1 |
      | -1 0  |
      | 0  -1 |
      | -1 1  |
      | 1  0  |
               6        2
o46 : Matrix ZZ  <--- ZZ
```

The package automatically selects one if the none is provided and the user writes

```
i47 : basisForFlowPolytope(bipartiteQuiver(2,3))
o47 = | -1 0  |
      | 0  -1 |
      | 1  1  |
      | 1  0  |
      | 0  1  |
      | -1 -1 |
               6        2
o47 : Matrix ZZ  <--- ZZ
```

We can recover the vertices of $\Delta(\theta)$ with respect to the basis induced by any spanning tree. For example, if Q is the bipartite graph and θ is the canonical weight, then $\Delta(\theta)$ is a hexagon. We obtain its vertices with respect to the basis induced by the stable tree with edges $T = \{0, 1, 4, 5\}$.

```
i48 : flowPolytopeVertices({-3,-3,2,2,2},bipartiteQuiver(2,3),Format=>{0,1,4,5})
o48 = {{0, 1}, {1, 1}, {0, -1}, {1, 0}, {-1, -1}, {-1, 0}}
o48 : List
```

This output interfaces with other Macaulay2 packages such as *Polyhedra* [6]. This allows the reader to translate results based on toric quiver computations into problems relating to polyhedral structures. For example, we can also explore which polytopes $\Delta(\theta)$ are reflexive. For example, given the quiver from the

complete graph with four vertices and its canonical weight $\{-3, -1, 1, 3\}$, then we test if the associated polytope is reflexive.

```
i49 : needsPackage("Polyhedra");
i50 : Q = toricQuiver completeGraph 4;
i51 : polyQ = convexHull transpose matrix flowPolytopeVertices({-3,-1,1,3},Q);
i52 : isReflexive polyQ
o52 = true
```

This result is expected because in fact the polytope $\Delta(\delta_Q)$ is always reflexive [1, Proposition 2.7].

4.2. ***Applications to moduli spaces.*** A thin sincere representation of a quiver Q assigns a one-dimensional vector space to each vertex in Q_0 and a linear map to each arrow in Q_1. The space of all representations of Q is $\mathrm{Rep}(Q) \cong \mathbb{C}^{Q_1}$, and their isomorphism classes are defined by action of $(\mathbb{C}^*)^{Q_0}$. Representations of quivers have a notion of θ-(semi)stability. Given a quiver Q with weight $\theta \in C(Q)$, there is a projective complex toric variety

$$\overline{M}(Q, \theta) := \mathrm{Rep}(Q) /\!\!/_\theta (\mathbb{C}^*)^{|Q_0|-1}$$

of dimension $|Q_1| - |Q_0| + 1$ parametrizing θ-semistable thin sincere representations up to isomorphism; see [12; 14]. Given a representation ω, we can define a subquiver $P := \mathrm{Supp}(\omega)$ such that $P_0 = Q_0$ and $P_1 = \{a \in Q_1 \mid \omega(a) \neq 0\}$. This quiver is called the support of ω, and ω is θ-stable precisely when the subquiver $\mathrm{Supp}(\omega)$ is θ-stable; see [12, Lemma 1.4] and discussion before [1, Section 2.2]. Therefore, we can describe the θ-stable and θ-semistable representations parametrized by every $\overline{M}(Q, \theta)$. Moreover, via toric geometry, our package can be used to study these moduli spaces.

Theorem 4.1 [12]. *If θ is in the interior of $C(Q)$, the compact toric variety $\overline{M}(Q, \theta)$ is not empty and its associated polytope is equal to $\Delta(\theta)$.*

In particular, the geometry of $C(Q)$ describes the birational geometry of these compactifications. Particular examples of algebraic varieties constructed via our package include the blow up of $\mathbb{P}^n$ along a linear subspace (see [16]) and toric compactifications of the moduli space of n labeled points in $\mathbb{P}^1$ (see [7]).

SUPPLEMENT. The online supplement contains version 0.1 of *ThinSincereQuivers*.

ACKNOWLEDGEMENT. We thank Daniel McKenzie and Eloísa Grifo for their insightful conversations related to this work. We also extend our gratitude to the referee, whose detailed feedback significantly improved our work. Gallardo appreciates the support provided by both Washington University at St. Louis and the University of California, Riverside. This material is based upon work supported by the National Science Foundation under grant DMS-2316749. Barker would like to express her thanks to the Department of Mathematics at Washington University in St. Louis for their support and access to computational resources.

REFERENCES.

[1] K. Altmann and L. Hille, "Strong exceptional sequences provided by quivers", *Algebr. Represent. Theory* **2**:1 (1999), 1–17. MR

[2] K. Altmann and D. van Straten, "Smoothing of quiver varieties", *Manuscripta Math.* **129**:2 (2009), 211–230. MR

[3] K. Altmann, B. Nill, S. Schwentner, and I. Wiercinska, "Flow polytopes and the graph of reflexive polytopes", *Discrete Math.* **309**:16 (2009), 4992–4999. MR

[4] M. A. Armenta and P.-M. Jodoin, "The representation theory of neural networks", 2020. arXiv 2007.12213

[5] W. Baldoni-Silva, J. A. De Loera, and M. Vergne, "Counting integer flows in networks", *Found. Comput. Math.* **4**:3 (2004), 277–314. MR

[6] R. Birkner and Lars Kastner, "Polyhedra: convex polyhedra", Macaulay2 package, available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages. Version 1.10.

[7] M. Blume and L. Hille, "Quivers and moduli spaces of pointed curves of genus zero", *Algebr. Comb.* **4**:1 (2021), 89–124. MR

[8] J. Burkart, D. Cook, C. Jansen, A. Taylor, and A. O'Keefe, "Graphs: graphs and directed graphs (digraphs)", Macaulay2 package, available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages. Version 0.3.2.

[9] A. Craw and G. G. Smith, "Projective toric varieties as fine moduli spaces of quiver representations", *Amer. J. Math.* **130**:6 (2008), 1509–1534. MR

[10] M. Domokos and D. Joó, "On the equations and classification of toric quiver varieties", *Proc. Roy. Soc. Edinburgh Sect. A* **146**:2 (2016), 265–295. MR

[11] T. Hausel and B. Sturmfels, "Toric hyperkähler varieties", *Doc. Math.* **7** (2002), 495–534. MR

[12] L. Hille, "Toric quiver varieties", pp. 311–325 in *Algebras and modules*, *II* (Geiranger, 1996), edited by I. Reiten et al., CMS Conf. Proc. **24**, Amer. Math. Soc., Providence, RI, 1998. MR

[13] L. Hille, "Quivers, cones and polytopes", *Linear Algebra Appl.* **365** (2003), 215–237. MR

[14] A. D. King, "Moduli of representations of finite-dimensional algebras", *Quart. J. Math. Oxford Ser.* (2) **45**:180 (1994), 515–530. MR

[15] A. Lundman and G. S. Ståhl, "LatticePolytopes", *Macaulay*2 package, available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages. Version 1.0.

[16] X. Qin, "Blow ups of $\mathbb{P}^n$ as quiver moduli for exceptional collections", 2018. arXiv 1804.09544

RECEIVED: 16 Jun 2021 REVISED: 19 Jun 2025 ACCEPTED: 20 Aug 2025

MARY BARKER:

marybarker@wustl.edu
Department of Mathematics and Statistics, Washington University, St. Louis, MO

PATRICIO GALLARDO:

pgallard@ucr.edu
Department of Mathematics, University of California, Riverside, CA, United States