Knot Group Representations Collected Results

Justin Lawrence

November 29, 2023

1 Torus Knots Group

The torus knot group has presentation [Muñ09]

$$\Gamma = \langle a, b \mid a^n = b^m \rangle \tag{1}$$

where $(n, m) = 1$. A full characterization of the $\mathrm{SL}_2(\mathbb{C})$ representations of this group can be found in [Muñ09], and a full characterization of the $\mathrm{SL}_3(\mathbb{C})$ representations in [MP16].

1.1 Irreducible $\mathrm{SL}_2(\mathbb{C})$ Representations

1.1.1 Diagonalizable Representations

Let $\rho \in \mathrm{Hom}(\Gamma, \mathrm{SL}_2(\mathbb{C}))$ be an irreducible representation, and assume that $A = \rho(a), B = \rho(b)$ are diagonalizable. That is

$$A \sim \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix} \qquad B \sim \begin{pmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{pmatrix}$$

Since $A, B \in \mathrm{SL}_2(\mathbb{C})$, we must get that $\alpha_1\alpha_2 = \beta_1\beta_2 = 1$, so we can simplify to

$$A \sim \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix} \qquad B \sim \begin{pmatrix} \beta & 0 \\ 0 & \beta^{-1} \end{pmatrix}$$

Since the equivalence class of a representation is not changed by a change of basis, we can assume that

$$A = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix}$$

Irreduciblility of ρ corresponds to A, B sharing no eigenspaces. Hence, we can choose (via re-scaling of the x and y axes) to make the eigenspaces of B

$$V_\beta = \mathrm{Span}_{\mathbb{C}} \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \qquad V_{\beta^{-1}} = \mathrm{Span}_{\mathbb{C}} \left\{ \begin{pmatrix} 1 \\ z \end{pmatrix} \right\}$$

where $z \neq 0, 1$. Furthermore, it is clear that the eigenspaces of A, B cannot be two dimensional, and hence $\alpha \neq \alpha^{-1}, \beta \neq \beta^{-1}$, implying that $\alpha, \beta \neq \pm 1$. Finally, we need for $A^n = B^m$. This is only possible if we merge the eigenspaces, so we need $\alpha^n = \alpha^{-n} = \beta^m = \beta^{-m}$, or more simply $\alpha^n = \beta^m = \pm 1$. Our choice of representation then amounts to a choice of tuple (α, β, z), following the restrictions

1. $\alpha^n = \beta^m = \pm 1$
2. $\alpha, \beta \neq \pm 1$
3. $z \neq 0, 1$

The equivalence relation on these tuples is

$$(\alpha, \beta, z) \sim (\alpha^{-1}, \beta, z^{-1}) \sim (\alpha, \beta^{-1}, z^{-1}) \sim (\alpha^{-1}, \beta^{-1}, z) \tag{2}$$

Finally, we consider the automorphism $\sigma \in \mathrm{Aut}(\Gamma)$ defined by $\sigma : a \mapsto a^{-1}, b \mapsto b^{-1}$. It's action on a representation is given by $(\alpha, \beta, z) \mapsto (\alpha^{-1}, \beta^{-1}, z)$. Hence, it acts trivially on the character variety, and should therefore act by conjugation on any particular representation $\rho(\Gamma)$. Indeed, we can find that $\sigma(X) = SXS^{-1}$, where

$$S = \begin{pmatrix} 0 & b \\ bz & 0 \end{pmatrix} \tag{3}$$

and $b \in \mathbb{C}^\times$ is arbitrary. For simplicity, we'll choose $b = 1$. Our choice of S (in this form) is uniquely defined by our choice of z. Note that two equivalent representations can have different S, although their two conjugation matrices will be equivalent as well.

1.1.2 Non-Diagonalizable Representations

This section consists solely of the following theorem.

Theorem 1.1. *There exists no irreducible representation $\rho \in \mathrm{Hom}(\Gamma, SL_2(\mathbb{C}))$ such that $A = \rho(a)$ or $B = \rho(b)$ are not diagonalizable.*

Proof. By Corollary 5.2.1, A is not diagonalizable if and only if B is not diagonalizable, since we need for $A^n = B^m$ to hold. Thus, we can assume that

$$A \sim \begin{pmatrix} \alpha & 1 \\ 0 & \alpha \end{pmatrix} \qquad B \sim \begin{pmatrix} \beta & 1 \\ 0 & \beta \end{pmatrix}$$

Irreduciblility then requires that $V_\alpha \cap V_\beta = \{\underline{0}\}$ (i.e. the eigenspaces are disjoint). $A^n = B^m$ implies that A^n, B^m have the same eigenspace. But by Corollary 5.2.1, A^n, B^m have the same eigenspaces as A, B, a contradiction. $\square$

1.2 Irreducible $\mathrm{SL}_3(\mathbb{C})$ Representations

1.2.1 Diagonalizable Representations

Let $\rho \in \mathrm{Hom}(\Gamma, \mathrm{SL}_3(\mathbb{C}))$ be an irreducible representation, and assume that $A = \rho(a), B = \rho(b)$ are both diagonalizable. Then we get that $A \sim \mathrm{diag}(\alpha_1, \alpha_2, \alpha_3)$, $B \sim \mathrm{diag}(\beta_1, \beta_2, \beta_3)$. Since $A, B \in \mathrm{SL}_3(\mathbb{C})$, we can see that $\prod_{j=1}^3 \alpha_j = \prod_{j=1}^3 \beta_j = 1$, so this can be simplified to $A \sim \mathrm{diag}(\alpha_1, \alpha_2, (\alpha_1\alpha_2)^{-1})$, $B \sim \mathrm{diag}(\beta_1, \beta_2, (\beta_1\beta_2)^{-1})$. Now irreduciblility requires that all the eigenspaces of A, B be disjoint. Since the direct sum of the eigenspaces of A or B span $\mathbb{C}^3$, it follows that no eigenspace can have dimension two or three. Hence, $\alpha_j \neq \alpha_k, \beta_j \neq \beta_k$. Furthermore, $A^n = B^m$ requires that the two matrices A^n, B^m have the same eigenspaces. Suppose that A^n is not scalar. Then it has an eigenspace of dimension one, which it shares with B^m. By Theorem 5.2, this must also be a shared eigenspace of A, B, which were assumed to have disjoint eigenspaces, a contradiction. Thus, A^n, B^m are scalar, that is $\alpha_j^n = \beta_k^m$ for any $j, k \in \{1, 2, 3\}$. Then $\alpha_1^n = \alpha_2^n$, and $\alpha_1^n = \alpha_1^{-n}\alpha_2^{-n}$ which implies that $\alpha_1^{3n} = 1$, with a similar result holding for each α_j, β_j. Finally, since the equivalence class of a representation is not changed by a change of basis, we can assume that $A = \mathrm{diag}(\alpha_1, \alpha_2, (\alpha_1\alpha_2)^{-1})$. Let $\underline{v}_1, \underline{v}_2, \underline{v}_3 \in \mathbb{C}^3$ be the eigenvectors of B. Irreducibility requires that all eigenspaces, and direct sums of two eigenspaces, be disjoint. Thus, for any $i, j, k, l \in \{1, 2, 3\}$, we require that $\mathrm{Span}_{\mathbb{C}}\{\underline{e}_i\} \neq \mathrm{Span}_{\mathbb{C}}\{\underline{v}_j\}$, and if $i \neq j, k \neq l$ then $\mathrm{Span}_{\mathbb{C}}\{\underline{e}_i, \underline{e}_j\} \neq \mathrm{Span}_{\mathbb{C}}\{\underline{v}_k, \underline{v}_l\}$ Our choice of representation equivalence class therefore amounts to a choice of tuple $(\alpha_1, \alpha_2, \beta_1, \beta_2, [\underline{v}_1], [\underline{v}_2], [\underline{v}_3])$ with restrictions

1. $\alpha_1^n = \alpha_2^n = \beta_1^m = \beta_2^m$, $\alpha_1^{3n} = 1$
2. $\alpha_1 \neq \alpha_2, \alpha_1^2 \neq \alpha_2^{-1}, \alpha_2^2 \neq \alpha_1^{-1}, \beta_1 \neq \beta_2, \beta_1^2 \neq \beta_2^{-1}, \beta_2^2 \neq \beta_1^{-1}$
3. $\mathrm{Span}_{\mathbb{C}}\{\underline{v}_1, \underline{v}_2, \underline{v}_3\} = \mathbb{C}^3$
4. $\forall i \neq j, k \neq l \in \{1, 2, 3\}, \mathrm{Span}_{\mathbb{C}}\{\underline{v}_i, \underline{v}_j\} \neq \mathrm{Span}_{\mathbb{C}}\{\underline{e}_k, \underline{e}_l\}$ and $\mathrm{Span}_{\mathbb{C}}\{\underline{v}_i\} \neq \mathrm{Span}_{\mathbb{C}}\{\underline{e}_k\}$

Let $\varphi_{ij} \in \mathrm{PGL}(3, \mathbb{C})$ be any transformation satisfying $\varphi([\underline{e}_i]) = [\underline{e}_j], \varphi([\underline{e}_j]) = [\underline{e}_i], \varphi([\underline{e}_k]) = [\underline{e}_k]$, where $k \neq i, j$. Then our equivalence relation on these tuples can be entirely defined by the following being equivalent

1. $(\alpha_1, \alpha_2, \beta_1, \beta_2, [\underline{v}_1], [\underline{v}_2], [\underline{v}_3])$
2. $(\alpha_1, (\alpha_1\alpha_2)^{-1}, \beta_1, \beta_2, \varphi_{23}([\underline{v}_1]), \varphi_{23}([\underline{v}_2]), \varphi_{23}([\underline{v}_3]))$
3. $(\alpha_2, \alpha_1, \beta_1, \beta_2, \varphi_{12}([\underline{v}_1]), \varphi_{12}([\underline{v}_2]), \varphi_{12}([\underline{v}_3]))$
4. $(\alpha_1, \alpha_2, \beta_1, (\beta_1\beta_2)^{-1}, [\underline{v}_1], [\underline{v}_3], [\underline{v}_2])$
5. $(\alpha_1, \alpha_2, \beta_2, \beta_1, [\underline{v}_2], [\underline{v}_1], [\underline{v}_3])$

Finally, we consider the action of the automorphism σ on these representations. In terms of tuples, this action is given by

$$(\alpha_1, \alpha_2, \beta_1, \beta_2, [\underline{v}_1], [\underline{v}_2], [\underline{v}_3]) \mapsto (\alpha_1^{-1}, \alpha_2^{-1}, \beta_1^{-1}, \beta_2^{-1}, [\underline{v}_1], [\underline{v}_2], [\underline{v}_3])$$

In general, this will act non-trivially on the character variety. In order for it to act trivially, we need for it to act by conjugation on any particular representation with the character of interest. Thus, our eigenvalues cannot change, so we must have one of $\alpha_2 = \alpha_1^{-1}, (\alpha_1\alpha_2)^{-1} = \alpha_1^{-1}$, and a similar result for B. Since we can permute the eigenvectors of A, B freely, we can choose that $\alpha_1^{-1} = \alpha_2, \beta_1^{-1} = \beta_2$. Then $A = \mathrm{diag}(\alpha, \alpha^{-1}, 1)$ and $B \sim \mathrm{diag}(\beta, \beta^{-1}, 1)$, where $\alpha = \alpha_1, \beta = \beta_1$. In this case, the action of σ is given by

$$(\alpha, \alpha^{-1}, \beta, \beta^{-1}, [\underline{v}_1], [\underline{v}_2], [\underline{v}_3]) \mapsto (\alpha^{-1}, \alpha, \beta^{-1}, \beta, [\underline{v}_1], [\underline{v}_2], [\underline{v}_3]) \sim (\alpha, \alpha^{-1}, \beta, \beta^{-1}, \varphi_{12}([\underline{v}_2]), \varphi_{12}([\underline{v}_1]), \varphi_{12}([\underline{v}_3]))$$

We can then divide our considerations up into the following cases.

Case 1 : Suppose that $\{\underline{e}_1, \underline{e}_2, \underline{e}_3, \underline{v}_3\}$ are in general position (see Definition 4.1). Then by Theorem 4.4, φ_{12} can be represented by the transformation $A \in \mathrm{GL}_3(\mathbb{C})$ given by

$$A = \begin{pmatrix} 0 & v_{31}v_{32}^{-1} & 0 \\ v_{32}v_{31}^{-1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

in the standard basis. Since $\underline{v}_3$ is in general position with the standard basis, it must have all non-zero components, so this is well-defined. Now, suppose that

$$\underline{v}_1 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

then we must get that

$$\underline{v}_2 = \begin{pmatrix} v_{31}v_{32}^{-1}b \\ v_{32}v_{31}^{-1}a \\ c \end{pmatrix}$$

So our conditions become

$$v_{21} = v_{31}v_{32}^{-1}v_{12} \qquad v_{22} = v_{32}v_{31}^{-1}v_{11} \qquad v_{23} = v_{13}$$

Case 2 : Suppose that $\{\underline{e}_1, \underline{e}_2, \underline{e}_3, \underline{v}_3\}$ are not in general position. By the proof of Theorem 4.3, we still must have φ_{12} represented by a transformation of the form

$$A = \begin{pmatrix} 0 & r_1 & 0 \\ r_2 & 0 & 0 \\ 0 & 0 & r_3 \end{pmatrix}$$

in the standard basis. Now, since $\underline{v}_3$ is not in general position with the standard basis, one of its components must be zero. In particular, since $A\underline{v}_3$ is a multiple of $\underline{v}_3$, it's clear that the third component must be zero. The other two components are clearly non-zero, so we can write that

$$\underline{v}_3 = \begin{pmatrix} 1 \\ a \\ 0 \end{pmatrix}$$

This in turn necessitates that each of $\underline{v}_1, \underline{v}_2$ have a non-zero third component, so we can write that

$$\underline{v}_1 = \begin{pmatrix} b \\ c \\ 1 \end{pmatrix} \qquad \underline{v}_2 = \begin{pmatrix} d \\ e \\ 1 \end{pmatrix}$$

Now, since $A\underline{v}_3$ is a multiple of $\underline{v}_3$, we need for

$$\underline{v}_3 = k \begin{pmatrix} r_1 a \\ r_2 \\ 0 \end{pmatrix}$$

for some $k \neq 0$. In particular, we must get that $k = r_1^{-1}a^{-1}$, so $a = r_1^{-1}r_2a^{-1}$ implying that $a^2 = r_1^{-1}r_2$. Since $A\underline{v}_1$ is a multiple of $\underline{v}_2$, we need for

$$\underline{v}_2 = k \begin{pmatrix} r_1 c \\ r_2 b \\ r_3 \end{pmatrix}$$

for some $k \neq 0$. In particular, the third component implies that $k = r_3^{-1}$, so we require that $d = r_3^{-1}r_1c$ and $e = r_3^{-1}r_2b$. Since A represents the same φ_{12} up to scaling, we may as well set $r_3 = 1$. Since $A\underline{v}_2$ is a multiple of $\underline{v}_1$, we need for

$$\underline{v}_1 = k \begin{pmatrix} r_1 e \\ r_2 d \\ 1 \end{pmatrix}$$

for some $k \neq 0$. In particular, the third component implies that $k = 1$, so we require that $b = r_1e$ and $c = r_2d$. This combined with the results above gives $er_2^{-1} = r_1e$, so $r_2 = r_1^{-1}$. We can derive the same result from $dr_1^{-1} = r_2d$. Thus, our transformation A now takes the form

$$A = \begin{pmatrix} 0 & \delta & 0 \\ \delta^{-1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for some $\delta \neq 0$. Note also that $a^2 = \delta^{-2}$, implying that $\delta = \pm a^{-1}$. So, our conditions become that

$$v_{21} = \pm v_{32}^{-1}v_{12} \qquad v_{22} = \pm v_{32}v_{11} \qquad v_{13} = v_{23}$$

which is equivalent to

$$v_{21} = \pm v_{31}v_{32}^{-1}v_{12} \qquad v_{22} = \pm v_{31}^{-1}v_{32}v_{11} \qquad v_{13} = v_{23}$$

our conditions from the first case. Using these, we prove the following theorem.

Theorem 1.2. *The set of all irreducible representations of Γ in $SL_3(\mathbb{C})$ on which σ acts trivially is closed relative to the set of all irreducible representations of Γ in $SL_3(\mathbb{C})$.*

Proof. Suppose it were not closed. Then there would exist a sequence of irreducible representations $\{\rho_n\}_{n\in\mathbb{N}}$ on which σ acts trivially, such that $\rho = \lim_{n\to\infty}\rho_n$ is an irreducible representation on which σ acts non-trivially. Since the only difference between the conditions on when σ acts trivially or not is on the eigenvectors of B, we can take the eigenvalues of A, B to be fixed in this sequence, along with the eigenvectors of A being the standard basis. We then split into three cases.

Case 1: Suppose that

$$\underline{v}_3 \to \begin{pmatrix} 1 \\ a \\ b \end{pmatrix}$$

where $a, b \neq 0$. Then we get that

$$\frac{v_{32}}{v_{31}} \to a \qquad\qquad \frac{v_{33}}{v_{32}} \to \frac{b}{a}$$

Since $v_{32} \neq 0$, this implies that from some point in the sequence onwards, we always need for $v_{33} \neq 0$. This puts us in the case of the standard basis and $\underline{v}_3$ being in general position, so we require that

$$\underline{v}_1 = \begin{pmatrix} v_{11} \\ v_{12} \\ v_{13} \end{pmatrix} \qquad\qquad \underline{v}_2 = \begin{pmatrix} \frac{v_{31}}{v_{32}} v_{12} \\ \frac{v_{32}}{v_{31}} v_{11} \\ v_{13} \end{pmatrix}$$

Now, without loss of generality, irreducibility of the limit requires that $\frac{v_{13}}{v_{11}}, \frac{v_{13}}{v_{12}} \not\to 0$. Suppose, without loss of generality, that $\frac{v_{13}}{v_{11}} \to c$ is finite. Then $\frac{v_{13}}{v_{11}} \frac{v_{31}}{v_{32}} \to \frac{c}{a} \neq 0$. If $\frac{v_{13}}{v_{12}} \to \infty$, then we're left with the limits

$$\underline{v}_1 \to \begin{pmatrix} 1 \\ 0 \\ c \end{pmatrix} \qquad \underline{v}_2 \to \begin{pmatrix} 0 \\ a \\ c \end{pmatrix} \qquad \underline{v}_3 \to \begin{pmatrix} 1 \\ a \\ b \end{pmatrix}$$

but this is a representation which σ acts trivially on if its irreducible, a contradiction. If $\frac{v_{13}}{v_{12}} \to d$ is finite, then we're left with the limits

$$\underline{v}_1 \to \begin{pmatrix} 1 \\ \frac{c}{d} \\ c \end{pmatrix} \qquad \underline{v}_2 \to \begin{pmatrix} \frac{c}{ad} \\ a \\ c \end{pmatrix} \qquad \underline{v}_3 \to \begin{pmatrix} 1 \\ a \\ b \end{pmatrix}$$

which will be a representation on which σ acts trivially, if it's irreducible. Thus, in this case we can never get a valid sequence.

Case 2: Suppose that

$$\underline{v}_3 \to \begin{pmatrix} 1 \\ a \\ 0 \end{pmatrix}$$

where $a \neq 0$. Then we get that

$$\frac{v_{32}}{v_{31}} \to a \qquad\qquad \frac{v_{33}}{v_{32}} \to 0$$

So we require that

$$\underline{v}_1 = \begin{pmatrix} v_{11} \\ v_{12} \\ v_{13} \end{pmatrix} \qquad\qquad \underline{v}_2 = \begin{pmatrix} \pm\frac{v_{31}}{v_{32}} v_{12} \\ \pm\frac{v_{32}}{v_{31}} v_{11} \\ v_{13} \end{pmatrix}$$

This sequence will fail to converge if the $\pm$ choice continues to switch, unless $v_{21}, v_{22} \to 0$, which produces a reducible representation. Thus, we can assume the sign chosen stays fixed in the sequence. Now, irreducibility of the limit requires that $\frac{v_{13}}{v_{11}}, \frac{v_{13}}{v_{12}} \not\to 0$. Suppose, without loss of generality, that $\frac{v_{13}}{v_{11}} \to c$ is finite. Then $\frac{v_{13}}{v_{11}} \frac{v_{31}}{v_{32}} \to \frac{c}{a} \neq 0$. If $\frac{v_{13}}{v_{12}} \to \infty$, then we're left with the limits

$$\underline{v}_1 \to \begin{pmatrix} 1 \\ 0 \\ c \end{pmatrix} \qquad \underline{v}_2 \to \begin{pmatrix} 0 \\ \pm a \\ c \end{pmatrix} \qquad \underline{v}_3 \to \begin{pmatrix} 1 \\ a \\ 0 \end{pmatrix}$$

but this is a representation which σ acts trivially on if its irreducible, a contradiction. If $\frac{v_{13}}{v_{12}} \to d$ is finite, then we're left with the limits

$$\underline{v}_1 \to \begin{pmatrix} 1 \\ \frac{c}{d} \\ c \end{pmatrix} \qquad \underline{v}_2 \to \begin{pmatrix} \pm\frac{c}{ad} \\ \pm a \\ c \end{pmatrix} \qquad \underline{v}_3 \to \begin{pmatrix} 1 \\ a \\ 0 \end{pmatrix}$$

which will be a representation on which σ acts trivially, if it's irreducible. Thus, in this case we can never get a valid sequence.

Case 3: Suppose, without loss of generality, that

$$\underline{v}_3 \to \begin{pmatrix} 1 \\ 0 \\ a \end{pmatrix}$$

where $a \neq 0$. Then we get that

$$\frac{v_{32}}{v_{31}} \to 0 \qquad \frac{v_{33}}{v_{31}} \to a$$

Since v_{33} converges to a non-zero value, we must have it always be non-zero after a certain point. So we require that

$$\underline{v}_1 = \begin{pmatrix} v_{11} \\ v_{12} \\ v_{13} \end{pmatrix} \qquad \underline{v}_2 = \begin{pmatrix} \frac{v_{31}}{v_{32}} v_{12} \\ \frac{v_{32}}{v_{31}} v_{11} \\ v_{13} \end{pmatrix}$$

Now, irreducibility of the limit requires that $\frac{v_{12}}{v_{11}}, \frac{v_{12}}{v_{13}} \not\to 0$, and that $\frac{v_{11}}{v_{12}}\left(\frac{v_{32}}{v_{31}}\right)^2 \not\to 0$. But $\frac{v_{32}}{v_{31}} \to 0$, and since $\frac{v_{12}}{v_{11}} \not\to 0$, it's inverse tends to a finite value, making $\frac{v_{11}}{v_{12}}\left(\frac{v_{32}}{v_{31}}\right)^2 \not\to 0$ an impossible condition. □

1.2.2 Non-Diagonalizable Representations

Again, we get the following theorem.

Theorem 1.3. *There exists no irreducible representation $\rho \in \mathrm{Hom}(\Gamma, SL_3(\mathbb{C}))$ such that $A = \rho(a)$ or $B = \rho(b)$ are not diagonalizable.*

Proof. By Corollary 5.2.1, we know that A, B must have the same Jordan forms. Thus, there are two cases to consider here.

Case 1: Suppose that

$$A \sim \begin{pmatrix} \alpha & 1 & 0 \\ 0 & \alpha & 1 \\ 0 & 0 & \alpha \end{pmatrix} \qquad B \sim \begin{pmatrix} \beta & 1 & 0 \\ 0 & \beta & 1 \\ 0 & 0 & \beta \end{pmatrix}$$

Since our representation is to be irreducible, we must have that the eigenspaces of A, B are disjoint. But by Corollary 5.2.1, A^n and B^m have the same eigenspaces as A, B, so $A^n \neq B^m$. Thus, this case is impossible.

Case 2: Since $A, B \in \mathrm{SL}_3(\mathbb{C})$, the only remaining possibility is the following case

$$A \sim \begin{pmatrix} \alpha & 1 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha^{-2} \end{pmatrix} \qquad B \sim \begin{pmatrix} \beta & 1 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \beta^{-2} \end{pmatrix}$$

Now, irreducibility implies that A, B share no eigenspaces or generalized eigenspaces. It's immediate from this that $\alpha \neq \alpha^{-2}$ and $\beta \neq \beta^{-2}$, as otherwise we'd have a two dimensional eigenspace. Since a change of basis doesn't change the equivalence class of a representation, we can assume that

$$A = \begin{pmatrix} \alpha & 1 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha^{-2} \end{pmatrix}$$

Let $\{\underline{v}_1, \underline{v}_2, \underline{v}_3\}$ be a Jordan basis of B. Since $A^n = B^m$, we need for them to have the same eigenspaces. Thus, we require that $\alpha^n = \alpha^{-2n} = \beta^m = \beta^{-2m}$. We further require, by Corollary 5.2.1, that $\mathrm{Span}_{\mathbb{C}}(\underline{e}_1, \underline{e}_3) = \mathrm{Span}_{\mathbb{C}}(\underline{v}_1, \underline{v}_3)$, so we can write that for some $a, b, c, d, e, f \in \mathbb{C}$,

$$\underline{v}_1 = \begin{pmatrix} a \\ 0 \\ b \end{pmatrix} \qquad \underline{v}_2 = \begin{pmatrix} c \\ 1 \\ d \end{pmatrix} \qquad \underline{v}_3 = \begin{pmatrix} e \\ 0 \\ f \end{pmatrix}$$

where $\underline{v}_1, \underline{v}_3$ are linearly independent from each other and $\underline{e}_1, \underline{e}_3$. However, we also know from Theorem 5.2 that the Jordan decomposition of A^n, B^m are the same as those of A, B. Since $A^n = B^m$, they must have the same Jordan decomposition. Thus, we get that $\mathrm{Span}_{\mathbb{C}}(\underline{v}_1, \underline{v}_2) = \mathrm{Span}_{\mathbb{C}}(\underline{e}_1, \underline{e}_2)$, so $b = d = 0$. But then $\underline{v}_1$ is parallel to $\underline{e}_1$, creating a shared eigenspace for A, B, breaking irreducibility. $\square$

2 Figure-8 Knot Group

The Figure-8 Knot Group has presentation [Whi73]

$$\Gamma = \langle a, b \mid aba^{-1}ba = bab^{-1}ab \rangle \tag{4}$$

It's also worth noting an essential result from [Whi73]

Theorem 2.1. *Let $\Gamma = \langle a, b \mid aba^{-1}ba = bab^{-1}ab \rangle$ be a presentation of the figure-8 knot group; let Λ denote the set of all subgroups of $SL_2(\mathbb{C})$ which represent Γ. An abelian group $G \in \Lambda$ if and only if G is cyclic; a nonabelian group $G \in \Lambda$ if and only if $G = \langle A, B \rangle$, where $\mathrm{Tr}(A) = \mathrm{Tr}(B) = x$ is an arbitrary complex number and*

$$\mathrm{Tr}(AB) = \frac{1 + x^2 \pm \sqrt{(x^2-1)(x^2-5)}}{2} \tag{5}$$

2.1 Fundamental Group and Symmetries

2.1.1 Deriving the Fundamental Group

From the following diagram

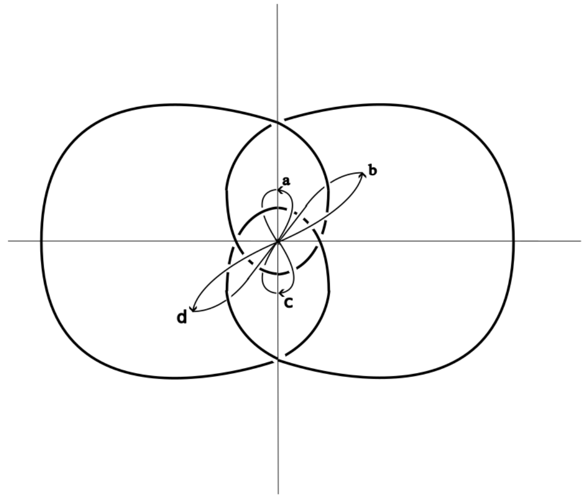

Figure 1: Figure-8 knot projection.

We get a presentation of the fundamental group

$$\Gamma = \langle a, b, c, d \mid ba = ac, da = bd, db = bc, ca = dc \rangle \tag{6}$$

This can be simplified to Equation 4 by noting that

$$c = a^{-1}ba \qquad\qquad d = ba^{-1}bab^{-1}$$

2.1.2 Symmetry Actions on the Fundamental Group

The simplest possible symmetry, denoted ρ^2, is a rotation of the knot by one half-turn about the axis going into the page in Figure 1. From Figure 1 we get that this acts by $\rho^2 : a \mapsto c, b \mapsto d$, or in the language of Equation 4

$$\rho^2 : a \mapsto a^{-1}ba, b \mapsto ba^{-1}bab^{-1} \tag{7}$$

Another symmetry, denoted σ, is given by a rotation of the knot by one half-turn about the vertical axis in Figure 1. This transforms the loops as in the following figure

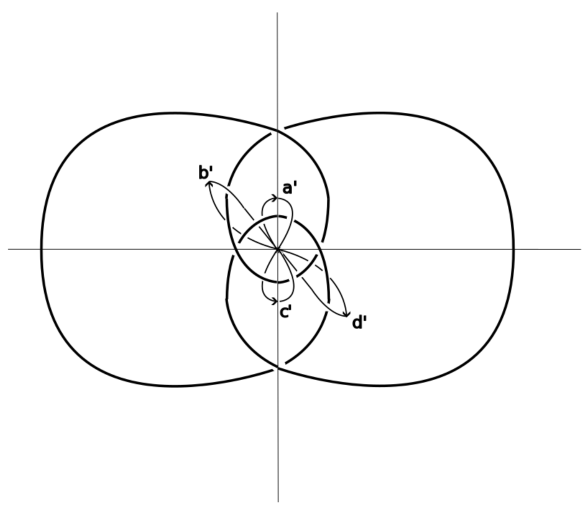

Figure 2: Figure-8 knot projection, with σ applied.

Reading off of Figure 2, we see that $\sigma : a \mapsto a^{-1}, b \mapsto a^{-1}c^{-1}a$, or

$$\sigma : a \mapsto a^{-1}, b \mapsto a^{-2}b^{-1}a^2 \tag{8}$$

Combining the two, we get a rotation $\rho^2\sigma$, which is a half-turn of the knot about the horizontal axis, with

$$\rho^2\sigma : a \mapsto a^{-1}b^{-1}a, b \mapsto a^{-1}b^{-1}a^{-1}ba \tag{9}$$

Now, we look at another presentation of this knot, given in Figure 3

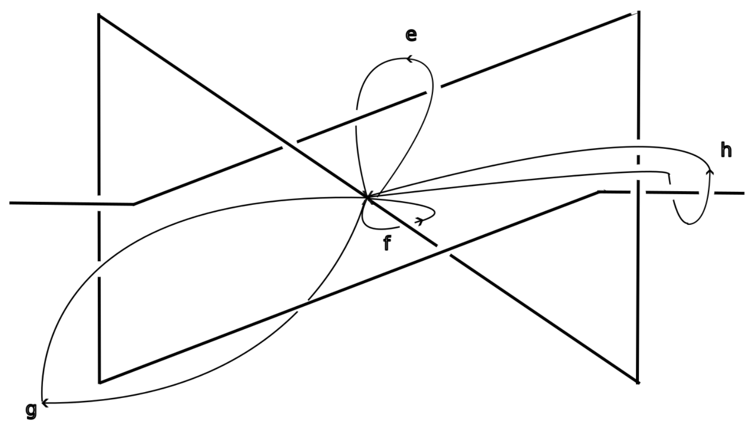

Figure 3: Figure-8 knot projection, line break, rotation, and deformations applied. The origin point of the loops is assumed to be above the plane of the screen.

The corresponding knot group has presentation

$$\Gamma = \langle e, f, g, h \mid eg = he, fe = hf, ge = fg, hg = fh \rangle \tag{10}$$

which we can put in the form of Equation 6, using e, h in place of a, b. Flipping Figure 3 one half-turn about the horizontal axis gives

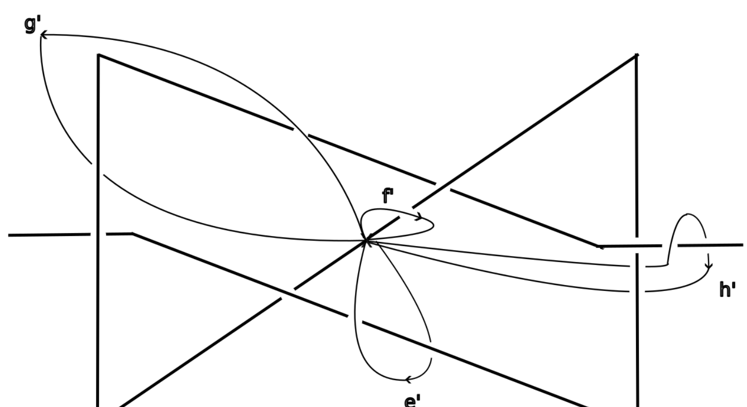

Figure 4: Figure 3 with a half-turn applied about the horizontal axis. The origin point of the loops is assumed to be below the plane of the screen.

Next, we reflect in the vertical/into the page plane to get

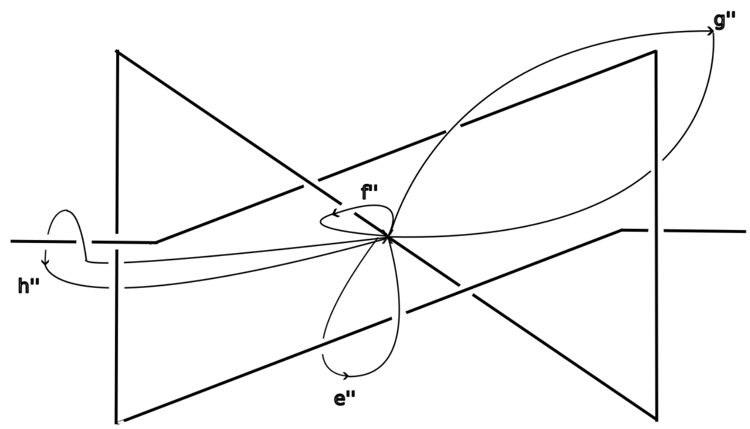

Figure 5: Figure 4 with a mirror applied across the vertical/into the screen plane axis. The origin point of the loops is assumed to be below the plane of the screen.

The loop f goes from the origin point in Figure 3 to the one in Figure 5 and back, and the transformation from Figure 3 to Figure 5 is the knot symmetry ρ. Thus, we can write that ρ acts by

$$\rho : e \mapsto f^{-1}gf, h \mapsto f^{-1}ghg^{-1}f$$

or equivalently, using $g = e^{-1}he, f = e^{-1}heh^{-1}e$ and the language of Equation 6, we get that

$$\rho : a \mapsto a^{-1}ba^{-1}bab^{-1}a, b \mapsto a^{-1}ba \tag{11}$$

2.2 Irreducible $\mathrm{SL}_2(\mathbb{C})$ Representations

2.2.1 Diagonalizable Representations

Cyclic representations are clearly reducible, so it is enough to consider only nonabelian representations. As shown in [Whi73], A, B are conjugate. Thus, as in subsubsection 1.1.1, we get that for some $\alpha \neq 0, \pm 1$

$$A = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix} \sim B$$

where the eigenspaces of B are

$$V_\alpha = \mathrm{Span}_{\mathbb{C}} \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \qquad V_{\alpha^{-1}} = \mathrm{Span}_{\mathbb{C}} \left\{ \begin{pmatrix} 1 \\ z \end{pmatrix} \right\}$$

with $z \neq 0, 1$. Let $x = \mathrm{Tr}(A) = \alpha + \alpha^{-1}$. Then by Theorem 2.1, we get a representation if and only if

$$\mathrm{Tr}(AB) = \frac{1 + x^2 \pm \sqrt{(x^2-1)(x^2-5)}}{2}$$

Through some arithmetic, we can show that this requires $x^2 \neq 5$ and

$$(x^2-5)z^2 - (x^2-5)(x^2-2)z - 1 = 0$$

Our choice of representation then amounts to a choice of tuple (α, k), where $k = \pm$ represents choosing from

$$z_k = \frac{(x^2-5)(x^2-2) \pm \sqrt{(x^2-5)^2(x^2-2)^2 + 4(x^2-5)}}{2(x^2-5)}$$

Our only restrictions are

1. $(\alpha + \alpha^{-1})^2 \neq 4, 5, \infty$

which also implies that $z_k \neq 0, 1$. Our equivalence is $(\alpha, z_k) \sim (\alpha^{-1}, z_k)$. It's also worth noting that we can equally view our representations as lying over the field

$$F = \mathrm{FF}\left(\frac{\mathbb{Q}[x,y,z]}{((x^2+y^2-3)z^2 - (x^2+y^2)(x^2+y^2-3)z - 1, xy-1)}\right) \tag{12}$$

2.2.2 Non-Diagonalizable Representations

Without loss of generality, we assume that A is not diagonalizable. As shown in [Whi73], A, B are conjugate. Clearly, we can get that

$$\begin{pmatrix} \pm 1 & 1 \\ 0 & \pm 1 \end{pmatrix} \sim A \sim B$$

from which it follows that B is not diagonalizable either. We first conjugate so that the eigenspace of B is $\mathrm{Span}_{\mathbb{C}}\{\underline{e}_2\}$. Since the representation is irreducible it follows that $\underline{e}_2$ is a generalized eigenvector of A, but not an eigenvector of A. Thus, conjugating so that $\underline{e}_1 = (A \mp \mathrm{Id}_2)\underline{e}_2$, we get that

$$A = \begin{pmatrix} \pm 1 & 1 \\ 0 & \pm 1 \end{pmatrix} \qquad B = \begin{pmatrix} \pm 1 & 0 \\ z & \pm 1 \end{pmatrix}$$

for some $z \neq 0$. By Theorem 2.1, the representation is valid if and only if

$$\mathrm{Tr}(AB) = \frac{1 + x^2 \pm \sqrt{(x^2-1)(x^2-5)}}{2}$$

Doing some arithmetic we arrive at the restriction $z = e^{i\frac{\pi}{3}}, e^{i\frac{5\pi}{3}}$. Therefore, up to conjugacy there are exactly four irreducible, non-diagonalizable representation of Γ in $\mathrm{SL}_2(\mathbb{C})$,which are given by

1. $$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 0 \\ e^{i\frac{\pi}{3}} & 1 \end{pmatrix}$$

2. $$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 0 \\ e^{i\frac{5\pi}{3}} & 1 \end{pmatrix}$$

3. $$A = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} \qquad B = \begin{pmatrix} -1 & 0 \\ e^{i\frac{\pi}{3}} & -1 \end{pmatrix}$$

4. $$A = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} \qquad B = \begin{pmatrix} -1 & 0 \\ e^{i\frac{5\pi}{3}} & -1 \end{pmatrix}$$

Conjugating again by the matrix

$$P = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

These go into the form

1. $$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B = -e^{i\frac{\pi}{3}} \begin{pmatrix} 1 & 1 \\ 1 & 1 - e^{i\frac{5\pi}{3}} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 - e^{i\frac{5\pi}{3}} & -1 \\ -1 & 1 \end{pmatrix}$$

2. $$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B = -e^{i\frac{5\pi}{3}} \begin{pmatrix} 1 & 1 \\ 1 & 1 - e^{i\frac{\pi}{3}} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 - e^{i\frac{\pi}{3}} & -1 \\ -1 & 1 \end{pmatrix}$$

3. $$A = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} \qquad B = -e^{i\frac{\pi}{3}} \begin{pmatrix} 1 & 1 \\ 1 & 1 - e^{i\frac{5\pi}{3}} \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 - e^{i\frac{5\pi}{3}} & -1 \\ -1 & 1 \end{pmatrix}$$

4.

$$A = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} \qquad B = -e^{i\frac{5\pi}{3}} \begin{pmatrix} 1 & 1 \\ 1 & 1 - e^{i\frac{\pi}{3}} \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 - e^{i\frac{\pi}{3}} & -1 \\ -1 & 1 \end{pmatrix}$$

which is in the same form as subsubsection 2.2.1, with $z = 1 - e^{i\frac{\pi}{3}}, 1 - e^{i\frac{5\pi}{3}}$. Note that the equation for z_k in that section will not give the correct values of z in this case, and our field from Equation 12 will hence no longer suffice, with no natural replacement.

2.2.3 Action Under Group Automorphisms

As verified in the raw code in subsection 5.3, all the group automorphisms from subsubsection 2.1.2 arising from symmetries in which the origin point of the loops is fixed act trivially on the equivalence classes of irreducible representations. The exact conjugation matrices needed to show this are also computed in the raw and clean code in subsection 5.3. The automorphism given by Equation 11 acts non-trivially on the group, as verified in the raw code of subsection 5.3, with an exact action yet to be determined. The results of the code can also be used to prove the following three theorems. For each theorem, let R be the ring of which Equation 12 is the fraction field, and consider only diagonalizable representations.

Theorem 2.2. *Let $P \in GL_2(\mathbb{C})$ be a matrix such that $\rho^2(X) = P^{-1}XP$, for any X in a given irreducible representation. Then* $\det(P)$ *is not a square in the field given by Equation 12.*

Proof. By the results of block 3 in the clean code of subsection 5.3, we can write that $\det(P) = a^2\frac{b}{c}$, where $a \in \mathbb{C}^\times$ is arbitrary and $b, c \in R$. Suppose that $\exists q \in F$ such that $q^2 = a^2\frac{b}{c}$. If this is true for any particular value of a, then $a^2 \in F$, so we may assume without loss of generality that $a^2 = 1$. Furthermore, $\exists m, n \in R$ such that $n \neq 0$ and $m^2 = n^2\frac{b}{c}$. We define the homomorphism $\varphi : R \to \mathbb{R}$, which is the identity on $\mathbb{Q}$, by the following rules

$$\varphi : \bar{x} \mapsto \sqrt{3}, \bar{y} \mapsto \frac{1}{\sqrt{3}}, \bar{z} \mapsto \frac{5 + 2\sqrt{13}}{3}$$

A quick arithmetic check shows that this is indeed a homomorphism. Now $m^2 = n^2\frac{b}{c}$ has a solution only if $\varphi(m^2) = \varphi(n^2\frac{b}{c})$ has a solution. That is, we must get a solution to

$$\varphi\Big(\frac{m}{n}\Big)^2 = \varphi\Big(\frac{b}{c}\Big)$$

But $\varphi\Big(\frac{m}{n}\Big) \in \mathbb{R}$, and by the result of block 4 in the clean code of subsection 5.3 $\varphi\Big(\frac{b}{c}\Big) < 0$, so this is not possible. $\square$

Theorem 2.3. *Let $P \in GL_2(\mathbb{C})$ be a matrix such that $\sigma(X) = P^{-1}XP$, for any X in a given irreducible representation. Then* $\det(P)$ *is not a square in the field given by Equation 12.*

Proof. By the results of block 15 in the clean code subsection 5.3, we can apply some simplifications to get that $\det(P) = -a^2\bar{z}\bar{x}^4$, where $a \in \mathbb{C}^\times$ is arbitrary. Suppose that $\exists q \in F$ such that $q^2 = -a^2\bar{z}\bar{x}^4$. If this is true for any particular value of a, then $a^2 \in F$, so we may assume without loss of generality that $a^2 = 1$. Furthermore, $\exists m, n \in R$ such that $n \neq 0$ and $m^2 = -n^2\bar{z}\bar{x}^2$. We define the homomorphism $\varphi : R \to \mathbb{R}$, which is the identity on $\mathbb{Q}$, by the following rules

$$\varphi : \bar{x} \mapsto \sqrt{3}, \bar{y} \mapsto \frac{1}{\sqrt{3}}, \bar{z} \mapsto \frac{5 + 2\sqrt{13}}{3}$$

A quick arithmetic check shows that this is indeed a homomorphism. Now $m^2 = -n^2\bar{z}\bar{x}^4$ has a solution only if $\varphi(m^2) = \varphi(-n^2\bar{z}\bar{x}^4)$ has a solution. That is, we must get a solution to

$$\varphi\Big(\frac{m}{n}\Big)^2 = -9 \cdot \frac{5 + 2\sqrt{13}}{3}$$

But $\varphi\Big(\frac{m}{n}\Big) \in \mathbb{R}$, so this is not possible. $\square$

Theorem 2.4. *Let $P \in GL_2(\mathbb{C})$ be a matrix such that $\rho^2\sigma(X) = P^{-1}XP$, for any X in a given irreducible representation. Then $\det(P)$ is not a square in the field given by Equation 12.*

Proof. By the results of block 25 in the clean code of subsection 5.3, $\det(P) = a^2(\bar{z}-1)$, where $a \in \mathbb{C}^\times$ is arbitrary. Suppose that $\exists q \in F$ such that $q^2 = a^2(\bar{z}-1)$. then $a^2 \in F$, so we may assume without loss of generality that $a^2 = 1$. Furthermore, $\exists m, n \in R$ such that $n \neq 0$ and $m^2 = (\bar{z}-1)n^2$. We define the homomorphism $\varphi : R \to \mathbb{R}$, which is the identity on $\mathbb{Q}$, by the following rules

$$\varphi : \bar{x} \mapsto \sqrt{3}, \bar{y} \mapsto \frac{1}{\sqrt{3}}, \bar{z} \mapsto \frac{5-2\sqrt{13}}{3}$$

A quick arithmetic check shows that this is indeed a homomorphism. Now $m^2 = (\bar{z}-1)n^2$ has a solution only if $\varphi(m^2) = \varphi((\bar{z}-1)n^2)$ has a solution. That is, we must get a solution to

$$\varphi\Big(\frac{m}{n}\Big)^2 = \frac{5-2\sqrt{13}}{3} - 1 < 0$$

But $\varphi\Big(\frac{m}{n}\Big) \in \mathbb{R}$, so this is not possible. $\square$

The next results are all to do with the field

$$E = \mathrm{FF}\Big(\frac{\mathbb{Q}[q,w]}{(w^2-(1+q^2)w+(2q^2-1))}\Big) \tag{13}$$

which arises naturally from the character variety of all representations [Whi73]. We have the relations $w = \mathrm{Tr}(AB), q = x + y$, which lead to a natural ramified covering of E by F. Note also that the polynomial generating the ideal being divided out in E is equivalent to $(x^2+y^2-3)z^2 - (x^2+y^2)(x^2+y^2-3)z - 1$ for diagonal representations. Finally, one can derive that for diagonal representations

$$z = \frac{w-2}{w+2-q^2} \tag{14}$$

So $z \in E$. This brings us to our two theorems.

Theorem 2.5. *For diagonal representations $x \notin E$.*

Proof. Since $z \in E$ depends on w and q, w can be written as a rational function of q, z, so $x \in E$ if and only if it can be written as a fraction in terms of q and z. Since $q = x + y$, we get that $(x^2+y^2-3)z^2 - (x^2+y^2)(x^2+y^2-3)z - 1 = 0$ is equivalent to $(q^2-5)z^2 - (q^2-2)(q^2-5)z - 1 = 0$, so the numerator and denominator of our fraction can be written as linear polynomials in z. That is, we can write that

$$x = \frac{az+b}{cz+d} \Rightarrow z(a-cx) + (b-dx) = 0$$

where $a, b, c, d \in \mathrm{FF}(\mathbb{Q}[q])$. We thus need for $a = cx$, which requires x to be a well-defined function of q. But since $yx = 1$, $q = x + x^{-1}$, which has two solutions for $x \in \mathbb{C}$ for each $q \in \mathbb{C}$, a contradiction. $\square$

Theorem 2.6. *For any conjugation matrix P which applies the action of the $\rho^2\sigma$, σ, or ρ^2 transformations, there exists some constants $n \in \mathbb{C}, r \in F^\times$ and some $a, b, c, d \in E$ such that $nrP = a\,\mathrm{Id}_2 + bA + cB + dAB$.*

Proof. As can be seen in blocks 5, 16 and 26 of the clean code in subsection 5.3, there are certainly $a, b, c, d \in F$ which make the statement true for $r = n = 1$. Thus, it suffices to find some $r \in F^\times$ such that $ra, rb, rc, rd \in E$. We do this first for the $\rho^2\sigma$ transformation. Choosing $r = x - y$, we can get that

$$ra = q^3 - 3q - qz \qquad rb = z + 3 - q^2 \qquad rc = (q^2-1)(z-1) \qquad rd = q(1-z)$$

as required. For the σ transformation, we choose $r = x^{-2}$ to get

$$ra = \frac{q^4-4q^2+2-(q^2-2)z}{q^2-4} \qquad rb = -\frac{q^3-3q-qz}{q^2-4} \qquad rc = \frac{(q^3-3q)(z-1)}{q^2-4} \qquad rd = \frac{(q^2-2)(1-z)}{q^2-4}$$

as required. From block 6 of the clean code in subsection 5.3, we have that for ρ^2 (ignoring denominators which lie in E)

$$\begin{aligned}
a &= -x(x-1)^4(x+1)^4(x^2+1)(x^4z - x^2z - x^2 + z)\\
b &= x^2(x-1)^4(x+1)^4(2x^4z - 3x^2z - x^2 + 2z)\\
c &= x^2(x^2-x-1)(x^2+x-1)(-x^4+2x^2z-1)(x^4z - x^2z - x^2 + z)\\
d &= -(x-1)^2(x+1)^2x^3(x^2+1)(x^4z - 3x^2z + x^2 + z)
\end{aligned}$$

We may factor each of these as follows

$$\begin{aligned}
a &= -x^4(x^2-1)^4(x+x^{-1})(z(x^2-1+x^{-2})-1) = -x^4(x^4-2x^2+1)^2q(z(q^2-3)-1)\\
&= -x^8(x^2-2+x^{-2})^2q(zq^2-3z-1) = -x^8(q^2-4)^2q(zq^2-3z-1)\\
b &= x^4(x^4-2x^2+1)^2(z(2x^2-3+2x^{-2})-1) = x^8(x^2-2+x^{-2})^2(z(2(q^2-2)-3)-1)\\
&= x^8(q^2-4)^2(z(2q^2-7)-1)\\
c &= x^6(x^4-2x^2+1)(2z-(x^2+x^{-2}))(z(x^2-1+x^{-2})-1) = x^8(q^2-4)(2z-(q^2-2))(z(q^2-3)-1)\\
d &= -x^6(x^2-1)^2(x+x^{-1})(z(x^2-3+x^{-2})+1) = -x^8(q^2-4)q(z(q^2-5)+1)
\end{aligned}$$

So multiplying each through by x^{-8} moves them into E, as required. The cummulative result here is to get that for $r = x^{-1}(x+1)(x-1)$

$$ra = -\frac{(q^2-4)^2q(zq^2-3z-1)}{z(z-1)^2} \qquad rb = \frac{(q^2-4)^2(z(2q^2-7)-1)}{z(z-1)^2}$$

$$rc = \frac{(q^2-4)(2z-q^2+2)(z(q^2-3)-1)}{z(z-1)} \qquad rd = -\frac{(q^2-4)q(z(q^2-5)+1)}{z(z-1)}$$

□

We can prove similar theorems for the non-diagonal representations. As mentioned before, non-diagonal representations do not have an analogue for Equation 12, but are still described by the field arising from the character variety given by Equation 13. One can also derive that for non-diagonalizable representations

$$z = (2-w)^{-1} + 1 \tag{15}$$

which allows us to prove the following theorem for non-diagonalizable representations

Theorem 2.7. *For any conjugation matrix P which applies the action of the $\rho^2\sigma$, σ, or ρ^2 transformations, there exists some $r \in \mathbb{C}$ and $a, b, c, d \in E$ such that $rP = a\,\mathrm{Id}_2 + bA + cB + dAB$.*

Proof. This follows immediately from Equation 15 and the results of blocks 9, 12, 19, 22, 29, and 32 of the clean code in subsection 5.3. □

Finally, we can unify these solutions with those from Theorem 2.6 by using the following tricks, along with Equation 14 and Equation 15.

For $\rho^2\sigma$, we multiply the result from Theorem 2.6 by $-\frac{w+2-q^2}{q^2-4}$, multiply the result for $q=-2$ by $-2(2-w)$, and substitute in w for z to get that in all cases

$$a = q(q^2-w-1) \qquad b = w - q^2 + 1 \qquad c = 1 - q^2 \qquad d = q$$

For σ, we multiply the result from Theorem 2.6 by $w+2-q^2$, multiply the result for $q = \pm 2$ by $(2-w)$, and substitute in w for z to get that in all cases

$$a = w(q^2-1) - q^2(q^2-2) \qquad b = q(q^2-w-1) \qquad c = q(q^2-3) \qquad d = 2 - q^2$$

Finally, since $\rho^2\sigma \cdot \sigma = \rho^2$, we can multiply our past two results (with the σ result coming first) to get the following for ρ^2 in all cases (this is done in block 42 in the clean code in subsection 5.3)

$$a = -q(q^2 - w)(q^2 - w - 2) \qquad b = (q^2 - w + 1)(q^2 - w - 2)$$
$$c = (q^2 - 1)(q^2 - w - 2) \qquad d = -q(q^2 - w - 2)$$

The above calculation also requires the following identities, which hold for all irreducible representations (these come from blocks 33-41 of the clean code from subsection 5.3, along with Equation 14 and Equation 15)

$$A^2 = qA - \mathrm{Id}_2 \qquad B^2 = qB - \mathrm{Id}_2 \qquad BA = (w - q^2)\,\mathrm{Id}_2 + qA + qB - AB$$

Furthermore, we can remove the common factor of $-(q^2 - w - 2)$ to get the simplified result

$$a = q(q^2 - w) \qquad b = w - q^2 - 1 \qquad c = 1 - q^2 \qquad d = q$$

All of these calculations are checked in blocks 43-53 of the clean code in subsection 5.3.

3 9-48 Knot Group

Knot 9-48 on the Rolfsen table is given by the presentation in Figure 6.

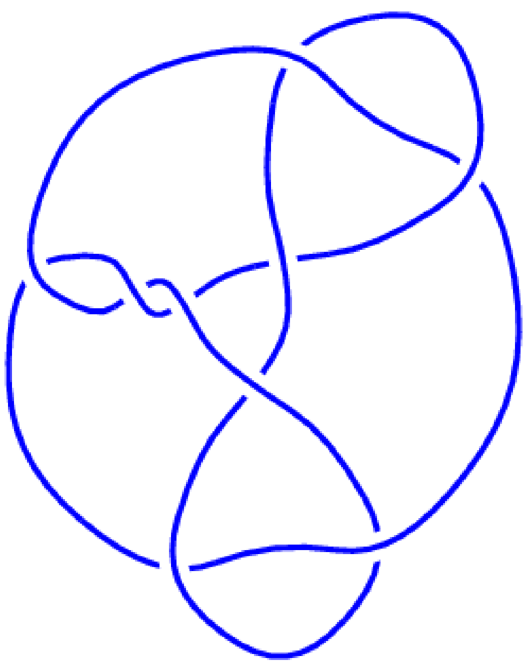

Figure 6: 9-48 knot projection, as given in [LM23].

A presentation of this knot in $\mathbb{R}^3$ is given in Figure 7

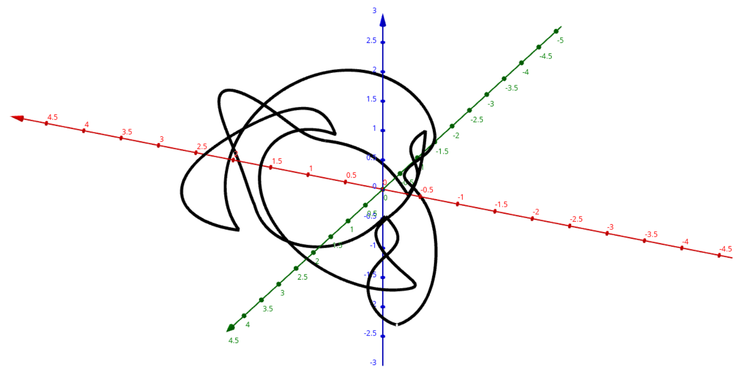

Figure 7: 9-48 knot presentation in $\mathbb{R}^3$.

Of course this $\mathbb{R}^3$ presentation was generated through trial and error, so it remains to prove that it is in fact knot 9-48. To do so, we first consider its projection, given in Figure 8.

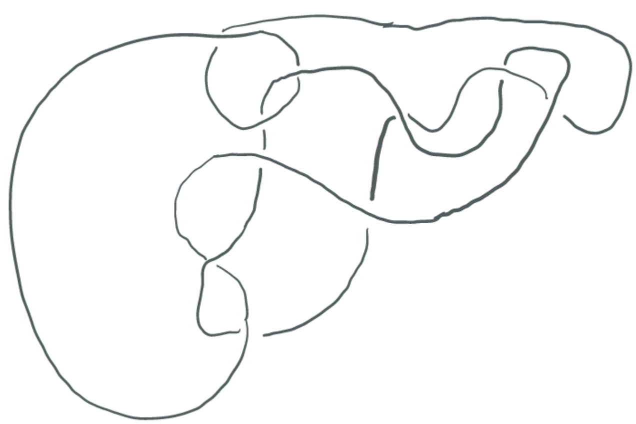

Figure 8: Projection of 9-48 knot presentation in Figure 7.

By sliding the middle arc with two over crossings down, we obtain the following presentation in Figure 9, with an arbitrarily chosen orientation and crossing numbers.

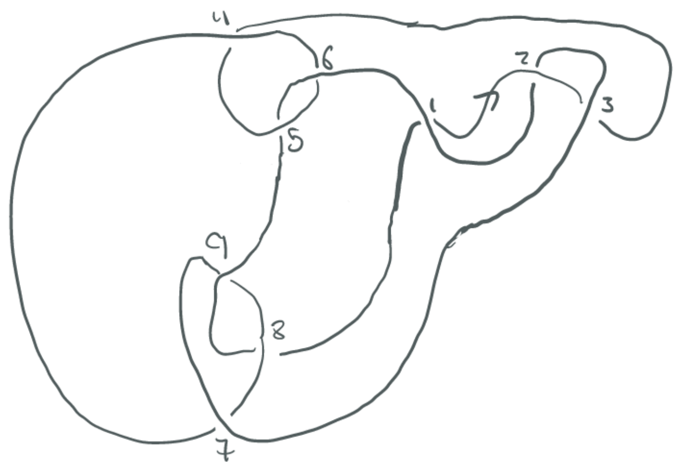

Figure 9: Projection of 9-48 knot presentation in Figure 7.

It can be verified then that this presentation has the same Gauss code as that given in Figure 6, with the crossing numbers given in Figure 10.

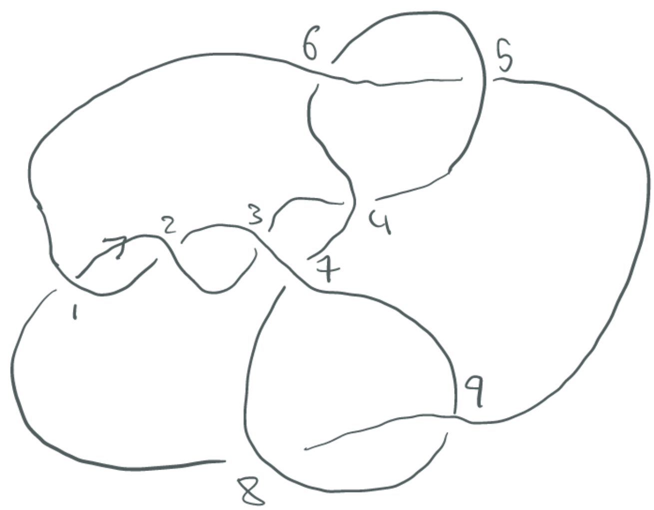

Figure 10: 9-48 knot projection, as given in [LM23].

making our knot either 9-48 or its mirror image, as desired.

3.1 Free 3-Period Symmetry

We spend this section building up to the following theorem.

Theorem 3.1. *The representation of 9-48 given in Figure 7 is freely 3-periodic.*

In order to do this, we first need to establish the precise parameterization of the knot at hand. We begin with the following parameterization of S^3

$$x_0(u,v,n) = \cos(u)\sin(n) \quad x_1(u,v,n) = \sin(u)\sin(n) \quad x_2(u,v,n) = \cos(v)\cos(n) \quad x_3(u,v,n) = \sin(v)\cos(n)$$

where $0 \leq u, v < 2\pi$ and $0 \leq n \leq \pi/2$. We may project this into $\mathbb{R}^3 \cup \{\infty\}$ by the equations

$$x(u,v,n) = \frac{x_0(u,v,n)}{1 - x_3(u,v,n)} \qquad y(u,v,n) = \frac{x_1(u,v,n)}{1 - x_3(u,v,n)} \qquad z(u,v,n) = \frac{x_2(u,v,n)}{1 - x_3(u,v,n)}$$

With this projection, any curve can be parameterized in the following form.

$$\underline{r}(t) = x(u(t), v(t), n(t))\underline{e}_1 + y(u(t), v(t), n(t))\underline{e}_2 + z(u(t), v(t), n(t))\underline{e}_3$$

where $u(t), v(t), n(t)$ are all functions. Using this, we can give the representation of 9-48 in $\mathbb{R}^3$ by the union of the following twelve curves (ignoring orientation).

$$\begin{aligned}
K_1 &: u(t) = t, v(t) = t, n(t) = \frac{\pi}{4} + \frac{1}{10}\sin\left(\frac{3}{2}t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
K_2 &: u(t) = t, v(t) = t, n(t) = \frac{\pi}{4} - \frac{1}{10}\sin\left(\frac{3}{2}t - \frac{\pi}{2}\right), \frac{2\pi}{3} \leq t \leq \frac{4\pi}{3} \\
K_3 &: u(t) = t, v(t) = t, n(t) = \frac{\pi}{4} + \frac{1}{10}\sin\left(\frac{3}{2}t - \frac{\pi}{2}\right), \frac{4\pi}{3} \leq t \leq 2\pi \\
S_1 &: u(t) = t, v(t) = t + \frac{2\pi}{3}, n(t) = \frac{\pi}{4} - \frac{1}{10}\sin\left(\frac{3}{2}t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
S_2 &: u(t) = t, v(t) = t + \frac{2\pi}{3}, n(t) = \frac{\pi}{4} + \frac{1}{10}\sin\left(\frac{3}{2}t - \frac{\pi}{2}\right), \frac{2\pi}{3} \leq t \leq \frac{4\pi}{3} \\
S_3 &: u(t) = t, v(t) = t + \frac{2\pi}{3}, n(t) = \frac{\pi}{4} - \frac{1}{10}\sin\left(\frac{3}{2}t - \frac{\pi}{2}\right), \frac{4\pi}{3} \leq t \leq 2\pi \\
L &: u(t) = \frac{1}{4}\sin(3t), v(t) = t, n(t) = \frac{\pi}{4} - \frac{1}{10}\sin\left(3t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
M &: u(t) = -\frac{1}{4}\sin(3t), v(t) = t, n(t) = \frac{\pi}{4} + \frac{1}{10}\sin\left(3t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
N &: u(t) = \frac{1}{4}\sin(3t) + \frac{2\pi}{3}, v(t) = t + \frac{2\pi}{3}, n(t) = \frac{\pi}{4} - \frac{1}{10}\sin\left(3t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
O &: u(t) = -\frac{1}{4}\sin(3t) + \frac{2\pi}{3}, v(t) = t + \frac{2\pi}{3}, n(t) = \frac{\pi}{4} + \frac{1}{10}\sin\left(3t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
P &: u(t) = \frac{1}{4}\sin(3t) + \frac{4\pi}{3}, v(t) = t + \frac{4\pi}{3}, n(t) = \frac{\pi}{4} - \frac{1}{10}\sin\left(3t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3} \\
Q &: u(t) = -\frac{1}{4}\sin(3t) + \frac{4\pi}{3}, v(t) = t + \frac{4\pi}{3}, n(t) = \frac{\pi}{4} + \frac{1}{10}\sin\left(3t - \frac{\pi}{2}\right), 0 \leq t \leq \frac{2\pi}{3}
\end{aligned}$$

We claim that the symmetry is given by $u \mapsto u + \frac{4\pi}{3}, v \mapsto v + \frac{4\pi}{3}, n \mapsto n$. It's clear that this map results in the following four cycles

$$\begin{array}{cc}
L \mapsto P \mapsto N \mapsto L & M \mapsto Q \mapsto O \mapsto M \\
K_1 \mapsto K_3 \mapsto K_2 \mapsto K_1 & S_1 \mapsto S_3 \mapsto S_2 \mapsto S_1
\end{array}$$

Demonstrating that this is in fact a period 3 symmetry of the knot. In fact, this symmetry fixes no point, and hence is freely 3-periodic. Thus, Theorem 3.1 is proven.

3.2 Knot Group and Transformation

From the following diagram

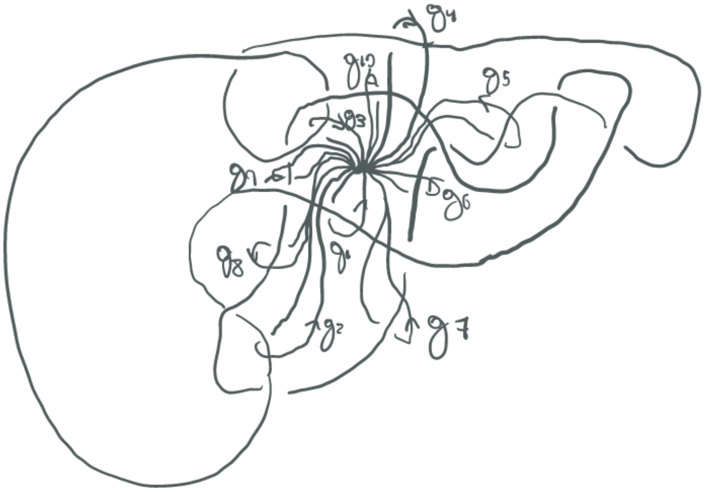

Figure 11: 9-48 knot projection, as given in Figure 7, with knot group generators drawn.

We get the following presentation of the knot group for 9-48.

$$\begin{aligned}\Gamma = \langle g_1, \dots, g_{10} \mid\ & g_1 g_8 = g_9 g_1, g_8 = g_2 g_8 g_1, g_2 g_8 = g_7 g_2, g_2 = g_4 g_2 g_3, \\ & g_{10} g_2 = g_3 g_{10}, g_3 g_{10} = g_9 g_3, g_1 g_4 = g_5 g_1, g_5 g_{10} = g_1 g_5, g_{10} g_5 = g_6 g_{10}, g_1 g_7 = g_6 g_1 \rangle\end{aligned}$$

These can be simplified to the following presentation, as calculated in the 9-48 code in subsection 5.3

$$\Gamma = \langle g_1, g_2, g_4 \mid g_2 g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2 g_1 g_2^{-1} g_4^{-1} g_1 g_4^{-1} g_1^{-1} g_4, g_2^{-1} g_1 g_4^{-1} g_1^{-1} g_4 g_1^{-1} g_2^{-1} g_4^{-1} g_2 g_1 g_4^{-1} g_1 g_4 g_1^{-1} \rangle \tag{16}$$

By parameterizing the loops of the knot group in Geogebra, we were able to find that the free 3-periodic symmetry κ acts on the fundamental group according to the following rules

$$\kappa : g_1 \mapsto g_9, g_2 \mapsto g_{10}^{-1}, g_4 \mapsto g_1$$

or in the form of Equation 16

$$\kappa : g_1 \mapsto g_1 g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2, g_2 \mapsto g_1 g_4^{-1} g_1 g_4 g_1^{-1}, g_4 \mapsto g_1 \tag{17}$$

Note that in theory, κ^3 should act by conjugation. κ^3 has its action calculated in the 9-48 code in subsection 5.3, and this indeed allows us to verify this property. We start with g_4

$$\begin{aligned}\kappa^3(g_4) &= g_1 (g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2)^2 g_1 g_2^{-1} g_4^{-1} (g_1 g_4^{-1} g_1^{-1} g_4)^2 g_1^{-1} \\ &= g_1 (g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2)^2 (g_1 g_2^{-1} g_4^{-1} g_1 g_4^{-1} g_1^{-1} g_4)(g_1 g_4^{-1} g_1^{-1} g_4) g_1^{-1} \\ &= g_1 (g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2)^2 (g_2^{-1} g_4^{-1} g_1 g_4^{-1} g_1^{-1} g_4 g_2^{-1})(g_1 g_4^{-1} g_1^{-1} g_4) g_1^{-1} \\ &= (g_1 g_4^{-1} g_1 g_4 g_1^{-1}) g_4 (g_1 g_4^{-1} g_1^{-1} g_4 g_1^{-1}) = (g_1 g_4^{-1} g_1 g_4 g_1^{-1}) g_4 (g_1 g_4^{-1} g_1 g_4 g_1^{-1})^2\end{aligned}$$

Note that using the relations calculated in the 9-48 code, we can re-write this as

$$\kappa^3(g_4) = g_{10} g_4 g_{10}^{-1} = (g_{10} g_4) g_4 (g_{10} g_4)^{-1}$$

We will in fact show that for any $x \in \Gamma$, $\kappa^3(x) = (g_{10} g_4) x (g_{10} g_4)^{-1}$. To do this, we first perform the preliminary calculation

$$\begin{aligned}\kappa(g_{10}) &= \kappa(g_1 g_4^{-1} g_1 g_4 g_1^{-1}) = g_9 g_1^{-1} g_9 g_1 g_9^{-1} \\ &= (g_1 g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2) g_1^{-1} (g_1 g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2) g_1 (g_2^{-1} g_4^{-1} g_1 g_4^{-1} g_1^{-1} g_4 g_1^{-1}) \\ &= g_1 g_4^{-1} g_1 g_4 g_1^{-1} g_4 (g_2 g_4^{-1} g_1 g_4 g_1^{-1} g_4 g_2 g_1 g_2^{-1} g_4^{-1} g_1 g_4^{-1} g_1^{-1} g_4) g_1^{-1} = g_{10} g_4 g_1^{-1}\end{aligned}$$

Using this and $\kappa(g_4) = g_1$, we can get that

$$\kappa^3(g_1) = \kappa^4(g_4) = \kappa(g_{10}g_4g_{10}^{-1}) = g_{10}g_4g_1^{-1}g_1g_1g_4^{-1}g_{10}^{-1} = (g_{10}g_4)g_1(g_{10}g_4)^{-1}$$

so κ^3 acts on g_1 by this conjugation. Finally, using all these results we get that

$$\kappa^3(g_2) = \kappa^2(g_{10}^{-1}) = \kappa(g_1g_4^{-1}g_{10}^{-1}) = g_9g_1^{-1}g_1g_4^{-1}g_{10}^{-1} = g_9g_4^{-1}g_{10}^{-1}$$

From the relations calculated in the 9-48 code, $g_9 = g_{10}g_4g_2$, so $\kappa^3(g_2) = (g_{10}g_4)g_2(g_{10}g_4)^{-1}$, completing the proof.

4 Transformations in CP^2

This section characterizes transformations of the complex projective space in $\mathbb{C}^3$. We start with some general concepts.

Definition 4.1. *We say that four vectors in $\mathbb{C}^3$ are in general position if none of them is in the span of two others.*

Lemma 4.2. *If $\{\underline{v}_j\}_{j=1}^4 \subset \mathbb{C}^3$ are in general position, then any set of three of them is spanning.*

Proof. It's clear that any pair of vectors cannot be collinear. Furthermore, if we pick any pair of vectors, then any third vector chosen will not be in their span, and hence together they form a basis for $\mathbb{C}^3$. □

Theorem 4.3. *If $\{\underline{v}_j\}_{j=1}^4 \subset \mathbb{C}^3$ are in general position, then any $\varphi \in PGL(3, \mathbb{C})$ is uniquely determined by its action on $\{[\underline{v}_j]\}_{j=1}^4$.*

Proof. Let $T \in \mathrm{Aut}_{\mathbb{C}}(\mathbb{C}^3)$ be a representative of φ, and let $\varphi([\underline{v}_j]) = [\underline{u}_j]$. Then $\exists r_j \in \mathbb{C}^\times$ such that $T(\underline{v}_j) = r_j\underline{u}_j$. Since φ is invariant under scalar multiplication of T, we can choose for $r_4 = 1$. By Lemma 4.2, $\{\underline{v}_j\}_{j=1}^3$ is a basis for $\mathbb{C}^3$. Hence, $\exists! u_{jk} \in \mathbb{C}$ such that $\underline{u}_j = \sum_{k=1}^3 u_{jk}\underline{v}_k$, and $\exists! v_{4j} \in \mathbb{C}$ such that $\underline{v}_4 = \sum_{j=1}^3 v_{4j}\underline{v}_j$. Gathering all this together, we get that

$$T(\underline{v}_4) = \underline{u}_4 = \sum_{j=1}^3 u_{4j}\underline{v}_j$$

$$T(\underline{v}_4) = \sum_{j=1}^3 v_{4j}r_j\underline{u}_j = \sum_{j=1}^3 v_{4j}r_j \sum_{k=1}^3 u_{jk}\underline{v}_k = \sum_{j=1}^3 \underline{v}_j \sum_{k=1}^3 v_{4k}r_k u_{kj}$$

Therefore, the r_j solve the system of equations

$$\begin{aligned} u_{41} &= r_1v_{41}u_{11} + r_2v_{42}u_{21} + r_3v_{43}u_{31} \\ u_{42} &= r_1v_{41}u_{12} + r_2v_{42}u_{22} + r_3v_{43}u_{32} \\ u_{43} &= r_1v_{41}u_{13} + r_2v_{42}u_{23} + r_3v_{43}u_{33} \end{aligned}$$

or more compactly

$$\begin{pmatrix} u_{41} \\ u_{42} \\ u_{43} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{pmatrix} \begin{pmatrix} r_1v_{41} \\ r_2v_{42} \\ r_3v_{43} \end{pmatrix}$$

Since T represents a transformation in PGL$(3, \mathbb{C})$, it is invertible, so $\{\underline{u}_1, \underline{u}_2, \underline{u}_3\}$ are linearly independent making the matrix invertible, and $\underline{u}_4 \neq \underline{0}$. Furthermore, since $\{\underline{v}_j\}_{j=1}^4 \subset \mathbb{C}^3$ are in general position, $\underline{v}_4$ cannot have a component of zero along any other $\underline{v}_k \in \{\underline{v}_j\}_{j=1}^3$, as otherwise it would be in the span of the two remaining vectors. Hence, this equations has a unique, non-zero solution for r_1, r_2, r_3, giving a unique transformation T and therefore a unique φ. □

Note that Theorem 4.3 does not guarantee the existence of a $\varphi \in$ PGL$(3, \mathbb{C})$ for any choice of $\underline{u}_j \in \mathbb{C}^4$, just the uniqueness of a well-specified transformation.

Theorem 4.4. *Let $\{\underline{v}_j\}_{j=1}^4 \subset \mathbb{C}^3$ be in general position, and specify $\varphi \in PGL(3, \mathbb{C})$ by the following four relations*

$$\varphi([\underline{v}_1]) = [\underline{v}_2] \qquad \varphi([\underline{v}_2]) = [\underline{v}_1] \qquad \varphi([\underline{v}_3]) = [\underline{v}_3] \qquad \varphi([\underline{v}_4]) = [\underline{v}_4]$$

Then in the basis $\{\underline{v}_1, \underline{v}_2, \underline{v}_3\}$, φ can be represented by a transformation of the form

$$T = \begin{pmatrix} 0 & v_{41}v_{42}^{-1} & 0 \\ v_{42}v_{41}^{-1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and φ is an involution.

Proof. Using the same notation as in Theorem 4.3, we get that $T(\underline{e}_1) = r_1\underline{e}_2, T(\underline{e}_2) = r_2\underline{e}_1, T(\underline{e}_3) = r_3\underline{e}_3$, so

$$v_{41} = r_2 v_{42} \qquad v_{42} = r_1 v_{41} \qquad v_{43} = r_3 v_{43}$$

So $r_1 = v_{41}^{-1}v_{42}, r_2 = v_{42}^{-1}v_{41}$, and $r_3 = 1$.Therefore, in the basis $\{\underline{v}_1, \underline{v}_2, \underline{v}_3\}$we get

$$T = \begin{pmatrix} 0 & v_{41}v_{42}^{-1} & 0 \\ v_{42}v_{41}^{-1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

as was to be shown. Since $T^2 = \mathrm{Id}_3$, φ is an involution. □

References

[LM23] C. Livingston and A. H. Moore, Knotinfo: Table of knot invariants, URL: `https://knotinfo.math.indiana.edu`, August 2023.

[Muñ09] V. Muñoz, The SL(2, ℂ)-character varieties of torus knots, *Revista Matemática Complutense* **22** no. 2 (2009), 489–497. `https://doi.org/10.48550/arXiv.0901.1783`.

[MP16] V. Muñoz and J. Porti, Geometry of the SL(3, ℂ)-character variety of torus knots, *Algebraic and Geometric Topology* **16** no. 1 (2016), 397–426. `https://doi.org/10.2140/agt.2016.16.397`.

[Whi73] A. Whittemore, On representations of the group of Listring's knot by subgroups of SL(2, ℂ), *Proceedings of the American Mathematical Society* **40** no. 2 (1973), 378–382. `https://doi.org/10.1090/S0002-9939-1973-0316585-3`.

5 Appendix

5.1 Jordan Decomposition

Lemma 5.1. *Let V be a vector space over an algebraically closed field F, and let $T \in \mathrm{End}_F(V)$. Then for any $n \in \mathbb{N}$, $\mathrm{Spec}_F(T^n) = \{\lambda^n \mid \lambda \in \mathrm{Spec}_F(T)\}$.*

Proof. First, suppose that $\lambda \in \mathrm{Spec}_F(T)$. Then $\exists \underline{v} \in V$ such that $(T - \lambda \,\mathrm{Id}_V)\underline{v} = \underline{0}$, so

$$(T^n - \lambda^n \,\mathrm{Id}_V)\underline{v} = \Big(\sum_{j=0}^{n-1} T^j \lambda^{n-j-1}\Big)(T - \lambda \,\mathrm{Id}_V)\underline{v} = \underline{0}$$

which implies that $\lambda^n \in \mathrm{Spec}_F(T^n)$, as required. Now, suppose that $\lambda \in \mathrm{Spec}_F(T^n)$. Then $\exists \underline{v} \in V$ such that $(T^n - \lambda \,\mathrm{Id}_V)\underline{v} = \underline{0}$. Now since F is algebraically closed, we can write that $(x^n - \lambda) = \prod_{i=1}^n (x - \omega_i)$, where $\omega_i \in F$ are nth roots of λ. Thus

$$\underline{0} = \Big(\prod_{i=1}^n (A - \omega_i \,\mathrm{Id}_V)\Big)\underline{v}$$

So $\ker(A - \omega_i \,\mathrm{Id}_V)$ is non-trivial for one of the ω_i, as required. □

Theorem 5.2. *Let $A \in GL_n(F)$, where $n \in \mathbb{N}$ and F is algebraically closed of characteristic zero. Considering F^n as a $F[\mu]$ module, with $F[\mu]$ acting by A, let the Jordan decomposition of F^n be $F^n = \bigoplus_{j=1}^{k} F[\mu]\underline{v}_{\lambda_j}$. Then this is also a Jordan decomposition of F^n as a $F[\mu]$ module acting by A^m.*

Proof. Since A acts independently on each $X_j = F[\mu]\underline{v}_{\lambda_j}$, so does A^m. Since F is algebraically closed, A^m has a Jordan decomposition on X_j. Let $\lambda_j \neq 0$ be the eigenvalue of A on X_j. By Lemma 5.1, the only eigenvalue of A^m on X_j is λ_j^m. Thus, the Jordan decomposition of A^m on X_j will be the same as that of A if and only if the dimension of the eigenspace of A^m on X_j, as an F-vector space, is one. If $\dim_F(X_j) = 1$, this is immediate. Otherwise, suppose this were not the case. The eigenvector $\underline{v} \in X_j$ of A is certainly an eigenvector of A^m, and let $\underline{v}' \in X_j$ be another linearly independent eigenvector of A^m. As an F-vector space, X_j has a basis

$$\{(A - \lambda_j \operatorname{Id}_{X_j})^k \underline{v}_{\lambda_j} \mid 0 \leq k < \dim_F(X_j)\}$$

Thus, we can write that for some $\alpha_j \in F$, $\underline{v}' = \sum_{k=0}^{d-1} \alpha_k (A - \lambda_j \operatorname{Id}_{X_j})^k \underline{v}_{\lambda_j}$, where $d = \dim_F(X_j)$. We'll use the shorthand $\underline{v}_k = (A - \lambda_j \operatorname{Id}_{X_j})^k \underline{v}_{\lambda_j}$, with $\underline{v}_k = \underline{0}$ for $k \geq d$. Note that with this notation, we get that $(A - \lambda_j \operatorname{Id}_{X_j})\underline{v}_k = \underline{v}_{k+1}$, and that $\underline{v} = \underline{v}_{d-1}$. Since $(A^m - \lambda_j^m \operatorname{Id}_{X_j})\underline{v}' = \underline{0}$, we get that

$$\begin{aligned}
\underline{0} = (A^m - \lambda_j^m \operatorname{Id}_{X_j}) \sum_{k=0}^{d-1} \alpha_k (A - \lambda_j \operatorname{Id}_{X_j})^k \underline{v}_{\lambda_j} &= \Big(\sum_{l=0}^{m-1} A^l \lambda_j^{m-l-1} \Big) \sum_{k=0}^{d-1} \alpha_k (A - \lambda_j \operatorname{Id}_{X_j})^{k+1} \underline{v}_{\lambda_j} \\
&= \Big(\sum_{l=0}^{m-1} A^l \lambda_j^{m-l-1} \Big) \sum_{k=1}^{d-1} \alpha_{k-1} (A - \lambda_j \operatorname{Id}_{X_j})^k \underline{v}_{\lambda_j}
\end{aligned}$$

Since $\underline{v}'$ is linearly independent from $\underline{v}$, there exists some $k < d-1$ such that $\alpha_k \neq 0$. In particular, there exists some smallest such α_k, call it α_r. Then we can get that

$$\begin{aligned}
\Big(\sum_{l=0}^{m-1} A^l \lambda_j^{m-l-1} \Big) \sum_{k=1}^{d-1} \alpha_{k-1} (A - \lambda_j \operatorname{Id}_{X_j})^k \underline{v}_{\lambda_j} &= \Big(\sum_{l=0}^{m-1} A^l \lambda_j^{m-l-1} \Big) \sum_{k=r+1}^{d-1} \alpha_{k-1} \underline{v}_k \\
&= \alpha_r \Big(\sum_{l=0}^{m-1} \lambda_j^{m-1} \Big) \underline{v}_{r+1} + \sum_{k=r+2}^{d-1} \beta_k \underline{v}_k
\end{aligned}$$

Where $\beta_k \in F$ are some scalars, and we use the shorthand that $\sum_{k=r+2}^{d-1} = 0$ if $r + 2 > d - 1$. Since $\alpha_r \neq 0$, $\lambda_j \neq 0$, F has characteristic zero, and the $\underline{v}_k$ are linearly independent, we get that $\alpha_r \Big(\sum_{l=0}^{m-1} \lambda_j^{m-1} \Big) \underline{v}_{r+1} + \sum_{k=r+2}^{d-1} \beta_k \underline{v}_k \neq \underline{0}$, a contradiction. ☐

Corollary 5.2.1. *Let $A \in GL_n(F)$, where $n \in \mathbb{N}$ and F is an algebraically closed field with characteristic zero. Then $\forall m \in \mathbb{N}$, the following are true.*

1. *A^m has the same Jordan block form as A, up to order of blocks.*
2. *For all $k \in \mathbb{N}$ and $\lambda \in \operatorname{Spec}_F(A)$, we get*

$$\ker((A - \lambda \operatorname{Id}_n)^k) \subseteq \ker((A^m - \lambda^m \operatorname{Id}_n)^k) = \sum_{\omega^m = \lambda^m} \ker((A - \omega \operatorname{Id}_n)^k) \tag{18}$$

Proof. We prove the two parts separately.

1. This is simply a restatement of A, A^m having the same Jordan decomposition, and hence follows immediately from Theorem 5.2.
2. Suppose that $\underline{v} \in \ker((A - \lambda \operatorname{Id}_n)^k)$. Then a quick computation confirms that

$$(A^m - \lambda^m \operatorname{Id}_n)^k \underline{v} = \Big(\sum_{j=0}^{m-1} A^j \lambda^{m-j-1} \Big)^k (A - \lambda \operatorname{Id}_n)^k \underline{v} = \underline{0}$$

So $\underline{v} \in \ker((A^m - \lambda^m \operatorname{Id}_n)^k)$, as required. Thus, $\ker((A - \lambda \operatorname{Id}_n)^k) \subseteq \ker((A^m - \lambda^m \operatorname{Id}_n)^k)$, and $\sum_{\omega^m = \lambda^m} \ker((A - \omega \operatorname{Id}_n)^k) \subseteq \ker((A^m - \lambda^m \operatorname{Id}_n)^k)$. Now, pick any $\lambda, \omega \in \operatorname{Spec}_F(A)$, with associated Jordan decomposition piece V_λ for the first. Let $\{\underline{v}_j\}_{j=1}^k$ be the natural Jordan basis for V_λ, that is the one with the property $(A - \lambda)\underline{v}_j = \underline{v}_{j+1}$, using the notation that $\underline{v}_j = \underline{0}$ for $j > k$. Then for any $\alpha_j \in F$, we note that

$$(A - \omega \operatorname{Id}_n) \sum_{j=1}^k \underline{v}_j = \sum_{j=1}^k (\lambda \underline{v}_j + \underline{v}_{j+1}) - \sum_{j=1}^k \omega \underline{v}_j \in V_\lambda$$

Thus, $(A - \omega)$ does not map between spaces in the Jordan decomposition. Since F is algebraically closed, for any $\lambda \in \operatorname{Spec}_F(A)$ we can get that

$$(A^m - \lambda^m \operatorname{Id}_n) = \prod_{j=1}^m (A - \omega_j \operatorname{Id}_n)$$

where $\omega_j^m = \lambda^m$. By the above result, none of these factors map between Jordan decomposition pieces, and therefore don't map between generalized eigenspaces. It follows that if a vector is in $\ker((A^m - \lambda^n \operatorname{Id}_n)^k)$, then its projection onto $\left(\sum_{\omega^m = \lambda^m} \ker((A - \omega \operatorname{Id}_n)^k)\right)^C$ must be zero. Thus, $\ker((A^m - \lambda^m \operatorname{Id}_n)^k) \subseteq \sum_{\omega^m = \lambda^m} \ker((A - \omega \operatorname{Id}_n)^k)$.

□

5.2 Free Groups

Lemma 5.3. *Let $A, B, C \in GL_2(\mathbb{C})$, with $A = \operatorname{diag}(\lambda, \lambda^{-1})$, where eigenvalues $\lambda, \lambda^{-1} \in \mathbb{C}^\times$ and $\operatorname{Tr}(B) = \operatorname{Tr}(C)$. Then $\operatorname{Tr}(AB) = \operatorname{Tr}(AC)$ if and only if $0 = \lambda^2(b_{11} - c_{11}) + (b_{22} - c_{22})$.*

Proof. We know that

$$AB = \begin{pmatrix} \lambda b_{11} & \lambda b_{12} \\ \lambda^{-1} b_{21} & \lambda^{-1} b_{22} \end{pmatrix} \qquad AC = \begin{pmatrix} \lambda c_{11} & \lambda c_{12} \\ \lambda^{-1} c_{21} & \lambda^{-1} c_{22} \end{pmatrix}$$

so our requirements becomes $\lambda b_{11} + \lambda^{-1} b_{22} = \lambda c_{11} + \lambda^{-1} c_{22}$, which is equivalent to the desired result. □

Lemma 5.4. *Suppose that $A, A' \in SL_2(\mathbb{C})$ are similar diagonalizable matrices, with $a_{11} = a'_{11}, a_{22} = a'_{22}$ and non-zero off-diagonals. Then there exists a diagonal $P \in GL_2(\mathbb{C})$ such that $A' = PAP^{-1}$.*

Proof. Let $P = \operatorname{diag}(x, y)$. Then we have that

$$PAP^{-1} = \operatorname{diag}(x, y) \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \operatorname{diag}(x^{-1}, y^{-1}) = \begin{pmatrix} a_{11} & xy^{-1} a_{12} \\ yx^{-1} a_{21} & a_{22} \end{pmatrix}$$

while

$$A' = \begin{pmatrix} a_{11} & a'_{12} \\ a'_{21} & a_{22} \end{pmatrix}$$

So $x = y a'_{12} a_{12}^{-1}$, and $y = x a'_{21} a_{21}^{-1} = y a'_{12} a_{12}^{-1} a'_{21} a_{21}^{-1}$. This is satisfied for any non-zero $y \in \mathbb{C}$, as long as $1 = a'_{12} a_{12}^{-1} a'_{21} a_{21}^{-1}$. But since $\det(A) = \det(A')$, $a_{12} a_{21} = a'_{12} a'_{21}$, so this is certainly the case, giving us an infinite number of x, y which work. □

Theorem 5.5. *Let Γ be the free group with two generators, a, b. Then for irreducible representations in $SL_2(\mathbb{C})$, the trace of the representatives for a, b, ab fully determines a representation up to conjugation, as long as none have trace ± 2 [Whi73].*

Proof. Let ρ be an arbitrary representation of Γ, where the traces above are all not ± 2. Since $\mathrm{Tr}(\rho(a)) \neq \pm 2$, $A = \rho(a)$ cannot have ± 1 as a repeated eigenvalue. Since the only possible repeated eigenvalues for elements of $\mathrm{SL}_2(\mathbb{C})$ are ± 1, this implies that A is diagonalizable with eigenvalues λ, λ^{-1}, where $\lambda \neq \pm 1$. Since trace is invariant under conjugation, we can choose that

$$A = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$$

Now $\lambda + \lambda^{-1} = \mu + \mu^{-1}$ if and only if $\mu = \lambda^{\pm 1}$, so it follows that any other representation with the same character must have the same eigenvalues for A. Thus, the map $A \mapsto A'$ can be represented by a change of basis (where A' is a representation with the same trace), and hence conjugation by some $P \in \mathrm{GL}_2(\mathbb{C})$. Let B' be the image of B in this same representation. Then we require that

$$\begin{aligned} \mathrm{Tr}(AB) = \mathrm{Tr}(A'B') &= \mathrm{Tr}\big(PAP^{-1}B'\big) = \mathrm{Tr}\big(PAP^{-1}PP^{-1}B'PP^{-1}\big) \\ &= \mathrm{Tr}\big(P^{-1}PAP^{-1}B'P\big) = \mathrm{Tr}\big(AP^{-1}B'P\big) \end{aligned}$$

By Lemma 5.3, if $C = P^{-1}B'P$ then we get that $0 = \lambda^2(b_{11} - c_{11}) + (b_{22} - c_{22})$. But $b_{11} + b_{22} = c_{11} + c_{22}$, so this is equivalent to saying that $\lambda^2 = 1$, unless $b_{11} = c_{11}, b_{22} = c_{22}$. Thus, since we assumed that $\lambda^2 \neq 1$, we get that $b_{11} = c_{11}, b_{22} = c_{22}$. Now (A, C) is also a valid, irreducible representation. Since both representations are irreducible, the off-diagonals of B, C must be non-zero. Furthermore, since $\det(B) = \det(C)$, we get that B, C are similar. Thus, by Lemma 5.4, there exists some diagonal $Q \in \mathrm{GL}_2(\mathbb{C})$ such that $C = QBQ^{-1}$. Since this Q commutes with A, we get that $(A, B) \sim (A, C)$, and hence $(A, B) \sim (A', B')$, as was to be shown. $\square$

5.3 Code

The raw code is used to verify that all solutions exist, but presents them in an unhelpful manner. The clean code assumes that all solutions exist, and under this assumptions presents them in a much nicer form.

Figure-8-Knot-Calculations-Raw

July 11, 2023

[1]:
```
# we start with the p transformation, diagonalizable
var('a b c d x') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
y = x + x^(-1)
z = ((y^2-5)*(y^2-2) + sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # +␣
↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A^(-1)*B*A*B^(-1)*A)*P^(-1) # P*p(A)*P^(-1)
D = P*(A^(-1)*B*A)*P^(-1) # P*p(B)*P^(-1)
# check that these are still conjugate
print(bool((A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvalues() == (A^(-1)*B*A).
↪eigenvalues()))
```

```
True
```

[2]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[2]:
```
[[a == 0, b == 0, c == 0, d == 0]]
```

[3]:
```
z = ((y^2-5)*(y^2-2) - sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # -␣
↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A^(-1)*B*A*B^(-1)*A)*P^(-1) # P*p(A)*P^(-1)
D = P*(A^(-1)*B*A)*P^(-1) # P*p(B)*P^(-1)
```

```
# check that these are still conjugate
print(bool((A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvalues() == (A^(-1)*B*A).
↪eigenvalues()))
```

```
True
```

[4]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[4]:
```
[[a == 0, b == 0, c == 0, d == 0]]
```

[5]:
```
# p transformation, non-diagonalizable
var('a b c d') # setup variables
z = 1-e^(i*pi/3) # first case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[6]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A^(-1)*B*A*B^(-1)*A)*P^(-1) # P*p(A)*P^(-1)
D = P*(A^(-1)*B*A)*P^(-1) # P*p(B)*P^(-1)
# check that these are still conjugate
print(bool((A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvalues() == (A^(-1)*B*A).
↪eigenvalues()))
C_eigen = (A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvectors_right()[0][1]
# extract list of eigenvectors for the eigenvalue
D_eigen = (A^(-1)*B*A).eigenvectors_right()[0][1]
print(len(C_eigen) == 1, len(D_eigen) == 1)
```

```
True
True True
```

[7]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[7]: [[a == 0, b == 0, c == 0, d == 0]]

[8]:
```
A = Matrix([[-1,1],[0,-1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A^(-1)*B*A*B^(-1)*A)*P^(-1) # P*p(A)*P^(-1)
D = P*(A^(-1)*B*A)*P^(-1) # P*p(B)*P^(-1)
# check that these are still conjugate
print(bool((A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvalues() == (A^(-1)*B*A).
 ↪eigenvalues()))
C_eigen = (A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvectors_right()[0][1]
# extract list of eigenvectors for the eigenvalue
D_eigen = (A^(-1)*B*A).eigenvectors_right()[0][1]
print(len(C_eigen) == 1, len(D_eigen) == 1)
```

```
True
True True
```

[9]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[9]: [[a == 0, b == 0, c == 0, d == 0]]

[10]:
```
z = 1-e^(i*5*pi/3) # second case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[11]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A^(-1)*B*A*B^(-1)*A)*P^(-1) # P*p(A)*P^(-1)
D = P*(A^(-1)*B*A)*P^(-1) # P*p(B)*P^(-1)
# check that these are still conjugate
print(bool((A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvalues() == (A^(-1)*B*A).
 ↪eigenvalues()))
C_eigen = (A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvectors_right()[0][1]
# extract list of eigenvectors for the eigenvalue
D_eigen = (A^(-1)*B*A).eigenvectors_right()[0][1]
print(len(C_eigen) == 1, len(D_eigen) == 1)
```

```
True
True True
```

[12]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[12]:
```
[[a == 0, b == 0, c == 0, d == 0]]
```

[13]:
```
A = Matrix([[-1,1],[0,-1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A^(-1)*B*A*B^(-1)*A)*P^(-1) # P*p(A)*P^(-1)
D = P*(A^(-1)*B*A)*P^(-1) # P*p(B)*P^(-1)
# check that these are still conjugate
print(bool((A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvalues() == (A^(-1)*B*A).
 ↪eigenvalues()))
C_eigen = (A^(-1)*B*A^(-1)*B*A*B^(-1)*A).eigenvectors_right()[0][1]
# extract list of eigenvectors for the eigenvalue
D_eigen = (A^(-1)*B*A).eigenvectors_right()[0][1]
print(len(C_eigen) == 1, len(D_eigen) == 1)
```

```
True
True True
```

[14]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[14]:
```
[[a == 0, b == 0, c == 0, d == 0]]
```

[15]:
```
# we start with the p^2 transformation, diagonalizable
var('a b c d x') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
y = x + x^(-1)
z = ((y^2-5)*(y^2-2) + sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # +␣
 ↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
```

```
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A)*P^(-1) # P*p^2(A)*P^(-1)
D = P*(B*A^(-1)*B*A*B^(-1))*P^(-1) # P*p^2(B)*P^(-1)
```

[16]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[16]:
```
[[a == r1, b == -2*(r1*x^4 - 3*r1*x^2 + r1)/(x^8 - 3*x^6 + x^4*sqrt((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 2*x^4
- 3*x^2 + 1), c == (64*r1*x^54 - 1536*r1*x^52 + 17280*r1*x^50 - 121600*r1*x^48 +
603200*r1*x^46 - 2256128*r1*x^44 + 6653952*r1*x^42 - 15996800*r1*x^40 +
32189440*r1*x^38 - 55406720*r1*x^36 + 83033024*r1*x^34 - 109819776*r1*x^32 +
129407040*r1*x^30 - 136602880*r1*x^28 + 129407040*r1*x^26 - 109819776*r1*x^24 +
83033024*r1*x^22 - 55406720*r1*x^20 + 32189440*r1*x^18 - 15996800*r1*x^16 +
6653952*r1*x^14 - 2256128*r1*x^12 + 603200*r1*x^10 - 121600*r1*x^8 +
17280*r1*x^6 - 1536*r1*x^4 + (r1*x^30 - 2*r1*x^28 + 2*r1*x^26)*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(7/2) -
2*(r1*x^38 - 12*r1*x^36 + 61*r1*x^34 - 176*r1*x^32 + 330*r1*x^30 - 434*r1*x^28 +
405*r1*x^26 - 256*r1*x^24 + 101*r1*x^22 - 22*r1*x^20 + 2*r1*x^18)*((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(5/2) +
64*r1*x^2 + (r1*x^46 - 14*r1*x^44 + 104*r1*x^42 - 572*r1*x^40 + 2493*r1*x^38 -
8450*r1*x^36 + 22008*r1*x^34 - 44368*r1*x^32 + 70288*r1*x^30 - 88700*r1*x^28 +
89774*r1*x^26 - 72676*r1*x^24 + 46461*r1*x^22 - 22938*r1*x^20 + 8472*r1*x^18 -
2240*r1*x^16 + 397*r1*x^14 - 42*r1*x^12 + 2*r1*x^10)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - 4*(2*r1*x^52 -
53*r1*x^50 + 642*r1*x^48 - 4756*r1*x^46 + 24332*r1*x^44 - 92119*r1*x^42 +
270338*r1*x^40 - 636641*r1*x^38 + 1236514*r1*x^36 - 2024108*r1*x^34 +
2837876*r1*x^32 - 3442899*r1*x^30 + 3630182*r1*x^28 - 3323575*r1*x^26 +
2627838*r1*x^24 - 1777596*r1*x^22 + 1015132*r1*x^20 - 480749*r1*x^18 +
184406*r1*x^16 - 55507*r1*x^14 + 12542*r1*x^12 - 1988*r1*x^10 + 196*r1*x^8 -
9*r1*x^6)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8))/(64*x^52 - 1536*x^50 + 17280*x^48 - 121600*x^46 + 603200*x^44 -
2256128*x^42 + 6653952*x^40 - 15996800*x^38 + 32189440*x^36 - 55406720*x^34 +
83033024*x^32 - 109819776*x^30 + 129407040*x^28 - 136602880*x^26 +
129407040*x^24 - 109819776*x^22 + 83033024*x^20 - 55406720*x^18 + 32189440*x^16
- 15996800*x^14 + 6653952*x^12 - 2256128*x^10 + 603200*x^8 - 121600*x^6 + (x^28
- 2*x^26 + x^24)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4
- 6*x^2 + 1)/x^8)^(7/2) + 17280*x^4 - 2*(x^36 - 12*x^34 + 60*x^32 - 166*x^30 +
289*x^28 - 344*x^26 + 289*x^24 - 166*x^22 + 60*x^20 - 12*x^18 + x^16)*((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(5/2) +
```

```
(x^44 - 14*x^42 + 103*x^40 - 544*x^38 + 2199*x^36 - 6754*x^34 + 15735*x^32 -
28184*x^30 + 39562*x^28 - 44208*x^26 + 39562*x^24 - 28184*x^22 + 15735*x^20 -
6754*x^18 + 2199*x^16 - 544*x^14 + 103*x^12 - 14*x^10 + x^8)*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - 1536*x^2
- 4*(2*x^50 - 53*x^48 + 644*x^46 - 4800*x^44 + 24780*x^42 - 94931*x^40 +
282576*x^38 - 676065*x^36 + 1334684*x^34 - 2219424*x^32 + 3157428*x^30 -
3884727*x^28 + 4159772*x^26 - 3884727*x^24 + 3157428*x^22 - 2219424*x^20 +
1334684*x^18 - 676065*x^16 + 282576*x^14 - 94931*x^12 + 24780*x^10 - 4800*x^8 +
644*x^6 - 53*x^4 + 2*x^2)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 -
14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 64), d == -(256*r1*x^70 - 7680*r1*x^68 +
109568*r1*x^66 - 992000*r1*x^64 + 6420736*r1*x^62 - 31751680*r1*x^60 +
125280000*r1*x^58 - 406922240*r1*x^56 + 1114721280*r1*x^54 - 2626434560*r1*x^52
+ 5409924864*r1*x^50 - 9875238400*r1*x^48 + 16154821632*r1*x^46 -
23897899520*r1*x^44 + 32191340544*r1*x^42 - 39688047360*r1*x^40 +
44939781120*r1*x^38 - 46830213120*r1*x^36 + 44939781120*r1*x^34 -
39688047360*r1*x^32 + 32191340544*r1*x^30 - 23897899520*r1*x^28 +
16154821632*r1*x^26 - 9875238400*r1*x^24 + 5409924864*r1*x^22 -
2626434560*r1*x^20 + 1114721280*r1*x^18 - 406922240*r1*x^16 + 125280000*r1*x^14
- 31751680*r1*x^12 + 6420736*r1*x^10 - 992000*r1*x^8 + 109568*r1*x^6 + (r1*x^38
- 2*r1*x^36 + 2*r1*x^34)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 +
13*x^4 - 6*x^2 + 1)/x^8)^(9/2) - 7680*r1*x^4 - (r1*x^46 - 16*r1*x^44 +
93*r1*x^42 - 286*r1*x^40 + 554*r1*x^38 - 740*r1*x^36 + 691*r1*x^34 - 428*r1*x^32
+ 159*r1*x^30 - 30*r1*x^28 + 2*r1*x^26)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 +
12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(7/2) - (r1*x^54 - 22*r1*x^52 +
192*r1*x^50 - 920*r1*x^48 + 2741*r1*x^46 - 5306*r1*x^44 + 6380*r1*x^42 -
3020*r1*x^40 - 5128*r1*x^38 + 14144*r1*x^36 - 18542*r1*x^34 + 16036*r1*x^32 -
9383*r1*x^30 + 3354*r1*x^28 - 392*r1*x^26 - 240*r1*x^24 + 129*r1*x^22 -
26*r1*x^20 + 2*r1*x^18)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 +
13*x^4 - 6*x^2 + 1)/x^8)^(5/2) + 256*r1*x^2 + (r1*x^62 - 28*r1*x^60 +
415*r1*x^58 - 3966*r1*x^56 + 26265*r1*x^54 - 126884*r1*x^52 + 466514*r1*x^50 -
1352582*r1*x^48 + 3185349*r1*x^46 - 6242502*r1*x^44 + 10373833*r1*x^42 -
14812980*r1*x^40 + 18314500*r1*x^38 - 19655326*r1*x^36 + 18272115*r1*x^34 -
14621306*r1*x^32 + 9964811*r1*x^30 - 5694660*r1*x^28 + 2669758*r1*x^26 -
996010*r1*x^24 + 283303*r1*x^22 - 57794*r1*x^20 + 7827*r1*x^18 - 740*r1*x^16 +
107*r1*x^14 - 22*r1*x^12 + 2*r1*x^10)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 +
12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - 4*(2*r1*x^68 - 97*r1*x^66 +
1862*r1*x^64 - 20691*r1*x^62 + 155298*r1*x^60 - 856015*r1*x^58 + 3654206*r1*x^56
- 12540581*r1*x^54 + 35587916*r1*x^52 - 85398048*r1*x^50 + 176456504*r1*x^48 -
318650912*r1*x^46 + 508956576*r1*x^44 - 725789698*r1*x^42 + 930576040*r1*x^40 -
1077970942*r1*x^38 + 1131391108*r1*x^36 - 1076954190*r1*x^34 + 929003848*r1*x^32
- 724398490*r1*x^30 + 508429788*r1*x^28 - 319277080*r1*x^26 + 177982184*r1*x^24
- 87211496*r1*x^22 + 37109582*r1*x^20 - 13510253*r1*x^18 + 4132066*r1*x^16 -
1037559*r1*x^14 + 207546*r1*x^12 - 31707*r1*x^10 + 3466*r1*x^8 - 241*r1*x^6 +
8*r1*x^4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8))/(256*x^70 - 7680*x^68 + 109568*x^66 - 992000*x^64 +
6420736*x^62 - 31751680*x^60 + 125280000*x^58 - 406922240*x^56 + 1114721280*x^54
- 2626434560*x^52 + 5409924864*x^50 - 9875238400*x^48 + 16154821632*x^46 -
```

```
23897899520*x^44 + 32191340544*x^42 - 39688047360*x^40 + 44939781120*x^38 -
46830213120*x^36 + 44939781120*x^34 - 39688047360*x^32 + 32191340544*x^30 -
23897899520*x^28 + 16154821632*x^26 - 9875238400*x^24 + 5409924864*x^22 -
2626434560*x^20 + 1114721280*x^18 - 406922240*x^16 + 125280000*x^14 -
31751680*x^12 + 6420736*x^10 - 992000*x^8 + 109568*x^6 + (x^38 - 2*x^36 + 2*x^34
- 2*x^32 + x^30)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4
- 6*x^2 + 1)/x^8)^(9/2) - (x^46 - 16*x^44 + 95*x^42 - 304*x^40 + 633*x^38 -
962*x^36 + 1118*x^34 - 1012*x^32 + 723*x^30 - 394*x^28 + 147*x^26 - 32*x^24 +
3*x^22)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 +
1)/x^8)^(7/2) - 7680*x^4 - (x^54 - 22*x^52 + 188*x^50 - 866*x^48 + 2410*x^46 -
4046*x^44 + 2886*x^42 + 4866*x^40 - 20575*x^38 + 40760*x^36 - 58280*x^34 +
66412*x^32 - 62535*x^30 + 49026*x^28 - 31602*x^26 + 16306*x^24 - 6498*x^22 +
1914*x^20 - 392*x^18 + 50*x^16 - 3*x^14)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 +
12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(5/2) + (x^62 - 28*x^60 + 413*x^58 -
3960*x^56 + 26648*x^54 - 132582*x^52 + 508745*x^50 - 1558836*x^48 + 3923170*x^46
- 8294472*x^44 + 14994931*x^42 - 23501356*x^40 + 32273487*x^38 - 39141778*x^36 +
42153258*x^34 - 40432676*x^32 + 34566769*x^30 - 26297758*x^28 + 17727275*x^26 -
10503232*x^24 + 5396682*x^22 - 2356040*x^20 + 849189*x^18 - 243056*x^16 +
52348*x^14 - 7806*x^12 + 685*x^10 - 20*x^8 - x^6)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 256*x^2 - 4*(2*x^68
- 97*x^66 + 1854*x^64 - 20466*x^62 + 152292*x^60 - 830570*x^58 + 3500542*x^56 -
11833218*x^54 + 32995864*x^52 - 77593803*x^50 + 156661264*x^48 - 275466906*x^46
+ 426508952*x^44 - 586027992*x^42 + 717792086*x^40 - 784477836*x^38 +
762365802*x^36 - 652228622*x^34 + 480467312*x^32 - 289405674*x^30 +
121327322*x^28 - 3966438*x^26 - 56045118*x^24 + 69990642*x^22 - 57567978*x^20 +
37001641*x^18 - 19391084*x^16 + 8359190*x^14 - 2946966*x^12 + 836512*x^10 -
186398*x^8 + 31356*x^6 - 3738*x^4 + 281*x^2 - 10)*sqrt((x^16 - 6*x^14 + 13*x^12
- 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8))], [a == r2, b ==
-(16*r2*x^42 - 240*r2*x^40 + 1632*r2*x^38 - 6784*r2*x^36 + 19760*r2*x^34 -
44048*r2*x^32 + 80032*r2*x^30 - 123632*r2*x^28 + 166480*r2*x^26 - 198096*r2*x^24
+ 209760*r2*x^22 - 198096*r2*x^20 + 166480*r2*x^18 - 123632*r2*x^16 +
80032*r2*x^14 - 44048*r2*x^12 + 19760*r2*x^10 - 6784*r2*x^8 + 1632*r2*x^6 -
240*r2*x^4 + 16*r2*x^2 + (r2*x^32 - 2*r2*x^30 - 9*r2*x^28 + 41*r2*x^26 -
89*r2*x^24 + 131*r2*x^22 - 155*r2*x^20 + 152*r2*x^18 - 124*r2*x^16 + 77*r2*x^14
- 34*r2*x^12 + 9*r2*x^10 - r2*x^8)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8
- 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - (r2*x^40 - 24*r2*x^38 + 208*r2*x^36
- 969*r2*x^34 + 2900*r2*x^32 - 6282*r2*x^30 + 10733*r2*x^28 - 15175*r2*x^26 +
18073*r2*x^24 - 18190*r2*x^22 + 15362*r2*x^20 - 10570*r2*x^18 + 5391*r2*x^16 -
1272*r2*x^14 - 992*r2*x^12 + 1471*r2*x^10 - 974*r2*x^8 + 396*r2*x^6 - 101*r2*x^4
+ 15*r2*x^2 - r2)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 +
13*x^4 - 6*x^2 + 1)/x^8))/(16*x^46 - 240*x^44 + 1648*x^42 - 7008*x^40 +
21200*x^38 - 49808*x^36 + 96496*x^34 - 160256*x^32 + 233584*x^30 - 303120*x^28 +
353296*x^26 - 371616*x^24 + 353296*x^22 - 303120*x^20 + 233584*x^18 -
160256*x^16 + 96496*x^14 - 49808*x^12 + 21200*x^10 - 7008*x^8 + 1648*x^6 -
240*x^4 + (3*x^36 - 14*x^34 + 23*x^32 - 18*x^30 - x^28 + 33*x^26 - 81*x^24 +
122*x^22 - 138*x^20 + 127*x^18 - 101*x^16 + 66*x^14 - 32*x^12 + 9*x^10 -
x^8)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 +
```

```
1)/x^8)^(3/2) + 16*x^2 - (3*x^44 - 48*x^42 + 338*x^40 - 1420*x^38 + 4126*x^36 -
9175*x^34 + 16695*x^32 - 25735*x^30 + 34209*x^28 - 39709*x^26 + 40495*x^24 -
36138*x^22 + 27833*x^20 - 18014*x^18 + 9232*x^16 - 3070*x^14 - 172*x^12 +
1121*x^10 - 859*x^8 + 373*x^6 - 99*x^4 + 15*x^2 - 1)*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)), c == (8*r2*x^40
- 120*r2*x^38 + 816*r2*x^36 - 3384*r2*x^34 + 9776*r2*x^32 - 21416*r2*x^30 +
37848*r2*x^28 - 56376*r2*x^26 + 72712*r2*x^24 - 82344*r2*x^22 + 82344*r2*x^20 -
72712*r2*x^18 + 56376*r2*x^16 - 37848*r2*x^14 + 21416*r2*x^12 - 9776*r2*x^10 +
3384*r2*x^8 - 816*r2*x^6 + 120*r2*x^4 - 8*r2*x^2 + (r2*x^30 - 2*r2*x^28 -
10*r2*x^26 + 42*r2*x^24 - 78*r2*x^22 + 100*r2*x^20 - 108*r2*x^18 + 99*r2*x^16 -
69*r2*x^14 + 33*r2*x^12 - 9*r2*x^10 + r2*x^8)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - (r2*x^38 -
16*r2*x^36 + 111*r2*x^34 - 450*r2*x^32 + 1228*r2*x^30 - 2488*r2*x^28 +
3995*r2*x^26 - 5201*r2*x^24 + 5428*r2*x^22 - 4394*r2*x^20 + 2490*r2*x^18 -
424*r2*x^16 - 1177*r2*x^14 + 1888*r2*x^12 - 1671*r2*x^10 + 984*r2*x^8 -
390*r2*x^6 + 100*r2*x^4 - 15*r2*x^2 + r2)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8))/(8*x^38 - 120*x^36 +
816*x^34 - 3384*x^32 + 9776*x^30 - 21416*x^28 + 37848*x^26 - 56376*x^24 +
72712*x^22 - 82344*x^20 + 82344*x^18 - 72712*x^16 + 56376*x^14 - 37848*x^12 +
21416*x^10 - 9776*x^8 + 3384*x^6 - 816*x^4 + (x^28 - 3*x^26 - x^24 + 11*x^22 -
21*x^20 + 27*x^18 - 29*x^16 + 24*x^14 - 15*x^12 + 6*x^10 - x^8)*((x^16 - 6*x^14
+ 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 120*x^2
- (x^36 - 17*x^34 + 126*x^32 - 548*x^30 + 1602*x^28 - 3444*x^26 + 5809*x^24 -
8008*x^22 + 9222*x^20 - 8958*x^18 + 7318*x^16 - 4908*x^14 + 2531*x^12 - 839*x^10
+ 48*x^8 + 110*x^6 - 56*x^4 + 12*x^2 - 1)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) - 8), d == -(8*r2*x^40 -
120*r2*x^38 + 816*r2*x^36 - 3384*r2*x^34 + 9776*r2*x^32 - 21416*r2*x^30 +
37848*r2*x^28 - 56376*r2*x^26 + 72712*r2*x^24 - 82344*r2*x^22 + 82344*r2*x^20 -
72712*r2*x^18 + 56376*r2*x^16 - 37848*r2*x^14 + 21416*r2*x^12 - 9776*r2*x^10 +
3384*r2*x^8 - 816*r2*x^6 + 120*r2*x^4 - 8*r2*x^2 + (r2*x^30 - 2*r2*x^28 -
10*r2*x^26 + 42*r2*x^24 - 78*r2*x^22 + 100*r2*x^20 - 108*r2*x^18 + 99*r2*x^16 -
69*r2*x^14 + 33*r2*x^12 - 9*r2*x^10 + r2*x^8)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - (r2*x^38 -
16*r2*x^36 + 111*r2*x^34 - 450*r2*x^32 + 1228*r2*x^30 - 2488*r2*x^28 +
3995*r2*x^26 - 5201*r2*x^24 + 5428*r2*x^22 - 4394*r2*x^20 + 2490*r2*x^18 -
424*r2*x^16 - 1177*r2*x^14 + 1888*r2*x^12 - 1671*r2*x^10 + 984*r2*x^8 -
390*r2*x^6 + 100*r2*x^4 - 15*r2*x^2 + r2)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8))/(8*x^40 - 120*x^38 +
816*x^36 - 3384*x^34 + 9776*x^32 - 21416*x^30 + 37848*x^28 - 56376*x^26 +
72712*x^24 - 82344*x^22 + 82344*x^20 - 72712*x^18 + 56376*x^16 - 37848*x^14 +
21416*x^12 - 9776*x^10 + 3384*x^8 - 816*x^6 + 120*x^4 + (x^30 - 3*x^28 - x^26 +
11*x^24 - 21*x^22 + 27*x^20 - 29*x^18 + 24*x^16 - 15*x^14 + 6*x^12 -
x^10)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 +
1)/x^8)^(3/2) - 8*x^2 - (x^38 - 17*x^36 + 126*x^34 - 548*x^32 + 1602*x^30 -
3444*x^28 + 5809*x^26 - 8008*x^24 + 9222*x^22 - 8958*x^20 + 7318*x^18 -
4908*x^16 + 2531*x^14 - 839*x^12 + 48*x^10 + 110*x^8 - 56*x^6 + 12*x^4 -
x^2)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2
```

```
+ 1)/x^8))], [a == 0, b == 0, c == 0, d == 0]]
```

[17]:
```
z = ((y^2-5)*(y^2-2) - sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # -
 ↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A)*P^(-1) # P*p^2(A)*P^(-1)
D = P*(B*A^(-1)*B*A*B^(-1))*P^(-1) # P*p^2(B)*P^(-1)
```

[18]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[18]:
```
[[a == r3, b == -2*(r3*x^4 - 3*r3*x^2 + r3)/(x^8 - 3*x^6 - x^4*sqrt((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 2*x^4
- 3*x^2 + 1), c == (64*r3*x^54 - 1536*r3*x^52 + 17280*r3*x^50 - 121600*r3*x^48 +
603200*r3*x^46 - 2256128*r3*x^44 + 6653952*r3*x^42 - 15996800*r3*x^40 +
32189440*r3*x^38 - 55406720*r3*x^36 + 83033024*r3*x^34 - 109819776*r3*x^32 +
129407040*r3*x^30 - 136602880*r3*x^28 + 129407040*r3*x^26 - 109819776*r3*x^24 +
83033024*r3*x^22 - 55406720*r3*x^20 + 32189440*r3*x^18 - 15996800*r3*x^16 +
6653952*r3*x^14 - 2256128*r3*x^12 + 603200*r3*x^10 - 121600*r3*x^8 +
17280*r3*x^6 - 1536*r3*x^4 - (r3*x^30 - 2*r3*x^28 + 2*r3*x^26)*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(7/2) +
2*(r3*x^38 - 12*r3*x^36 + 61*r3*x^34 - 176*r3*x^32 + 330*r3*x^30 - 434*r3*x^28 +
405*r3*x^26 - 256*r3*x^24 + 101*r3*x^22 - 22*r3*x^20 + 2*r3*x^18)*((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(5/2) +
64*r3*x^2 - (r3*x^46 - 14*r3*x^44 + 104*r3*x^42 - 572*r3*x^40 + 2493*r3*x^38 -
8450*r3*x^36 + 22008*r3*x^34 - 44368*r3*x^32 + 70288*r3*x^30 - 88700*r3*x^28 +
89774*r3*x^26 - 72676*r3*x^24 + 46461*r3*x^22 - 22938*r3*x^20 + 8472*r3*x^18 -
2240*r3*x^16 + 397*r3*x^14 - 42*r3*x^12 + 2*r3*x^10)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 4*(2*r3*x^52 -
53*r3*x^50 + 642*r3*x^48 - 4756*r3*x^46 + 24332*r3*x^44 - 92119*r3*x^42 +
270338*r3*x^40 - 636641*r3*x^38 + 1236514*r3*x^36 - 2024108*r3*x^34 +
2837876*r3*x^32 - 3442899*r3*x^30 + 3630182*r3*x^28 - 3323575*r3*x^26 +
2627838*r3*x^24 - 1777596*r3*x^22 + 1015132*r3*x^20 - 480749*r3*x^18 +
184406*r3*x^16 - 55507*r3*x^14 + 12542*r3*x^12 - 1988*r3*x^10 + 196*r3*x^8 -
9*r3*x^6)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8))/(64*x^52 - 1536*x^50 + 17280*x^48 - 121600*x^46 + 603200*x^44 -
2256128*x^42 + 6653952*x^40 - 15996800*x^38 + 32189440*x^36 - 55406720*x^34 +
```

```
83033024*x^32 - 109819776*x^30 + 129407040*x^28 - 136602880*x^26 +
129407040*x^24 - 109819776*x^22 + 83033024*x^20 - 55406720*x^18 + 32189440*x^16
- 15996800*x^14 + 6653952*x^12 - 2256128*x^10 + 603200*x^8 - 121600*x^6 - (x^28
- 2*x^26 + x^24)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4
- 6*x^2 + 1)/x^8)^(7/2) + 17280*x^4 + 2*(x^36 - 12*x^34 + 60*x^32 - 166*x^30 +
289*x^28 - 344*x^26 + 289*x^24 - 166*x^22 + 60*x^20 - 12*x^18 + x^16)*((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(5/2) -
(x^44 - 14*x^42 + 103*x^40 - 544*x^38 + 2199*x^36 - 6754*x^34 + 15735*x^32 -
28184*x^30 + 39562*x^28 - 44208*x^26 + 39562*x^24 - 28184*x^22 + 15735*x^20 -
6754*x^18 + 2199*x^16 - 544*x^14 + 103*x^12 - 14*x^10 + x^8)*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) - 1536*x^2
+ 4*(2*x^50 - 53*x^48 + 644*x^46 - 4800*x^44 + 24780*x^42 - 94931*x^40 +
282576*x^38 - 676065*x^36 + 1334684*x^34 - 2219424*x^32 + 3157428*x^30 -
3884727*x^28 + 4159772*x^26 - 3884727*x^24 + 3157428*x^22 - 2219424*x^20 +
1334684*x^18 - 676065*x^16 + 282576*x^14 - 94931*x^12 + 24780*x^10 - 4800*x^8 +
644*x^6 - 53*x^4 + 2*x^2)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 -
14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 64), d == -(256*r3*x^70 - 7680*r3*x^68 +
109568*r3*x^66 - 992000*r3*x^64 + 6420736*r3*x^62 - 31751680*r3*x^60 +
125280000*r3*x^58 - 406922240*r3*x^56 + 1114721280*r3*x^54 - 2626434560*r3*x^52
+ 5409924864*r3*x^50 - 9875238400*r3*x^48 + 16154821632*r3*x^46 -
23897899520*r3*x^44 + 32191340544*r3*x^42 - 39688047360*r3*x^40 +
44939781120*r3*x^38 - 46830213120*r3*x^36 + 44939781120*r3*x^34 -
39688047360*r3*x^32 + 32191340544*r3*x^30 - 23897899520*r3*x^28 +
16154821632*r3*x^26 - 9875238400*r3*x^24 + 5409924864*r3*x^22 -
2626434560*r3*x^20 + 1114721280*r3*x^18 - 406922240*r3*x^16 + 125280000*r3*x^14
- 31751680*r3*x^12 + 6420736*r3*x^10 - 992000*r3*x^8 + 109568*r3*x^6 - (r3*x^38
- 2*r3*x^36 + 2*r3*x^34)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 +
13*x^4 - 6*x^2 + 1)/x^8)^(9/2) - 7680*r3*x^4 + (r3*x^46 - 16*r3*x^44 +
93*r3*x^42 - 286*r3*x^40 + 554*r3*x^38 - 740*r3*x^36 + 691*r3*x^34 - 428*r3*x^32
+ 159*r3*x^30 - 30*r3*x^28 + 2*r3*x^26)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 +
12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(7/2) + (r3*x^54 - 22*r3*x^52 +
192*r3*x^50 - 920*r3*x^48 + 2741*r3*x^46 - 5306*r3*x^44 + 6380*r3*x^42 -
3020*r3*x^40 - 5128*r3*x^38 + 14144*r3*x^36 - 18542*r3*x^34 + 16036*r3*x^32 -
9383*r3*x^30 + 3354*r3*x^28 - 392*r3*x^26 - 240*r3*x^24 + 129*r3*x^22 -
26*r3*x^20 + 2*r3*x^18)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 +
13*x^4 - 6*x^2 + 1)/x^8)^(5/2) + 256*r3*x^2 - (r3*x^62 - 28*r3*x^60 +
415*r3*x^58 - 3966*r3*x^56 + 26265*r3*x^54 - 126884*r3*x^52 + 466514*r3*x^50 -
1352582*r3*x^48 + 3185349*r3*x^46 - 6242502*r3*x^44 + 10373833*r3*x^42 -
14812980*r3*x^40 + 18314500*r3*x^38 - 19655326*r3*x^36 + 18272115*r3*x^34 -
14621306*r3*x^32 + 9964811*r3*x^30 - 5694660*r3*x^28 + 2669758*r3*x^26 -
996010*r3*x^24 + 283303*r3*x^22 - 57794*r3*x^20 + 7827*r3*x^18 - 740*r3*x^16 +
107*r3*x^14 - 22*r3*x^12 + 2*r3*x^10)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 +
12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 4*(2*r3*x^68 - 97*r3*x^66 +
1862*r3*x^64 - 20691*r3*x^62 + 155298*r3*x^60 - 856015*r3*x^58 + 3654206*r3*x^56
- 12540581*r3*x^54 + 35587916*r3*x^52 - 85398048*r3*x^50 + 176456504*r3*x^48 -
318650912*r3*x^46 + 508956576*r3*x^44 - 725789698*r3*x^42 + 930576040*r3*x^40 -
1077970942*r3*x^38 + 1131391108*r3*x^36 - 1076954190*r3*x^34 + 929003848*r3*x^32
```

```
- 724398490*r3*x^30 + 508429788*r3*x^28 - 319277080*r3*x^26 + 177982184*r3*x^24
- 87211496*r3*x^22 + 37109582*r3*x^20 - 13510253*r3*x^18 + 4132066*r3*x^16 -
1037559*r3*x^14 + 207546*r3*x^12 - 31707*r3*x^10 + 3466*r3*x^8 - 241*r3*x^6 +
8*r3*x^4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8))/(256*x^70 - 7680*x^68 + 109568*x^66 - 992000*x^64 +
6420736*x^62 - 31751680*x^60 + 125280000*x^58 - 406922240*x^56 + 1114721280*x^54
- 2626434560*x^52 + 5409924864*x^50 - 9875238400*x^48 + 16154821632*x^46 -
23897899520*x^44 + 32191340544*x^42 - 39688047360*x^40 + 44939781120*x^38 -
46830213120*x^36 + 44939781120*x^34 - 39688047360*x^32 + 32191340544*x^30 -
23897899520*x^28 + 16154821632*x^26 - 9875238400*x^24 + 5409924864*x^22 -
2626434560*x^20 + 1114721280*x^18 - 406922240*x^16 + 125280000*x^14 -
31751680*x^12 + 6420736*x^10 - 992000*x^8 + 109568*x^6 - (x^38 - 2*x^36 + 2*x^34
- 2*x^32 + x^30)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4
- 6*x^2 + 1)/x^8)^(9/2) + (x^46 - 16*x^44 + 95*x^42 - 304*x^40 + 633*x^38 -
962*x^36 + 1118*x^34 - 1012*x^32 + 723*x^30 - 394*x^28 + 147*x^26 - 32*x^24 +
3*x^22)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 +
1)/x^8)^(7/2) - 7680*x^4 + (x^54 - 22*x^52 + 188*x^50 - 866*x^48 + 2410*x^46 -
4046*x^44 + 2886*x^42 + 4866*x^40 - 20575*x^38 + 40760*x^36 - 58280*x^34 +
66412*x^32 - 62535*x^30 + 49026*x^28 - 31602*x^26 + 16306*x^24 - 6498*x^22 +
1914*x^20 - 392*x^18 + 50*x^16 - 3*x^14)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 +
12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(5/2) - (x^62 - 28*x^60 + 413*x^58 -
3960*x^56 + 26648*x^54 - 132582*x^52 + 508745*x^50 - 1558836*x^48 + 3923170*x^46
- 8294472*x^44 + 14994931*x^42 - 23501356*x^40 + 32273487*x^38 - 39141778*x^36 +
42153258*x^34 - 40432676*x^32 + 34566769*x^30 - 26297758*x^28 + 17727275*x^26 -
10503232*x^24 + 5396682*x^22 - 2356040*x^20 + 849189*x^18 - 243056*x^16 +
52348*x^14 - 7806*x^12 + 685*x^10 - 20*x^8 - x^6)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 256*x^2 + 4*(2*x^68
- 97*x^66 + 1854*x^64 - 20466*x^62 + 152292*x^60 - 830570*x^58 + 3500542*x^56 -
11833218*x^54 + 32995864*x^52 - 77593803*x^50 + 156661264*x^48 - 275466906*x^46
+ 426508952*x^44 - 586027992*x^42 + 717792086*x^40 - 784477836*x^38 +
762365802*x^36 - 652228622*x^34 + 480467312*x^32 - 289405674*x^30 +
121327322*x^28 - 3966438*x^26 - 56045118*x^24 + 69990642*x^22 - 57567978*x^20 +
37001641*x^18 - 19391084*x^16 + 8359190*x^14 - 2946966*x^12 + 836512*x^10 -
186398*x^8 + 31356*x^6 - 3738*x^4 + 281*x^2 - 10)*sqrt((x^16 - 6*x^14 + 13*x^12
- 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8))], [a == r4, b ==
-(16*r4*x^42 - 240*r4*x^40 + 1632*r4*x^38 - 6784*r4*x^36 + 19760*r4*x^34 -
44048*r4*x^32 + 80032*r4*x^30 - 123632*r4*x^28 + 166480*r4*x^26 - 198096*r4*x^24
+ 209760*r4*x^22 - 198096*r4*x^20 + 166480*r4*x^18 - 123632*r4*x^16 +
80032*r4*x^14 - 44048*r4*x^12 + 19760*r4*x^10 - 6784*r4*x^8 + 1632*r4*x^6 -
240*r4*x^4 + 16*r4*x^2 - (r4*x^32 - 2*r4*x^30 - 9*r4*x^28 + 41*r4*x^26 -
89*r4*x^24 + 131*r4*x^22 - 155*r4*x^20 + 152*r4*x^18 - 124*r4*x^16 + 77*r4*x^14
- 34*r4*x^12 + 9*r4*x^10 - r4*x^8)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8
- 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + (r4*x^40 - 24*r4*x^38 + 208*r4*x^36
- 969*r4*x^34 + 2900*r4*x^32 - 6282*r4*x^30 + 10733*r4*x^28 - 15175*r4*x^26 +
18073*r4*x^24 - 18190*r4*x^22 + 15362*r4*x^20 - 10570*r4*x^18 + 5391*r4*x^16 -
1272*r4*x^14 - 992*r4*x^12 + 1471*r4*x^10 - 974*r4*x^8 + 396*r4*x^6 - 101*r4*x^4
+ 15*r4*x^2 - r4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 +
```

```
13*x^4 - 6*x^2 + 1)/x^8))/(16*x^46 - 240*x^44 + 1648*x^42 - 7008*x^40 +
21200*x^38 - 49808*x^36 + 96496*x^34 - 160256*x^32 + 233584*x^30 - 303120*x^28 +
353296*x^26 - 371616*x^24 + 353296*x^22 - 303120*x^20 + 233584*x^18 -
160256*x^16 + 96496*x^14 - 49808*x^12 + 21200*x^10 - 7008*x^8 + 1648*x^6 -
240*x^4 - (3*x^36 - 14*x^34 + 23*x^32 - 18*x^30 - x^28 + 33*x^26 - 81*x^24 +
122*x^22 - 138*x^20 + 127*x^18 - 101*x^16 + 66*x^14 - 32*x^12 + 9*x^10 -
x^8)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 +
1)/x^8)^(3/2) + 16*x^2 + (3*x^44 - 48*x^42 + 338*x^40 - 1420*x^38 + 4126*x^36 -
9175*x^34 + 16695*x^32 - 25735*x^30 + 34209*x^28 - 39709*x^26 + 40495*x^24 -
36138*x^22 + 27833*x^20 - 18014*x^18 + 9232*x^16 - 3070*x^14 - 172*x^12 +
1121*x^10 - 859*x^8 + 373*x^6 - 99*x^4 + 15*x^2 - 1)*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)), c == (8*r4*x^40
- 120*r4*x^38 + 816*r4*x^36 - 3384*r4*x^34 + 9776*r4*x^32 - 21416*r4*x^30 +
37848*r4*x^28 - 56376*r4*x^26 + 72712*r4*x^24 - 82344*r4*x^22 + 82344*r4*x^20 -
72712*r4*x^18 + 56376*r4*x^16 - 37848*r4*x^14 + 21416*r4*x^12 - 9776*r4*x^10 +
3384*r4*x^8 - 816*r4*x^6 + 120*r4*x^4 - 8*r4*x^2 - (r4*x^30 - 2*r4*x^28 -
10*r4*x^26 + 42*r4*x^24 - 78*r4*x^22 + 100*r4*x^20 - 108*r4*x^18 + 99*r4*x^16 -
69*r4*x^14 + 33*r4*x^12 - 9*r4*x^10 + r4*x^8)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + (r4*x^38 -
16*r4*x^36 + 111*r4*x^34 - 450*r4*x^32 + 1228*r4*x^30 - 2488*r4*x^28 +
3995*r4*x^26 - 5201*r4*x^24 + 5428*r4*x^22 - 4394*r4*x^20 + 2490*r4*x^18 -
424*r4*x^16 - 1177*r4*x^14 + 1888*r4*x^12 - 1671*r4*x^10 + 984*r4*x^8 -
390*r4*x^6 + 100*r4*x^4 - 15*r4*x^2 + r4)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8))/(8*x^38 - 120*x^36 +
816*x^34 - 3384*x^32 + 9776*x^30 - 21416*x^28 + 37848*x^26 - 56376*x^24 +
72712*x^22 - 82344*x^20 + 82344*x^18 - 72712*x^16 + 56376*x^14 - 37848*x^12 +
21416*x^10 - 9776*x^8 + 3384*x^6 - 816*x^4 - (x^28 - 3*x^26 - x^24 + 11*x^22 -
21*x^20 + 27*x^18 - 29*x^16 + 24*x^14 - 15*x^12 + 6*x^10 - x^8)*((x^16 - 6*x^14
+ 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 120*x^2
+ (x^36 - 17*x^34 + 126*x^32 - 548*x^30 + 1602*x^28 - 3444*x^26 + 5809*x^24 -
8008*x^22 + 9222*x^20 - 8958*x^18 + 7318*x^16 - 4908*x^14 + 2531*x^12 - 839*x^10
+ 48*x^8 + 110*x^6 - 56*x^4 + 12*x^2 - 1)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) - 8), d == -(8*r4*x^40 -
120*r4*x^38 + 816*r4*x^36 - 3384*r4*x^34 + 9776*r4*x^32 - 21416*r4*x^30 +
37848*r4*x^28 - 56376*r4*x^26 + 72712*r4*x^24 - 82344*r4*x^22 + 82344*r4*x^20 -
72712*r4*x^18 + 56376*r4*x^16 - 37848*r4*x^14 + 21416*r4*x^12 - 9776*r4*x^10 +
3384*r4*x^8 - 816*r4*x^6 + 120*r4*x^4 - 8*r4*x^2 - (r4*x^30 - 2*r4*x^28 -
10*r4*x^26 + 42*r4*x^24 - 78*r4*x^22 + 100*r4*x^20 - 108*r4*x^18 + 99*r4*x^16 -
69*r4*x^14 + 33*r4*x^12 - 9*r4*x^10 + r4*x^8)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + (r4*x^38 -
16*r4*x^36 + 111*r4*x^34 - 450*r4*x^32 + 1228*r4*x^30 - 2488*r4*x^28 +
3995*r4*x^26 - 5201*r4*x^24 + 5428*r4*x^22 - 4394*r4*x^20 + 2490*r4*x^18 -
424*r4*x^16 - 1177*r4*x^14 + 1888*r4*x^12 - 1671*r4*x^10 + 984*r4*x^8 -
390*r4*x^6 + 100*r4*x^4 - 15*r4*x^2 + r4)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8))/(8*x^40 - 120*x^38 +
816*x^36 - 3384*x^34 + 9776*x^32 - 21416*x^30 + 37848*x^28 - 56376*x^26 +
72712*x^24 - 82344*x^22 + 82344*x^20 - 72712*x^18 + 56376*x^16 - 37848*x^14 +
```

```
21416*x^12 - 9776*x^10 + 3384*x^8 - 816*x^6 + 120*x^4 - (x^30 - 3*x^28 - x^26 +
11*x^24 - 21*x^22 + 27*x^20 - 29*x^18 + 24*x^16 - 15*x^14 + 6*x^12 -
x^10)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 +
1)/x^8)^(3/2) - 8*x^2 + (x^38 - 17*x^36 + 126*x^34 - 548*x^32 + 1602*x^30 -
3444*x^28 + 5809*x^26 - 8008*x^24 + 9222*x^22 - 8958*x^20 + 7318*x^18 -
4908*x^16 + 2531*x^14 - 839*x^12 + 48*x^10 + 110*x^8 - 56*x^6 + 12*x^4 -
x^2)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2
+ 1)/x^8))], [a == 0, b == 0, c == 0, d == 0]]
```

[19]:
```
# p^2 transformation, non-diagonalizable
var('a b c d') # setup variables
z = 1-e^(i*pi/3) # first case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[20]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*B*A*P^(-1)
D = P*B*A^(-1)*B*A*B^(-1)*P^(-1)
```

[21]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[21]:
```
[[a == 0, b == r5, c == -1/2*r5*(I*sqrt(3) - 1), d == 0], [a == 0, b == 0, c ==
0, d == 0]]
```

[22]:
```
A = Matrix([[-1,1],[0,-1]]) # setup - eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*B*A*P^(-1)
D = P*B*A^(-1)*B*A*B^(-1)*P^(-1)
```

[23]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[23]:
```
[[a == r6, b == 1/4*r6*(I*sqrt(3) - 7), c == 1/2*r6, d == -r6], [a == 0, b == 0,
c == 0, d == 0]]
```

[24]:
```
z = 1-e^(i*5*pi/3) # second case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[25]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*B*A*P^(-1)
D = P*B*A^(-1)*B*A*B^(-1)*P^(-1)
```

[26]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[26]:
```
[[a == 0, b == r7, c == 1/2*r7*(I*sqrt(3) + 1), d == 0], [a == 0, b == 0, c ==
0, d == 0]]
```

[27]:
```
A = Matrix([[-1,1],[0,-1]]) # setup - eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*B*A*P^(-1)
D = P*B*A^(-1)*B*A*B^(-1)*P^(-1)
```

[28]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[28]:
```
[[a == r8, b == -1/4*r8*(I*sqrt(3) + 7), c == 1/2*r8, d == -r8], [a == 0, b ==
0, c == 0, d == 0]]
```

[29]:
```
# sigma transformation, diagonalizable
var('a b c d x') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
y = x + x^(-1)
```

```
z = ((y^2-5)*(y^2-2) + sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # +␣
↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*A^(-1)*P^(-1) # P*sigma(A)*P^(-1)
D = P*(A^(-2)*B^(-1)*A^2)*P^(-1) # P*sigma(B)*P^(-1)
```

[30]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[30]:
```
[[a == 0, b == r9, c == 1/2*(r9*x^10 - 3*r9*x^8 + r9*x^6*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 2*r9*x^6 -
3*r9*x^4 + r9*x^2)/(x^4 - 3*x^2 + 1), d == 0], [a == 0, b == 0, c == 0, d == 0]]
```

[31]:
```
z = ((y^2-5)*(y^2-2) - sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # -␣
↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*A^(-1)*P^(-1) # P*sigma(A)*P^(-1)
D = P*(A^(-2)*B^(-1)*A^2)*P^(-1) # P*sigma(B)*P^(-1)
```

[32]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[32]:
```
[[a == 0, b == r10, c == 1/2*(r10*x^10 - 3*r10*x^8 - r10*x^6*sqrt((x^16 - 6*x^14
+ 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 2*r10*x^6 -
3*r10*x^4 + r10*x^2)/(x^4 - 3*x^2 + 1), d == 0], [a == 0, b == 0, c == 0, d ==
0]]
```

[33]:
```
# sigma transformation, non-diagonalizable
var('a b c d') # setup variables
z = 1-e^(i*pi/3) # first case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[34]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*P^(-1)
D = P*(A^(-2))*(B^(-1))*(A^2)*P^(-1)
```

[35]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[35]:
```
[[a == r11, b == 0, c == 0, d == -r11], [a == 0, b == 0, c == 0, d == 0]]
```

[36]:
```
A = Matrix([[-1,1],[0,-1]]) # setup - eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*P^(-1)
D = P*(A^(-2))*(B^(-1))*(A^2)*P^(-1)
```

[37]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[37]:
```
[[a == r12, b == -4*r12, c == 0, d == -r12], [a == 0, b == 0, c == 0, d == 0]]
```

[38]:
```
z = 1-e^(i*5*pi/3) # second case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[39]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*P^(-1)
D = P*(A^(-2))*(B^(-1))*(A^2)*P^(-1)
```

[40]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[40]:
```
[[a == r13, b == 0, c == 0, d == -r13], [a == 0, b == 0, c == 0, d == 0]]
```

[41]:
```
A = Matrix([[-1,1],[0,-1]]) # setup - eigenvalue case
B = Q*A*Q^(-1)
C = P*A^(-1)*P^(-1)
D = P*(A^(-2))*(B^(-1))*(A^2)*P^(-1)
```

[42]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[42]:
```
[[a == r14, b == -4*r14, c == 0, d == -r14], [a == 0, b == 0, c == 0, d == 0]]
```

[43]:
```
# p^2 * sigma transformation, diagonalizable
var('a b c d x') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
y = x + x^(-1)
z = ((y^2-5)*(y^2-2) + sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # +
 ↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)*P^(-1) # P*p^2sigma(A)*P^(-1)
D = P*(A^(-1)*B^(-1)*A^(-1)*B*A)*P^(-1) # P*p^2sigma(B)*P^(-1)
```

[44]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
```

```
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[44]:
```
[[a == r15, b == -r15/x^2, c == 1/4*(4*r15*x^24 - 44*r15*x^22 + 208*r15*x^20 -
568*r15*x^18 + 1032*r15*x^16 - 1388*r15*x^14 + r15*x^12*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) +
1512*r15*x^12 - 1388*r15*x^10 + 1032*r15*x^8 - 568*r15*x^6 + 208*r15*x^4 -
44*r15*x^2 + (3*r15*x^20 - 26*r15*x^18 + 91*r15*x^16 - 174*r15*x^14 +
212*r15*x^12 - 174*r15*x^10 + 91*r15*x^8 - 26*r15*x^6 + 3*r15*x^4)*sqrt((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) +
4*r15)/(x^20 - 11*x^18 + 52*x^16 - 141*x^14 + 247*x^12 - 296*x^10 + 247*x^8 -
141*x^6 + 52*x^4 - 11*x^2 + (x^16 - 8*x^14 + 24*x^12 - 34*x^10 + 24*x^8 - 8*x^6
+ x^4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8) + 1), d == -1/8*(8*r15*x^24 - 80*r15*x^22 + 344*r15*x^20 -
864*r15*x^18 + 1480*r15*x^16 - 1936*r15*x^14 + r15*x^12*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) +
2096*r15*x^12 - 1936*r15*x^10 + 1480*r15*x^8 - 864*r15*x^6 + 344*r15*x^4 -
80*r15*x^2 + (7*r15*x^20 - 50*r15*x^18 + 147*r15*x^16 - 250*r15*x^14 +
292*r15*x^12 - 250*r15*x^10 + 147*r15*x^8 - 50*r15*x^6 + 7*r15*x^4)*sqrt((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) +
8*r15)/(x^24 - 10*x^22 + 43*x^20 - 108*x^18 + 185*x^16 - 242*x^14 + 262*x^12 -
242*x^10 + 185*x^8 - 108*x^6 + 43*x^4 - 10*x^2 + (x^20 - 7*x^18 + 20*x^16 -
33*x^14 + 38*x^12 - 33*x^10 + 20*x^8 - 7*x^6 + x^4)*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 1)], [a == r16,
b == -(4*r16*x^32 - 40*r16*x^30 + 176*r16*x^28 - 476*r16*x^26 + 948*r16*x^24 -
1544*r16*x^22 + 2148*r16*x^20 - 2600*r16*x^18 + 2768*r16*x^16 - 2600*r16*x^14 +
2148*r16*x^12 - 1544*r16*x^10 + 948*r16*x^8 - 476*r16*x^6 + 176*r16*x^4 -
40*r16*x^2 - (r16*x^22 - 2*r16*x^20 + 3*r16*x^18 - 3*r16*x^16 + r16*x^14 -
r16*x^12)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2
+ 1)/x^8)^(3/2) + (r16*x^30 - 4*r16*x^28 + 23*r16*x^24 - 66*r16*x^22 +
130*r16*x^20 - 190*r16*x^18 + 219*r16*x^16 - 208*r16*x^14 + 171*r16*x^12 -
116*r16*x^10 + 62*r16*x^8 - 21*r16*x^6 + 3*r16*x^4)*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 4*r16)/(4*x^34
- 40*x^32 + 176*x^30 - 476*x^28 + 948*x^26 - 1544*x^24 + 2148*x^22 - 2600*x^20 +
2768*x^18 - 2600*x^16 + 2148*x^14 - 1544*x^12 + 948*x^10 - 476*x^8 + 176*x^6 -
40*x^4 + (x^22 - x^20 + 2*x^18 - x^16 + x^14)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 4*x^2 + (3*x^30 -
21*x^28 + 63*x^26 - 124*x^24 + 197*x^22 - 255*x^20 + 278*x^18 - 255*x^16 +
197*x^14 - 124*x^12 + 63*x^10 - 21*x^8 + 3*x^6)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)), c == (r16*x^12 -
3*r16*x^10 + 3*r16*x^8 - 4*r16*x^6 + 3*r16*x^4 - 3*r16*x^2 + (r16*x^8 +
r16*x^4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8) + r16)/(x^8 - 3*x^6 + x^4*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 2*x^4 - 3*x^2 + 1), d ==
-r16], [a == 0, b == 0, c == 0, d == 0]]
```

[45]:
```
z = ((y^2-5)*(y^2-2) - sqrt((y^2-5)^2*(y^2-2)^2+4*(y^2-5)))/(2*(y^2-5)) # -␣
↪root case
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)*P^(-1) # P*p^2sigma(A)*P^(-1)
D = P*(A^(-1)*B^(-1)*A^(-1)*B*A)*P^(-1) # P*p^2sigma(B)*P^(-1)
```

[46]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[46]:
```
[[a == r17, b == -r17/x^2, c == 1/4*(4*r17*x^24 - 44*r17*x^22 + 208*r17*x^20 -
568*r17*x^18 + 1032*r17*x^16 - 1388*r17*x^14 - r17*x^12*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) +
1512*r17*x^12 - 1388*r17*x^10 + 1032*r17*x^8 - 568*r17*x^6 + 208*r17*x^4 -
44*r17*x^2 - (3*r17*x^20 - 26*r17*x^18 + 91*r17*x^16 - 174*r17*x^14 +
212*r17*x^12 - 174*r17*x^10 + 91*r17*x^8 - 26*r17*x^6 + 3*r17*x^4)*sqrt((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) +
4*r17)/(x^20 - 11*x^18 + 52*x^16 - 141*x^14 + 247*x^12 - 296*x^10 + 247*x^8 -
141*x^6 + 52*x^4 - 11*x^2 - (x^16 - 8*x^14 + 24*x^12 - 34*x^10 + 24*x^8 - 8*x^6
+ x^4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8) + 1), d == -1/8*(8*r17*x^24 - 80*r17*x^22 + 344*r17*x^20 -
864*r17*x^18 + 1480*r17*x^16 - 1936*r17*x^14 - r17*x^12*((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) +
2096*r17*x^12 - 1936*r17*x^10 + 1480*r17*x^8 - 864*r17*x^6 + 344*r17*x^4 -
80*r17*x^2 - (7*r17*x^20 - 50*r17*x^18 + 147*r17*x^16 - 250*r17*x^14 +
292*r17*x^12 - 250*r17*x^10 + 147*r17*x^8 - 50*r17*x^6 + 7*r17*x^4)*sqrt((x^16 -
6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) +
8*r17)/(x^24 - 10*x^22 + 43*x^20 - 108*x^18 + 185*x^16 - 242*x^14 + 262*x^12 -
242*x^10 + 185*x^8 - 108*x^6 + 43*x^4 - 10*x^2 - (x^20 - 7*x^18 + 20*x^16 -
33*x^14 + 38*x^12 - 33*x^10 + 20*x^8 - 7*x^6 + x^4)*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 1)], [a == r18,
b == -(4*r18*x^32 - 40*r18*x^30 + 176*r18*x^28 - 476*r18*x^26 + 948*r18*x^24 -
1544*r18*x^22 + 2148*r18*x^20 - 2600*r18*x^18 + 2768*r18*x^16 - 2600*r18*x^14 +
2148*r18*x^12 - 1544*r18*x^10 + 948*r18*x^8 - 476*r18*x^6 + 176*r18*x^4 -
40*r18*x^2 + (r18*x^22 - 2*r18*x^20 + 3*r18*x^18 - 3*r18*x^16 + r18*x^14 -
r18*x^12)*((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2
+ 1)/x^8)^(3/2) - (r18*x^30 - 4*r18*x^28 + 23*r18*x^24 - 66*r18*x^22 +
130*r18*x^20 - 190*r18*x^18 + 219*r18*x^16 - 208*r18*x^14 + 171*r18*x^12 -
```

```
116*r18*x^10 + 62*r18*x^8 - 21*r18*x^6 + 3*r18*x^4)*sqrt((x^16 - 6*x^14 +
13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 4*r18)/(4*x^34
- 40*x^32 + 176*x^30 - 476*x^28 + 948*x^26 - 1544*x^24 + 2148*x^22 - 2600*x^20 +
2768*x^18 - 2600*x^16 + 2148*x^14 - 1544*x^12 + 948*x^10 - 476*x^8 + 176*x^6 -
40*x^4 - (x^22 - x^20 + 2*x^18 - x^16 + x^14)*((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)^(3/2) + 4*x^2 - (3*x^30 -
21*x^28 + 63*x^26 - 124*x^24 + 197*x^22 - 255*x^20 + 278*x^18 - 255*x^16 +
197*x^14 - 124*x^12 + 63*x^10 - 21*x^8 + 3*x^6)*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8)), c == (r18*x^12 -
3*r18*x^10 + 3*r18*x^8 - 4*r18*x^6 + 3*r18*x^4 - 3*r18*x^2 - (r18*x^8 +
r18*x^4)*sqrt((x^16 - 6*x^14 + 13*x^12 - 14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 -
6*x^2 + 1)/x^8) + r18)/(x^8 - 3*x^6 - x^4*sqrt((x^16 - 6*x^14 + 13*x^12 -
14*x^10 + 12*x^8 - 14*x^6 + 13*x^4 - 6*x^2 + 1)/x^8) + 2*x^4 - 3*x^2 + 1), d ==
-r18], [a == 0, b == 0, c == 0, d == 0]]
```

[47]:
```
# p^2*sigma transformation, non-diagonalizable
var('a b c d') # setup variables
z = 1-e^(i*pi/3) # first case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[48]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)*P^(-1) # P*p^2sigma(A)*P^(-1)
D = P*(A^(-1)*B^(-1)*A^(-1)*B*A)*P^(-1) # P*p^2sigma(B)*P^(-1)
```

[49]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[49]:
```
[[a == 0, b == r19, c == 1/2*r19*(I*sqrt(3) - 1), d == 0], [a == 0, b == 0, c ==
0, d == 0]]
```

[50]:
```
A = Matrix([[-1,1],[0,-1]]) # setup - eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)*P^(-1) # P*p^2sigma(A)*P^(-1)
D = P*(A^(-1)*B^(-1)*A^(-1)*B*A)*P^(-1) # P*p^2sigma(B)*P^(-1)
```

[51]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
```

```
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[51]: [[a == r20, b == -1/4*r20*(I*sqrt(3) + 9), c == 1/2*r20, d == -r20], [a == 0, b == 0, c == 0, d == 0]]

[52]:
```
z = 1-e^(i*5*pi/3) # second case
P = Matrix([[a,b],[c,d]]) # setup common matrices
Q = Matrix([[1,1],[1,z]])
```

[53]:
```
A = Matrix([[1,1],[0,1]]) # setup + eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)*P^(-1) # P*p^2sigma(A)*P^(-1)
D = P*(A^(-1)*B^(-1)*A^(-1)*B*A)*P^(-1) # P*p^2sigma(B)*P^(-1)
```

[54]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[54]: [[a == 0, b == r21, c == -1/2*r21*(I*sqrt(3) + 1), d == 0], [a == 0, b == 0, c == 0, d == 0]]

[55]:
```
A = Matrix([[-1,1],[0,-1]]) # setup - eigenvalue case
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)*P^(-1) # P*p^2sigma(A)*P^(-1)
D = P*(A^(-1)*B^(-1)*A^(-1)*B*A)*P^(-1) # P*p^2sigma(B)*P^(-1)
```

[56]:
```
eq1 = A[0,0] == C[0,0] # solve system for a, b, c, d
eq2 = A[0,1] == C[0,1]
eq3 = A[1,0] == C[1,0]
eq4 = A[1,1] == C[1,1]
eq5 = B[0,0] == D[0,0]
eq6 = B[0,1] == D[0,1]
eq7 = B[1,0] == D[1,0]
eq8 = B[1,1] == D[1,1]
solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], a, b, c, d)
```

[56]: [[a == r22, b == 1/4*r22*(I*sqrt(3) - 9), c == 1/2*r22, d == -r22], [a == 0, b
== 0, c == 0, d == 0]]

Figure-8-Knot-Calculations-Clean

July 19, 2023

[1]:
```
# p^2 transformation, diagonalizable, general case. these methods only work if␣
↪a solution exists
var('a b c d x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
R = factor(C)[0][0]
L = factor(D)[0][0]
sola=1
solb = factor(solve(R == L, b)[0].rhs().substitute(a=sola)) # there is only one␣
↪solution
R = factor(C)[1][0]
L = factor(D)[1][0]
sold = factor(solve(R == L, d)[0].rhs())
R = factor(E)[0][0]
L = factor(F)[0][0]
solc = factor(solve(R == L, c)[0].rhs().substitute(a=sola).substitute(b=solb))
sold = sold.substitute(a=sola).substitute(c=solc)
Psol = Matrix([[sola, solb],[solc, sold]]) # there's a constant factor not␣
↪shown here in front of P
print(Psol)
det = Psol.determinant()
```

```
[
1
-1/(x^2*z)]
[-(x^10*z - 2*x^8*z^2 - 2*x^8*z - x^8 + 4*x^6*z^2 + 5*x^6*z + x^6 - 4*x^4*z^2 -
5*x^4*z - x^4 + x^2*z^2 + 4*x^2*z - z)/(x^6*(z - 1)^2)  (x^10*z - 2*x^8*z^2 -
2*x^8*z - x^8 + 4*x^6*z^2 + 5*x^6*z + x^6 - 4*x^4*z^2 - 5*x^4*z - x^4 + x^2*z^2
+ 4*x^2*z - z)/(x^8*(z - 1)^2)]
```

[2]:
```
# check that the conjugation matrix works
assume((x^2+x^(-2)-3)*z^2-(x^2+x^(-2))*(x^2+x^(-2)-3)*z-1==0) # sage needs this␣
↪info
assert(Psol*(A^(-1)*B*A) == A*Psol)
assert(Psol*(B*A^(-1)*B*A*B^(-1)) == B*Psol)
```

[3]:
```
R = PolynomialRing(QQ, 'x, y, z') # find the determinant in the field F
R.inject_variables()
I = R.ideal([((x+y)^2-5)*z^2-((x+y)^2-5)*((x+y)^2-2)*z-1,x*y-1])
Q = R.quotient(I)
F = Q.fraction_field()
det_in_field = F(det)
print(simplify(det_in_field))
```

```
Defining x, y, z
(-xbar^8 + ybar^4*zbar^4 + ybar^4*zbar^3 - xbar^2*zbar^5 - ybar^2*zbar^5 +
ybar^4*zbar^2 - xbar^2*zbar^4 - 4*ybar^2*zbar^4 - 15*ybar^4*zbar -
3*ybar^2*zbar^3 + 3*zbar^5 - 2*xbar^4 - 4*xbar^2*zbar^2 + 13*ybar^2*zbar^2 +
4*zbar^4 + 40*xbar^2*zbar + 39*ybar^2*zbar + 2*zbar^3 - 6*xbar^2 - 38*zbar^2 -
13*zbar - 15)/(-ybar^4*zbar^4 - 2*ybar^4*zbar^3 + xbar^2*zbar^5 + ybar^2*zbar^5
- 4*ybar^4*zbar^2 + 2*xbar^2*zbar^4 + 5*ybar^2*zbar^4 + 8*ybar^4*zbar +
3*xbar^2*zbar^3 + 9*ybar^2*zbar^3 - 3*zbar^5 + xbar^4 + 10*xbar^2*zbar^2 +
2*ybar^2*zbar^2 - 7*zbar^4 - 23*xbar^2*zbar - 21*ybar^2*zbar - 12*zbar^3 +
3*xbar^2 + 16*zbar^2 + 4*zbar + 8)
```

[4]:
```
f = R.hom([sqrt(3), 1/sqrt(3), (5+2*sqrt(13))/3], RR) # define the needed␣
↪homomorphism from R -> Reals
top, bottom = R(det.numerator()), R(det.denominator()) # extract the numerator␣
↪and denominator
print(f(top), f(bottom)) # map them to the reals
```

```
-763.599691833407 1012.29946070846
```

[5]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a + b*x + c/(z-1)*(x*z-1/x) + d/(z-1)*(x^2*z-1) == Psol[0,0]
eq2 = c/(z-1)*(1/x-x) + d/(z-1)*(1-x^2) == Psol[0,1] # [0,1] component
eq3 = c/(z-1)*z*(x-1/x) + d/(z-1)*z*(1-x^(-2)) == Psol[1,0]
eq4 = a + b/x + c/(z-1)*(z/x-x) + d/(z-1)*(x^(-2)*z-1) == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == (x^10 + x^6 + x^4 - (4*x^8 - 5*x^6 + 5*x^4 - x^2)*z^3 + (4*x^10 - 5*x^8 +
13*x^6 - 7*x^4 + 5*x^2 - 1)*z^2 - (x^12 - x^10 + 6*x^8 - x^6 + 4*x^4)*z)/((x^8 -
x^6)*z^3 - 2*(x^8 - x^6)*z^2 + (x^8 - x^6)*z), b == -(x^10 + x^6 - (5*x^8 -
7*x^6 + 5*x^4 - x^2)*z^3 + (4*x^10 - 3*x^8 + 9*x^6 - 8*x^4 + 5*x^2 - 1)*z^2 -
(x^12 - x^10 + 7*x^8 - 3*x^6 + 2*x^4)*z)/((x^9 - x^7)*z^3 - 2*(x^9 - x^7)*z^2 +
(x^9 - x^7)*z), c == (x^6 + 2*(x^6 - x^4 + x^2)*z^2 + x^2 - (x^8 - x^6 + 4*x^4 -
x^2 + 1)*z)/((x^5 - x^3)*z^2 - (x^5 - x^3)*z), d == -(x^6 + (2*x^6 - 2*x^4 +
```

```
x^2)*z^2 - (x^8 - x^6 + 4*x^4 - 3*x^2 + 1)*z)/((x^6 - x^4)*z^2 - (x^6 - x^4)*z)]
]
```

[6]:
```
# do some manual substitution, multiplying through by a common factor of␣
↪x^7(x^2-1)(x^2 + x^-2 -3),
# and only looking at numerators
u = (x^2+x^(-2))*z + (x^2+x^(-2)-3)^(-1) # z^2
v = ((x^2+x^(-2))^2+(x^2+x^(-2)-3)^(-1))*z + (x^2+x^(-2))*(x^2+x^(-2)-3)^(-1)␣
↪#z^3
a_new = (x^2+x^(-2)-3)*x*(x^10 + x^6 + x^4 - (4*x^8 - 5*x^6 + 5*x^4 - x^2)*v +␣
↪(4*x^10 - 5*x^8 + 13*x^6 - 7*x^4 + 5*x^2 - 1)*u - (x^12 - x^10 + 6*x^8 - x^6␣
↪+ 4*x^4)*z)
b_new = -(x^2+x^(-2)-3)*(x^10 + x^6 - (5*x^8 - 7*x^6 + 5*x^4 - x^2)*v + (4*x^10␣
↪- 3*x^8 + 9*x^6 - 8*x^4 + 5*x^2 - 1)*u - (x^12 - x^10 + 7*x^8 - 3*x^6 +␣
↪2*x^4)*z)
c_new = (x^2+x^(-2)-3)*x^4*(x^6 + 2*(x^6 - x^4 + x^2)*u + x^2 - (x^8 - x^6 +␣
↪4*x^4 - x^2 + 1)*z)
d_new = -(x^2+x^(-2)-3)*x^3*(x^6 + (2*x^6 - 2*x^4 + x^2)*u - (x^8 - x^6 + 4*x^4␣
↪- 3*x^2 + 1)*z)
print(factor(a_new))
print(factor(b_new))
print(factor(c_new))
print(factor(d_new))
```

```
(-1) * x * (x - 1)^4 * (x + 1)^4 * (x^2 + 1) * (x^4*z - x^2*z - x^2 + z)
x^2 * (x - 1)^4 * (x + 1)^4 * (2*x^4*z - 3*x^2*z - x^2 + 2*z)
x^2 * (x - 1)^2 * (x + 1)^2 * (x^2 - x + 1) * (x^2 + x + 1) * (x^4*z - 3*x^2*z +
x^2 + z)
(-1) * (x - 1)^2 * (x + 1)^2 * x^3 * (x^2 + 1) * (x^4*z - 3*x^2*z + x^2 + z)
```

[7]:
```
# p^2 transformation, non-diagonalizable, general case (+1 eigenvalue). these␣
↪methods only work if a solution exists
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
R = factor(C)[0][0]
L = factor(D)[0][0]
sola = 0
```

```
sold = 0
solb = 1
solc = factor(solve(R == L, c)[0].rhs().substitute(b=solb).substitute(a=sola).
 ↪substitute(d=sold))
Psol = Matrix([[sola, solb],[solc, sold]]) # there's a constant factor not␣
 ↪shown here in front of P
print(Psol)
det = Psol.determinant()
```

```
[          0           1]
[-1/(z - 1)            0]
```

[8]:
```
# check that the conjugation matrix works
assume(z^2-z+1==0)
assert(Psol*(A^(-1)*B*A) == A*Psol)
assert(Psol*(B*A^(-1)*B*A*B^(-1)) == B*Psol)
```

[9]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a + b + c/(z-1)*(z-2) + d/(z-1)*(z-3) == Psol[0,0]
eq2 = b + c/(z-1) + d/(z-1)*(z+1) == Psol[0,1] # [0,1] component
eq3 = -c/(z-1) - d/(z-1) == Psol[1,0]
eq4 = a + b + c/(z-1)*z + d/(z-1)*z == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == -2*(2*z - 1)/(z - 1), b == (3*z - 2)/(z - 1), c == 3, d == -2]
]
```

[10]:
```
# p^2 transformation, non-diagonalizable, general case (-1 eigenvalue). these␣
 ↪methods only work if a solution exists
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
R = factor(C)[1][1]
L = factor(D)[1][1]
sold = 1
solc = factor(solve(R == L, c)[0].rhs().substitute(d=sold))
R = factor(E)[1][1]
```

```
L = factor(F)[1][1]
solb = factor(solve(R == L, b)[0].rhs().substitute(c=solc).substitute(d=sold))
R = factor(C)[0][0]
L = factor(D)[0][0]
sola = factor(solve(R == L, a)[0].rhs().substitute(b=solb).substitute(c=solc).
 ↪substitute(d=sold))
Psol = Matrix([[sola, solb],[solc, sold]]) # there's a constant factor not␣
 ↪shown here in front of P
print(Psol)
det = Psol.determinant()
```

```
[1/4*(z^3 - 5*z^2 + 9*z - 4)/(z - 1)^2       1/2*(2*z^2 - 6*z + 3)/(z - 1)^2]
[                                 -1/2                                      1]
```

[11]:
```
# check that the conjugation matrix works
assume(z^2-z+1==0)
assert(Psol*(A^(-1)*B*A) == A*Psol)
assert(Psol*(B*A^(-1)*B*A*B^(-1)) == B*Psol)
```

[12]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a - b - c/(z-1)*z + d == Psol[0,0]
eq2 = b + c/(z-1) - d == Psol[0,1] # [0,1] component
eq3 = -c/(z-1) + d/(z-1) == Psol[1,0]
eq4 = a - b + c/(z-1)*(2-z) + d/(z-1)*(z-2) == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == -1/4*(z^4 - 9*z^3 + 21*z^2 - 16*z + 4)/(z^2 - 2*z + 1), b == -1/4*(z^4 -
7*z^3 + 17*z^2 - 14*z + 4)/(z^2 - 2*z + 1), c == -1/4*(z^3 - 7*z^2 + 13*z -
6)/(z - 1), d == -1/4*(z^3 - 5*z^2 + 9*z - 4)/(z - 1)]
]
```

[13]:
```
# sigma transformation, diagonalizable, general case. these methods only work␣
 ↪if a solution exists
clear_vars()
forget()
var('a b c d x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*A^(-1)
D = A*P
E = P*(A^(-2)*B^(-1)*A^2)
F = B*P
```

```
sola = 0
solb = 1
sold = 0
R = factor(E)[0][0]
L = factor(F)[0][0]
solc = factor(solve(R == L, c)[0].rhs().substitute(a=sola).substitute(b=solb))
Psol = Matrix([[sola, solb],[solc, sold]])
print(Psol) # there's a constant factor not shown here in front of P
det = Psol.determinant()
```

```
[    0     1]
[x^4*z     0]
```

[14]:
```
# check that the conjugation matrix works
assert(Psol*A^(-1) == A*Psol)
assert(Psol*(A^(-2)*B^(-1)*A^2) == B*Psol)
```

[15]:
```
R = PolynomialRing(QQ, 'x, y, z') # find the determinant in the field F
R.inject_variables()
I = R.ideal([((x+y)^2-5)*z^2-((x+y)^2-5)*((x+y)^2-2)*z-1,x*y-1])
Q = R.quotient(I)
F = Q.fraction_field()
det_in_field = F(det) # from here I'm not sure what to do, sage can't factor in
    ↪the field
print(simplify(det_in_field))
```

```
Defining x, y, z
ybar^4*zbar - xbar^2*zbar^2 - ybar^2*zbar^2 - 3*xbar^2*zbar - 3*ybar^2*zbar +
3*zbar^2 + 2*zbar + 1
```

[16]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a + b*x + c/(z-1)*(x*z-1/x) + d/(z-1)*(x^2*z-1) == Psol[0,0]
eq2 = c/(z-1)*(1/x-x) + d/(z-1)*(1-x^2) == Psol[0,1] # [0,1] component
eq3 = c/(z-1)*z*(x-1/x) + d/(z-1)*z*(1-x^(-2)) == Psol[1,0]
eq4 = a + b/x + c/(z-1)*(z/x-x) + d/(z-1)*(x^(-2)*z-1) == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == (x^8 - (x^6 + x^2)*z + 1)/(x^4 - 2*x^2 + 1), b == -(x^7 - (x^5 + x^3)*z +
x)/(x^4 - 2*x^2 + 1), c == -(x^7 - (x^7 + x)*z + x)/(x^4 - 2*x^2 + 1), d == (x^6
+ x^2 - (x^6 + x^2)*z)/(x^4 - 2*x^2 + 1)]
]
```

[17]:
```
# sigma transformation, non-diagonalizable, general case (+1 eigenvalue). these
    ↪methods only work if a solution exists
clear_vars()
forget()
```

```
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
C = P*A^(-1)
D = A*P
E = P*(A^(-2)*B^(-1)*A^2)
F = B*P
Psol = Matrix([[1,0],[0,-1]])
det = Psol.determinant()
```

[18]:
```
# check that the conjugation matrix works
assert(Psol*A^(-1) == A*Psol)
assert(Psol*(A^(-2)*B^(-1)*A^2) == B*Psol)
```

[19]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a + b + c/(z-1)*(z-2) + d/(z-1)*(z-3) == Psol[0,0]
eq2 = b + c/(z-1) + d/(z-1)*(z+1) == Psol[0,1] # [0,1] component
eq3 = -c/(z-1) - d/(z-1) == Psol[1,0]
eq4 = a + b + c/(z-1)*z + d/(z-1)*z == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == -2*z - 1, b == 2*z, c == 2*z - 2, d == -2*z + 2]
]
```

[20]:
```
# sigma transformation, non-diagonalizable, general case (-1 eigenvalue). these␣
↪methods only work if a solution exists
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
C = P*A^(-1)
D = A*P
E = P*(A^(-2)*B^(-1)*A^2)
F = B*P
Psol = Matrix([[1,-4],[0,-1]])
det = Psol.determinant()
```

[21]:
```
# check that the conjugation matrix works
assert(Psol*A^(-1) == A*Psol)
assert(Psol*(A^(-2)*B^(-1)*A^2) == B*Psol)
```

[22]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a - b - c/(z-1)*z + d == Psol[0,0]
eq2 = b + c/(z-1) - d == Psol[0,1] # [0,1] component
eq3 = -c/(z-1) + d/(z-1) == Psol[1,0]
eq4 = a - b + c/(z-1)*(2-z) + d/(z-1)*(z-2) == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == -2*z - 1, b == -2*z, c == -2*z + 2, d == -2*z + 2]
]
```

[23]:
```
# p^2*sigma transformation, diagonalizable, general case. these methods only␣
↪work if a solution exists
clear_vars()
forget()
var('a b c d x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)
D = A*P
E = P*(A^(-1)*B^(-1)*A^(-1)*B*A)
F = B*P
R = factor(C)[0][0]
L = factor(D)[0][0]
sola = 1
solb = factor(solve(R == L, b)[0].rhs().substitute(a=sola)) # there is only one␣
↪solution
R = factor(C)[1][0]
L = factor(D)[1][0]
sold = factor(solve(R == L, d)[0].rhs())
R = factor(E)[0][0]
L = factor(F)[0][0]
solc = factor(solve(R == L, c)[0].rhs().substitute(a=sola).substitute(b=solb))
sold = sold.substitute(a=sola).substitute(c=solc)
Psol = Matrix([[sola,solb],[solc,sold]]) # there's a constant factor not shown␣
↪here in front of P
print(Psol)
det = Psol.determinant()
```

```
[      1 -1/x^2]
[ x^2*z      -1]
```

[24]:
```
# check that the conjugation matrix works
assert(Psol*(A^(-1)*B^(-1)*A) == A*Psol)
assert(Psol*(A^(-1)*B^(-1)*A^(-1)*B*A) == B*Psol)
```

[25]:
```
R = PolynomialRing(QQ, 'x, y, z') # find the determinant in the field F
R.inject_variables()
I = R.ideal([((x+y)^2-5)*z^2-((x+y)^2-5)*((x+y)^2-2)*z-1,x*y-1])
Q = R.quotient(I)
F = Q.fraction_field()
det_in_field = F(det) # from here I'm not sure what to do, sage can't factor in␣
↪the field
print(simplify(det_in_field))
```

```
Defining x, y, z
zbar - 1
```

[26]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a + b*x + c/(z-1)*(x*z-1/x) + d/(z-1)*(x^2*z-1) == Psol[0,0]
eq2 = c/(z-1)*(1/x-x) + d/(z-1)*(1-x^2) == Psol[0,1] # [0,1] component
eq3 = c/(z-1)*z*(x-1/x) + d/(z-1)*z*(1-x^(-2)) == Psol[1,0]
eq4 = a + b/x + c/(z-1)*(z/x-x) + d/(z-1)*(x^(-2)*z-1) == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == (x^6 - (x^4 + x^2)*z + 1)/(x^4 - x^2), b == -(x^4 - x^2*z - x^2 + 1)/(x^3
- x), c == -(x^4 + x^2 - (x^4 + x^2 + 1)*z + 1)/(x^3 - x), d == (x^2 - (x^2 +
1)*z + 1)/(x^2 - 1)]
]
```

[27]:
```
# p^2*sigma transformation, non-diagonalizable, general case (+1 eigenvalue).␣
↪these methods only work if a solution exists
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)
D = A*P
E = P*(A^(-1)*B^(-1)*A^(-1)*B*A)
F = B*P
R = factor(C)[0][0]
```

```
L = factor(D)[0][0]
sola = 0
sold = 0
solb = 1
solc = factor(solve(R == L, c)[0].rhs().substitute(b=solb).substitute(a=sola).
 ↪substitute(d=sold))
Psol = Matrix([[sola, solb],[solc, sold]]) # there's a constant factor not␣
 ↪shown here in front of P
print(Psol)
det = Psol.determinant()
```

```
[        0          1]
[1/(z - 1)          0]
```

[28]:

```
# check that the conjugation matrix works
assert(Psol*(A^(-1)*B^(-1)*A) == A*Psol)
assert(Psol*(A^(-1)*B^(-1)*A^(-1)*B*A) == B*Psol)
```

[29]:

```
# find P as a E-linear sum of I, A, B, AB
eq1 = a + b + c/(z-1)*(z-2) + d/(z-1)*(z-3) == Psol[0,0]
eq2 = b + c/(z-1) + d/(z-1)*(z+1) == Psol[0,1] # [0,1] component
eq3 = -c/(z-1) - d/(z-1) == Psol[1,0]
eq4 = a + b + c/(z-1)*z + d/(z-1)*z == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == 2*z/(z - 1), b == -z/(z - 1), c == -3, d == 2]
]
```

[30]:

```
# p^2*sigma transformation, non-diagonalizable, general case (-1 eigenvalue).␣
 ↪these methods only work if a solution exists
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
P = Matrix([[a,b],[c,d]])
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
C = P*(A^(-1)*B^(-1)*A)
D = A*P
E = P*(A^(-1)*B^(-1)*A^(-1)*B*A)
F = B*P
R = factor(C)[1][1]
L = factor(D)[1][1]
sold = 1
solc = factor(solve(R == L, c)[0].rhs().substitute(d=sold))
```

```
R = factor(E)[1][1]
L = factor(F)[1][1]
solb = factor(solve(R == L, b)[0].rhs().substitute(c=solc).substitute(d=sold))
R = factor(C)[0][0]
L = factor(D)[0][0]
sola = factor(solve(R == L, a)[0].rhs().substitute(b=solb).substitute(c=solc).
 ↪substitute(d=sold))
Psol = Matrix([[sola, solb],[solc, sold]]) # there's a constant factor not␣
 ↪shown here in front of P
print(Psol)
det = Psol.determinant()
```

```
[          -1 -1/2*z + 5/2]
[        -1/2            1]
```

[31]:
```
# check that the conjugation matrix works
assert(Psol*(A^(-1)*B^(-1)*A) == A*Psol)
assert(Psol*(A^(-1)*B^(-1)*A^(-1)*B*A) == B*Psol)
```

[32]:
```
# find P as a E-linear sum of I, A, B, AB
eq1 = a - b - c/(z-1)*z + d == Psol[0,0]
eq2 = b + c/(z-1) - d == Psol[0,1] # [0,1] component
eq3 = -c/(z-1) + d/(z-1) == Psol[1,0]
eq4 = a - b + c/(z-1)*(2-z) + d/(z-1)*(z-2) == Psol[1,1]
sol = solve([eq1, eq2, eq3, eq4], a, b, c, d)
print(sol)
```

```
[
[a == z, b == 1/2*z, c == 3/2*z - 3/2, d == z - 1]
]
```

[33]:
```
# some algebra calculations, diagonalizable case
clear_vars()
forget()
var('a b c d x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = A*B
T = A^2
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[33]: [[a == -1, b == (x^2 + 1)/x, c == 0, d == 0]]

[34]:
```
clear_vars()
forget()
var('a b c d x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = A*B
T = B^2
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[34]: [[a == -1, b == 0, c == (x^2 + 1)/x, d == 0]]

[35]:
```
clear_vars()
forget()
var('a b c d x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
C = A*B
T = B*A
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[35]: [[a == (x^4 - 2*x^2*z + 1)/(x^2*z - x^2), b == (x^2 + 1)/x, c == (x^2 + 1)/x, d == -1]]

[36]:
```
# some algebra calculations, non-diagonalizable case (positive eigenvalue)
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
```

```
C = A*B
T = A^2
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[36]: [[a == -1, b == 2, c == 0, d == 0]]

[37]:
```
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
C = A*B
T = B^2
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[37]: [[a == -1, b == 0, c == 2, d == 0]]

[38]:
```
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
C = A*B
T = B*A
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[38]: [[a == -(2*z - 1)/(z - 1), b == 2, c == 2, d == -1]]

[39]:
```
# some algebra calculations, non-diagonalizable case (negative eigenvalue)
clear_vars()
forget()
```

```
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
C = A*B
T = A^2
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[39]: [[a == -1, b == -2, c == 0, d == 0]]

[40]:
```
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
C = A*B
T = B^2
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[40]: [[a == -1, b == 0, c == -2, d == 0]]

[41]:
```
clear_vars()
forget()
var('a b c d z') # setup variables and matrices
assume(z != 0, z != 1)
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
C = A*B
T = B*A
eq1 = a + b*A[0,0] + c*B[0,0] + d*C[0,0] == T[0,0]
eq2 = b*A[0,1] + c*B[0,1] + d*C[0,1] == T[0,1]
eq3 = b*A[1,0] + c*B[1,0] + d*C[1,0] == T[1,0]
eq4 = a + b*A[1,1] + c*B[1,1] + d*C[1,1] == T[1,1]
solve([eq1, eq2, eq3, eq4], a, b, c, d)
```

[41]: [[a == -(2*z - 1)/(z - 1), b == -2, c == -2, d == -1]]

[42]:
```
# some more algebra
clear_vars()
forget()
var('q w')
a_1 = w*(q^2-1)-q^2*(q^2-2)
b_1 = q*(q^2-w-1)
c_1 = q*(q^2-3)
d_1 = 2-q^2
a_2 = q*(q^2-w-1)
b_2 = w-q^2+1
c_2 = 1-q^2
d_2 = q
a = a_1*a_2 - b_1*b_2 + (w-q^2)*c_1*b_2 - q*d_1*b_2 - c_1*c_2 - q*c_1*d_2 -
 ↪d_1*d_2
b = a_1*b_2 + a_2*b_1 + q*b_1*b_2 + q*c_1*b_2 + w*d_1*b_2 - d_1*c_2 + c_1*d_2
c = a_1*c_2 + a_2*c_1 - b_1*d_2 + q*c_1*b_2 + d_1*b_2 + q*c_1*c_2 + w*c_1*d_2
d = d_1*a_2 + a_1*d_2 + b_1*c_2 + q*b_1*d_2 - c_1*b_2 + q*d_1*c_2 + w*d_1*d_2
print(factor(a))
print(factor(b))
print(factor(c))
print(factor(d))
```

```
-(q^2 - w)*(q^2 - w - 2)*q
(q^2 - w + 1)*(q^2 - w - 2)
(q^2 - w - 2)*(q + 1)*(q - 1)
-(q^2 - w - 2)*q
```

[43]:
```
# checking the final results for p^2, diagonal case
clear_vars()
forget()
var('x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
q = x + x^(-1)
w = (z*(q^2-2)-2)/(z-1)
a = q*(q^2-w)
b = w-q^2-1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
```

```
F = B*P
assert(C == D)
assume((x^2+x^(-2)-3)*z^2-(x^2+x^(-2))*(x^2+x^(-2)-3)*z-1==0) # sage needs this␣
↪info
assert(E == F)
```

[44]:

```
# checking the final results for p^2, non-diagonalizable case (+1 eigenvalue)
clear_vars()
forget()
z = 1-e^(i*pi/3)
q = 2
w = 2-1/(z-1)
a = q*(q^2-w)
b = w-q^2-1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
assert(C == D)
assert(E == F)
```

[45]:

```
# checking the final results for p^2, non-diagonalizable case (+1 eigenvalue)
clear_vars()
forget()
z = 1-e^(i*5*pi/3)
q = 2
w = 2-1/(z-1)
a = q*(q^2-w)
b = w-q^2-1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
assert(C == D)
assert(E == F)
```

[46]:
```
# checking the final results for p^2, non-diagonalizable case (-1 eigenvalue)
clear_vars()
forget()
z = 1-e^(i*pi/3)
q = -2
w = 2-1/(z-1)
a = q*(q^2-w)
b = w-q^2-1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
assert(C == D)
assert(E == F)
```

[47]:
```
# checking the final results for p^2, non-diagonalizable case (-1 eigenvalue)
clear_vars()
forget()
z = 1-e^(i*5*pi/3)
q = -2
w = 2-1/(z-1)
a = q*(q^2-w)
b = w-q^2-1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B*A)
D = A*P
E = P*(B*A^(-1)*B*A*B^(-1))
F = B*P
assert(C == D)
assert(E == F)
```

[48]:
```
# checking the final results for sigma, diagonal case
clear_vars()
forget()
var('x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
```

```
assume(z != 0, z != 1)
q = x + x^(-1)
w = (z*(q^2-2)-2)/(z-1)
a = w*(q^2-1)-q^2*(q^2-2)
b = q*(q^2-w-1)
c = q*(q^2-3)
d = 2-q^2
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*A^(-1)
D = A*P
E = P*(A^(-2)*B^(-1)*A^2)
F = B*P
assert(C == D)
assert(E == F)
```

[49]:

```
# checking the final results for sigma, non-diagonal case (+1 eigenvalue)
clear_vars()
forget()
var('z') # setup variables and matrices
assume(z != 0, z != 1)
q = 2
w = 2-1/(z-1)
a = w*(q^2-1)-q^2*(q^2-2)
b = q*(q^2-w-1)
c = q*(q^2-3)
d = 2-q^2
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*A^(-1)
D = A*P
E = P*(A^(-2)*B^(-1)*A^2)
F = B*P
assert(C == D)
assert(E == F)
```

[50]:

```
# checking the final results for sigma, non-diagonal case (-1 eigenvalue)
clear_vars()
forget()
var('z') # setup variables and matrices
assume(z != 0, z != 1)
q = -2
w = 2-1/(z-1)
```

```
a = w*(q^2-1)-q^2*(q^2-2)
b = q*(q^2-w-1)
c = q*(q^2-3)
d = 2-q^2
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*A^(-1)
D = A*P
E = P*(A^(-2)*B^(-1)*A^2)
F = B*P
assert(C == D)
assert(E == F)
```

[51]:
```
# checking the final results for p^2*sigma, diagonal case
clear_vars()
forget()
var('x z') # setup variables and matrices
assume(x != 0, x^2 != 4, x^2 != 5)
assume(z != 0, z != 1)
q = x + x^(-1)
w = (z*(q^2-2)-2)/(z-1)
a = q*(q^2-w-1)
b = w-q^2+1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[x,0],[0,x^(-1)]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B^(-1)*A)
D = A*P
E = P*(A^(-1)*B^(-1)*A^(-1)*B*A)
F = B*P
assert(C == D)
assert(E == F)
```

[52]:
```
# checking the final results for p^2*sigma, non-diagonal case (+1 eigenvalue)
clear_vars()
forget()
var('z') # setup variables and matrices
assume(z != 0, z != 1)
q = 2
w = 2-1/(z-1)
a = q*(q^2-w-1)
b = w-q^2+1
```

```
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[1,1],[0,1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B^(-1)*A)
D = A*P
E = P*(A^(-1)*B^(-1)*A^(-1)*B*A)
F = B*P
assert(C == D)
assert(E == F)
```

[53]:
```
# checking the final results for p^2*sigma, non-diagonal case (-1 eigenvalue)
clear_vars()
forget()
var('z') # setup variables and matrices
assume(z != 0, z != 1)
q = -2
w = 2-1/(z-1)
a = q*(q^2-w-1)
b = w-q^2+1
c = 1-q^2
d = q
Q = Matrix([[1,1],[1,z]])
A = Matrix([[-1,1],[0,-1]])
B = Q*A*Q^(-1)
P = a*A*A^(-1) + b*A + c*B + d*A*B
C = P*(A^(-1)*B^(-1)*A)
D = A*P
E = P*(A^(-1)*B^(-1)*A^(-1)*B*A)
F = B*P
assert(C == D)
assert(E == F)
```

9-48-Calculations

August 18, 2023

[1]:
```
# calculations for knot 9-48
# first, define the knot group by presentation
F.<g_1,g_2,g_3,g_4,g_5,g_6,g_7,g_8,g_9,g_10> = FreeGroup()
eq1 = g_8^(-1)*g_1^(-1)*g_9*g_1
eq2 = g_8^(-1)*g_2*g_8*g_1
eq3 = g_8^(-1)*g_2^(-1)*g_7*g_2
eq4 = g_2^(-1)*g_4*g_2*g_3
eq5 = g_2^(-1)*g_10^(-1)*g_3*g_10
eq6 = g_10^(-1)*g_3^(-1)*g_9*g_3
eq7 = g_4^(-1)*g_1^(-1)*g_5*g_1
eq8 = g_10^(-1)*g_5^(-1)*g_1*g_5
eq9 = g_5^(-1)*g_10^(-1)*g_6*g_10
eq10 = g_7^(-1)*g_1^(-1)*g_6*g_1
G = F / [eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq9,eq10]
print(G)
```

```
Finitely presented group < g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_10 |
g_8^-1*g_1^-1*g_9*g_1, g_8^-1*g_2*g_8*g_1, g_8^-1*g_2^-1*g_7*g_2,
g_2^-1*g_4*g_2*g_3, g_2^-1*g_10^-1*g_3*g_10, g_10^-1*g_3^-1*g_9*g_3,
g_4^-1*g_1^-1*g_5*g_1, g_10^-1*g_5^-1*g_1*g_5, g_5^-1*g_10^-1*g_6*g_10,
g_7^-1*g_1^-1*g_6*g_1 >
```

[2]:
```
# simplify the group presentation
hk = G.simplification_isomorphism()
H = G.simplified()
print(hk)
print(H)
```

```
Generic morphism:
  From: Finitely presented group < g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9,
g_10 | g_8^-1*g_1^-1*g_9*g_1, g_8^-1*g_2*g_8*g_1, g_8^-1*g_2^-1*g_7*g_2,
g_2^-1*g_4*g_2*g_3, g_2^-1*g_10^-1*g_3*g_10, g_10^-1*g_3^-1*g_9*g_3,
g_4^-1*g_1^-1*g_5*g_1, g_10^-1*g_5^-1*g_1*g_5, g_5^-1*g_10^-1*g_6*g_10,
g_7^-1*g_1^-1*g_6*g_1 >
  To:   Finitely presented group < g_1, g_2, g_4 |
g_2*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-1*g_4,
g_2^-1*g_1*g_4^-1*g_1^-1*g_4*g_1^-1*g_2^-1*g_4^-1*g_2*g_1*g_4^-1*g_1*g_4*g_1^-1
>
  Defn: g_1 |--> g_1
```

```
        g_2 |--> g_2
        g_3 |--> g_2^-1*g_4^-1*g_2
        g_4 |--> g_4
        g_5 |--> g_1*g_4*g_1^-1
        g_6 |--> g_1*g_4^-1*g_1*(g_4*g_1^-1)^2
        g_7 |--> g_4^-1*g_1*g_4*g_1^-1*g_4
        g_8 |--> g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1
        g_9 |--> g_1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2
        g_10 |--> g_1*g_4^-1*g_1*g_4*g_1^-1
Finitely presented group < g_1, g_2, g_4 |
g_2*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-1*g_4,
g_2^-1*g_1*g_4^-1*g_1^-1*g_4*g_1^-1*g_2^-1*g_4^-1*g_2*g_1*g_4^-1*g_1*g_4*g_1^-1
>
```

[3]:
```
# create homomorphism on free group
f = F.hom([g_9, g_10^-1, g_10*g_1^-1*g_10^-1, g_1, g_9*g_1*g_9^-1,
 ↪g_9*g_1^-1*g_9*(g_1*g_9^-1)^2, g_1^-1*g_9*g_1*g_9^-1*g_1,
 ↪g_1^-1*g_9*g_1*g_9^-1*g_1*g_10^-1*g_9,
 ↪g_9*g_1^-1*g_9*g_1*g_9^-1*g_1*g_10^-1, g_9*g_1^-1*g_9*g_1*g_9^-1])
```

[4]:
```
# apply it 3 times then move into simplified group
print(hk(G(f(f(f(g_1))))))
print("")
print(hk(G(f(f(f(g_9))))))
print("")
print(hk(G(f(f(f(g_4))))))
```

```
g_1*(g_4^-1*g_1*g_4*g_1^-1*g_4*g_2)^2*g_1*g_2^-1*g_4^-1*(g_1*g_4^-1*g_1^-
1*g_4)^2*g_1^-1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-
1*g_4^2*g_2*g_1*(g_4^-1*g_1*g_4*g_1^-1)^2*g_4*g_2*g_1^-1*g_2^-1*g_4^-1*g_1*g_4^-
1*g_1^-1*g_4*g_1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1^-1*(g_2^-1*g_4^-1*g_1*g_4^-
1*g_1^-1*g_4)^2*g_1^-1

g_1*(g_4^-1*g_1*g_4*g_1^-1*g_4*g_2)^2*g_1*g_2^-1*g_4^-1*(g_1*g_4^-1*g_1^-
1*g_4)^2*g_1^-1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-
1*g_4^2*g_2*g_1*(g_4^-1*g_1*g_4*g_1^-1)^2*g_4*g_2*g_1^-1*g_2^-1*g_4^-1*g_1*g_4^-
1*g_1^-1*g_4*g_1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1^-1*(g_2^-1*g_4^-1*g_1*g_4^-
1*g_1^-1*g_4)^2*g_1^-1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-1*g_4^-1*g_1*g_4^-
1*g_1^-1*g_4^2*g_2*g_1*(g_4^-1*g_1*g_4*g_1^-1)^2*g_4*g_2*g_1^-1*g_2^-1*g_4^-
1*g_1*g_4^-1*g_1^-1*g_4*g_1*g_4*g_2*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-
1*g_4^-1*g_1*g_4^-1*g_1^-1*g_4*g_1^-1*g_4^-1*g_1*g_4*g_1^-1*g_4*g_2*g_1*g_2^-
1*g_4^-1*(g_1*g_4^-1*g_1^-1*g_4)^2*g_1^-1*g_2^-1*g_4^-2*g_1*g_4*g_1^-
1*g_4*g_2*g_1^-1*g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-1*g_4*g_1*(g_4^-1*g_1*g_4*g_1^-
1*g_4*g_2)^2*g_1*g_2^-1*g_4^-1*(g_1*g_4^-1*g_1^-1*g_4)^2*g_1^-1*g_2^-1*g_4^-
2*g_1*g_4*g_1^-1*g_4*g_2*g_1^-1*g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-1*g_4*g_1*(g_4^-
1*g_1*g_4*g_1^-1)^2*g_4*g_2*g_1^-1*(g_2^-1*g_4^-1*g_1*g_4^-1*g_1^-1*g_4)^2*g_1^-
1
```

```
g_1*(g_4^-1*g_1*g_4*g_1^-1*g_4*g_2)^2*g_1*g_2^-1*g_4^-1*(g_1*g_4^-1*g_1^-
1*g_4)^2*g_1^-1
```