Original Research Article

Roots in Motion: How Directional Bias Alters Twist Knot Trajectories

Zijiao Zheng

International School of Beijing, 10 An Hua Street, Shunyi, Beijing, 101318, China

ABSTRACT

A directional bias parameter q is introduced into a random walk model on knot diagrams in order to study how it alters the algebraic root trajectories of the deformed Alexander polynomial for the twist knot family $J(2, 2n)$. Using absorbing Markov chains, generalized Alexander-type polynomials are derived for the four smallest twist knots: trefoil (3_1), figure-eight (4_1), three-twist (5_2), and stevedore (6_1). The roots are computed for $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and visualized on the complex plane. The unit circle $|z| = 1$ and the imaginary axis $\mathrm{Re}(z) = 0$ are treated as algebraic reference boundaries; crossings of these boundaries are reported as observed algebraic events. Within the tested range, the trefoil shows no unit-circle crossing, the figure-eight shows one outward crossing between $q = 0.3$ and $q = 0.5$, and the three-twist crosses the unit circle exactly at $q = 0.5$ (modulus 0.664 at $q = 0.3$ and modulus 1.830 at $q = 0.7$), and the stevedore shows two inward crossings at $q \in \{0.1, 0.3\}$ and $q \in \{0.7, 0.9\}$ and is the only knot with imaginary-axis crossings and a change in the number of real roots. Based on these four case studies, a conjecture is identified, not a proven result, that the number and type of algebraic reference boundary crossings may increase with crossing number for the twist knot family. This algebraic framework provides a method for tracking root trajectories in topological systems. However, the exact connection between polynomial roots and Markov-chain absorption is not established here and is left for future work.

Keywords: Alexander polynomial; Markov chains; twist knots; root trajectories; boundary crossing; directional bias; complex plane; knot invariants

INTRODUCTION

When is the last time you tried to resolve a tangled pair of headphones, a knotted shoelace, or a twisted electrical cord? These everyday messes look random and useless, but knot theory turns them into rigorous mathematics that has served as a practical tool in biology, chemistry, and physics for modeling knotted structures since the 20th century. For example, DNA loops and twists affect gene expression, enzymes fold into knots that determine their function, and polymer physics and quantum field theory also borrow knot theory insights.

To study knots mathematically, topologists extract and idealize them by joining the loose ends to form a closed circle. A knot is a continuous and closed loop without self-intersections, S^1, embedded in the three-dimensional Euclidean space, $\mathbb{R}^3$ (1). A knot diagram is a two-dimensional projection of a knot that demonstrates the overpass, which is the strand on top, and the underpass, which is the strand below, at each crossing. Two knots are equivalent, or isomorphic, if one can be deformed into the other without cutting or passing through itself. To distinguish knots, topologists use knot invariants, which are algebraic or numerical quantities that stay unchanged

Corresponding author: Zijiao Zheng, E-mail: 17801772381@163.com.
Copyright: © 2026 Zijiao Zheng. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
Accepted November 11, 2026
https://doi.org/10.70251/HYJR2348.46625634

under such deformations. If two knots have different invariants, they are different knots, or non-isomorphic. But the converse, where two knots have the same invariant, cannot guarantee their equivalence because different knots might have several identical invariants, and they can only be distinguished when two different invariants are found.

The Alexander polynomial is a powerful invariant that assigns a polynomial $\Delta(t)$ with integer coefficients to each knot (2). Its roots carry topological information, and their behavior under deformation has been a popular research area. For a knot with n crossings, the normalized Alexander polynomial has degree at most $n - 1$ (3). The computation may generate a polynomial multiplied by $\pm t^k$, where $k \in \mathbb{Z}$, depending on the choice of diagram and labeling, but the degree is at most $n - 1$ once $\pm t^k$ is removed. Although the Alexander polynomial is stronger than numerical invariants like the knot determinant, it cannot distinguish a knot from its mirror image (4).

The standard Alexander polynomial is calculated from an oriented knot diagram by extracting the determinant of an $(n - 1) \times (n - 1)$ minor of the Alexander matrix using cofactor expansion. When deriving the generalized Alexander-type polynomial $\Delta(t; q)$, the same linear algebra techniques are used since the coefficient matrix of the difference equations that model each crossing is structurally isomorphic to the deformed Alexander matrix.

The family of twist knots $J(2, 2n)$ is studied, which are knots formed by adding consecutive half-twists to a loop before closing it. Since the number of half-twists can grow arbitrarily large, the family has a clear and simple structure parameterized by crossing number , which makes twist knots an ideal testing ground for topological conjectures. The four smallest members are analyzed, excluding their mirror images, to ease calculations: the trefoil (3_1), figure-eight (4_1), three-twist (5_2), and stevedore (6_1). The four knots have crossing numbers $n = 3$, 4, 5, and 6, respectively.

Probability theory offers a dynamic alternative to fixed topological invariants (5). In Markov chains (6), a particle performs random walks along the knot's strands. The states of the system are the oriented arcs, or unbroken strands, of the knot diagram. At each crossing, the particle follows an outgoing strand with a specific probability. The Law of Total Probability says the chance of arriving at an outgoing strand equals the sum of the chances from all incoming paths, while conditional probability dictates each transition. A Markov chain is a stochastic process in which the next state depends only on the current state, not on the history of the walk. An absorbing Markov chain contains at least one absorbing state that, once the particle enters it, it never returns to any other state.

Previous research has explored connections among knot theory, random walks, polynomial invariants, and dynamical systems. Lin and Wang (7) developed a random-walk formulation on knot diagrams and related the Alexander polynomial to an Ihara–Selberg-type zeta function involving weighted cycles. However, their work did not examine how directional bias in crossing weights affects polynomial roots. Nawata et al. (8) investigated recursive structures in generalized polynomial invariants of twist knots, providing an algebraic framework for comparing members of this family, but did not analyze root trajectories under a continuously varying parameter. Noguchi (9) examined relationships between Alexander polynomials and dynamical or number-theoretic properties, including entropy and the growth of homological invariants, without establishing a direct dynamical interpretation of the parameter-dependent root trajectories considered here.

Collectively, these studies establish the mathematical foundation for probabilistic and algebraic approaches to knot invariants. However, how directional bias in crossing weights influences the root trajectories of generalized Alexander-type polynomials across selected twist knots remains insufficiently explored, particularly their crossings of reference boundaries in the complex plane.

To address this gap, the present study introduces a directional bias parameter into a random-walk construction on twist-knot diagrams. Generalized Alexander-type polynomials are derived for four knots: the trefoil, figure-eight, three-twist, and stevedore, and their complex roots are tracked as varies. The analysis focuses on root trajectories and crossings of the unit circle and imaginary axis, treating these strictly as algebraic observations rather than indicators of physical or stochastic stability. By comparing these four constructions, the study seeks to identify patterns in polynomial-root trajectories and propose a tentative conjecture for future investigation. The findings are limited to the selected knots and parameter range, and broader generalizations require further calculations or rigorous mathematical proof.

METHODS AND MATERIALS

Probabilistic Framework on Knot Diagrams

Twist knots $J(2, 2n)$ are modeled as absorbing Markov chains. Each oriented arc of the knot diagram represents a state where a probabilistic particle can be located. At every crossing, the probabilities of a particle moving to the next available strands strictly sum to less than 1. The missing fraction is assigned to a single absorbing state,

written , which the particle enters on leaving the knot and never leaves. This dissipation is naturally built into the bias parameter q, so the chain terminates rather than looping indefinitely. The random walks in this study follow Lin and Wang's established rules (7).

Formally, $M(t, q) = I - W(t, q)$ with $W(t, q) \geq 0$, and every row of $M(t, q)$ sums to the absorption probability of the corresponding crossing. Adding Γ as an $(n + 1)$-st coordinate therefore gives the full transition matrix, in block form, $P(t, q) = [W; (t, q)\ \ a\ (t, q)\ 0 \cdots 0\ 1]$, in which every row sums to 1 and $a\ (t, q)$ is the column of absorption probabilities $1 - \sum_j W_{ij}\ (t, q)$. The admissible parameter domain is $0 < t < 1$ and $0 < q < 1$, within which every entry of $W(t, q)$ and $a\ (t, q)$ is non-negative.

Topological Signs and Directional Bias

Crossings are classified by their topological sign, which is positive or negative. If one travels along the underpass, a positive crossing has the overpass moving from left to right, while a negative crossing has the overpass moving from right to left. The overpass is the single, continuous strand that does not break at the crossing. The strand that breaks is the underpass, creating one incoming underpass and one outgoing underpass. At any crossing, the probability of a particle landing on the outgoing underpass is solved for. To evaluate the asymmetry, the directional bias parameter $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ is applied. A particle can land on the outgoing underpass via two incoming paths:

1. It was on the incoming underpass and continued straight with transmission probability t.
2. It was on the continuous overpass and transferred down to the outgoing underpass. The bias parameter q splits the baseline transfer probability $1 - t$ depending on the crossing's sign:
    I. Positive Crossing: The transfer probability from the over-arc down to the under-arc is $q(1 - t)$. Thus, the probability of probability is $1 - (t + q(1 - t)) = (1 - q)(1 - t)$.
    II. Negative Crossing: The transfer probability from the over-arc down to the under-arc is $(1 - q)\ (1 - t)$. The probability of absorption is $1 - (t + (1 - q)\ (1 - t)) = q(1 - t)$.

This creates controlled asymmetry, for example, if $q = 0.9$, the particle is highly likely to jump down from the overpass at positive crossings but severely restricted from doing so at negative crossings.

The Difference Equation

The Law of Total Probability states that the total chance of being on the outgoing strand x_j is the sum of all the mutually exclusive incoming paths. For a positive crossing, $x_j = tx_i + q(1 - t)\ x_k$ where x_j is the probability of the particle being on the outgoing underpass strand, tx_i is the probability that the particle came from the incoming underpass x_i and traveled straight t, and $q(1 - t)\ x_k$ is the probability that the particle came from the continuous overpass x_k and transferred down onto the outgoing strand. To satisfy the properties of an absorbing Markov chain, this relationship can be formalized as a local transition probability matrix. Mapping the incoming states to the outgoing underpass state gives the matrix

$$[x_j] = [t\ q(1 - t)] \begin{bmatrix} x_i \\ x_k \end{bmatrix}$$

For a negative crossing, $x_j = tx_i + (1 - q)(1 - t)\ x_k$, which is

$$[x_j] = [(t\ (1 - q)(1 - t)] \begin{bmatrix} x_i \\ x_k \end{bmatrix}$$

Figure 1 shows a labeled diagram for both crossings. Each strand is a state with a transition weight.

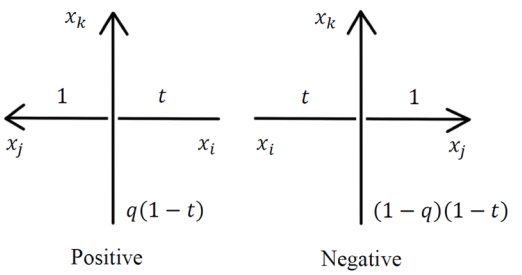


Figure 1. *Positive and negative crossings labeled with states and transition weights.*

Algorithmic Generation of the Coefficient Matrix

By structuring the difference equations for all n crossings of a given twist knot, an $n \times n$ transition coefficient matrix $M_{t,q}$ can be extracted. Because the spatial relations at the crossings in this model are identical to the standard Wirtinger presentation, the probabilistic difference equations generate the same system of linear relations as the deformed Alexander matrix. Therefore, mapping standard topological signs to the t and q variables produces a coefficient matrix with the same structure as this traditional type of matrix.

Extracting the Deformed Polynomial

The polynomial $\Delta(t; q)$ is extracted from a selected $(n - 1) \times (n - 1)$ minor of the deformed Alexander matrix.

In the undeformed Alexander polynomial case, different minors differ only by a unit $\pm t^k$, so the invariant is well defined. In the deformed case, the parameter appears in the matrix entries, and different minors can yield polynomials with different q-dependence.

It is therefore not claimed here that the polynomial is invariant under the choice of minor, labeling, or diagram. Instead, it chooses a specific minor for each knot and studies the root trajectories of the resulting polynomial. The selection criterion is that the chosen minor retains the bias parameter q through cofactor expansion. It must also yield a polynomial satisfying $\Delta(1; q) = 1$, which is a normalization check in Results. Systematically analyzing how the choice of minor affects deformed polynomials is future work.

Data Analysis

The framework for extracting the Alexander-type polynomials allows for high computational efficiency since the symbolic matrix determinant is calculated once per knot to derive the generalized formula $\Delta(t; q)$. To evaluate the four twist knots, five q-values are substituted into the generalized formulae, generating 20 distinct polynomials.

Because manual root extraction for polynomials of degree up to five is impractical, Mathematica was used to automate the calculation of all complex roots. Once calculated, these roots are visualized on the complex plane, where the horizontal axis represents the real part, Re, and the vertical axis represents the imaginary part, Im. Parameterized root trajectories are generated on the complex plane. The bias parameter q acts as a temporal variable that, as it scales from 0.1 to 0.9, the changing coordinates of each root trace a distinct, continuous path. These trajectories are visually indicated using directional arrows and color gradients from red to purple to represent the trajectory created by the increasing q-value.

To ensure analytical rigor, visual inspection of shifting roots is not performed. For each discrete q-value, the root modulus $|z| = \sqrt{(\mathrm{Re}^2 + \mathrm{Im}^2)}$ and their rate of displacement across the complex plane are calculated. The goal is to track these metrics to identify potential root-trajectory thresholds, which are specific thresholds where a continuous deformation in q forces a root to cross critical algebraic boundaries.

The modulus measures a root's distance from the origin, telling how close it is to crossing a critical boundary. The rate of displacement measures sensitivity, telling how violently or slowly a root moves when the bias parameter q varies. Because q acts as a discrete temporal variable, the rate of displacement can be calculated as the change in the root's position divided by the change in q. For a root z that moves from z_1 to z_2 as q increases from q_1 to q_2, the rate of displacement equals the difference quotient $\frac{z_2 - z_1}{q_2 - q_1}$.

In discrete probabilistic systems, these critical boundaries are usually the unit circle $|z| = 1$ (10) or the imaginary axis $\mathrm{Re}(z) = 0$ (11). A root crossing these boundaries is observed as an algebraic root event. Unit-circle crossings are reported purely in terms of root moduli exceeding or shrinking below 1 and imaginary-axis crossings in terms of the real part changing sign. In highly asymmetrical systems, root trajectories can become noisy and hard to track. In such cases, the dominant root, which is the root with the largest modulus, can be tracked as the primary indicator of the algebraic trajectory.

Limits of Interpretation

Many roots of the deformed Alexander-type polynomial $\Delta(t; q)$ are complex values of the parameter t for which the determinant of the selected minor equals zero. They are not, in general, eigenvalues of the transient transition matrix of the absorbing Markov chain. The unit-circle and imaginary-axis crossings reported are therefore algebraic events in the root trajectories. A rigorous connection between root location and absorption probability remains an open question.

RESULTS

To demonstrate how the bias parameter q creates asymmetry in transition weights, the generalized polynomials are calculated for the four twist knots and evaluated at each value of q. To build the deformed matrices, each crossing is assigned its own row. For clarity, the equations are arranged in the way that the outgoing underpass strand always serves as the anchor of the difference equation, with a coefficient of 1.

Polynomial Calculations

The Trefoil Knot (3_1)

The calculation begins with the right-handed trefoil (Figure 2A), which has three positive crossings.

Let the sequence of arcs be x_1, x_2, x_3. Because all crossings are positive, the transfer probability for the overpass is always $q(1 - t)$. Writing down the difference equations for each crossing:

Crossing 1 (Outputs to x_1): $x_1 = tx_2 + q(1 - t)\, x_3 \Longrightarrow x_1 - tx_2 - q(1 - t)\, x_3 = 0$

Crossing 2 (Outputs to x_2): $x_2 = tx_3 + q(1 - t)\, x_1 \Longrightarrow -q(1$

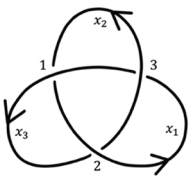


Figure 2A

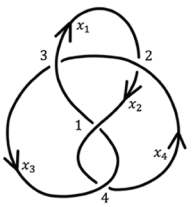


Figure 2B

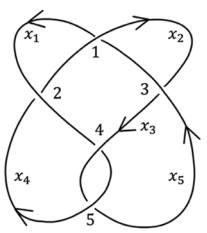


Figure 2C

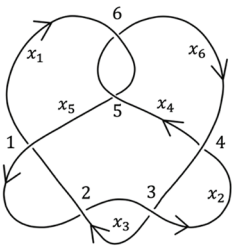


Figure 2D

Figure 2. *Labeled and oriented diagrams for the four twist knots. (A) Right-handed trefoil (3_1); (B) figure-eight (4_1); (C) right-handed three-twist (5_2); (D) stevedore (6_1). In each panel, labels an oriented arc and the numbered labels identify the crossing. Arrows indicate strand orientation.*

$-t)\,x_1 + x_2 - tx_3 = 0$

Crossing 3 (Outputs to x_3): $x_3 = tx_1 + q(1-t)\,x_2 \Longrightarrow -tx_1 - q(1-t)\,x_2 + x_3 = 0$

Placing the coefficients of x_1, x_2, x_3 into columns generates a 3 × 3 Alexander-type matrix with 1s on the left-to-right diagonal.

$$M = \begin{pmatrix} 1 & -t & -q(1-t) \\ -q(1-t) & 1 & -t \\ -t & -q(1-t) & 1 \end{pmatrix}$$

Because the knot is a closed loop, the full system is redundant. Deleting the third row and third column gives the selected minor M_{33} that contains the q value.

$$M_{33} = \begin{pmatrix} 1 & -t \\ -q(1-t) & 1 \end{pmatrix}$$

The determinant is calculated by cross-multiplying and subtracting.

$$\Delta(t;q) = (1)(1) - (-t)(-q(1-t)) = 1 - qt(1-t) = qt^2 - qt + 1$$

Substituting the five q-values gives the trefoil polynomials in Table 1.

The Figure-Eight Knot (4_1)

The figure-eight knot (Figure 2B) contains four crossings. Two are positive and two are negative.

Following Figure 2B, the difference equations could be written as follows.

Crossing 1 (Negative, outputs to x_1): $x_1 - (1-q)(1-t)\,x_2 - tx_3 = 0$

Crossing 2 (Positive, outputs to x_2): $-tx_1 + x_2 - q(1-t)\,x_4 = 0$

Crossing 3 (Positive, outputs to x_3): $-q(1-t)x_1 + x_3 - tx_4 = 0$

Crossing 4 (Negative, outputs to x_4): $-tx_2 - (1-q)(1-t)\,x_3 + x_4 = 0$

Aligning these equations gives a 4 × 4 matrix. If an arc is not involved in a crossing, it is given a coefficient of 0.

$$M = \begin{pmatrix} 1 & -(1-q)(1-t) & -t & 0 \\ -t & 1 & 0 & -q(1-t) \\ -q(1-t) & 0 & 1 & -t \\ 0 & -t & -(1-q)(1-t) & 1 \end{pmatrix}$$

Table 1. *Alexander-type polynomials after substituting all q values for the four twist knots.*

q Values	Twist Knots			
	Trefoil (3_1)	**Figure-Eight (4_1)**	**Three-Twist (5_2)**	**Stevedore (6_1)**
0.1	$0.1t^2 - 0.1t + 1$	$0.99t^3 + 0.02t^2 - 0.11t + 0.1$	$0.999t^4 - 0.097t^3 - 0.003t^2 + 0.101t$	$0.0361t^5 + 0.6156t^4 + 1.3356t^3 - 2.2914t^2 + 1.1331t + 0.171$
0.3	$0.3t^2 - 0.3t + 1$	$0.91t^3 + 0.18t^2 - 0.39t + 0.3$	$0.973t^4 - 0.219t^3 - 0.081t^2 + 0.327t$	$0.2601t^5 + 0.9996t^4 - 0.3864t^3 - 0.3094t^2 + 0.0791t + 0.357$
0.5	$0.5t^2 - 0.5t + 1$	$0.75t^3 + 0.5t^2 - 0.75t + 0.5$	$0.875t^4 - 0.125t^3 - 0.375t^2 + 0.625t$	$0.5625t^5 + 0.7500t^4 - 0.7500t^3 + 0.3750t^2 - 0.3125t + 0.375$
0.7	$0.7t^2 - 0.7t + 1$	$0.51t^3 + 0.98t^2 - 1.19t + 0.7$	$0.657t^4 + 0.329t^3 - 1.029t^2 + 1.043t$	$0.8281t^5 + 0.3276t^4 - 0.4944t^3 + 0.3666t^2 - 0.3009t + 0.273$
0.9	$0.9t^2 - 0.9t + 1$	$0.19t^3 + 1.62t^2 - 1.71t + 0.9$	$0.271t^4 + 1.287t^3 - 2.187t^2 + 1.629t$	$0.9801t^5 + 0.0396t^4 - 0.1284t^3 + 0.1166t^2 - 0.1069t + 0.099$

Deleting the first row and third column produces:

$$M_{13} = \begin{pmatrix} -t & 1 & -q(1-t) \\ -q(1-t) & 0 & -t \\ 0 & -t & 1 \end{pmatrix}$$

Taking the determinant by expanding along the first column yields:

$$\Delta(t; q) = -q^2t^3 + 2q^2t^2 - q^2t + t^3 - qt + q$$

The Three-Twist Knot (5_2)

The three-twist knot has five positive crossings (Figure 2C).

The Alexander-type matrix is written as:

$$M = \begin{pmatrix} 1 & -q(1-t) & 0 & 0 & -t \\ -q(1-t) & 1 & 0 & -t & 0 \\ 0 & -t & 1 & 0 & -q(1-t) \\ -t & 0 & -q(1-t) & 1 & 0 \\ 0 & 0 & -t & -q(1-t) & 1 \end{pmatrix}$$

To ease the minor calculation, let $a = -q(1 - t)$ and $b = -t$. Then, M_{15} is extracted and expanded along the first column. To evaluate the two 3 × 3 matrix determinants, they are multiplied by their front coefficient, and $-q(1 - t)$ and $-t$ are substituted back. Combining and simplifying the components lead to:

$$\Delta(t; q) = -q^3t^4 + 3q^3t^3 - 3q^3t^2 + q^3t - qt^3 + t^4 + qt$$

The Stevedore Knot (6_1)

The stevedore knot has six crossings, four negative and two positive (Figure 2D).

The Alexander-type matrix is written as:

$$M = \begin{pmatrix} 1 & 0 & -t & 0 & -(1-q)(1-t) & 0 \\ 0 & 1 & -(1-q)(1-t) & 0 & -t & 0 \\ 0 & -(1-q)(1-t) & 1 & 0 & 0 & -t \\ 0 & -t & 0 & 1 & 0 & -(1-q)(1-t) \\ -t & 0 & 0 & -q(1-t) & 1 & 0 \\ -q(1-t) & 0 & 0 & -t & 0 & 1 \end{pmatrix}$$

M_{51} is extracted and expanded along the last row, then the same variable substitution method and matrix evaluation method for the 5_2 knot are applied to obtain the general polynomial:

$$\begin{aligned} \Delta(t; q) = {} & q^4t^5 - 4q^4t^4 - 4q^3t^5 + 6q^4t^3 + 16q^3t^4 + 4q^2t^5 - 4q^4t^2 \\ & - 25q^3t^3 - 20q^2t^4 + q^4t + 19q^3t^2 + 36q^2t^3 + 8qt^4 \\ & - 7q^3t - 31q^2t^2 - 20qt^3 + q^3 + 14q^2t + 20qt^2 + 3t^3 \\ & - 3q^2 - 10qt - 4t^2 + 2q + 2t \end{aligned}$$

All 20 polynomials are collected in Table 1 after substituting q. The Alexander polynomial has the convention $\Delta(1)=\pm1$ (4), which provides a check on the polynomials. All 20 polynomials equal 1 at t=1, providing a consistency check that they were extracted correctly.

Data Processing

Processing the data in Mathematica, four root trajectory diagrams (Figure 3–6) are generated for the twist knots. Roots are matched between consecutive q-values by minimum-distance pairing, $z_i\ (q_{k+1}) \sim z_j\ (q_k)$ chosen to minimize $|z_i\ (q_{k+1}) - z_j\ (q_k)|$. This rule also determines which branch each member of a newly formed conjugate pair follows. When two real roots converge and become a conjugate pair, the two members have identical real parts, so a distance test cannot separate them; the pairing is then fixed by continuity of the ordering at the previous sampled value, with the root that carried the lower label at q_k keeping the lower label at q_{k+1}. These labels are a bookkeeping convention only and do not affect any reported modulus or boundary crossing.

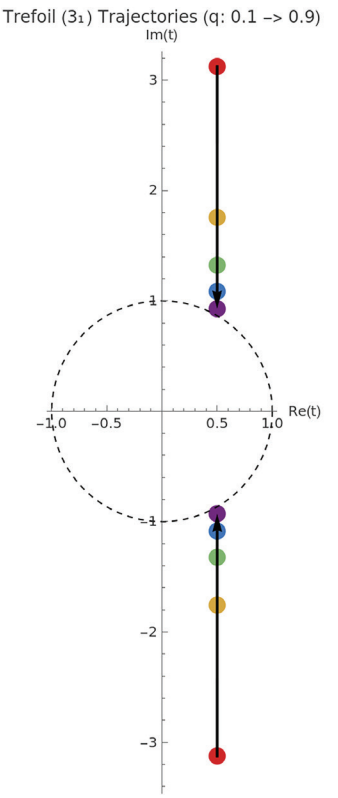


Figure 3. *Trefoil (3_1) root trajectories as increases from 0.1 to 0.9. Each colored dot is a root of $\Delta(t; q) = 0$ at a sampled -value. The color gradient runs from red ($q = 0.1$) to purple ($q = 0.9$), and the black arrows follow the direction of increasing q. The dashed circle is the unit circle $|z| = 1$. The horizontal axis is Re(t) and the vertical axis is Im(t). Trajectories are numbered in order of increasing imaginary part, starting with the most negative.*

Trefoil Knot (3_1) Observations

The quadratic polynomial produces two roots, which form a complex conjugate pair symmetric about the real axis, with real part fixed at 0.5. Their moduli decrease from to as increases, with the trajectories approaching the unit circle but never crossing it. No trajectory approaches or recedes from the imaginary axis. The rate of displacement drops steadily as increases, from 6.83 for $q \in \{0.1, 0.3\}$ down to 0.79 for $q \in \{0.7, 0.9\}$. No algebraic boundary transition occurs within the tested range.

Figure-Eight Knot (4_1) Observations

Three trajectories appear since the polynomial has a degree of three. Trajectory 2 is real and moves leftward along the negative real axis. The other two form a complex conjugate pair, symmetric about the real axis and staying inside the unit circle throughout. Trajectory 2 crosses the unit circle outward between q=0.3 where $|z|$=0.979 and q=0.5 where $|z|$=1.572, marking the first unit-circle crossing in the family. No trajectory crosses the imaginary axis since all move farther from it as q increases. The real trajectory accelerates with q up to 33.16 at the last shift, while the complex pair decelerates down to 0.083.

Three-Twist Knot (5_2) Observations

Four trajectories appear since this is a quartic polynomial. Trajectory 2 is real and crosses the unit circle exactly at q = 0.5, where its modulus is 1. At the sampled points immediately below and above, its moduli are approximately 0.664 at q = 0.3 and 1.830 at q = 0.7. The crossing sits exactly at the sampled value, the upper endpoint of the bracketing interval, and therefore occurs later than the figure-eight's crossing. Trajectory 3 is also real but stays at the origin for all q. Trajectories 1 and 4 form a complex conjugate pair and stay inside the unit circle. No trajectory crosses the imaginary axis. The rate of displacement mirrors that of the figure-eight knot, where the nonzero real trajectory accelerates up to 21.88, while the complex pair decelerates to 0.32.

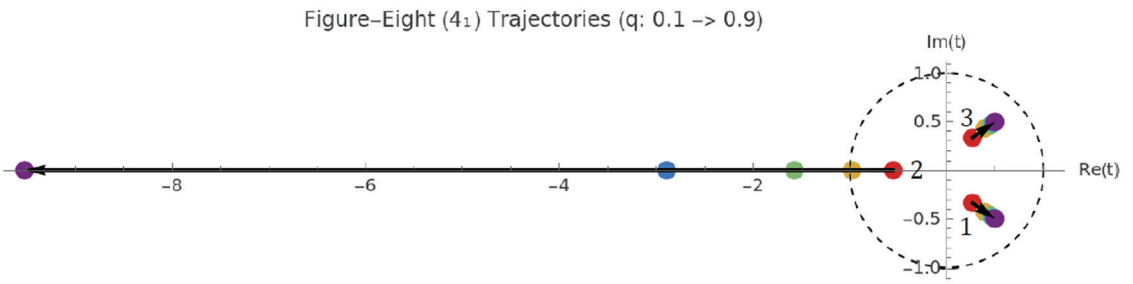


Figure 4. *Figure-eight (4_1) root trajectories. Each colored dot is a root of $\Delta(t; q) = 0$ at a sampled q-value; The color gradient runs from red (q = 0.1) to purple (q = 0.9); black arrows indicate the direction of increasing ; the dashed circle is the unit circle $|z| = 1$; horizontal and vertical axes are Re(t) and Im(t); trajectories are numbered in order of increasing imaginary part, starting with the most negative.*

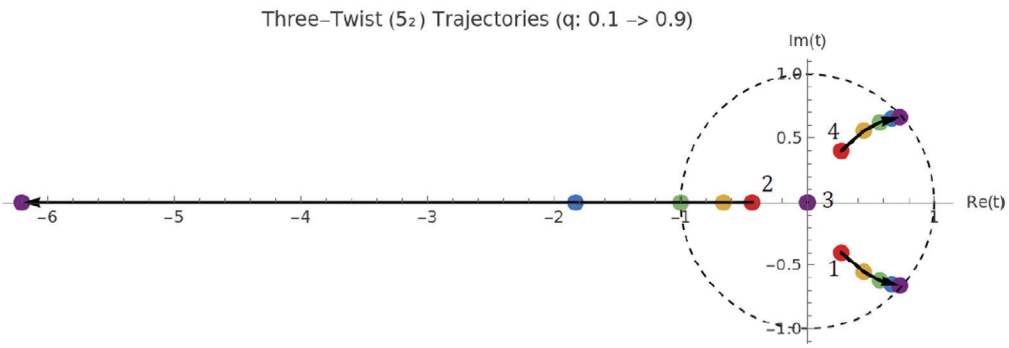


Figure 5. *Three-twist (5_2) root trajectories. Each colored dot is a root of $\Delta(t; q) = 0$ at a sampled q-value; The color gradient runs from red (q = 0.1) to purple (q = 0.9); black arrows indicate the direction of increasing q; the dashed circle is the unit circle $|z| = 1$; horizontal and vertical axes are Re(t) and Im(t); trajectories are numbered in order of increasing imaginary part, starting with the most negative.*

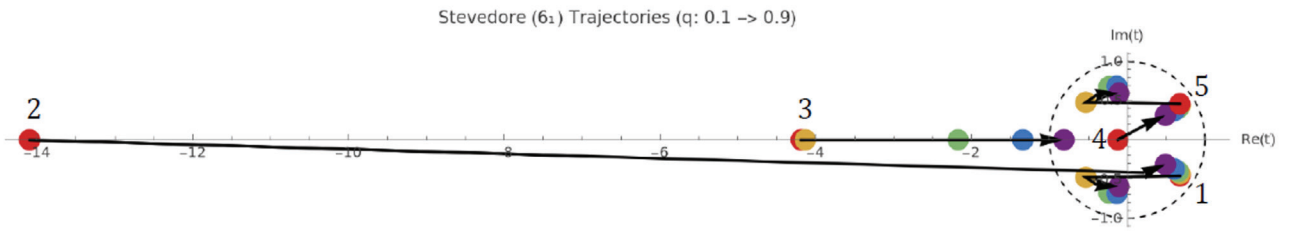


Figure 6. *Stevedore (6_1) root trajectories. Each colored dot is a root of $\Delta(t; q) = 0$ at a sampled q-value; The color gradient runs from red ($q = 0.1$) to purple ($q = 0.9$); black arrows indicate the direction of increasing q; the dashed circle is the unit circle $|z| = 1$; horizontal and vertical axes are Re(t) and Im(t); trajectories are numbered in order of increasing imaginary part, starting with the most negative.*

Stevedore Knot (6_1) Observations

Five trajectories appear since this is a quintic polynomial. Only trajectory 3 is real, moving along the negative real axis from −4.19 to −0.81, and it is the only trajectory that does not cross the imaginary axis. Trajectories 1 and 5 form a complex conjugate pair for all q, and trajectories 2 and 4 start as real roots at q=0.1 before becoming a second conjugate pair at q=0.3. Two trajectories cross the unit circle inward. Trajectory 2 crosses between q=0.1 and q=0.3, dropping from $|z| = 14.10$ to $|z| = 0.80$, while trajectory 3 between q=0.7 and q=0.9, dropping from $|z|$= 1.34 to $|z| = 0.81$. This is the only knot in the study with imaginary-axis crossings and where the number of real roots changes as q grows from 0.1 to 0.3.

DISCUSSION

A consistent root-trajectory pattern is identified within the tested range $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ across the four tested knots. The trefoil shows no algebraic boundary transition, the figure-eight knot shows one outward crossing of the unit circle between $q = 0.3$ and $q = 0.5$, the three-twist knot crosses the unit circle outward exactly at $q = 0.5$, and the stevedore knot shows two inward crossings of the unit circle at $q \in \{0.1, 0.3\}$ and $q \in \{0.7, 0.9\}$, with four trajectories crossing the imaginary axis. For the twist knot family $J(2, 2n)$, the number and type of algebraic boundary crossings may increase with crossing number. It is emphasized that these are observations from four cases, not a proven family-wide result. This study extends the probabilistic framework of Lin and Wang (7) by demonstrating that a directional bias in transition weights systematically alters the coefficients and roots of the deformed Alexander polynomial. The four twist knots reveal a consistent pattern in how the deformed Alexander polynomial roots respond to the bias parameter q, with the unit circle $|z| = 1$ serving frequently as an algebraic boundary.

For the twist knot family $J(2, 2n)$, as the crossing number increases, the number and type of algebraic boundary crossings appear to increase across the four tested knots, as summarized in Table 2. The trefoil with $n = 3$ displays the simplest behavior, where its two complex conjugate roots remain outside the unit circle for all tested q, with moduli decreasing as q increases but with real parts never changing. These roots approach the unit circle but never cross it, so no boundary crossing occurs. The figure-eight shows the first crossing, where trajectory 2 crosses outward between $q = 0.3$ and $q = 0.5$. The three-twist trajectory 2 crosses the unit circle exactly at $q = 0.5$ (moduli ~0.664 at $q = 0.3$, 1 at $q = 0.5$, ~1.830 at $q = 0.7$). The stevedore with $n = 6$ shows the most complicated behavior, with two inward unit-circle crossings, where trajectory 2 crosses between $q = 0.1$ and $q = 0.3$ and trajectory 3 crosses between $q = 0.7$ and $q = 0.9$. The stevedore is also the only knot where trajectories cross the imaginary axis and where the number of real roots changes as q increases from 0.1 to 0.3. As n increases from 4 to 6, the highest threshold range shifts from (0.3, 0.5) to (0.7, 0.9). This suggests that the threshold may grow with crossing number, though neither the existence nor the rate of such growth can be established from four case studies. Additionally, the number of thresholds grows from one to two and the direction of the crossing flips from outward to inward.

The recursive structure that Nawata et al. (8) found in twist knot polynomials, where larger twist knots are built from the trefoil and figure-eight by appending twisting

Table 2. *Analysis summary for the four tested twist knots, including crossing number, algebraic boundary crossing, and root movement direction.*

Twist Knot	Crossing Number, n	Boundary Crossing,	Direction
Trefoil	3	None	None
Figure-Eight	4	$0.3 < q_c < 0.5$	Outward
Three-Twist	5	$q_c = 0.5$	Outward
Stevedore	6	$0.1 < q_c < 0.3$ and $0.7 < q_c < 0.9$	Inward

factors, is consistent with the progression of thresholds observed here. The three-twist is built from the trefoil, and the stevedore is from the figure-eight, so the observed thresholds follow the same incremental pattern. Noguchi's (9) connection between Alexander polynomial roots and dynamical entropy is not invoked in the present analysis. Only the positions of the roots relative to the unit circle and imaginary axis are tracked, and they are treated purely as algebraic coordinates.

This study is restricted to four small twist knots with q sampled at 0.2 intervals. Finer sampling could refine the identified threshold intervals or reveal additional transitions within them. The conjecture remains untested for knots with more than six crossings, where the pattern may not hold. Modulus calculations replace visual identification of crossing points, but some subjectivity remains because a crossing can be located only to within the sampling interval. Finally, the explanations offered are intuitive, whereas a rigorous proof would require a general expression for the deformed Alexander polynomial of the entire twist knot family.

CONCLUSION

A directional bias parameter q is introduced into random walks on knot diagrams in order to analyze how it alters the root trajectories of the deformed Alexander polynomial for the four smallest twist knots. The unit circle $|z| = 1$ frequently serves as a reference boundary for the algebraic root trajectories. The results show a pattern across the four case studies that, as crossing number increases, the number and type of reference boundary crossings may increase. The trefoil shows no transition, the figure-eight shows one outward transition between $q = 0.3$ and $q = 0.5$, and the three-twist crosses exactly at $q = 0.5$, and the stevedore shows two inward transitions at $q \in \{0.1, 0.3\}$ and $q \in \{0.7, 0.9\}$, along with imaginary-axis crossings and a change in the number of real roots at $q \in \{0.1, 0.3\}$.

These findings point to a conjecture that the algebraic root-trajectory behavior of the twist knot family may depend systematically on crossing number, with more complex knots exhibiting more complicated algebraic movement on the complex plane, but remains untested beyond four knots. Future work may determine whether this hypothesis holds beyond the four examples studied here. The main limitations are the small sample size of four knots and the discrete q-sampling at 0.2 intervals for only five values. Future work could extend the analysis to higher-crossing twist knots, refine the threshold intervals with finer sampling, and work toward a formal proof using a general expression for the deformed Alexander polynomial of the family. The connection between root behavior and Markov chain absorption probabilities could also be formally investigated. Overall, this framework bridges algebraic knot invariants and algebraic systems with probabilities, providing a method for tracking algebraic root trajectories in these types of topological systems.

ACKNOWLEDGEMENTS

This paper was supported by the Lumiere Education research program. Thanks are given to Professor Sourav Sarkar at the University of Cambridge, and Kiana Manian and Nicholas Burns at Lumiere Education for their guidance and help in making it possible.

CONFLICT OF INTEREST

The author declares that there are no conflicts of interest related to this work.

REFERENCES

1. Adams CC. The knot book: An elementary introduction to the mathematical theory of Knots. Providence, R.I: American Mathematical Society;

2010. ISBN: 978-0821836781.
2. Bai S. Alexander polynomials of ribbon knots and Virtual Knots [Internet]. 2026. Available from: https://arxiv.org/abs/2103.07128v3 (accessed on 2026-08-12)
3. Belousov Y. Explicit formulas for the Alexander polynomial of pretzel knots [Internet]. 2026. Available from: https://arxiv.org/abs/2502.10370v4 (accessed on 2026-08-17)
4. Weisstein EW. Alexander Polynomial [Internet]. Available from: https://mathworld.wolfram.com/AlexanderPolynomial.html (accessed on 2026-08-29)
5. Hoel PG, Port SC, Stone CJ. Introduction to Probability Theory. Boston: Houghton Mifflin; 1996. ISBN: 978-0395046364.
6. Norris JR. Markov chains. New York, US: Cambridge University Press; 2012. ISBN: 978-0-521-63396-3.
7. Lin X-S, Wang Z. Random Walk on Knot Diagrams, Colored Jones Polynomial and Ihara-Selberg Zeta Function [Internet]. 1998. Available from: https://arxiv.org/pdf/math/9812039v1 (accessed on 2026-07-03)
8. Nawata S, Ramadevi P, Zodinmawia, Sun X. Super-A-polynomials for Twist Knots [Internet]. 2013. Available from: https://arxiv.org/pdf/1209.1409v4 (accessed on 2026-07-09)
9. Noguchi A. Zeros of the Alexander polynomial of knot [Internet]. 2005. Available from: https://arxiv.org/pdf/math/0503252v2 (accessed on 2026-07-09)
10. Hamilton JD. Time Series Analysis. Princeton, New Jersey: Princeton University Press; 2006. ISBN: 978-0-691-04289-3.
11. Ogata K. Modern Control Engineering. Prentice Hall; 2022. ISBN: 978-0136156734.

APPENDIX

The following Mathematica code was used to construct the deformed Alexander matrices, compute their determinants, extract their roots and other algebraic information, and generate the root-trajectory plots.

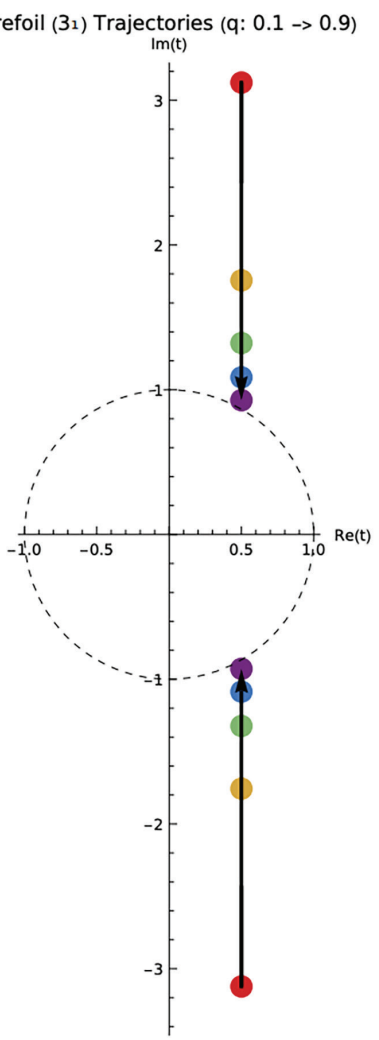
Appendix A: Trefoil Knot

```
In[36]:= (* ========================================================== *)
         (* TREFOIL KNOT (3_1)                 *)
         (* Polynomial: Δ(t;q) = q t^2 - q t + 1                       *)
         (* Arrows: Black, point from q=0.1 (red) to q=0.9 (purple)  *)
         (* ========================================================== *)

In[37]:= ClearAll[trefoil, qVals, roots, traj1, traj2, pts1, pts2, colors];

In[38]:= (* 1. Polynomial *)
         trefoil[q_, t_] := q t^2 - q t + 1;

In[39]:= (* 2. q values *)
         qVals = {0.1, 0.3, 0.5, 0.7, 0.9};

In[40]:= (* 3. Compute roots, sort by imaginary part *)
         roots = Table[{q, SortBy[t /. NSolve[trefoil[q, t] == 0, t], Im]}, {q, qVals}];

In[41]:= (* 4. Extract lower and upper branches *)
         traj1 = roots[[All, 2, 1]];
         traj2 = roots[[All, 2, 2]];

In[43]:= (* 5. Convert to coordinate pairs *)
         pts1 = Table[{Re[traj1[[i]]], Im[traj1[[i]]]}, {i, 5}];
         pts2 = Table[{Re[traj2[[i]]], Im[traj2[[i]]]}, {i, 5}];

In[45]:= (* 6. Color gradient: Red → Purple *)
         colors = Table[ColorData["Rainbow"][1 - (i - 1)/4], {i, 5}];

In[46]:= (* 7. Point plots with colored dots *)
         scatter1 = ListPlot[
          Table[Style[pts1[[i]], colors[[i]], PointSize[0.075]], {i, 5}],
          PlotRange → All, AspectRatio → Automatic
         ];
         scatter2 = ListPlot[
          Table[Style[pts2[[i]], colors[[i]], PointSize[0.075]], {i, 5}],
          PlotRange → All, AspectRatio → Automatic
         ];

In[48]:= (* 8. Arrows: Black *)
         arrow1 = Graphics[{Thick, Black, Arrowheads[0.08], Arrow[pts1]}];
         arrow2 = Graphics[{Thick, Black, Arrowheads[0.08], Arrow[pts2]}];

In[50]:= (* 9. Unit circle *)
         circle = Graphics[{Dashed, Circle[{0, 0}, 1]}];
```

In[51]:=
```
(* 10. Combine all layers *)
finalPlot = Show[
 scatter1, scatter2, arrow1, arrow2, circle,
 Axes → True,
 AxesLabel → {"Re(t)", "Im(t)"},
 AxesOrigin → {0, 0},
 PlotLabel → "Trefoil (3₁) Trajectories (q: 0.1 -> 0.9)",
 PlotRange → All,
 AspectRatio → Automatic
];
```

In[52]:=
```
(* Display the plot *)
finalPlot
```

Out[52]=



In[53]:=
```
(* 11. Moduli *)
moduli1 = Abs[traj1]; moduli2 = Abs[traj2];
Print["Moduli (lower): ", moduli1];
Print["Moduli (upper): ", moduli2];
```

```
Moduli (lower): {3.16228, 1.82574, 1.41421, 1.19523, 1.05409}
Moduli (upper): {3.16228, 1.82574, 1.41421, 1.19523, 1.05409}
```

In[56]:=
```
(* 12. Unit-circle crossing *)
Print["Lower |z| > 1? ", Map[# > 1 &, moduli1]];
Print["Upper |z| > 1? ", Map[# > 1 &, moduli2]];
```

```
Lower |z| > 1? {True, True, True, True, True}
Upper |z| > 1? {True, True, True, True, True}
```

In[58]:=
```
(* 13. Imaginary-axis crossing *)
reals1 = Re[traj1]; reals2 = Re[traj2];
Print["Real parts (lower): ", reals1];
Print["Real parts (upper): ", reals2];
Print["Lower Re < 0? ", Map[# < 0 &, reals1]];
Print["Upper Re < 0? ", Map[# < 0 &, reals2]];
Print["Lower Re == 0? ", Map[# == 0 &, reals1]];
Print["Upper Re == 0? ", Map[# == 0 &, reals2]];
```

```
Real parts (lower): {0.5, 0.5, 0.5, 0.5, 0.5}
Real parts (upper): {0.5, 0.5, 0.5, 0.5, 0.5}
Lower Re < 0? {False, False, False, False, False}
Upper Re < 0? {False, False, False, False, False}
Lower Re == 0? {False, False, False, False, False}
Upper Re == 0? {False, False, False, False, False}
```

In[65]:=
```
(* 14. Displacement rates and speeds *)
rates1 = Table[(traj1[[i + 1]] - traj1[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates2 = Table[(traj2[[i + 1]] - traj2[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
Print["Rates (lower): ", rates1];
Print["Rates (upper): ", rates2];
Print["Speeds (lower): ", Abs[rates1]];
Print["Speeds (upper): ", Abs[rates2]];
```

```
Rates (lower): {0. + 6.83278 i, 0. + 2.16533 i, 0. + 1.18628 i, 0. + 0.788298 i}
Rates (upper): {0. - 6.83278 i, 0. - 2.16533 i, 0. - 1.18628 i, 0. - 0.788298 i}
Speeds (lower): {6.83278, 2.16533, 1.18628, 0.788298}
Speeds (upper): {6.83278, 2.16533, 1.18628, 0.788298}
```

Appendix B: Figure-Eight Knot

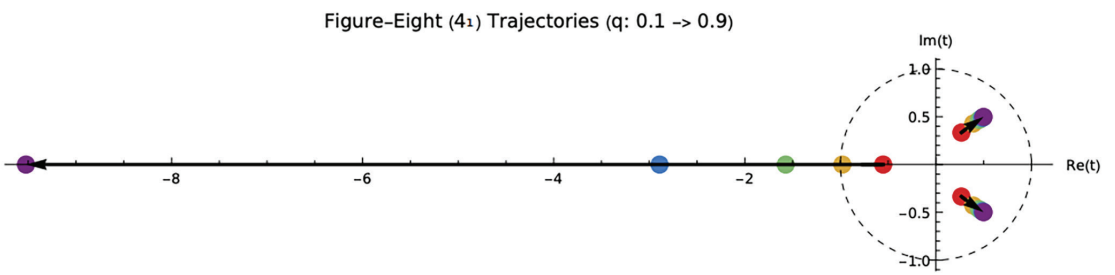
```
In[62]:= (* ========================================================== *)
         (* FIGURE-EIGHT KNOT (4_1)                                    *)
         (* Polynomial: Δ(t;q) = (1-q²)t³ + 2q²t² - (q²+q)t + q        *)
         (* Arrows: Black, point from q=0.1 (red) to q=0.9 (purple)    *)
         (* ========================================================== *)

In[63]:= ClearAll[figure8, qVals, roots, traj1, traj2, traj3, pts1, pts2, pts3, colors];

In[64]:= (* 1. Polynomial *)
         figure8[q_, t_] := (1 - q^2) t^3 + 2 q^2 t^2 - (q^2 + q) t + q;

In[65]:= (* 2. q values *)
         qVals = {0.1, 0.3, 0.5, 0.7, 0.9};

In[66]:= (* 3. Compute roots, sort by imaginary part *)
         roots = Table[{q, SortBy[t /. NSolve[figure8[q, t] == 0, t], Im]}, {q, qVals}];

In[67]:= (* 4. Extract three branches (lowest, middle, highest imaginary) *)
         traj1 = roots[[All, 2, 1]];
         traj2 = roots[[All, 2, 2]];
         traj3 = roots[[All, 2, 3]];

In[70]:= (* 5. Convert to coordinate pairs *)
         pts1 = Table[{Re[traj1[[i]]], Im[traj1[[i]]]}, {i, 5}];
         pts2 = Table[{Re[traj2[[i]]], Im[traj2[[i]]]}, {i, 5}];
         pts3 = Table[{Re[traj3[[i]]], Im[traj3[[i]]]}, {i, 5}];

In[73]:= (* 6. Color gradient: Red → Purple *)
         colors = Table[ColorData["Rainbow"][1 - (i - 1)/4], {i, 5}];

In[74]:= (* 7. Point plots with colored dots *)
         scatter1 = ListPlot[
          Table[Style[pts1[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
          PlotRange → All, AspectRatio → Automatic
         ];
         scatter2 = ListPlot[
          Table[Style[pts2[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
          PlotRange → All, AspectRatio → Automatic
         ];
         scatter3 = ListPlot[
          Table[Style[pts3[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
          PlotRange → All, AspectRatio → Automatic
         ];

In[77]:= (* 8. Arrows: Black *)
         arrow1 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts1]}];
         arrow2 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts2]}];
         arrow3 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts3]}];
```

In[80]:=
```
(* 9. Unit circle *)
circle = Graphics[{Dashed, Circle[{0, 0}, 1]}];
```

In[81]:=
```
(* 10. Combine all layers *)
finalPlot = Show[
 scatter1, scatter2, scatter3, arrow1, arrow2, arrow3, circle,
 Axes → True,
 AxesLabel → {"Re(t)", "Im(t)"},
 AxesOrigin → {0, 0},
 PlotLabel → "Figure-Eight (4₁) Trajectories (q: 0.1 -> 0.9)",
 PlotRange → All,
 AspectRatio → Automatic
];
```

In[82]:=
```
(* Display the plot *)
finalPlot
```

Out[82]=



In[83]:=
```
(* 11. Moduli *)
moduli1 = Abs[traj1]; moduli2 = Abs[traj2]; moduli3 = Abs[traj3];
Print["Moduli (lowest): ", moduli1];
Print["Moduli (middle): ", moduli2];
Print["Moduli (highest): ", moduli3];
```

```
Moduli (lowest): {0.427623, 0.580221, 0.651152, 0.688871, 0.705253}
Moduli (middle): {0.552387, 0.979247, 1.57233, 2.89236, 9.52357}
Moduli (highest): {0.427623, 0.580221, 0.651152, 0.688871, 0.705253}
```

In[87]:=
```
(* 12. Unit-circle crossing *)
Print["Lowest |z| > 1? ", Map[# > 1 &, moduli1]];
Print["Middle |z| > 1? ", Map[# > 1 &, moduli2]];
Print["Highest |z| > 1? ", Map[# > 1 &, moduli3]];
```

```
Lowest |z| > 1? {False, False, False, False, False}
Middle |z| > 1? {False, False, True, True, True}
Highest |z| > 1? {False, False, False, False, False}
```

In[90]:=
```
(* 13. Imaginary-axis crossing *)
reals1 = Re[traj1]; reals2 = Re[traj2]; reals3 = Re[traj3];
Print["Real parts (lowest): ", reals1];
Print["Real parts (middle): ", reals2];
Print["Real parts (highest): ", reals3];
Print["Lowest Re < 0? ", Map[# < 0 &, reals1]];
Print["Middle Re < 0? ", Map[# < 0 &, reals2]];
Print["Highest Re < 0? ", Map[# < 0 &, reals3]];
Print["Lowest Re == 0? ", Map[# == 0 &, reals1]];
Print["Middle Re == 0? ", Map[# == 0 &, reals2]];
Print["Highest Re == 0? ", Map[# == 0 &, reals3]];
```

```
Real parts (lowest): {0.266093, 0.390723, 0.452831, 0.485395, 0.498625}
Real parts (middle): {-0.552387, -0.979247, -1.57233, -2.89236, -9.52357}
Real parts (highest): {0.266093, 0.390723, 0.452831, 0.485395, 0.498625}
Lowest Re < 0? {False, False, False, False, False}
Middle Re < 0? {True, True, True, True, True}
Highest Re < 0? {False, False, False, False, False}
Lowest Re == 0? {False, False, False, False, False}
Middle Re == 0? {False, False, False, False, False}
Highest Re == 0? {False, False, False, False, False}
```

In[100]:=
```
(* 14. Displacement rates and speeds *)
rates1 = Table[(traj1[[i + 1]] - traj1[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates2 = Table[(traj2[[i + 1]] - traj2[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates3 = Table[(traj3[[i + 1]] - traj3[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
Print["Rates (lowest): ", rates1];
Print["Rates (middle): ", rates2];
Print["Rates (highest): ", rates3];
Print["Speeds (lowest): ", Abs[rates1]];
Print["Speeds (middle): ", Abs[rates2]];
Print["Speeds (highest): ", Abs[rates3]];
```

```
Rates (lowest):
 {0.62315 - 0.470981 i, 0.310543 - 0.194851 i, 0.162821 - 0.104476 i, 0.0661489 - 0.049717 i}
Rates (middle): {-2.1343, -2.96541, -6.60015, -33.156}
Rates (highest):
 {0.62315 + 0.470981 i, 0.310543 + 0.194851 i, 0.162821 + 0.104476 i, 0.0661489 + 0.049717 i}
Speeds (lowest): {0.781114, 0.366611, 0.193458, 0.0827494}
Speeds (middle): {2.1343, 2.96541, 6.60015, 33.156}
Speeds (highest): {0.781114, 0.366611, 0.193458, 0.0827494}
```

Appendix C: Three-Twist Knot

```
In[1]:= (* ========================================================== *)
        (* THREE-TWIST KNOT (5_2)           *)
        (* Polynomial: Δ(t;q) = t^4 + qt - qt^3 + q^3 t (1-t)^3     *)
        (* Arrows: Black, point from q=0.1 (red) to q=0.9 (purple)  *)
        (* ========================================================== *)

In[2]:= ClearAll[threeTwist, qVals, roots, traj1, traj2, traj3, traj4,
            pts1, pts2, pts3, pts4, colors];

In[3]:= (* 1. Polynomial *)
        threeTwist[q_, t_] := t^4 + q t - q t^3 + q^3 t (1 - t)^3;

In[4]:= (* 2. q values *)
        qVals = {0.1, 0.3, 0.5, 0.7, 0.9};

In[5]:= (* 3. Compute roots, sort by imaginary part *)
        roots = Table[{q, SortBy[t /. NSolve[threeTwist[q, t] == 0, t], Im]}, {q, qVals}];

In[6]:= (* 4. Extract four branches (lowest to highest imaginary) *)
        traj1 = roots[[All, 2, 1]];
        traj2 = roots[[All, 2, 2]];
        traj3 = roots[[All, 2, 3]];
        traj4 = roots[[All, 2, 4]];

In[10]:= (* 5. Convert to coordinate pairs *)
        pts1 = Table[{Re[traj1[[i]]], Im[traj1[[i]]]}, {i, 5}];
        pts2 = Table[{Re[traj2[[i]]], Im[traj2[[i]]]}, {i, 5}];
        pts3 = Table[{Re[traj3[[i]]], Im[traj3[[i]]]}, {i, 5}];
        pts4 = Table[{Re[traj4[[i]]], Im[traj4[[i]]]}, {i, 5}];

In[14]:= (* 6. Color gradient: Red → Purple *)
        colors = Table[ColorData["Rainbow"][1 - (i - 1) / 4], {i, 5}];
```

In[15]:=
```
(* 7. Point plots with colored dots *)
scatter1 = ListPlot[
 Table[Style[pts1[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter2 = ListPlot[
 Table[Style[pts2[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter3 = ListPlot[
 Table[Style[pts3[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter4 = ListPlot[
 Table[Style[pts4[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
```

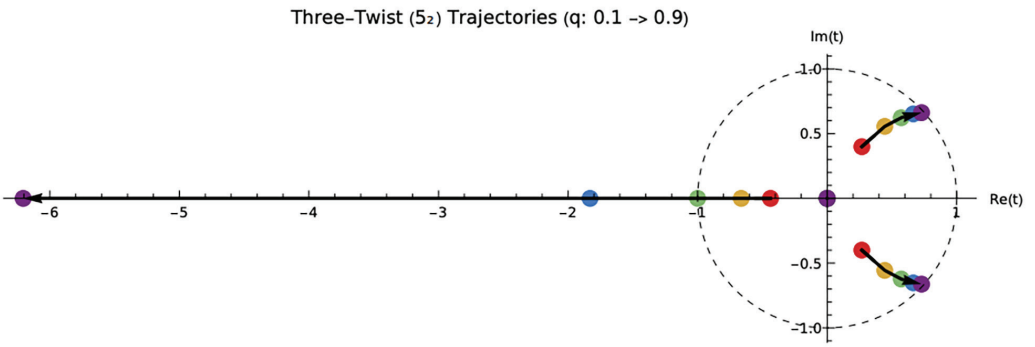
In[19]:=
```
(* 8. Arrows: Black, reversed to point from q=0.1 to q=0.9 *)
arrow1 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts1]}];
arrow2 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts2]}];
arrow3 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts3]}];
arrow4 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts4]}];
```

In[23]:=
```
(* 9. Unit circle *)
circle = Graphics[{Dashed, Circle[{0, 0}, 1]}];
```

In[24]:=
```
(* 10. Combine all layers *)
finalPlot = Show[
 scatter1, scatter2, scatter3, scatter4,
 arrow1, arrow2, arrow3, arrow4, circle,
 Axes → True,
 AxesLabel → {"Re(t)", "Im(t)"},
 AxesOrigin → {0, 0},
 PlotLabel → "Three-Twist (5₂) Trajectories (q: 0.1 -> 0.9)",
 PlotRange → All,
 AspectRatio → Automatic
];
```

In[25]:=
```
(* Display the plot *)
finalPlot
```

Out[25]=



In[26]:=
```
(* 11. Moduli *)
moduli1 = Abs[traj1]; moduli2 = Abs[traj2];
moduli3 = Abs[traj3]; moduli4 = Abs[traj4];
Print["Moduli (lowest): ", moduli1];
Print["Moduli (lower-middle): ", moduli2];
Print["Moduli (upper-middle): ", moduli3];
Print["Moduli (highest): ", moduli4];
```

```
Moduli (lowest): {0.480641, 0.711655, 0.845154, 0.931311, 0.9842}
Moduli (lower-middle): {0.437637, 0.663584, 1., 1.83033, 6.20562}
Moduli (upper-middle): {0., 0., 0., 0., 0.}
Moduli (highest): {0.480641, 0.711655, 0.845154, 0.931311, 0.9842}
```

In[32]:=
```
(* 12. Unit-circle crossing *)
Print["Lowest |z| > 1? ", Map[# > 1 &, moduli1]];
Print["Lower-middle |z| > 1? ", Map[# > 1 &, moduli2]];
Print["Upper-middle |z| > 1? ", Map[# > 1 &, moduli3]];
Print["Highest |z| > 1? ", Map[# > 1 &, moduli4]];
```

```
Lowest |z| > 1? {False, False, False, False, False}
Lower-middle |z| > 1? {False, False, False, True, True}
Upper-middle |z| > 1? {False, False, False, False, False}
Highest |z| > 1? {False, False, False, False, False}
```

In[36]:=
```
(* 13. Imaginary-axis crossing *)
reals1 = Re[traj1]; reals2 = Re[traj2];
reals3 = Re[traj3]; reals4 = Re[traj4];
Print["Real parts (lowest): ", reals1];
Print["Real parts (lower-middle): ", reals2];
Print["Real parts (upper-middle): ", reals3];
Print["Real parts (highest): ", reals4];
Print["Lowest Re < 0? ", Map[# < 0 &, reals1]];
Print["Lower-middle Re < 0? ", Map[# < 0 &, reals2]];
Print["Upper-middle Re < 0? ", Map[# < 0 &, reals3]];
Print["Highest Re < 0? ", Map[# < 0 &, reals4]];
Print["Lowest Re == 0? ", Map[# == 0 &, reals1]];
Print["Lower-middle Re == 0? ", Map[# == 0 &, reals2]];
Print["Upper-middle Re == 0? ", Map[# == 0 &, reals3]];
Print["Highest Re == 0? ", Map[# == 0 &, reals4]];
```

```
Real parts (lowest): {0.267367, 0.44433, 0.571429, 0.664785, 0.728272}
Real parts (lower-middle): {-0.437637, -0.663584, -1., -1.83033, -6.20562}
Real parts (upper-middle): {0., 0., 0., 0., 0.}
Real parts (highest): {0.267367, 0.44433, 0.571429, 0.664785, 0.728272}
Lowest Re < 0? {False, False, False, False, False}
Lower-middle Re < 0? {True, True, True, True, True}
Upper-middle Re < 0? {False, False, False, False, False}
Highest Re < 0? {False, False, False, False, False}
Lowest Re == 0? {False, False, False, False, False}
Lower-middle Re == 0? {False, False, False, False, False}
Upper-middle Re == 0? {True, True, True, True, True}
Highest Re == 0? {False, False, False, False, False}
```

In[50]:=
```
(* 14. Displacement rates and speeds *)
rates1 = Table[(traj1[[i + 1]] - traj1[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates2 = Table[(traj2[[i + 1]] - traj2[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates3 = Table[(traj3[[i + 1]] - traj3[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates4 = Table[(traj4[[i + 1]] - traj4[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
Print["Rates (lowest): ", rates1];
Print["Rates (lower-middle): ", rates2];
Print["Rates (upper-middle): ", rates3];
Print["Rates (highest): ", rates4];
Print["Speeds (lowest): ", Abs[rates1]];
Print["Speeds (lower-middle): ", Abs[rates2]];
Print["Speeds (upper-middle): ", Abs[rates3]];
Print["Speeds (highest): ", Abs[rates4]];
```

```
Rates (lowest):
 {0.884818 - 0.782428 i, 0.635491 - 0.334005 i, 0.46678 - 0.147642 i, 0.317437 - 0.0489534 i}
Rates (lower-middle): {-1.12974, -1.68208, -4.15165, -21.8765}
Rates (upper-middle): {0., 0., 0., 0.}
Rates (highest):
 {0.884818 + 0.782428 i, 0.635491 + 0.334005 i, 0.46678 + 0.147642 i, 0.317437 + 0.0489534 i}
Speeds (lowest): {1.18114, 0.717919, 0.489573, 0.32119}
Speeds (lower-middle): {1.12974, 1.68208, 4.15165, 21.8765}
Speeds (upper-middle): {0., 0., 0., 0.}
Speeds (highest): {1.18114, 0.717919, 0.489573, 0.32119}
```

APPENDIX D: STEVEDORE KNOT

In[436]:=
```
(* ========================================================= *)
(* STEVEDORE KNOT (6_1)                *)
(* Polynomial: Δ(t;q) = q^4 t^5 - 4q^4 t^4 - 4q^3 t^5       *)
(*              + 6q^4 t^3 + 16q^3 t^4 + 4q^2 t^5           *)
(*              - 4q^4 t^2 - 25q^3 t^3 - 20q^2 t^4          *)
(*              + q^4 t + 19q^3 t^2 + 36q^2 t^3 + 8q t^4    *)
(*              - 7q^3 t - 31q^2 t^2 - 20q t^3              *)
(*              + q^3 + 14q^2 t + 20q t^2 + 3t^3            *)
(*              - 3q^2 - 10q t - 4t^2 + 2q + 2t             *)
(* Arrows: Black, point from q=0.1 (red) to q=0.9 (purple)  *)
(* ========================================================= *)
```

In[437]:=
```
ClearAll[stevedore, qVals, roots,
    traj1, traj2, traj3, traj4, traj5,
    pts1, pts2, pts3, pts4, pts5, colors];
```

In[438]:=
```
(* 1. Polynomial *)
stevedore[q_, t_] :=
 q^4 t^5 - 4 q^4 t^4 - 4 q^3 t^5 + 6 q^4 t^3 +
   16 q^3 t^4 + 4 q^2 t^5 - 4 q^4 t^2 - 25 q^3 t^3 - 20 q^2 t^4 + q^4 t +
   19 q^3 t^2 + 36 q^2 t^3 + 8 q t^4 - 7 q^3 t - 31 q^2 t^2 - 20 q t^3 +
   q^3 + 14 q^2 t + 20 q t^2 + 3 t^3 - 3 q^2 - 10 q t - 4 t^2 + 2 q + 2 t;
```

In[439]:=
```
(* 2. q values *)
qVals = {0.1, 0.3, 0.5, 0.7, 0.9};
```

In[440]:=
```
(* 3. Compute roots, sort by imaginary part *)
roots = Table[{q, SortBy[t /. NSolve[stevedore[q, t] == 0, t], Im]}, {q, qVals}];
```

In[441]:=
```
(* 4. Extract five branches (lowest to highest imaginary) *)
traj1 = roots[[All, 2, 1]];
traj2 = roots[[All, 2, 2]];
traj3 = roots[[All, 2, 3]];
traj4 = roots[[All, 2, 4]];
traj5 = roots[[All, 2, 5]];
```

In[446]:=
```
(* 5. Convert to coordinate pairs *)
pts1 = Table[{Re[traj1[[i]]], Im[traj1[[i]]]}, {i, 5}];
pts2 = Table[{Re[traj2[[i]]], Im[traj2[[i]]]}, {i, 5}];
pts3 = Table[{Re[traj3[[i]]], Im[traj3[[i]]]}, {i, 5}];
pts4 = Table[{Re[traj4[[i]]], Im[traj4[[i]]]}, {i, 5}];
pts5 = Table[{Re[traj5[[i]]], Im[traj5[[i]]]}, {i, 5}];
```

In[451]:=
```
(* 6. Color gradient: Red (q=0.1) → Purple (q=0.9) *)
colors = Table[ColorData["Rainbow"][1 - (i - 1)/4], {i, 5}];
```

In[452]:=
```
(* 7. Point plots with colored dots *)
scatter1 = ListPlot[
 Table[Style[pts1[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter2 = ListPlot[
 Table[Style[pts2[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter3 = ListPlot[
 Table[Style[pts3[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter4 = ListPlot[
 Table[Style[pts4[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
scatter5 = ListPlot[
 Table[Style[pts5[[i]], colors[[i]], PointSize[0.0175]], {i, 5}],
 PlotRange → All, AspectRatio → Automatic
];
```

In[457]:=
```
(* 8. Arrows: Black, pointing from q=0.1 to q=0.9 *)
arrow1 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts1]}];
arrow2 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts2]}];
arrow3 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts3]}];
arrow4 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts4]}];
arrow5 = Graphics[{Thick, Black, Arrowheads[0.02], Arrow[pts5]}];
```

In[462]:=
```
(* 9. Unit circle *)
circle = Graphics[{Dashed, Circle[{0, 0}, 1]}];
```

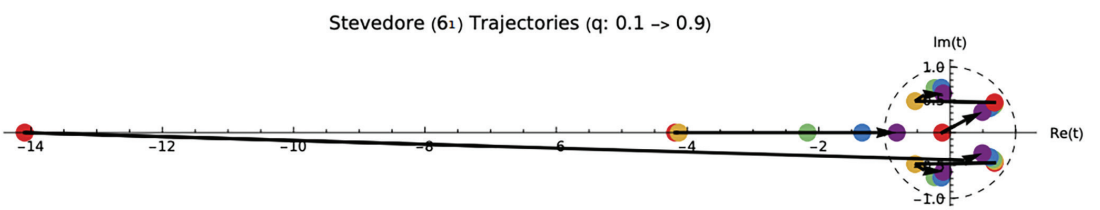
In[463]:=

```
(* 10. Combine all layers *)
finalPlot = Show[
 scatter1, scatter2, scatter3, scatter4, scatter5,
 arrow1, arrow2, arrow3, arrow4, arrow5, circle,
 Axes → True,
 AxesLabel → {"Re(t)", "Im(t)"},
 AxesOrigin → {0, 0},
 PlotLabel → "Stevedore (6₁) Trajectories (q: 0.1 -> 0.9)",
 PlotRange → All,
 AspectRatio → Automatic
];
```

In[464]:=

```
(* Display the plot *)
finalPlot
```

Out[464]=



In[465]:=

```
(* 11. Moduli *)
moduli1 = Abs[traj1]; moduli2 = Abs[traj2];
moduli3 = Abs[traj3]; moduli4 = Abs[traj4]; moduli5 = Abs[traj5];
Print["Moduli (traj 1): ", moduli1];
Print["Moduli (traj 2): ", moduli2];
Print["Moduli (traj 3): ", moduli3];
Print["Moduli (traj 4): ", moduli4];
Print["Moduli (traj 5): ", moduli5];
```

```
Moduli (traj 1): {0.817945, 0.716527, 0.718799, 0.697493, 0.603483}
Moduli (traj 2): {14.0979, 0.804233, 0.770676, 0.711019, 0.583804}
Moduli (traj 3): {4.1881, 4.13332, 2.17245, 1.34041, 0.81377}
Moduli (traj 4): {0.119914, 0.804233, 0.770676, 0.711019, 0.583804}
Moduli (traj 5): {0.817945, 0.716527, 0.718799, 0.697493, 0.603483}
```

In[472]:=

```
(* 12. Unit-circle crossing *)
Print["Traj1 |z| > 1? ", Map[# > 1 &, moduli1]];
Print["Traj2 |z| > 1? ", Map[# > 1 &, moduli2]];
Print["Traj3 |z| > 1? ", Map[# > 1 &, moduli3]];
Print["Traj4 |z| > 1? ", Map[# > 1 &, moduli4]];
Print["Traj5 |z| > 1? ", Map[# > 1 &, moduli5]];
```

```
Traj1 |z| > 1? {False, False, False, False, False}
Traj2 |z| > 1? {True, False, False, False, False}
Traj3 |z| > 1? {True, True, True, True, False}
Traj4 |z| > 1? {False, False, False, False, False}
Traj5 |z| > 1? {False, False, False, False, False}
```

In[477]:=

```
(* 13. Imaginary-axis crossing *)
reals1 = Re[traj1]; reals2 = Re[traj2];
reals3 = Re[traj3]; reals4 = Re[traj4]; reals5 = Re[traj5];
Print["Real parts (traj 1): ", reals1];
Print["Real parts (traj 2): ", reals2];
Print["Real parts (traj 3): ", reals3];
Print["Real parts (traj 4): ", reals4];
Print["Real parts (traj 5): ", reals5];
Print["Traj1 Re < 0? ", Map[# < 0 &, reals1]];
Print["Traj2 Re < 0? ", Map[# < 0 &, reals2]];
Print["Traj3 Re < 0? ", Map[# < 0 &, reals3]];
Print["Traj4 Re < 0? ", Map[# < 0 &, reals4]];
Print["Traj5 Re < 0? ", Map[# < 0 &, reals5]];
Print["Traj1 Re == 0? ", Map[# == 0 &, reals1]];
Print["Traj2 Re == 0? ", Map[# == 0 &, reals2]];
Print["Traj3 Re == 0? ", Map[# == 0 &, reals3]];
Print["Traj4 Re == 0? ", Map[# == 0 &, reals4]];
Print["Traj5 Re == 0? ", Map[# == 0 &, reals5]];
```

```
Real parts (traj 1): {0.676627, -0.533137, -0.236605, -0.133193, -0.105074}
Real parts (traj 2): {-14.0979, 0.67823, 0.656163, 0.605596, 0.491757}
Real parts (traj 3): {-4.1881, -4.13332, -2.17245, -1.34041, -0.81377}
Real parts (traj 4): {-0.119914, 0.67823, 0.656163, 0.605596, 0.491757}
Real parts (traj 5): {0.676627, -0.533137, -0.236605, -0.133193, -0.105074}
Traj1 Re < 0? {False, True, True, True, True}
Traj2 Re < 0? {True, False, False, False, False}
Traj3 Re < 0? {True, True, True, True, True}
Traj4 Re < 0? {True, False, False, False, False}
Traj5 Re < 0? {False, True, True, True, True}
Traj1 Re == 0? {False, False, False, False, False}
Traj2 Re == 0? {False, False, False, False, False}
Traj3 Re == 0? {False, False, False, False, False}
Traj4 Re == 0? {False, False, False, False, False}
Traj5 Re == 0? {False, False, False, False, False}
```

In[494]:=

```
(* 14. Displacement rates and speeds *)
rates1 = Table[(traj1[[i + 1]] - traj1[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates2 = Table[(traj2[[i + 1]] - traj2[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates3 = Table[(traj3[[i + 1]] - traj3[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates4 = Table[(traj4[[i + 1]] - traj4[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
rates5 = Table[(traj5[[i + 1]] - traj5[[i]]) / (qVals[[i + 1]] - qVals[[i]]), {i, 4}];
Print["Rates (traj 1): ", rates1];
Print["Rates (traj 2): ", rates2];
Print["Rates (traj 3): ", rates3];
Print["Rates (traj 4): ", rates4];
Print["Rates (traj 5): ", rates5];
Print["Speeds (traj 1): ", Abs[rates1]];
Print["Speeds (traj 2): ", Abs[rates2]];
Print["Speeds (traj 3): ", Abs[rates3]];
Print["Speeds (traj 4): ", Abs[rates4]];
Print["Speeds (traj 5): ", Abs[rates5]];
```

```
Rates (traj 1):
 {-6.04882 - 0.0957332 i, 1.48266 - 1.0001 i, 0.517056 - 0.0295771 i, 0.140599 + 0.451962 i}
Rates (traj 2):
 {73.8805 - 2.16099 i, -0.110331 + 0.139909 i, -0.252835 + 0.15828 i, -0.569199 + 0.289573 i}
Rates (traj 3): {0.273894, 9.80435, 4.1602, 2.6332}
Rates (traj 4):
 {3.99072 + 2.16099 i, -0.110331 - 0.139909 i, -0.252835 - 0.15828 i, -0.569199 - 0.289573 i}
Rates (traj 5):
 {-6.04882 + 0.0957332 i, 1.48266 + 1.0001 i, 0.517056 + 0.0295771 i, 0.140599 - 0.451962 i}
Speeds (traj 1): {6.04958, 1.78843, 0.517901, 0.473326}
Speeds (traj 2): {73.9121, 0.178178, 0.298292, 0.638624}
Speeds (traj 3): {0.273894, 9.80435, 4.1602, 2.6332}
Speeds (traj 4): {4.53825, 0.178178, 0.298292, 0.638624}
Speeds (traj 5): {6.04958, 1.78843, 0.517901, 0.473326}
```

Appendix E: A unified verification that checks the general polynomial calculation, polynomial at each value, satisfaction of , and roots at each value.

```
(* ========================================================== *)
(* UNIFIED POLYNOMIAL VERIFICATION       *)
(* For knots: 3_1, 4_1, 5_2, 6_1                              *)
(* ========================================================== *)
```

In[29]:=
```
ClearAll["Global`*"];
```

```
(* ---- Check Δ(1;q)=1 ---- *)
checkAtOne[polyFunc_, qVals_] := Table[(polyFunc[qv] /. t → 1) == 1, {qv, qVals}];
```

In[31]:=
```
(* ---- 1. TREFOIL (3_1) ---- *)
trefoilMatrix[q_, t_] := {
 {1, -t, -q(1 - t)},
 {-q(1 - t), 1, -t},
 {-t, -q(1 - t), 1}
};
trefoilPoly[q_, t_] := Expand[Det[trefoilMatrix[q, t][[1 ;; 2, 1 ;; 2]]];
```

In[33]:=
```
(* ---- 2. FIGURE-EIGHT (4_1) ---- *)
figure8Matrix[q_, t_] := {
 {1, -(1 - q)(1 - t), -t, 0},
 {-t, 1, 0, -q(1 - t)},
 {-q(1 - t), 0, 1, -t},
 {0, -t, -(1 - q)(1 - t), 1}
};
figure8Poly[q_, t_] := Expand[Det[figure8Matrix[q, t][[{2, 3, 4}, {1, 2, 4}]]];
```

In[35]:=
```
(* ---- 3. THREE-TWIST (5_2) ---- *)
threeTwistMatrix[q_, t_] := {
 {1, -q(1 - t), 0, 0, -t},
 {-q(1 - t), 1, 0, -t, 0},
 {0, -t, 1, 0, -q(1 - t)},
 {-t, 0, -q(1 - t), 1, 0},
 {0, 0, -t, -q(1 - t), 1}
};
threeTwistPoly[q_, t_] := Expand[Det[threeTwistMatrix[q, t][[2 ;; 5, 1 ;; 4]]];
```

In[37]:=
```
(* ---- 4. STEVEDORE (6_1) ---- *)
stevedoreMatrix[q_, t_] := {
 {1, 0, -t, 0, -(1 - q)(1 - t), 0},
 {0, 1, -(1 - q)(1 - t), 0, -t, 0},
 {0, -(1 - q)(1 - t), 1, 0, 0, -t},
 {0, -t, 0, 1, 0, -(1 - q)(1 - t)},
 {-t, 0, 0, -q(1 - t), 1, 0},
 {-q(1 - t), 0, 0, -t, 0, 1}
};
stevedorePoly[q_, t_] :=
  Expand[Det[stevedoreMatrix[q, t][[{1, 2, 3, 4, 6}, {2, 3, 4, 5, 6}]]];
```

In[39]:=
```
(* ---- Collect all knots ---- *)
knots = {
 {"Trefoil (3_1)", trefoilPoly},
 {"Figure-Eight (4_1)", figure8Poly},
 {"Three-Twist (5_2)", threeTwistPoly},
 {"Stevedore (6_1)", stevedorePoly}
};
```

In[40]:=
```
qVals = {0.1, 0.3, 0.5, 0.7, 0.9};
```

In[41]:=
```
(* ---- Loop and print ---- *)
Do[
 knotName = knots[[i, 1]];
 polyFunc = knots[[i, 2]];

 Print["
", knotName];
 Print["General polynomial: ", polyFunc[q, t]];
 Print["Substituted polynomials:"];
 Do[
  Print["  q=", qv, ": ", polyFunc[qv, t]];
 , {qv, qVals}];

 Print["Check Δ(1;q)=1 for all q: ", checkAtOne[polyFunc, qVals]];
 Print["Roots for each q:"];
 Do[
  roots = t /. NSolve[polyFunc[qv, t] == 0, t];
  Print["  q=", qv, ": ", roots];
 , {qv, qVals}];
 Print["-----------------------------------"];
,
{i, Length[knots]}
];
```

```
Trefoil (3_1)
General polynomial: 1 - q t + q t^2
Substituted polynomials:
  q=0.1: 1 - 0.1 t + 0.1 t^2
  q=0.3: 1 - 0.3 t + 0.3 t^2
```

Check Δ(1;q)=1 for all q: {trefoilPoly[0.1] == 1, trefoilPoly[0.3] == 1,
trefoilPoly[0.5] == 1, trefoilPoly[0.7] == 1, trefoilPoly[0.9] == 1}

Roots for each q:

q=0.1: $\{0.5 - 3.1225\,i,\ 0.5 + 3.1225\,i\}$

q=0.3: $\{0.5 - 1.75594\,i,\ 0.5 + 1.75594\,i\}$

q=0.5: $\{0.5 - 1.32288\,i,\ 0.5 + 1.32288\,i\}$

q=0.7: $\{0.5 - 1.08562\,i,\ 0.5 + 1.08562\,i\}$

q=0.9: $\{0.5 - 0.927961\,i,\ 0.5 + 0.927961\,i\}$

Figure-Eight (4_1)

General polynomial: $q - q\,t - q^2\,t + 2\,q^2\,t^2 + t^3 - q^2\,t^3$

Substituted polynomials:

q=0.1: $0.1 - 0.11\,t + 0.02\,t^2 + 0.99\,t^3$

q=0.3: $0.3 - 0.39\,t + 0.18\,t^2 + 0.91\,t^3$

q=0.5: $0.5 - 0.75\,t + 0.5\,t^2 + 0.75\,t^3$

q=0.7: $0.7 - 1.19\,t + 0.98\,t^2 + 0.51\,t^3$

q=0.9: $0.9 - 1.71\,t + 1.62\,t^2 + 0.19\,t^3$

Check Δ(1;q)=1 for all q: {figure8Poly[0.1] == 1, figure8Poly[0.3] == 1,
figure8Poly[0.5] == 1, figure8Poly[0.7] == 1, figure8Poly[0.9] == 1}

Roots for each q:

q=0.1: $\{-0.552387,\ 0.266093 - 0.334747\,i,\ 0.266093 + 0.334747\,i\}$

q=0.3: $\{-0.979247,\ 0.390723 - 0.428944\,i,\ 0.390723 + 0.428944\,i\}$

q=0.5: $\{-1.57233,\ 0.452831 - 0.467914\,i,\ 0.452831 + 0.467914\,i\}$

q=0.7: $\{-2.89236,\ 0.485395 - 0.488809\,i,\ 0.485395 + 0.488809\,i\}$

q=0.9: $\{-9.52357,\ 0.498625 - 0.498752\,i,\ 0.498625 + 0.498752\,i\}$

Three-Twist (5_2)

General polynomial: $q\,t + q^3\,t - 3\,q^3\,t^2 - q\,t^3 + 3\,q^3\,t^3 + t^4 - q^3\,t^4$

Substituted polynomials:

q=0.1: $0.101\,t - 0.003\,t^2 - 0.097\,t^3 + 0.999\,t^4$

q=0.3: $0.327\,t - 0.081\,t^2 - 0.219\,t^3 + 0.973\,t^4$

q=0.5: $0.625\,t - 0.375\,t^2 - 0.125\,t^3 + 0.875\,t^4$

q=0.7: $1.043\,t - 1.029\,t^2 + 0.329\,t^3 + 0.657\,t^4$

q=0.9: $1.629\,t - 2.187\,t^2 + 1.287\,t^3 + 0.271\,t^4$

```
Check Δ(1;q)=1 for all q: {threeTwistPoly[0.1] == 1, threeTwistPoly[0.3] == 1,
  threeTwistPoly[0.5] == 1, threeTwistPoly[0.7] == 1, threeTwistPoly[0.9] == 1}
Roots for each q:
  q=0.1: {-0.437637, 0., 0.267367 - 0.399413 i, 0.267367 + 0.399413 i}
  q=0.3: {-0.663584, 0., 0.44433 - 0.555899 i, 0.44433 + 0.555899 i}
  q=0.5: {-1., 0., 0.571429 - 0.6227 i, 0.571429 + 0.6227 i}
  q=0.7: {-1.83033, 0., 0.664785 - 0.652228 i, 0.664785 + 0.652228 i}
  q=0.9: {-6.20562, 0., 0.728272 - 0.662019 i, 0.728272 + 0.662019 i}
------------------------------------

Stevedore (6_1)
General polynomial:
 2 q - 3 q^2 + q^3 + 2 t - 10 q t + 14 q^2 t - 7 q^3 t + q^4 t - 4 t^2 + 20 q t^2 - 31 q^2 t^2 + 19 q^3 t^2 - 4 q^4 t^2 + 3 t^3 -
  20 q t^3 + 36 q^2 t^3 - 25 q^3 t^3 + 6 q^4 t^3 + 8 q t^4 - 20 q^2 t^4 + 16 q^3 t^4 - 4 q^4 t^4 + 4 q^2 t^5 - 4 q^3 t^5 + q^4 t^5
Substituted polynomials:
  q=0.1: 0.171 + 1.1331 t - 2.2914 t^2 + 1.3356 t^3 + 0.6156 t^4 + 0.0361 t^5
  q=0.3: 0.357 + 0.0791 t - 0.3094 t^2 - 0.3864 t^3 + 0.9996 t^4 + 0.2601 t^5
  q=0.5: 0.375 - 0.3125 t + 0.375 t^2 - 0.75 t^3 + 0.75 t^4 + 0.5625 t^5
  q=0.7: 0.273 - 0.3009 t + 0.3666 t^2 - 0.4944 t^3 + 0.3276 t^4 + 0.8281 t^5
  q=0.9: 0.099 - 0.1069 t + 0.1166 t^2 - 0.1284 t^3 + 0.0396 t^4 + 0.9801 t^5
Check Δ(1;q)=1 for all q: {stevedorePoly[0.1] == 1, stevedorePoly[0.3] == 1,
  stevedorePoly[0.5] == 1, stevedorePoly[0.7] == 1, stevedorePoly[0.9] == 1}
Roots for each q:
  q=0.1: {-14.0979, -4.1881, -0.119914, 0.676627 - 0.459576 i, 0.676627 + 0.459576 i}
  q=0.3: {-4.13332, -0.533137 - 0.478722 i,
  -0.533137 + 0.478722 i, 0.67823 - 0.432198 i, 0.67823 + 0.432198 i}
  q=0.5: {-2.17245, -0.236605 - 0.678742 i,
  -0.236605 + 0.678742 i, 0.656163 - 0.404217 i, 0.656163 + 0.404217 i}
  q=0.7: {-1.34041, -0.133193 - 0.684657 i,
  -0.133193 + 0.684657 i, 0.605596 - 0.372561 i, 0.605596 + 0.372561 i}
  q=0.9: {-0.81377, -0.105074 - 0.594265 i,
  -0.105074 + 0.594265 i, 0.491757 - 0.314646 i, 0.491757 + 0.314646 i}
------------------------------------
```