How AI Entered Our Collaboration on M_{23} and How We Found the Case

Shaowu Zhang

This is a preliminary personal account; our official disclosure will appear in the final version of the paper.

August 11, 2026

1 A Personal Response

Since we posted our paper realizing M_{23} as a Galois group over $\mathbb{Q}$, many people have asked how we used AI and what led us to consider the triple $(2A, 23A, 23B)$. Here is my short answer, adapted from an earlier Zulip post. I hope this helps clarify some of the questions at this early stage while we continue revising our arXiv paper.

We started with four-tuples. Building on earlier work, we extended the existing computations and developed a fairly complete picture of the possible genus-zero braid orbits. Hafner's Theorem 5.1 led us from these four-tuples to the following five generating triples:

$$(2A, 23A, 23B), \quad (2A, 14A, 14B), \quad (2A, 15A, 15B), \quad (2A, 11A, 11B), \quad (2A, 8A, 8A).$$

Each of these gives rise to a length-two, genus-zero braid orbit for the corresponding four-tuple.

The case $(2A, 8A, 8A)$ was the easiest to compute because its associated quotient curves have genus 0. We could handle it directly with BelyiDB. In this case, however, the Galois action on the resulting covers was transitive, so there was no fixed point of the kind we needed.

The case $(2A, 23A, 23B)$ was different. Its associated curves are non-hyperelliptic curves of genus 4. BelyiDB gave us the numerical data, but not the exact defining equations, so we had to recover those equations ourselves.

Our first attempt used triangle-group coordinates. We tried 40, 60, and then 90 digits of precision. Even at 90 digits, and with height bounds approaching 10^{36}, we could not recognize the coefficients. The AI systems suggested pushing the precision higher. At that point, however, our computational resources were already limited, so we stopped and asked whether the real problem was our choice of coordinates.

We then realized that coordinates defined using the Belyi map are equivariant under the Galois action. This removes the $O(\varepsilon^2)$ ambiguity coming from the original coordinates. The change was dramatic. With the new coordinates, 40 digits were enough to recognize the quadratic equation P, and its coefficients had heights below 10^3. This was far below the height bounds approaching 10^{36} that we had already tried. At this point, $(2A, 23A, 23B)$ clearly stood out among the five candidates.

We still had to recover the cubic equation Q. The reliable digits left after recognizing P were not enough to recognize Q directly. The initial AI suggestion was again to increase the precision. Instead, we developed a separate reduction algorithm and recovered Q from the same 40-digit data. The equations P and Q define the image of the genus-4 curve X under its canonical embedding into $\mathbb{P}^3$.

Some of the most important decisions were clearly mathematical judgments made by the human collaborators. These included deciding what to compute, stopping an increasingly expensive calculation, changing the strategies, interpreting the numerical evidence, and deciding where to direct our limited resources.

Our multi-agent AI workflows helped us generate and test code, analyze data, work with existing packages, organize large computations, and monitor and recover from failures. This saved us a great deal of time. But the overall strategy, and the decisions to redirect it when something was not working, came from the human collaborators.

We also tried more autonomous approaches at the beginning of the project. Within our limited AI-usage budget, these attempts did not make meaningful progress and did not give us much guidance about what to try next. The workflow that eventually succeeded was not end-to-end AI autonomy. It was a tightly controlled collaboration in which AI made the computational work much faster, while the human collaborators continued to choose and revise the mathematical direction.

For a sense of the computational scale, the 60-digit computation of orbit 7 took approximately 1.71 CPU-hours. All the computations that followed were completed within one day and used fewer than 12 CPU-hours in total. These included recognizing and analyzing the coefficients of the quadratic equation P and the cubic equation Q, computing the Belyi map, carrying out the descent to $\mathbb{Q}$, and constructing a polynomial

$$F(T, V) \in \mathbb{Z}[T, V]$$

whose splitting field over $\mathbb{Q}(T)$ has Galois group M_{23}. We then specialized T to a rational value and used the PARI/GP routines `polredabs` and `polredbest` to obtain a lower-height degree-23 polynomial

$$f(x) \in \mathbb{Z}[x].$$

The final mathematical claims were verified rigorously in Magma and PARI/GP without AI. No text in the manuscript was written by AI, although we did use multiple AI agents for internal checking and proofreading.

The main lesson for me is that more computing power alone would not have solved the problem. The best analogy is deep-space observation: AI gave us a powerful and precise telescope, but the team's mathematical knowledge told us where to point it, when further magnification would not help and how to interpret what came into view. Once we chose the right object and the right coordinates, the relevant structure became visible.

The result grew out of many fruitful discussions and our exploration of human-AI collaboration. I hope this personal account gives a clearer picture of how that collaboration worked in practice.

The full paper is available on arXiv.