RANDOMIZED ALGORITHMS FOR VERIFYING MONODROMY GROUPS

JUHEE KIM (JOINT WORK WITH TAYLOR BRYSIEWICZ)

Numerically computing the monodromy group of parametrized polynomial systems is a fundamental problem in numerical algebraic geometry. In practice, it is done by sampling permutations via numerical homotopy continuation, where n solution paths are tracked along loops in the base parameter space. Each loop induces a permutation of the solutions, yielding elements of the symmetric group S_n, and the monodromy group $G \leq S_n$ is generated by these permutations. However, numerical path-tracking is inherently error-prone and path crossings can occur, producing incorrect permutations. If G is a proper subgroup of S_n, then the inclusion of a single erroneous permutation in the generating set for G can lead to an incorrect result, making monodromy group computations extremely sensitive to numerical errors. While certified path-tracking methods exist which guarantee the correctness of the computed group, they are typically computationally expensive.

We propose an efficient alternative for recovering G, under a simplified probabilistic framework. In our generalized setting, we consider the problem of recovering some unknown permutation group $G \leq S_n$ from an error-prone sampling process which returns an erroneous permutation with probability p, and otherwise a random element of G with probability $1-p$. Here, an erroneous permutation is interpreted as a random element of S_n. We model the output of this sampling process as an S_n-valued random variable $X = X(G,p)$:

$$\Pr(X=\sigma) = \begin{cases} (1-p)\frac{1}{|G|} + p\frac{1}{n!} & \sigma \in G, \\ p\frac{1}{n!} & \sigma \notin G. \end{cases}$$

In practice, one may attempt to recover G by taking k independent samples $X_1, \ldots, X_k$ and returning the group they generate - we call this the naive algorithm. This method is, however, sensitive to errors as mentioned previously. The naive algorithm is randomized, and has some probability $\gamma(G,p,k)$ of success. If this base success rate is greater than $\frac{1}{2}$, then it can be amplified via repetition and taking the majority vote to achieve an arbitrarily high success rate. In our algorithm, we provide error detection methods based on properties of G that improves $\gamma(G,p,k)$ beyond $\frac{1}{2}$, and thus allows for successful amplification. We additionally provide algorithms to extract properties of G such as transitivity and orbit structures that can be used in error detection. We analyze the performance of our algorithm for various classes of groups and error rates, and demonstrate its effectiveness in recovering G in cases where the naive algorithm fails.