Software evolution often requires the untangling of code. Particularly challenging and error-prone is the task of separating computations that are intertwined in a loop. The lack of automatic tools for such transformations complicates maintenance and hinders reuse.

We present a theory and implementation of fine slicing, a method for computing executable program slices that can be finely tuned, and can be used to extract non-contiguous pieces of code and untangle loops. Unlike previous solutions, it supports temporal abstraction of series of values computed in a loop in the form of newly-created sequences. Fine slicing has proved useful in capturing meaningful subprograms and has enabled the creation of an advanced computation-extraction algorithm and its implementation in a prototype refactoring tool for Cobol and Java in Eclipse.

Aharon Abadi has been a research staff member at IBM Research – Haifa since 2007. He works on a range of topics, including static program analysis, refactoring, program transformation, automatic code fix, semantic search, and software verification. His work has been included in IBM products. His most recent work has been on inter-procedural static analysis, inter-procedural slicing, and inter-procedural domination and post-domination.