

The Effect of Sparse Matrix Multiplication on Analytical Ultracentrifugation Analysis Using Ultrascan Software

Zachary Adam Ozer

Analytical ultracentrifugation (AUC) is a powerful technique for determining self-association and interaction properties of macromolecules within solution. In order to characterize the interaction and association of a sample, AUC scan data must be gathered from a wide array of experiments and then inserted into a matrix, whereupon a nonlinear least squares (NLS) algorithm is used to fit the parameters globally. However, not all datasets contain data for all parameters. In experiments for which the value of the variable is undetermined, it is assigned a value of zero, thus ensuring that it does not contribute to the fitted value. In most cases, this results in a sparse data matrix, that is, one that is populated mainly by the zeros. Yet, the NLS algorithm is an extremely time consuming process. Each improvement in the fit requires that the entirety of the data matrix be multiplied by its transpose, an operation which entails the application of a series of mathematical operations to every entry within the matrix, including the vast numbers of zeros. The purpose of this project was to optimize the efficiency of analysis for analytical ultracentrifugation data, specifically as it is utilized in the Ultrascan software. The hypothesis of this experiment stated that substituting a highly repetitive, but computationally simple matrix multiplication algorithm with a more complex process, such as a sparse matrix multiplication algorithm, would provide a notable improvement in efficiency, as only entries which contribute to the NLS algorithms fit will be processed.

The resulting improvement in overall efficiency was staggering. For the largest dataset, one involving 38,121 data points, the original algorithm required an average of 87,797.9 ms per iteration with a standard deviation of 57,388.5 ms per iteration. The sparse matrix algorithm, alternatively, required only 737.56 ms per iteration with a standard deviation of 309.04. Furthermore, the time per operation of the sparse matrix algorithm was one third of that used by the original algorithm. Although the algorithmic improvements appear quite impressive, the conversion to a sparse matrix data structure yielded results that are even more phenomenal. For the largest dataset, 38,121 data points, the compressed data required two orders of magnitude less space than the uncompressed data, dropping from 46.05 megabytes to 761.62 kilobytes. Although it is difficult to attribute the improvement in efficiency to either the data structure or the improved multiplication algorithm, since both are required for either to perform their function, it seems that the benefit of the new implementation is a result of the improved data structure.