

# Implementation of a Large-scale Eigenvalue Solver

**Intern:** Sam Handler

**Adviser:** Yousef Saad

I have implemented a eigenvalue solver for large, sparse matrices. The algorithm used (the Lanczos Algorithm) allows the approximation of the smallest (or largest) eigenvalues of large matrices, in far less time than would be needed for exact calculations of the same values. There are many fields where such a solver could be applicable; some examples are quantum mechanics, structural engineering, and graph theory. Although matrices typically have many eigenvalues, in most cases only the smallest or largest eigenvalues are of interest; the Lanczos method provides a very efficient way of calculating these.

Since this calculation requires the storage of thousands of large vectors, it must make efficient use of disk storage. My implementation does this by keeping as many vectors in RAM as possible, and attempting to load vectors from the disk before they are needed.

This implementation can handle matrices of any reasonable size; the code should correctly handle matrices with up to billions of rows. The practical limitations of this implementation are the user's disk size and patience; the former limits matrix size to about one million rows on a typical workstation.

However, the implementation is currently limited to running on a single processor. I hope to later expand it to run on larger, multiple processor machines, taking advantage of the parallel nature of the algorithm.

Aside from additional programming experience, this internship has given me knowledge of some of the software tools and programming techniques used in large-scale, high-performance computing. I hope to apply these techniques while further developing the software I have started this summer, as well as on future projects.