

# SimTKlapack

## A Library for solving linear algebra problems

Jack Middleton\*

*ROUGH DRAFT*, April 4, 2006

### Abstract

This document describes the goals and design decisions behind the SimTKlapack library, which is a linear algebra library for FORTRAN and C/C++ programmers. The goal is to provide an accurate, high performance library for solving linear algebra problems. [Note: this document is only partially written at the moment.]

1	Purpose of this document .....	1
2	Goals .....	1
	2.1 Performance.....	1
	2.2 Accuracy.....	1
3	Design issues.....	1
	3.1 Why we chose ATLAS.....	1
	3.2 Shared libraries.....	1
	3.3 Updates to ATLAS.....	2
4	Implementation.....	2
	4.1 Build system structure .....	2
	4.2 Regression tests .....	2
5	Using LAPACK .....	2
	5.1 Documentation.....	2
	5.2 Examples .....	2
	Acknowledgments.....	2
	References.....	2

## 1 Purpose of this document

To describe the goals, design issues, implementation and use of the SimTKlapack library.

## 2 Goals

The overall goal of this project is to provide developers and researchers access to high quality numerical methods software for solving linear algebra problems. This requires that the software be accurate, robust, and easy to use and be free of licensing issues.

---

\* Department of Bioengineering, Stanford University, [jacklm@stanford.edu](mailto:jacklm@stanford.edu), (650) 736-1378.

## 2.1 Performance

Our goal is to provide high performance software that performs well on large problems (matrices greater than 100x100) running on small tightly connected SMP systems.

## 2.2 Accuracy and robustness

The software must provide accurate results and be stable across a wide range of problems.

## 2.3 Ease of use

The library must be easy to install on our target platforms (Linux, Mac and Windows) and the programming interfaces be easy to use. This requires good documentation, and examples.

## 2.4 No licensing issues

Our goal is to provide software that does not have any licensing issues associated with it. This will allow developers to use it to create products without dealing with any licensing encumbrances.

## 3 Design issues

The project goals stated above created a number of design issues which are explained below.

### 3.1 Why we chose ATLAS

ATLAS (Automatically Tuned Linear Algebra Software)[2] is a high performance code generate system for linear algebra. We chose ATLAS to build our implementation of LAPACK because it produces very fast, high quality code for the target platform and has no licensing restrictions. The ATLAS build system generates a set of candidates for each routine based on parameters such as cache size, number of registers and supported instruction sets. It then times each candidate routine and selects the fastest candidate.

### 3.2 Shared libraries

SimTKlapack is deployed as a single shared library. We decided on a shared library instead of a static library because applications can take advantage of any bug fixes or performance improvements simply by installing the new library. The application does not have to be relinked.

### 3.3 Updates to ATLAS

Because ATLAS is actively being developed we need to be able to easily incorporate new versions of ATLAS into our build system. Our goal is to replace our existing version of ATLAS with a new version when updates are available without making major changes to our build system.

## 4 Implementation

### 4.1 Build system structure

The build system for ATLAS is very complex because its timing and comparison measurements are integrated into its build process. We decided to make as few changes to the ATLAS build process as possible while still allowing cross-platform builds. To accomplish this we use cmake custom commands to pass arguments to the ALTAS configure program and then initiate the ATLAS build process. ALTAS generates a set of static libraries which implement BLAS and a subset of the LAPACK API. Cmake is used to compile the source for the remaining LAPACK API and combines these objects with the ATLAS built objects into the SimTKlapack library.

In order to build ATLAS on our nightly build system, without user intervention, two types of changes needed to be made to ATLAS. The first change modified ATLAS's configure program to stop prompting the user to input various parameters such as build directories and the number of CPU's. New switches were added to the configure program to allow passing in any parameters that the configure program could not determine at runtime. The second type of change involved modifying routines used in comparing the execution time of candidate routines to produce consistent timing results. When ATLAS compares execution times of candidate routines it runs the each candidate multiple times to verify that the results are consistent and are not affected by other processes running on the build system. If ATLAS encounters inconsistent results the build stops. Changes were made to force the tests to be rerun if inconsistencies are encountered and to adjust the estimated FLOPS counts that are used to determine the size of the matrices used in the timing tests.

All ATLAS source code is stored in the subversion source tree as a single compressed tar image. At build time, the tar file is untarred and any ATLAS files that need to be changed are overwritten before the build starts.

### 4.2 Regression tests

## 5 Using LAPACK

### 5.1 Documentation

### 5.2 Examples

## Acknowledgments

This work was funded by the National Institutes of Health through the NIH Roadmap for Medical Research,<sup>1</sup> Grant U54 GM072970.

## 6 References

- [1] LAPACK Users' Guide, Third Ed, by Anderson, E. and Bai, Z. and Bischof, C. and Blackford, S. and Demmel, J. and Dongarra, J. and Du Croz, J. and Greenbaum, A. and Hammarling, S. and McKenney, A. Society for Industrial and Applied Mathematics, 1999
- [2] Automatically Tuned Linear Algebra Software, by R. Clint Whaley and Jack Dongarra, Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999
- [3] Information on the National Centers for Biomedical Computing can be obtained from: <http://nihroadmap.nih.gov/bioinformatics>

---

<sup>1</sup> Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>.