SimTKlapack README

Version 1.1

Jack Middleton

June 7, 2006

1 What is SimTKlapack

SimTKlapack is a high performance linear algebra library for C/C++ and FORTRAN programmers. SimTKlapack implements the LAPACK (Linear Algebra PACKage) [1] and BLAS (Basic Linear Algebra Subprograms) [2] standards and provides routines for solving systems of simultaneous linear equations, eigenvalue problems, singular value problems and least-squares solutions of linear systems of equations. SimTKlapack also provides the LU, Cholesky, QR, SVD, Schur, generalized Schur matrix factorizations and operations for estimating condition numbers. It supports both dense and banded matrices and can operate on real and complex matrices in single or double precision. SimTKlapack is internally threaded and will automatically take advantage of multiple processors.

.

2 What is in this download

The download file includes the SimTKlapack shared library, header file, and example programs for using SimTKlapack. The download file is organized into 4 directories: lib, include, doc and examples. The lib directory contains the SimTKlapack shared library. On Linux systems this is named libSimTKlapack.so and on Mac it is named libSimTKlapack.dylib. On Windows there are two files; SimTKlapack.dll and SimTKlapack.lib. The include directory contains the SimTKlapack.h header file which contains the function declarations for C and C++ programs to call the LAPACK FORTRAN routines. The doc directory contains this file. The examples directory contains example programs that use the SimTKlapack.h header file and the SimTKlapack shared library.

3 Which file should I download?

Because SimTKlapack has been tuned for specific hardware platforms, download files are provided for each hardware platform. The name of the download file indicates the system SimTKlapack was tuned for. The format of each file name is: SimTKlapack_OS_32/64bit_processor_NCPU. The OS field indicates the operating system SimTKlapack was built on (Windows, Linux, or Mac). The 32/64bit indicates if the library was built to support 32 or 64 bit applications. The processor field indicates which processor SimTKlapack was tuned for. NCPU indicates how many processors in an SMP SimTKlapack can take advantage of. Files that do not have a NCPU field always use one processor. For example, a file named SimTK_Mac_PPCG5_2 was tuned for a 2 processor, Power PC G5 running the Mac OSX operating system. For help more help on determining the correct file to download for your system consult the Download_help.pdf document.

4 How to use SimTKlapack

SimTKlapack implements the standard FORTRAN 77 interface for LAPACK and BLAS which can also be called from C and C++ programs. Instructions on how C and C++ programs can call SimTKlapack are provided in the README file in the examples directory of this download.

5 Where to go for more information

To learn more about using the LAPACK API a good place to start is the "LAPACK User's Guide" third edition. You can either buy a hardcopy version (ISBN: 0898714478) or use the online version at: http://www.netlib.org/lapack/lug. Also the Numerical Algorithms Group (NAG) [4] has a nice set of FORTRAN example for using LAPACK. UNIX man pages can also be downloaded from http://www.netlib.org/lapack/manpages.tgz.

6 SimTKlapack performance

The performance of SimTKlapack vs. coding it yourself if very good. For example, a matrix multiply of a 2000x2000 single precision matrix is over 200 times faster in SimTKlapack on a 4 processor AMD opteron than using the code from Numerical Recipes [5] compiled with all optimizations. SimTKlapack is optimized for large problems (matrices larger than 50x50). If your application does many linear algebra operations on small problems, other linear algebra packages may give better performance.

6 Known Limitations

There are no known limitations in SimTKlapack 1.0. Version 0.9.1 of SimTKlapack had a few routines that were not thread safe, however, these have been fixed in Version 1.0.

7 Changes to the LAPACK API

In order to make SimTKlapack thread safe arguments needed to be added at the end of the parameter list for routines that were doing reverse communication with the calling routine. The values of the additional arguments are set in the routine itself and should be initialized to zero in the calling routine before the first time the routine is called. The routines which have additional arguments are: (s/d/c/z)lacon which added: J, JUMP, ITER. The (s/d)lasq4 routines which added the argument G. The (s/d)lasq3 routines added the TTYPE, DMIN1, DMIN2, DN, DN1, DN2, TAU arguments

Also note that the auxiliary routine slamch returns a double value when called from C programs.

8 References

[1] http://www.netlib.org/lapack

[2] http://www.netlib.org/blas

[3] http://math-atlas.sourceforge.net

[4] http://www.nag.com/lapack-ex/lapack-ex.html

[5] "Numerical Recipes in C++: The Art of Scientific Computing", Cambridge University Press; 2 edition