Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Note

Choice of linear algebra libraries for NRLib

Note no SAND/02/09

Authors Per Røe, Ragnar Hauge, Ariel Almendral Vazquez

Date February, 2009

Norsk Regnesentral

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modelling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Norsk Regnesentral
Norwegian Computing Center
Postboks 114, Blindern
NO-0314 Oslo, Norway

Besøksadresse
Office address
Gaustadalléen 23
NO-0373 Oslo, Norway

Telefon · telephone
(+47) 22 85 25 00
Telefaks · telefax
(+47) 22 69 76 60

Internett · internet
www.nr.no
E-post · e-mail
nr@nr.no

Title	**Choice of linear algebra libraries for NRLib**
Authors	**Per Røe, Ragnar Hauge, Ariel Almendral Vazquez**
Date	February
Year	2009
Publication number	SAND/02/09

Abstract

This document describes alternatives for linear algebra libraries that were considered for use in NRLib, and the rationale for choosing the combination of Intel MKL and FLENS.

Keywords	Linear Algebra, C++, NRLib
Target group	Internal
Availability	Open
Project number	1049
Research field	Numerics, programming
Number of pages	11

Contents

1 Introduction

Currently Sand is using several different linear algebra packages in different projects. Some projects use self-implemented functions based on numerical recipes while some other projects use Matrix template library (MTL) or Boost uBLAS.

In association with the compilation of the NRLib code library several linear algebra libraries was evaluated for use as a standard linear algebra library for Sand. To ensure stability and high performance we opted for a solution based on LAPACK and BLAS with a C++ interface on top.

2 LAPACK and BLAS implementation

In addition to the original reference implementation of LAPACK and BLAS there exist several optimized implementations, both general and vendor specific.

We evaluated four alternatives, the LAPACK and BLAS reference implementations, ATLAS, Intel math kernel libraries (MKL), and AMD core math library (ACML).

2.1 Reference implementations

Web-site: http://www.netlib.org/lapack/ http://www.netlib.org/blas/

This is the original reference implementations for BLAS and LAPACK in FORTRAN.

Pros and Cons:

+ Free.
- Implemented in FORTRAN – complicated to compile, especially on Windows.
- Relatively slow, not optimized for modern platforms.

2.2 Automatically Tuned Linear Algebra Software (ATLAS)

Web-site: http://math-atlas.sourceforge.net/

ATLAS is a highly optimized implementation of BLAS and parts of LAPACK. Testing is done during compilation to optimize the compiled library for the platform which it is compiled on.

Pros and Cons:

+ Free.
+ Fast. Performance similar to the vendor-optimized libraries.
- Needs Cygwin for Windows compilation.
- Not a complete LAPACK implementation, reference LAPACK must also be installed to get a complete LAPACK library.

2.3 Intel math kernel libraries MKL

Web-site: http://www.intel.com/cd/software/products/asmo-na/eng/307757.htm

Intel MKL is Intel's optimized math libraries for the Intel processors, offering very good performance on Intel processors, and also good performance on AMD processors. Intel MKL consists of implementations of LAPACK, BLAS, ScaLAPACK in addition to libraries for solving of sparse equation systems and an FFT library.

A license must be purchased for commercial use. A 2-seat floating license costs around NOK 6000,- and should handle 10-25 developers, depending on compilation time, since the license is only checked during compilation and linking.

Pros and Cons:

+ Fast.
+ Available both on Windows and Linux.
+ Comprehensive.
- Needs to buy a license.

2.4 ACML

Web-site: http://developer.amd.com/cpu/Libraries/acml/Pages/default.aspx

ACML is AMD's math library, containing a BLAS and LAPACK implementation in addition to an FFT library. It is fairly comprehensive, but does not contain a CBLAS interface. It also seems more complicated to compile and link with ACML than Inter MKL.

Pros and Cons:

+ Fast.
+ Free.
+ Available both on Windows and Linux.
+ Comprehensive, although less so than Intel MKL.
- Allegedly complicated in use.

2.5 Conclusion

NR chose to use Intel MKL. The main reasons for doing this were performance and ease to use. Although a license is needed, the price is affordable.

3 C++ linear algebra library

There has been very many attempts on making a good linear algebra library for C++, however many of these libraries were never completed, and are not in active development. The following libraries were discarded for this reason:

- Blitz++ - http://www.oonumerics.org/blitz/

- TNT - http://math.nist.gov/tnt/

We were looking for a library that gave simple linear algebra functionality and a C++ interface to BLAS/LAPACK.

3.1 Boost uBLAS

Web-site: http://www.boost.org/doc/libs/1_38_0/libs/numeric/ublas/doc/index.htm

uBLAS is a C++ template library that provides BLAS functionality. It is part of the boost package. uBLAS is based on the use of expression templates. It is possible to use uBLAS as a layer on top of BLAS and LAPACK using "bindings".

Pros and Cons:

+ Widely used.
+ Under continuously development.
+ Fairly comprehensive
+ LAPACK and BLAS bindings
- Dependencies on rest of Boost – need many (100-200) files from boost to compile.
- Quite difficult code
- Hard to extend.

3.2 The Matrix Template Library (MTL)

Web-site: http://www.osl.iu.edu/research/mtl/

MTL is a C++ linear library based on expression templates. It has a pretty nice syntax for easy operations. NR has used MTL v.2 in some projects but have experienced some cases where the calculated results were wrong. MTL v4 is under development, but is still in a early phase.

Pros and Cons:

+ Nice notation.
+ Fairly comprehensive
- No LAPACK and BLAS bindings
- Only BLAS functionality, no LAPACK functionality
- Quite difficult code
- Hard to extend.

3.3 NewMat

Web-site: http://www.robertnz.net/nm_intro.htm

NewMat is a fairly simple matrix library for C++.

Pros and Cons:

+ Simple.
+ Has the most important functionality.
- No LAPACK and BLAS bindings
- Only BLAS functionality, no LAPACK functionality

3.4 FLENS

Web-site: http://flens.sourceforge.net/

Flens is a light-weight C++-interface for BLAS and LAPACK. It provides a nice interface, while all the calculations are done in BLAS and LAPACK. It has a one-to-one mapping with BLAS, it is only possible to compile expressions that it is possible to represent as a single BLAS call, or a series of BLAS calls without the use of any temporary variables.

Pros and Cons:

+ Very nice notation.
+ Easy to extend.
+ Only an interface on top of LAPACK and BLAS.
+ Readable code.
- Limited support – few developers.
- Unfinished.

3.5 GMM++

Web-site: http://home.gna.org/getfem/gmm_intro

GMM++ is a C++ linear algebra library similar to MTL. It is fairly comprehensive and still developed. It can be used on its own, but also includes interfaces to BLAS, LAPACK, ATLAS and SuperLU.

Pros and Cons:

+ Comprehensive.
+ LAPACK and BLAS
- No operator overloading.
- Limited support – few developers.
- Quite complex

3.6 Lapack++

Web-site: http://lapackpp.sourceforge.net/

Lapack++ is a C++-interface for LAPACK. Originally developed by NIST where it was superseded by TNT. Lapack++ was not considered, mainly due to thread safety issues, see http://www.crystalclearsoftware.com/cgi-bin/boost_wiki/wiki.pl?Linear_Algebra_With_UBLAS

3.7 Conclusion

Of the libraries listed here, only uBLAS, FLENS and GM++ were considered as suitable for NRLib. uBLAS was discarded due to the extensive dependencies with other parts of the boost library, and due to the complexity of extending the bindings to LAPACK or ATLAS.

FLENS and GMM++ seem pretty good, but are both still under development, with small groups of developers. FLENS was chosen for two reasons:

- Its limited goal of only being a BLAS/LAPACK interface suited us.

- Due to its limited scope, the design is simple and nice, and we can do extensions, modifications and bugfixes on our own.