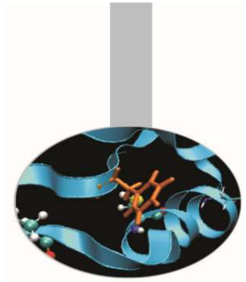
Numerical libraries

M.Cremonesi

May 2016

Numerical libraries



Numerical libraries are collections of functions that implement a variety of mathematical algorithms. These may include low level operations such as matrix-vector arithmetics or random functions, but also more complicated algorithms such as Fast Fourier Transforms or Minimization Problems.

Numerical libraries

Linear algebra operations are among the most common problems solved in numerical libraries. Typical operations are:

Scalar products: $$s = \sum_i a_i . b$$

Linear Systems: $$\mathrm{A}_{\mathrm{ij}} \cdot x_j = b_i$$

Eigenvalue Equations: $$\mathrm{A}_{\mathrm{ij}} \cdot x_j = \alpha \cdot x_i$$

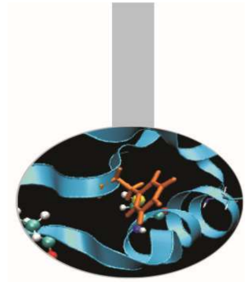
Numerical libraries

Libraries should be used in programs:

- To avoid code repeating
- To enhance program functionality
- To avert numerical errors
- To gain efficiency

As far as parallel computing is concerned many versions of numerical libraries are available to run efficiently in different computer environments.

Numerical libraries

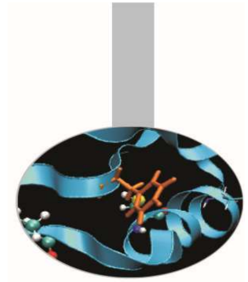


Many numerical libraries have been written to solve linear system equations efficiently.

Linear problems are of the kind: find x := A . x = b

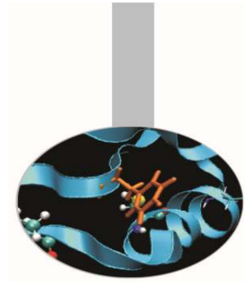
Not linear problems may be solved with a sequence of linear problems.

Numerical libraries



Solving a linear system with Gaussian elimination can take a lot of time and memory, for large matrices.

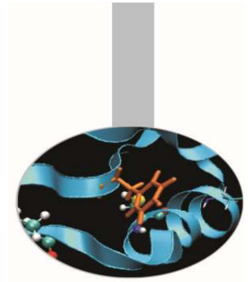
This is why many libraries use iterative solvers. They are based on finding solution of the problem by calculating successive approximations, even though convergence can not always be guaranteed.



Numerical libraries

Iterative solvers are faster and use less memory but a check for correctness must be computed at each step, and a pre-conditioner is usually needed.

The condition number associated to a linear equation $A \cdot x = b$ gives a bound on how inaccurate the solution x will be after approximate solution. The value of the condition number depends on the properties of the matrix A. It is not related to round-off errors nor accuracy in computing floating point operations. Instead it could be interpreted as the rate at which the solution x will change with respect to a change in b. This rate should be as closed to 1 as possible.



Numerical libraries

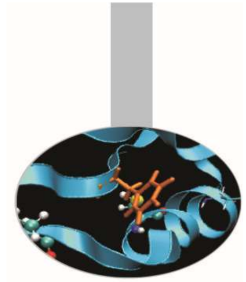
A pre-conditioner of a matrix A is a matrix P such that $P^{-1} \cdot A$ has a condition number smaller than A.

In many practical problems matrix A is:

- Large
- Sparse
- Symmetric positive definite

Linear problems come from a variety of geometric and engineering problems; they are of interest in image processing too. They are often encountered whenever PDE equations have to be solved.

Numerical libraries



BLAS/CBLAS (Basic Linear Algebra Subprograms)

This is one of the first written numerical libraries and include a lot of low level matrix and vector operations. They have been optimized and many implementations exist, each one tuned to a specific computer architecture. For this reason BLAS routines are used by many other libraries.

Language: FORTRAN, C

Availability: public domain

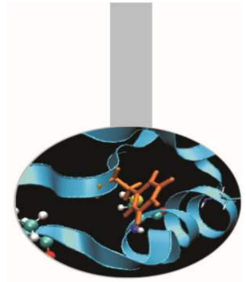
Developers: Jack Dongarra, ORNL and Eric Grosse, Bell Labs

Distributors: NETLIB

Ref.: The University of Tennessee at Knoxville and Bell Laboratories

http://www.netlib.org/blas/

BLAS



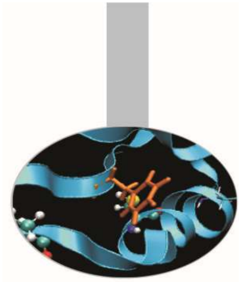
Level 1 BLAS are FORTRAN (66/77) subroutines for computing basic scalar and vector operations. They represent the conclusion of a collaborative project that ended in 1977.

Level 2 BLAS subroutines have been written for implementing vector-matrix operations; they were developed in 1984-1986.

Level 3 BLAS are Fortran subroutines for computing matrix-matrix operations and are available since 1988.

Level 2 and level 3 BLAS exploit vector and hierarchical memory architectures to the greatest extent. For this reason they are widely used by LAPACK, a well-known linear algebra package.

BLAS

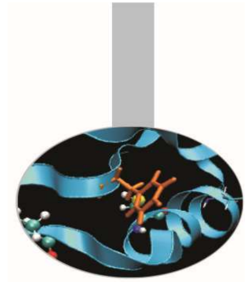


BLAS subroutines have been written to be applied to real and complex data, either simple and double precision.

Functionalities

Scalar and vector operations:

- *Dot product, vector norm, sum, sum of squares, min/max value and location*
- *Givens and Jacobi rotation, Householder transform*
- *Scaling, plane rotation*
- *Copy, swap, permutation, gather and scatter*



BLAS

Functionalities

Matrix-vector operations:

- *Matrix-vector products*
- *Triangular solve*

Matrix operations:

- *Norms, diagonal scaling*
- *Product, Triangular multiply and solve*
- *Copy, transpose, permutation*

Numerical libraries

BLACS (Basic Linear Algebra Communication Subprograms)

This is a set of subroutines that implement low level matrix and vector operations on distributed memory platforms. The library has been developed to be efficiently portable on several computing platforms. It can use different communication interfaces including MPI. It is used by other higher level parallel libraries, among those is scaLAPACK.

Language: C, FORTRAN

Availability: public domain

Developers: Jack J. Dongarra and R. Clint Whaley

Distributors: NETLIB

Ref.: The University of Tennessee at Knoxville and Bell Laboratories

http://www.netlib.org/blacs/

Numerical libraries

LAPACK/LAPACKE

These libraries provide Fortran/C subroutines for solving systems of simultaneous linear equations, least-squares of linear systems of equations, eigenvalue problems, and singular value problems. The aim of the LAPACK project was to substitute EISPACK and LINPACK libraries in order to run efficiently on shared-memory vector and parallel processors.

Language: FORTRAN, C

Availability: public domain

Developers: Jack Dongarra, ORNL and Eric Grosse, Bell Labs

Distributors: NETLIB

Ref.: The University of Tennessee at Knoxville and Bell Laboratories

http://www.netlib.org/lapack/

LAPACK

LAPACK library is designed to exploit at the higher extent the Basic Linear Algebra Subprograms (BLAS), with a preference to Level 3 BLAS. This is for running as much as efficiently on modern computing architectures.

Highly efficient machine-specific implementations of BLAS library should be available on each computing platform. Alternatively, ATLAS tool is available for generating optimized BLAS installation for the computer in use.

It is recalled that a Fortran 77 implementation of the BLAS library is available from netlib, however it should not be used for production because will not usually perform at maximum on many computing environments.

LAPACK

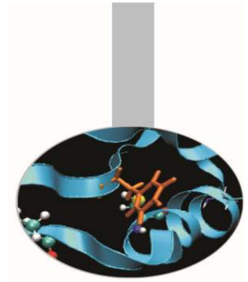
As LAPACK subroutines are Fortran based, the elements of matrices are memorized in column wise order, at least for the dense ones. Packed storage formats are used for triangular or banded matrices.

Lower triangular matrices: elements are memorized column wise; columns are packed as length-decreasing blocks of data.

A(1,1) - 00					
A(2,1) - 01	A(2,2) - 06				
A(3,1) - 02	A(3,2) - 07	A(3,3) - 11			
A(4,1) - 03	A(4,2) - 08	A(4,3) - 12	A(4,4) - 15		
A(5,1) - 04	A(5,2) - 09	A(5,3) - 13	A(5,4) - 16	A(5,5) - 18	
A(6,1) - 05	A(6,2) - 10	A(6,3) - 14	A(6,4) - 17	A(6,5) - 19	A(6,6) - 20

Indices of elements and memory addresses.

SuperComputing Applications and Innovation



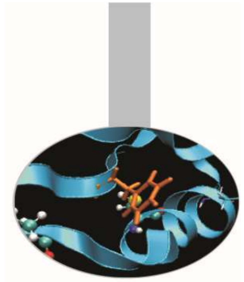
LAPACK

Alternatively they could be memorized in a so called *Rectangular Full Packed Storage Format*. As an example:

A(4,4) – 00	A(5,4) – 07	A(6,4) – 14			
A(1,1) – 01	A(5,5) – 08	A(6,5) – 15			
A(2,1) – 02	A(2,2) – 09	A(6,6) – 16			
A(3,1) – 03	A(3,2) – 10	A(3,3) – 17			
A(4,1) – 04	A(4,2) – 11	A(4,3) – 18	A(4,4)		
A(5,1) – 05	A(5,2) – 12	A(5,3) – 19	A(5,4)	A(5,5)	
A(6,1) – 06	A(6,2) – 13	A(6,3) – 20	A(6,4)	A(6,5)	A(6,6)

In the table red elements have been moved to blue positions in order to reduce memory requirements.

LAPACK



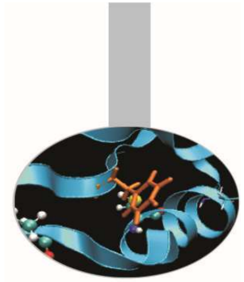
Utility subroutines are provided in LAPACK library to help transforming matrices from a triangular format to an other:

DTPTTF copies a triangular matrix A from standard packed format (TP) to rectangular full packed format (TF).

DTPTTR copies a triangular matrix A from standard packed format (TP) to standard full format (TR).

DTRTTF copies a triangular matrix A from standard full format (TR) to rectangular full packed format (TF) .

DTRTTP copies a triangular matrix A from standard full format (TR) to standard packed format (TP).



LAPACK

Band matrices can be memorized in *Banded Storage Format*.

As an example, if A(N,N), N=6, has a lower band of length KL=2 and an upper band of length KU=1:

A(1,1) – 00	A(1,2) – 06				
A(2,1) – 01	A(2,2) – 07	A(2,3) – 13			
A(3,1) – 02	A(3,2) – 08	A(3,3) – 14	A(3,4) – 20		
	A(4,2) – 09	A(4,3) – 15	A(4,4) – 21	A(4,5) – 27	
		A(5,3) – 16	A(5,4) – 22	A(5,5) – 28	A(5,6) – 34
			A(6,4) – 23	A(6,5) – 29	A(6,6) – 35

It is “packed” as a 2*KL+KU+1=6 x N matrix:

Empty - 00	Empty – 06	Empty - 12	Empty - 18	Empty - 24	Empty - 30
Empty - 01	Empty – 07	Empty - 13	Empty - 19	Empty - 25	Empty - 31
Empty - 02	A(1,2) – 08	A(2,3) – 14	A(3,4) – 20	A(4,5) – 26	A(5,6) – 32
A(1,1) – 03	**A(2,2) – 09**	**A(3,3) – 15**	**A(4,4) – 21**	**A(5,5) – 27**	**A(6,6) – 33**
A(2,1) – 04	A(3,2) – 10	A(4,3) – 16	A(5,4) – 22	A(6,5) – 28	Empty - 34
A(3,1) – 05	A(4,2) – 11	A(5,3) – 17	A(6,4) – 23	Empty - 29	Empty - 35

Numerical libraries

ATLAS (Automatically Tuned Linear Algebra Software)

This is not a new library but a software that provides optimal linear algebra software. At now it provides a complete BLAS library and a very small subset of the LAPACK library. ATLAS performance should be equivalent to machine-specific tuned libraries.

Language: FORTRAN, C

Availability: public domain / open source

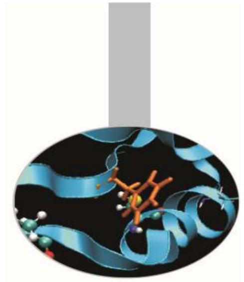
Developers: R. Clint Whaley, Antoine Petitet, Jack Dongarra

Distributors: sourceforge.net

Ref.: http://math-atlas.sourceforge.net/

http://www.netlib.org/atlas/

Numerical libraries



GotoBLAS

This is the result of the effort of Kazushige Gotō, a researcher at the Texas Advanced Computing Center (University of Texas at Austin). He hand optimized assembly routines of the BLAS library in order to run faster on many supercomputers. The work is somehow continued by the projects GotoBLAS2 and OpenBLAS.

Ref.: http://www.tacc.utexas.edu/tacc-projects/gotoblas2

http://xianyi.github.com/OpenBLAS/

SCAI
SuperComputing Applications and Innovation

Numerical libraries

PLASMA (Parallel Linear Algebra Software for Multi-core Architectures)

This library contains FORTRAN and C functions for the solution of linear equations. They have been written to be efficient on multi-core processors in a share memory node. The library has been developed to supercede LAPACK but lacks of some functions, among these eigenvalues and sparse and band matrices.

Language: FORTRAN, C

Availability: public domain

Developers: several

Distributors: University of Tennessee

Ref.: Dep. Electrical Engineering and Computer Science, University of Tennessee at Knoxville

http://www.netlib.org/plasma/

Numerical libraries

LAPACK++

C++ interfaces for the solution of linear equations and eigenproblems. It was meant to replace LAPACK calls in C++ programs.

Language: C++

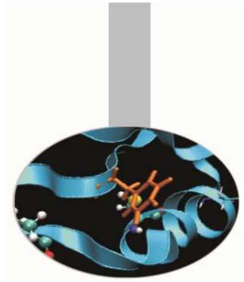
Availability: public domain

Developers: Roldan Pozo, Mathematical and Computational Sciences Division, NIST

Distributors: NETLIB

Ref.: The University of Tennessee at Knoxville and Bell Laboratories

http://www.netlib.org/lapack++/



Numerical libraries

TNT (Template Numerical Toolkit)

This is a collection of interfaces and implementations of objects useful for numerical computing in C++. It defines interfaces for basic data structures, such as multidimensional arrays and sparse matrices, commonly used in numerical applications. The package deals with many of the portability and maintenance problems with C++ codes.

Language: C++

Availability: public domain

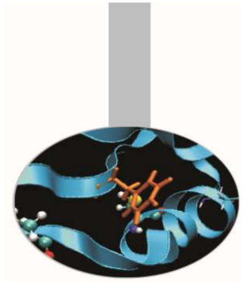
Developers: Roldan Pozo, Mathematical and Computational Sciences Division, NIST

Distributors: NETLIB

Ref.: National Institute of Standards and Technology

http://math.nist.gov/tnt

TNT



TNT provides a distinction between interfaces and implementations of components. For example, there is a TNT interface for two-dimensional arrays which describes how individual elements are accessed; however, there can be several implementations of such an interface.

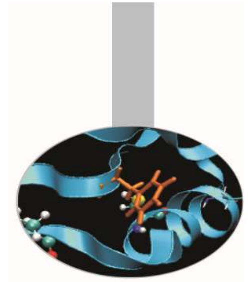
A few example lines:

```
Array2D< double > A(M,N, 0.0);

for (i=0; i < M; i++)
   for (j=0; j < N; j++)
       A[i][j] = f(i,j);

Array2D< double > B = A.copy
```

Numerical libraries



ScaLAPACK

High performance linear algebra subroutines specifically designed for parallel distributed memory platforms. ScaLAPACK solves dense and banded linear systems, least squares problems, eigenvalue problems, and singular value problems.

Languages: FORTRAN, C

Availability: public domain

Developers: several

Distributors: NETLIB

Ref.: The University of Tennessee at Knoxville and Bell Laboratories

http://www.netlib.org/scalapack/

ScaLAPACK

The qualities of the ScaLAPACK package are:

efficiency (hopefully on any platform)

scalability (at growing of number of processors and problem size)

reliability (results closed to serial execution)

portability (functionality and efficiency)

flexibility (in order to be useful in many situations)

and ease of use (interface as closed as possible to LAPACK).

ScaLAPACK is written in Fortran (with a bit of C code) and is based on the communication libraries MPI and PVM.

The functionality and efficiency of this package depend upon the installations of BLAS, LAPACK and BLACS.

Numerical libraries

AZTEC

This is a parallel library of iterative solution methods and preconditioners. Its main aim is to provide tools that facilitate handling the distributed data structures. It contains a collection of data transformation tools, as well as query functions of the data structures.

Language: FORTRAN, C

Availability: free

Developers: Ray S. Tuminaro, John N. Shadid, Mike Heroux

Distributors: Sandia National Laboratories

Ref.: Sandia National Laboratories

http://www.cs.sandia.gov/CRF/aztec1.html

Numerical libraries

FFTPACK

Fortran subroutines to compute fast Fourier transforms of periodic and other symmetric sequences. It includes complex, real, sine, cosine, and quarter-wave transforms.

Language: FORTRAN

Availability: public domain

Developers: Paul N. Swarztrauber, National Center for Atmospheric Research, Boulder, CO

Distributors: NETLIB

Ref.: The University of Tennessee at Knoxville and Bell Laboratories

http://www.netlib.org/fftpack/

Numerical libraries

FFTW (Fastest Fourier Transform in the West)

C functions to compute the discrete Fourier transform (DFT) in one or more dimensions, with arbitrary input size. Real and complex data are supported.

Language: C

Availability: public domain

Developers: Matteo Frigo and Steven G. Johnson, MIT

Distributors: FFTW

Ref.: Massachusetts Institute of Technology

http://www.fftw.org/

Numerical libraries

ARPACK/PARPACK (Arnoldi Package)

Fortran77 subroutines written to compute large scale eigenvalue problems. The library was developed to compute eigenvalues and eigenvectors of a general n by n matrix A. It is best when applied to large sparse matrices or matrices A such that a matrix-vector product requires order n rather than the usual order n^2 floating point operations.

Language: FORTRAN

Availability: public domain

Developers: Rich Lehoucq, Kristi Maschhoff, Danny Sorensen, Chao Yang

Distributors: Rice University

Ref.: Computational & Applied Mathematics, Rice University, Houston

http://www.caam.rice.edu/software/ARPACK/

Numerical libraries

ELPA (Eigenvalue SoLvers for Petaflop-Applications)

Implements an efficient eigenvalue solver for petaflop applications, suitable for large problems, where direct solvers are inefficient.

Language: FORTRAN

Availability: GNU license

Developers: multi-disciplinary consortium of partners

Distributors: Max-Planck-Gesellschaft

Ref.: http://elpa.mpcdf.mpg.de/

Numerical libraries

SLATEC Common Mathematical Library

FORTRAN subroutines to solve a variety of mathematical problems. It was developed to run efficiently on many parallel computing platforms.

The package includes the functionalities of BLAS, LINPACK, EISPACK and also contains interpolation and optimization algorithms, quadrature, statistics and other.

Language: FORTRAN

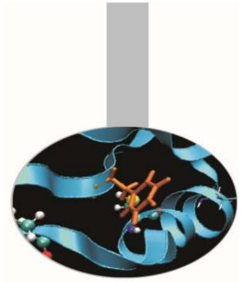
Availability: public domain

Developers: several

Distributors: NETLIB

Ref.: National Institute of Standards and Technology (NIST), Gaithersburg, Maryland.

http://www.netlib.org/slatec/



Numerical libraries

GSL (GNU Scientific Library)

Collection of C/C++ functions that implements many mathematical algorithms. More than 1000 functions are available including linear equation solvers, FFT, statistical functions, random numbers, differential equations, minimization, quadrature and other.

Language: C/C++

Availability: GNU license

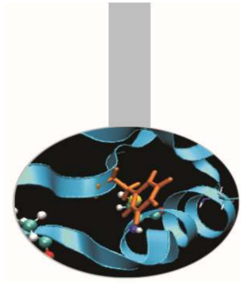
Developers: several

Distributors: GNU

Ref.: GNU

http://www.gnu.org/software/gsl/

SuperComputing Applications and Innovation



Numerical libraries

METIS

Collection of programs for partitioning graphs, finite element meshes and fill reducing orderings for sparse matrices. The algorithms implemented are based on the multilevel recursive-bisection, multilevel k-way, and multi-constraint partitioning schemes.

Language: C

Availability: free for academy and research

Developers: George Karypis Lab

Distributors: George Karypis Lab

Ref.: Department of Computer Science & Engineering, University of Minnesota

http://glaros.dtc.umn.edu/gkhome/views/metis

Numerical libraries

ParMETIS

MPI-based parallel library that extends the functionalities of METIS and includes functions suited for parallel Adaptive Mesh Refinement computations and large scale numerical simulations.

Language: C

Availability: free for academy and research

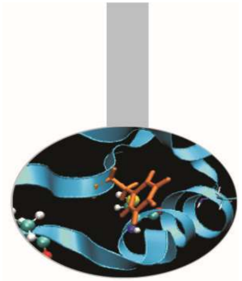
Developers: George Karypis Lab

Distributors: George Karypis Lab

Ref.: Department of Computer Science & Engineering, University of Minnesota

http://glaros.dtc.umn.edu/gkhome/views/metis

Numerical libraries



PETSC

This is a suite of data structures and routines for the parallel solution of programs based on partial differential equations. It supports MPI, shared memory pthreads, and NVIDIA GPUs, as well as hybrid parallelism. It is suitable for use in large-scale applications.

Language: C, C++, Fortran, Python

Availability: open source

Developers: several

Distributors: Mathematics and Computer Science Division, Argonne National Laboratory

Ref.: Argonne National Laboratory

http://www.mcs.anl.gov/petsc/

Numerical libraries

UMFPACK

It is a collection of C functions for solving unsymmetric sparse linear systems, A·x = b. It makes use of the Unsymmetric-pattern MultiFrontal method and direct sparse LU factorization. UMFPACK relies on the Level-3 BLAS for improving its performance.

Language: C/Fortran

Availability: GNU GPL license.

Developers: several

Distributors: Computer & Information Science & Engineering, University of Florida

Ref.: University of Florida

http://www.cise.ufl.edu/research/sparse/umfpack/

Numerical libraries

MKL (Math kernel Library)

This is a mathematical library of highly optimized, thread-parallel routines. It includes BLAS, LAPACK, ScaLAPACK1, sparse solvers, FFTs, vector operations, and other.

Language: Fortran, C, C++

Availability: proprietary

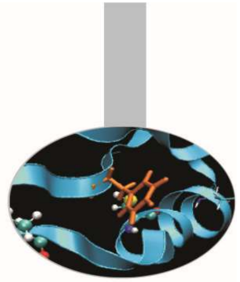
Developers: Intel

Distributors: Intel

Ref.: Intel

http://software.intel.com/en-us/articles/intel-mkl/

Numerical libraries



AMD (AMD Core Math Library)

It incorporates BLAS, LAPACK, FFT routines and many other mathematical functions. This library is designed for reaching good performances on AMD platforms.

Language: Fortran, C, C++

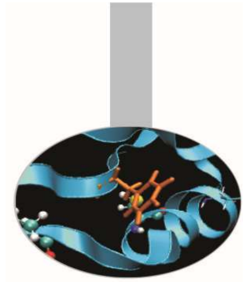
Availability: proprietary

Developers: AMD

Distributors: AMD, Portland Group

Ref.: AMD

http://developer.amd.com/libraries/acml/



Numerical libraries

In order of using a library in an application, the developer must pay attention to use the correct calling syntax.

In the linking phase the correct version of the installed library should be used: the developer should know where and how the software has been installed on the computing platform.

Furthermore there may be specific modalities for compiling and linking a program that makes use of a software library, particularly if it is a proprietary implementation. Some examples follow.

BLAS

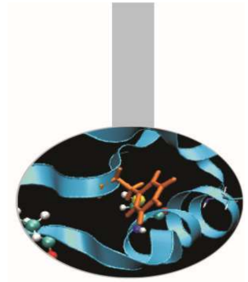
Intel: ifort <program> -L$MKLROOT/lib/intel64 -lguide -lpthread -lmkl

PGI: pgf77 <program> -L$PGI_ROOT/lib -lacml

GNU: gfortran <program> -L$BLAS_ROOT/lib -lblas

SCAI
SuperComputing Applications and Innovation

Numerical libraries



Examples of linking LAPACK library

LAPACK

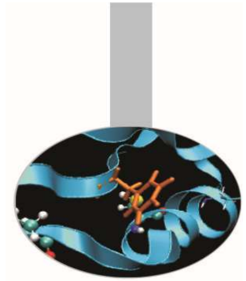
Intel: ifort <program> -L${MKLROOT}/lib/intel64 -lguide -lpthread –lmkl

icc -I${MKLROOT}/include <program> -L${MKLROOT}/lib/intel64 \

-lmkl_intel_lp64 -lmkl_core -lmkl_sequential -lpthread -lm

PGI: pgf77 <program> -llapack -lblas

GNU: gfortran <program> -L$LIB_ROOT/lib -llapack –lrefblas



Numerical libraries

Examples of linking ScaLAPACK library

ScaLAPACK

Intel: mpif77 <program> -L$MKLROOT/lib/intel64 -lmkl_scalapack_lp64 \
-lmkl_blacs_openmpi -lmkl_intel_lp64 -lmkl_intel_thread \
-lmkl_core -liomp5 -lpthread

PGI: pgf77 <program> –Mmpi=mpich –Mscalapack

GNU: gfortran -L$LIB_ROOT/lib -lscalapack

If Intel compilers are to be used one may refer to the page http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/ for linking MKL libraries