

Libraries for High-Performance Computing and other purposes

Overview

- ❑ Why use libraries?
- ❑ Where to search?
- ❑ Commercial libraries
- ❑ Special purpose libraries

Why use libraries?

- ❑ Why should we invent the wheel again and again and again ...?
- ❑ Let us (re-)use what others have developed over many years ...
- ❑ ... with
 - ❑ reliable results
 - ❑ good performance
 - ❑ a lot of manpower

Where to search?

- ❑ Search engines like Google
 - ❑ need to know the name quite exactly
 - ❑ or a good description of what you want to achieve
 - ❑ a lot of irrelevant hits
- ❑ A better approach:
 - ❑ go to well known websites that have collected information over years
 - ❑ “Trust the old people!”

Places to start a search



- ❑ <http://netlib.org>

- ❑ Netlib is a collection of mathematical software, papers, and databases.
- ❑ Netlib is mainly dedicated to
 - ❑ Linear Algebra routines
 - ❑ the work by Jack Dongarra and friends
- ❑ up-to-date?

Places to start a search

Guide to Available Mathematical Software

❑ <http://gams.nist.gov/>



- ❑ extensive catalog of mathematical software
- ❑ nice division into classes and subclasses of problems
- ❑ other sites use the GAMS taxonomy as well, e.g. Netlib

Places to start a search

Collected Algorithms (CALGO)



Association for
Computing Machinery

Advancing Computing as a Science & Profession

- ❑ <http://calgo.acm.org/>
- ❑ Algorithms published in ACM journals:
 - ❑ “This software is refereed for originality, accuracy, robustness, completeness, portability, and lasting value.”

Places to start a search

Software/hardware vendors:

- ❑ Intel: <http://software.intel.com/>
 - ❑ look for MKL (Math Kernel Library)
- ❑ AMD: <http://developer.amd.com/>
 - ❑ look for ACML (Advanced Core Math Library)

Trust an “old” man

Nick Trefethen

- ❑ <http://people.maths.ox.ac.uk/trefethen/>
- ❑ Professor (emeritus) of Numerical Analysis at Oxford University Computing Laboratory
- ❑ Collection of links:
 - ❑ <http://people.maths.ox.ac.uk/trefethen/tools.html>

Commercial Libraries

- ❑ Numerical Algorithms Group (NAG): <http://www.nag.co.uk/>
 - ❑ well established general purpose library
- ❑ IMSL: <http://www.roguewave.com/products-services/imsl-numerical-libraries>
 - ❑ well established general purpose library
- ❑ Harwell Subroutine Library (HSL): <http://www.hsl.rl.ac.uk/>
 - ❑ free versions for academic research and teaching

A free library

GNU Scientific Library (GSL)

- ❑ <http://www.gnu.org/software/gsl/>
 - ❑ *“A numerical library for C and C++ programmers. It is free software under the GNU General Public License.”*
- ❑ has a lot of features, but not always optimized
- ❑ ships with many Linux distros

Special purpose libraries

Linear Algebra

- ❑ LAPACK & BLAS

- ❑ from Netlib.org or vendor

- ❑ ATLAS

- ❑ Automatically Tuned Linear Algebra Software

- ❑ <http://math-atlas.sourceforge.net/>

- ❑ GotoBLAS: fast implementation of BLAS

- ❑ by Kazushige Goto, TACC (now: Intel)

- ❑ now: OpenBLAS <http://www.openblas.net/>

Special purpose libraries

Linear Algebra

- ❑ ARPACK: library for large and sparse eigenvalue problems
 - ❑ <https://www.arpack.org/>
 - ❑ ARPACK++: C++ interface to ARPACK
- ❑ MatrixMarket:
 - ❑ <http://math.nist.gov/MatrixMarket/>
 - ❑ collection of tools and data sets

Special purpose libraries

Fast Fourier Transforms

- ❑ FFTW: <http://www.fftw.org/>
 - ❑ Fastest Fourier Transform in the West
 - ❑ “*FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data.*”
 - ❑ state-of-the-art FFT library
 - ❑ also in parallel (MPI and OpenMP)

Special purpose libraries

Parallel Solvers:

- ❑ PETSc: Portable, Extensible Toolkit for Scientific Computation
 - ❑ <https://petsc.org/release/>
- ❑ MUMPS: MULTifrontal Massively Parallel sparse direct Solver
 - ❑ <https://mumps-solver.org/index.php>

Good advice

If in doubt, don't hesitate to ask an expert, e.g. in your local Scientific Computing/Numerical Analysis group or some researchers working with similar problems