LAPACK 3, LAPACK 95, and Fortran 90 linkage issues

Ed Anderson
Lockheed Martin Services Inc.
Anderson.Edward@epa.gov
May 22, 2001

What is LAPACK?

LAPACK is a collection of Fortran subroutines for solving linear systems, linear least squares problems, and matrix eigenvalue problems.

Since the first web counters were installed, LAPACK has been the most accessed link on netlib (www.netlib.org):

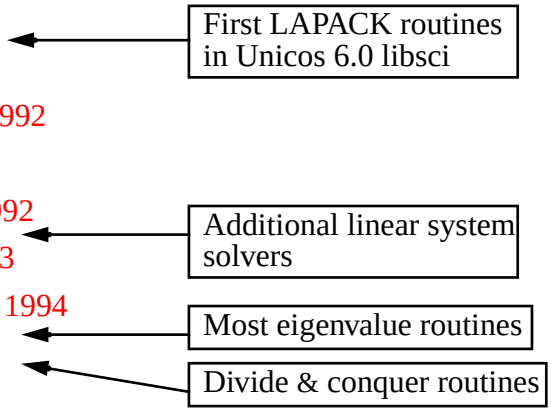
Library name	Number of accesses
lapack	9,521,387
lapack/lug	5,045,459
pvm3	4,471,797
scalapack	4,077,517
slatec	2,765,538

LAPACK Users' Guide online: **`http://www.netlib.org/lapack/lug/`**

LAPACK and libsci

LAPACK releases

Beta-1	April, 1989
Beta-2	March, 1990
Beta-3	August, 1991
Version 1.0	February 29, 1992
Version 1.0a	June 30, 1992
Version 1.0b	October 31, 1992
Version 1.1	March 31, 1993
Version 2.0	September 30, 1994
Version 3.0	June 30, 1999
Version 3.0 (update)	October 31, 1999
Version 3.0 (update)	May 31, 2000



libsci improvements to LAPACK

Solve right hand sides one at a time if few in number (xxxTRS)

up to 8X faster

Use vectorizable code for scaled sum of squares (xLASSQ)

up to 40X faster

Remove unnecessary code from balancing routines (xGEBAL)

up to 4X faster

Inline Level 1 BLAS, avoid unnecessary scaling in inverse iteration (xSTEIN)

up to 3X faster

Plus many other small improvements up to 2X

Performance Improvements to LAPACK for the Cray Scientific Library, (A. and Fahey), LAPACK Working Note 126, April 1997.

What's new in LAPACK 3

- faster SVD using divide-and-conquer (xGESDD)
- faster routines for solving rank-deficient least squares problems
  - using QR with column pivoting (xGELSY)
  - using SVD with divide-and-conquer (xGELSD)
- new routines for generalized symmetric eigenproblem
  - faster routines using divide-and-conquer (SSYGVD, CHEGVD, etc.)
  - routines based on bisection/inverse iteration (SSYGVX, CHEGVX, etc.)
- faster routines for the symmetric eigenproblem using the "relatively robust eigenvector algorithm" (xSTEGR, SSYEVR/CHEEVR)
- new drivers for the generalized nonsymmetric eigenproblem (xGGES, xGGESX, xGGEV, xGGEVX)
- solver for generalized Sylvester equation (xTGSYL)
- blocked version of xTZRQF (xTZRZF, plus SORMRZ/CUNMRZ)
- 79 new man pages

LAPACK 3 supplement to libsci

- Includes all new LAPACK 3 routines
- Replaces some libsci routines where necessary
  - Fix bugs
  - Compute workspace in WORK(1) (needed by LAPACK 90)
- Defines loader mappings
  - Map SLAXYZ to SLAXYZ@ in libsci
  - Map DUVXYZ to SUVXYZ for use with `-dp`
- Includes "lapack3" module
- Tested on CRAY C90 and CRAY T3E
- Not supported or endorsed by Cray or LAPACK group

Installing LAPACK 3 on CRAY machines, Dec. 1999,
`http://www.cs.utk.edu/~eanderso/lapack3.html`

Porting notes

Some new LAPACK software is biased towards IEEE arithmetic.

- xSTEGR (skipped by driver routines if non-IEEE)
- SLASQ1 -- SLASQ6 (new interface can't be mapped to libsci versions)

Workspace queries for LWORK = -1 were incompletely implemented.

- LWORK = -1 must be recognized as a workspace query
- If a query, calculate workspace and return amount needed in WORK(1)
- Workspace calculations (using upper bounds) added to xGEESX, xGGESX

Bugs in LAPACK or libsci LAPACK have been reported.

- xHGEQZ: one rotation is incorrectly applied
- xSTEBZ: FUDGE factor needs to be a little bigger
- xLASSQ: LAPACK may fail to scale small x(i), libsci may overflow if SUMSQ is large and x(i) is small
- xSTEIN: libsci version uses CGS orthogonalization instead of MGS

Renaming LAPACK routines

LAPACK auxiliary routines that are not documented in libsci have only an internal entry point ending in @ (for example, SLASWP@).

Applications ported from other platforms may be written in double precision and use double precision names (such as DGEMM).

Both problems can be resolved by the loader:

- C90: **`f90 -dp -Wl"-Dequiv=SGEMM(DGEMM)" dcode.f`**
- T3E: **`f90 -dp -Wl"-Dequiv(DGEMM)=SGEMM" dcode.f`**

The LAPACK 3 supplement to libsci provides files of loader directives:

`blasdp2sp.segldr`	**`blasdp2sp.cld`**
`lapackaux.segldr`	**`lapackaux.cld`**
`lapackdp2sp.segldr`	**`lapackdp2sp.cld`**

Then the command is

`f90 -dp -Wl"<directives_file>" dcode.f`

Enhancements to LAPACK

xLARTG (Generate Givens Rotations): Given f, g, compute c and s such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \qquad c^2 + s^2 = 1$$

- libsci version is compatible with BLAS SROTG
- Both LAPACK and libsci versions are discontinuous in r
- New version is only available in the LAPACK 3 supplement (see *Discontinuous Plane Rotations and the Symmetric Eigenvalue Problem*, LAPACK Working Note 150, Dec. 2000)

xGEBAL (Balancing for the nonsymmetric eigenvalue problem)

- LAPACK and libsci scale by factors of 10
- New version scales by factors of 8 for better accuracy
- With additional inlining, version in supplement is actually faster

Test suite improvements

- Replace test code for xGEBAL/xGEBAK (in progress)
- Add tests of reduction routines and their associated orthogonal transformations
- Print the header if the number of M and N values is zero and THRESH is also 0 (to see what tests were performed)
- Substitute Level 3 BLAS/LAPACK calls for Level 2
- Clean up formatting of error tests

Installing the libsci LAPACK 3 supplement

1. Copy the **`lapack.tgz`** file from netlib and the latest patch file from **`http://www.cs.utk.edu/~eanderso/lapack3.html`**
2. Follow the instructions for untarring and building the package.
3. Install it somewhere like **`/usr/local/lib/LAPACK.`**
4. Put the “lapack3” module in **`/opt/modulefiles.`**
5. Load the “lapack3” module and compile with
   `f90 -llapack3 ...`

Loading the lapack3 module performs the following actions:

Append **`/usr/local/lib/LAPACK/man`** to MANPATH

Append **`/usr/local/lib/LAPACK/lib`** to LD_LIBRARY_PATH_C90

LAPACK 95

LAPACK95 is a collection of Fortran 90 wrappers for LAPACK.

LAPACK95 also contains interface blocks for both the Fortran 77 and new Fortran 90 calling sequences.

F77_LAPACK:

- Leading S, D, C, or Z replaced by LA_
- All other arguments are the same

F90_LAPACK:

- Leading S, D, C, or Z replaced by LA_
- Option arguments are omitted if they can be inferred
- No integer arguments specifying array sizes or dimensions
- No workspace arguments (workspace is dynamically allocated)
- Many arguments are optional

Sample usage

Example: SGEEV is a driver routine to find the eigenvalues and, optionally, eigenvectors of a real matrix A.

F77 interface:

```
USE F77_LAPACK
CALL LA_GEEV( 'N', 'V', N, A, LDA, WR, WI, VL, LDVL, &
              VR, LDVR, WORK, LWORK, INFO )
```

F90 interface:

```
USE F90_LAPACK
CALL LA_GEEV( A, WR, WI, VR=VR, INFO=INFO )
```

In both interfaces, the type of data (real or complex) and its kind (32-bit or 64-bit) is inferred from the type of the input arguments.

Installing LAPACK 95 on CRAY machines

For some useful patches, see

Installing LAPACK 90/95 on CRAY machines, Sept. 2000,
`http://www.cs.utk.edu/~eanderso/lapack90.html`

User instructions at NESC:

1. Load the LAPACK modules:

   `module load lapack3 lapack95`

2. When compiling, use the f90 -p option to specify a location for the precompiled LAPACK95 modules:

   `f90 -p ${LAPACK95LIB}/liblapack95.a -c prog.f90`

3. Link the local LAPACK 3 supplement to libsci when loading:

   `f90 -llapack3 prog.o`

Fortran 90 linkage issues

1. Fortran 90 modules, like “include” files in C, are needed at compile time.

   Need an environment variable for use with Modules.

2. The location of included modules is hard-coded into the compiled object files.

   This is probably not necessary.

3. Resizing options are a problem everywhere.

   If the compiler can detect different options, it should be able to combine them into one object file.

Issue #1: no MODULE_SEARCH_PATH

At NESC, the LAPACK 95 library is in

```
/usr/local/lib/LAPACK/lib/liblapack95.a
```

or

```
/usr/local/lib/LAPACK/lib/3.4/liblapack95.a
```

To link to it, users must load the “lapack95” or “lapack95.3.4” module and use:

```
f90 -p ${LAPACK95LIB}/liblapack95.a mycode.f
```

My complaint(s):

- There is no MODULE_PATH that can be set in the modulefile.
- There is no shorthand (like `-llapack95`) to indicate that the module site is an archive file.
- Once the module is pulled in, there is no need to specify the library at link time, so LD_LIBRARY_PATH is ignored.

Issue #2: non-relocatable modules

The compiler records the location of Fortran modules as specified via the -p option in the object file. This complicates the building of libraries.

dir2/mline.f90

```
module mline
   type point
      real :: x = 0.0
      real :: y = 0.0
   end type point
   type line
     type(point) :: t1, t2
   end type line
end module mline
```

dir1/mplane.f90

```
module mplane
   use mline
   type plane
      type(line) :: g1, g2
   end type plane
end module mplane
```

./plp.f90

```
program plp
   use mplane
   type(plane) :: p1
   p1 = plane(line(point(1.,2.),point(3.,4.)), &
              line(point(5.,6.),point(7.,8.)))
end
```


non-relocatable modules, cont.

What happens if we compile the modules and then move them?

```
% cd dir2;      f90 -c mline.f90
% cd ../dir1;   f90 -p ../dir2 -c mplane.f90
% cd ..;        mv dir1/mplane.o dir2/mline.o dir3
% f90 -p dir3 -c plp.f90

% f90 -o plp plp.o dir3/mplane.o dir3/mline.o
cld-412 cld: WARNING
  The USE module `MLINE', referenced from relocatable
  object `dir3/mplane.o:MPLANE', was defined in file
  `./../dir2/mline.o' but the file was not found.
cld-431 cld: WARNING
  The resulting output file `plp' is not executable
  because of previous WARNING messages.
```

A workaround is to create a symbolic link to `mline.o` in the local directory before compiling `mplane.o` (since `mplane.o` and `mline.o` end up together).

Issue #3: data resizing by the compiler

When porting from IBM to Cray, a popular compile option is

```
f90 -dp foo.f
```

When porting from Cray to IBM, you may want

```
xlf90 -qrealsize=8 -qintsize=8 foo.f
```

Not all combinations are detected by the loader

- Cray loader detects `-dp` and won't combine dp and non-dp modules.
- IBM loader detects `-q64`, but not other resizing options.

It would be nice to have a library compilation mode that includes all sizing combinations in one object file.

Last slide

You need newer LAPACK software than is in libsci if you want

- LAPACK 3 compatibility
- LAPACK 95

LAPACK continues to evolve.

- See `http://www.netlib.org/lapack/release_notes.html`
- There may be an LAPACK 4.

Fortran modules feature requests:

- INCLUDE_PATH for modules
- Unified library for all possible data resizing options