LAPACK 3, LAPACK 95, and Fortran 90 linkage issues

Edward Anderson, Lockheed Martin Services Inc.,
Anderson.Edward@epa.gov

ABSTRACT: *The Cray Scientific Library (libsci) implementation of the linear algebra package LAPACK differs from the standard netlib version in several important and occasionally annoying ways. The latest LAPACK releases, LAPACK 3 and LAPACK 95, include new subroutines not in libsci, new tests that better exercise the existing software, and new Fortran 90 interfaces that allow dynamic memory allocation of scratch space. On the other hand, libsci features performance enhancements to parts of LAPACK and years of software engineering that have filtered out some IEEE arithmetic bias in the original code. This paper presents highlights of a publically available LAPACK 3 supplement to libsci that attempts to strike a balance between the new and the old. It also introduces LAPACK 95, a Fortran 90 interface to LAPACK, and presents several open issues related to the linking of Fortran 90 modules.*

1 What is LAPACK?

LAPACK [4] is a collection of Fortran subroutines for solving linear systems, linear least squares problems, and matrix eigenvalue problems. It has its roots in the Level 2 and 3 BLAS (Basic Linear Algebra Subprograms [9,10]), which are sets of Fortran kernels for performing matrix-vector and matrix-matrix operations. While originally targeted at high-end vector machines like the early Crays, the BLAS and LAPACK also have proved to be well-suited to cache-based microprocessors. Since its public introduction in 1992, LAPACK has undergone several updates and has contributed to other numerical software packages such as ScaLAPACK [6] and ARPACK [11]. With 9.5 million hits as of April 2001 (and another 5 million to the Users' Guide), LAPACK is the most requested package on netlib [7].

2 LAPACK in libsci

The Cray Scientific Library [8] was one of the earliest adopters of LAPACK. The first LAPACK routines actually appeared in libsci in 1991, using interfaces from one of the widely circulated beta releases of LAPACK software. Most of the linear system solvers and least squares routines were added soon after LAPACK's public release in 1992, and the eigensystem routines that were current as of LAPACK version 2.0 (released in late 1994) appeared in 1995 and 1996.

The Cray Scientific Library implementation of LAPACK is unique for its many performance improvements [5]. These include

- Solving for right-hand sides one at a time (xxxTRS) if they are few in number -- *up to 8X faster*
- Using vectorizable code for the scaled sum of squares (xLASSQ) -- *up to 40X faster*
- Removing unnecessary code from the balancing routines (xGEBAL) -- *up to 4X faster*
- Inlining Level 1 BLAS and avoiding unnecessary scaling in inverse iteration (xSTEIN) -- *up to 3X faster*

Many other small improvements have led to lesser speedups, up to a factor of 2. Some of the Cray-specific enhancements have been adopted in subsequent LAPACK releases and are not mentioned here.

3 What's new in LAPACK 3

The first major update to LAPACK in nearly five years occurred with the release of LAPACK version 3.0 in July, 1999. The most significant new routines in LAPACK 3 are:

- faster singular value decomposition (SVD) computed by a divide-and-conquer algorithm (xGESDD)
- faster routines for solving rank-deficient least squares problems:
  - using QR with column pivoting (xGELSY)
  - using the SVD based on divide-and-conquer (xGELSD)
- new routines for the generalized symmetric eigenproblem:
  - faster routines based on divide-and-conquer (xHEGVD/xSYGVD, xHPGVD/xSPGVD, xHBGVD/xSBGVD)
  - routines based on bisection/inverse iteration, for computing part of the spectrum (xHEGVX/xSYGVX, xHPGVX/xSPGVX, xHBGVX/xSBGVX)
- new routines for the symmetric eigenproblem using the "relatively robust eigenvector algorithm" (xSTEGR, CHEEVR/SSYEVR, SSTEVR)
- new drivers for the generalized nonsymmetric eigenproblem (xGGES, xGGESX, xGGEV, xGGEVX)

- solver for generalized Sylvester equation (xTGSYL) and some associated computational routines (xTGEXC, xTGSEN, xTGSNA)
- a blocked version of xTZRQF (xTZRZF), and associated xORMRZ/xUNMRZ

The update also includes some 79 new man pages.

4 LAPACK 3 supplement to libsci

In order to provide the functionality of LAPACK 3 without losing the performance improvements of libsci, I have created an LAPACK 3 supplement to libsci [1]. The supplement takes the form of another library that should be linked before libsci at compile time. It includes all the subroutines new to LAPACK 3 and replaces some existing libsci routines where necessary to fix bugs or to add workspace calculations. The update package also includes files of loader directives, to assist with mapping double precision names to single precision, and an "lapack3" module to aid its installation under modules. The LAPACK 3 supplement to libsci has been tested on a CRAY C90 and a CRAY T3E at the U. S. EPA's High-Performance Computing Center, but it is not supported or endorsed by Cray, Lockheed Martin, EPA, or the LAPACK group.

4.1 Porting notes

Included among the new routines in LAPACK 3 are some documented as working only on systems with IEEE arithmetic that propagate NaNs and infinities, a definition that excludes both CRAY PVP and CRAY MPP systems. These subroutines have not been fully tested on the Crays, but they are included in the supplement for completeness and should be regarded as research codes for experimentation and further study. In particular, the "relatively robust" eigenvalue routine xSTEGR relies on IEEE arithmetic, and calls to it are skipped in the driver routines on the Crays.

A new feature of LAPACK 3 that was not exploited until LAPACK 95 is the ability to query a routine to determine how much workspace it requires. In certain subroutines that contain both an array argument WORK (sometimes called RWORK or IWORK) and an integer argument LWORK indicating its length, the previously invalid value of LWORK = -1 is now recognized on input as a workspace query. During a workspace query, the LAPACK routine performs no work, but only computes the required amount of workspace and returns it in WORK(1). This feature allows the calling routine to dynamically allocate exactly as much workspace as it needs, instead of statically reserving the maximum amount of workspace it may need. Because the workspace queries were not exercised in the LAPACK 3 test software, a number of bugs were not detected until the advent of LAPACK 95. The supplement fixes the workspace calculations introduced in LAPACK 3 and extends them to include the subroutines xGEESX and xGGESX using upper bounds on the workspace requirements.

Porting LAPACK 3 and LAPACK 95 to the Crays also uncovered some bugs in LAPACK or the libsci versions of LAPACK routines, which are fixed in the supplement. They include

- xHGEQZ: one rotation is incorrectly applied
- xSTEBZ: the FUDGE factor needs to be a little bigger for T3E
- xLASSQ: LAPACK may fail to scale small x(i), and libsci may overflow if SUMSQ is large and x(i) is small
- xSTEIN: the libsci version inadvertantly switched to Classical Gram-Schmidt orthogonalization instead of the more well-behaved Modifed Gram-Schmidt orthogonalization in the coincident eigenvalue case

The continuously updated LAPACK release notes (`http://www.netlib.org/lapack/release_notes.html`) may include other bug fixes not in the libsci supplement.

4.2 Renaming LAPACK routines in libsci

In the Cray Scientific Library version of LAPACK, auxiliary routines that were not thought to be of general interest were renamed to <LAPACK_NAME>@ in order to make their entry points internal to the library. These internal entry points were then optimized by removing argument checking and occasionally making algorithmic improvments [5]. However, some users have complained about the inconvenience of not being able to call auxiliary routines by their LAPACK names. A related problem is that Cray Scientific Library routines that operate on 64-bit arrays have names that begin with S or C, while library routines on other systems require double precision to use 64-bit arithmetic, and so may call subroutines with names beginning with D or Z.

Both problems can be solved by using loader options to map calls from one name to another. For example, a program that declares arrays to be DOUBLE PRECISION and calls DGEMM could be compiled with double precision disabled (f90 -dp) and instructed to replace calls to DGEMM with calls to SGEMM, all without modifying the code. The form of the loader mappings is

PVP (segldr):

```
f90 -dp -Wl"-Dequiv=SGEMM(DGEMM)" dcode.f
```

T3E (cld):

```
f90 -dp -Wl"-Dequiv(DGEMM)=SGEMM" dcode.f
```

Because there are so many such mappings, it is convenient to put them in a file and pass the name of the file to the loader via a f90 command line option, for example,

```
f90 -Wl"blasdp2sp.cld".
```

The following directives files included in the supplement provide some of the most commonly used mappings:

PVP:

`blasdp2sp.segldr` -- map double precision BLAS names to single

`lapackaux.segldr` -- map LAPACK@ names to regular LAPACK names

`lapackdp2sp.segldr` -- map double precision LAPACK names to single (includes BLAS and LAPACK auxiliary routine mappings)

T3E:

`blasdp2sp.cld`

`lapackaux.cld`

`lapackdp2sp.cld`

4.3 Enhancements to LAPACK

Most of the algorithmic enhancements to LAPACK in libsci or in the LAPACK 3 supplement to libsci have been done for performance reasons, but a few are concerned with numerical behavior. Two recent such changes will be discussed here, involving the auxiliary routines xLARTG, which generate Givens rotations, and the balancing routines xGEBAL.

The problem of generating Givens rotations accurately and efficiently is a subject in itself [3]. Most of the existing software to compute these simple transformations, including the versions in the BLAS, EISPACK, LAPACK, and libsci, use different formulas depending on the relative magnitudes of the inputs. These differences create lines of discontinuity in the function domain which makes the algorithm sensitive to small perturbations in the input data or small errors made in the course of the computation. The discontinuities can be removed by choosing the signs of the computed cosine and sine to always make the output value r positive. In the real case, the new version of SLARTG can be described by

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

while in the complex case, where c is real but s and r are complex, the new version of CLARTG can be described by

$$\begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad c^2 + s \cdot \bar{s} = 1, \quad Re(r) \geq 0$$

where $\bar{s}$ is the conjugate of s. The constraints on r are new and are sufficient to resolve the signs of c and s in a way that makes r continuous.

Subroutines for balancing (xGEBAL in LAPACK) are applied to general nonsymmetric matrices in order to isolate eigenvalues and make the magnitudes of elements in corresponding rows and columns nearly equal. This can reduce the 1-norm of a matrix and improve the accuracy of its computed eigenvalues and/or eigenvectors. The balancing step is applied iteratively, scaling the rows and columns up or down by some factor until they are nearly equal. In EISPACK, a scaling factor of 2 was used to avoid introducing any rounding errors, but in LAPACK the factor was increased to 10 to achieve faster convergence. A recent proposal suggests using a factor of 8 to again avoid rounding errors. This proposal is implemented in the LAPACK 3 supplement to libsci, along with enough additional inlining to compensate for the performance degradation due to scaling by a smaller factor.

5 LAPACK 95

LAPACK 95 [12] is a collection of Fortran 90 wrappers for the Fortran 77-style interfaces in LAPACK. It also contains interface blocks for both the Fortran 77 and new Fortran 90 calling sequences, defined in modules. Both the F77_LAPACK and the F90_LAPACK interfaces support a generic subroutine name, in which the leading S, D, C, or Z is replaced by the prefix "LA_" and the specific routine to call is determined from the type of the array arguments at compile time. In the F77_LAPACK interface, the list of arguments is the same as in LAPACK, but the F90_LAPACK interface compacts the argument list as much as possible using the following conventions:

- Option arguments are omitted if they can be inferred from the presence or absence of other arguments
- Integer arguments specifying array sizes or dimensions are omitted
- Workspace arguments are omitted (all workspace is dynamically allocated)
- Many arguments are optional

For example, the driver routine SGEEV computes the eigenvalues and, optionally, eigenvectors of a matrix A. In the Fortran 77 interface, both the array of left eigenvectors and the array of right eigenvectors must be provided, and option arguments tell whether or not the eigenvectors are to be computed. In the Fortran 90 interface, the arrays of eigenvectors are provided only if they are desired. To illustrate, the calls to compute the eigenvalues and only the right eigenvectors of a real matrix A would be as follows:

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 )
```

The syntax for "positional arguments" is required for VR in the F90 interface because if only one eigenvector array were present, the compiler would assume it was the first optional array, which is VL.

Tips for installing LAPACK 95 on a CRAY machine are available in a separate report [2].

6 Fortran 90 linkage issues

While Fortran 90 itself is hardly new, implementation-dependent features such as how Fortran modules are linked with program units that use them are still being sorted out. Some issues that arose in porting the latest LAPACK software to the Crays are as follows:

1) Fortran 90 modules, like include files in C, are needed at compile time. The Cray solution is to provide an option "-p <pathname>" to tell the compiler where to find the pre-compiled module files. However, there is no corresponding environment variable that can be set to indicate a Fortran module search path. Other vendors use the -I option for both C include files and Fortran module files, for which the INCLUDE_PATH environment variable already exists. If the -p option is to be the Cray standard, then a MODULE_PATH environment variable is needed.
2) Compiled module files need to be relocatable. This is particularly important for libraries, which may document one module interface to their users while using several other modules internally in a nested fashion. These internal modules may be compiled in one location but installed or archived somewhere else, and the original location, whether absolute or relative, may not be accessible to the end users. Currently, the location of any module files specified via the -p option is hard-coded into the object file that uses it, and this must be circumvented to make the files relocatable.
3) Resizing options are a problem everywhere. Program units compiled with different data size options, such as -dp on the Crays or any combination of -q64, -qrealsize, or -qintsize on an IBM, are typically incompatible with each other, and the compiler may not be able to detect these differences. Ideally, the compiler would flag all such resizing options separately. Then it might be possible to generate one super object file for use in libraries that contains all possible combinations of data sizing options, instead of compiling a different library for each set of options as is the case today.

7 Summary

Every software developer knows that there is a conflict between change and stability. The success of LAPACK has resulted in ongoing funding and continual additions, revisions, and improvements, which make it difficult for vendor libraries to remain compatible. The solution proposed here for Cray platforms is an LAPACK supplement to libsci, which adds in the new interfaces without overwriting all of the old, well-tested ones. The advantages of this approach are its timeliness -- bug fixes may appear in the supplement before they appear in an LAPACK update -- and its targeting of the Cray programming environment, so that the package can be built without even changing the Makefiles. The disadvantages are its complete lack of support and the tendency of the author to make more than the minimal changes needed for correctness. Feedback on the usefulness of this LAPACK supplement would be welcome.

References

[1] E. Anderson, *Installing LAPACK 3 on CRAY Machines*, online technical report, Dec. 1999. (`http://www.cs.utk.edu/~eanderso/lapack3.html`)

[2] E. Anderson, *Installing LAPACK 90/95 on CRAY Machines*, online technical report, Sept. 2000. (`http://www.cs.utk.edu/~eanderso/lapack95.html`)

[3] E. Anderson, *Discontinuous Plane Rotations and the Symmetric Eigenvalue Problem*, LAPACK Working Note 150, University of Tennessee, CS-00-454, Dec. 2000.

[4] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide, Third Edition*, SIAM, Philadelphia, 1999.

[5] E. Anderson and M. Fahey, *Performance Improvements to LAPACK for the Cray Scientific Library*, LAPACK Working Note 126, University of Tennessee, CS-97-359, April 1997.

[6] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*, SIAM, Philadelphia, 1997.

[7] S. Browne, J. Dongarra, E. Grosse, and T. Rowan, *The Netlib Mathematical Software Repository*, D-Lib Magazine, September 1995. (`http://www.netlib.org/srwn/srwn21.html`)

[8] Cray Research Inc., *Scientific Libraries Reference Manual (SR-2081)*, 1997.

[9] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. Duff, *A Set of Level 3 Basic Linear Algebra Subprograms*, ACM Trans. Math. Soft., 16(1):1-17, March 1990.

[10] J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson, *An Extended Set of FORTRAN Basic Linear Algebra Subprograms*, ACM Trans. Math. Soft., 14(1):1-17, March 1988.

[11] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide*, SIAM, Philadelphia, 1998.

[12] J. Wasniewski and J. Dongarra, *High Performance Linear Algebra Package -- LAPACK90*, LAPACK Working Note 134, University of Tennessee, CS-98-383, March 1998.