.

LAPACK Working Note #16

Results from the Initial Release of LAPACK *

* This work was supported in part by the
National Science Foundation under grant no. NSF ASC-8715728.

Edward Anderson and Jack Dongarra

Abstract

This report details our results and experiences from the first round of testing of LAPACK. A list of the known bugs in LAPACK and the BLAS is provided, all of which will be corrected in the next release. Selected timing results from the test sites are also presented to give some indication of the performance expected from LAPACK.

The initial release of software from LAPACK was sent to test sites in April-May 1989. This release included routines for solving systems of equations for general matrices, positive definite matrices, and symmetric or Hermitian indefinite matrices in general, packed, or banded storage. Also included were routines to perform the QR factorization of a general matrix, numerous auxiliary routines, a test package, and a timing package[1, 3]. These routines represent only a fraction of the software that will be a part of LAPACK. For more information on the scope and purpose of the LAPACK project, see [2]. Subsequent releases of LAPACK will also include corrections of the errors uncovered during this first round of testing and reported in this paper. We received complete sets of data in one or more precisions from 25 test sites covering 22 different models of computers. Several test sites reported results from more than one machine. In addition, a few test sites examined the code in more detail and suggested improvements, which we have incorporated into the code for the next release. We are grateful to all the people and organizations who contributed to the first round of testing. With apologies to anyone we may have missed, here is a list of the people who sent us their results:

Organization	Name	Computer
Alliant Computer Systems	Martin Lewitt	Alliant FX/80
AT&T	Linda Kaufman	Vax 8550
CSRD, Champaign-Urbana	Mike Berry	Alliant FX/80
CERFACS	Michel Dayde	CRAY 2 ETA-10P Alliant FX/80
CERN	Federico Carminati	Vax 8800 IBM 3090
Convex Computer Corp.	Dave Dodson	Convex C210

Cornell University	Adolfy Hoisie	IBM 3090-600E/VF
Cray Research	Phuong Vu Sandy Carney	CRAY 2-S CRAY Y-MP/8
CWI, Amsterdam	Dik Winter Margreet Louter-Nool Walter Lioen	CDC CYBER 205 CDC CYBER 995 Alliant FX-4 IBM 3090 NEC SX2
FPS Computing	Robert Larson	FPS Model 511
Harwell Laboratory	S. Marlow	CRAY 2
HNSX Supercomputers	Takayuki Sasakura	NEC SX2-400
IBM ECSEC, Rome	Giuseppe Radicati	IBM 3090-600E/VF
IMSL	Richard Lehoucq Jing Li	CRAY 2/418 CRAY Y-MP/832 CRAY X-MP-48 NEC-SX2A IBM 3090-600E/VF
University of Karlsruhe	Klaus Geers	Siemans/Fujitsu VP 400-EX
University of Kentucky	Anne Leigh	IBM 3090-300E/VF
MasPar Computer Corp.	Kenneth Jacobsen	Vax 3100
University of Michigan	Len Harding	IBM 3090/VF
NASA Ames Research Center	King Lee Horst Simon David Bailey	CRAY Y-MP/8
NCAR, Boulder	Adrianne Link	CRAY X-MP-18
NCSA, Champaign-Urbana	Beth Richardson	CRAY X-MP/48
Pittsburgh Supercomputer Center	John Burkardt	CRAY Y-MP 8/32
Rice University	Michael Pearlman	MIPS M/120-5 Sun 3/180
San Diego Supercomputer Center	Bob Leary	CRAY X-MP/48
Purdue University	Brad Lucier	Ardent Titan
Rutgers University	Sam Yu Dan Kowalski	BBN Butterfly GP1000
Stellar Computer	Charles Valentine	Stellar GS1000
Sun Microsystems	Shing Ma	Sun SPARCstation 1
SCRI, Florida State University	Sy-Shin Lo	ETA-10G
University of Toronto	Edgar Smart	Sun 4
Yale University	Stan Eisenstat	Celerity 1260D Multiflow Trace 14/200

To achieve high performance, LAPACK requires an efficient implementation of the Level 1, 2, and 3 BLAS. Many people used the Fortran BLAS without modification and observed poor performance from LAPACK as a result. We emphasize that the Fortran BLAS available from *netlib* and included on the first release tape provide definitions of the BLAS operations but represent a generic implementation for any computer environment; hence, they may not be very efficient. In particular, many of the loops in the Fortran BLAS routines are not optimized by vectorizing compilers; some minor modifications to the code, such as selectively interchanging inner and outer loops, would improve performance considerably. The first round of testing of LAPACK can be judged a success if it has helped encourage computer manufacturers and other interested researchers to work on optimizing the BLAS. An optimized library of these basic kernels would benefit not only the LAPACK library, but also anyone developing software in a similar style. Designing parallel programs around the BLAS gives the algorithm developer larger building blocks, thereby reducing development time and improving the clarity of the program. The larger granularity of tasks in the Level 2 and 3 BLAS also frees the programmer from much of the burden of managing the memory hierarchy in a high-performance computer. Since most of the computational work is transferred to a library routine, the high-level routine should provide nearly optimal performance, as well as portability to any other machine with a similar library of the BLAS. Cray Research is, to date, the only company that has implemented tuned versions of all of the Level 1, 2, and 3 BLAS in its scientific library. Other companies are following suit. Of those that returned results, Convex and Alliant have each implemented optimized versions of all of the Level 1 BLAS, most of the Level 2 BLAS, and at least the matrix-matrix multiply from the Level 3 BLAS. IBM's ESSL library currently includes the Level 1 BLAS and selected routines from the Level 2 and 3 BLAS. Our hope is that by the time LAPACK is released to the public in 1991, the BLAS will be implemented efficiently on a wide range of machines. The following list describes bugs that were discovered in the first release. A leading ‘‘x’’ in the routine name indicates that the error occurred in all four precisions. xGETRF and xGBTRF: These factorization routines incorrectly handled $M \times N$ rectangular matrices when $M \neq N$. The test package tested only square matrices. CSYTRF, ZSYTRF, CSPTRF, and ZSPTRF: The procedure used in the factorization routines for complex symmetric indefinite matrices could fail if one of the 2 x 2 diagonal blocks were not diagonalizable. A test of this case has been added, and alternative code has been supplied to perform the rank-2 update. xSBTRF, CHBTRF, and ZHBTRF: The factorization of a symmetric indefinite banded matrix could overflow the allocated amount of storage. Fill-in outside the band is not bounded when symmetric pivoting is used. The xSB and xHB paths will not be included in Release 2. R1MACH and D1MACH: The auxiliary routine R1MACH, a subroutine to determine machine parameters that called the single-precision version of MACHAR, returned incorrect results for CDC/ETA machines. On other machines, unusual behavior was observed if this subroutine was compiled with optimization. A new auxiliary routine, called SLAMCH (DLAMCH), is being developed to compute machine parameters. xGETRF: The loop DO 30 (apply interchanges to previous blocks) should be executed even if INFO = 0. Also, the index of the first (rather than the last) zero pivot should be returned. xPOTRS, xPPTRS, and xPBTRS: B should be of dimension (LDB, NRHS), and LDB must be at least max(1,N). xPBTRS, xPBCON: Error code 6 should occur if LDA < KD+1 (formerly tested that LDA < N). xPBTRS, xPOTRS, xPPTRI, CPPTRS, ZPPTRS: The second argument to XERBLA should be -INFO, instead of INFO. In xPBTRS, the first argument to XERBLA should be xPBTRS (was xPOTRS). CLARF and ZLARF: Line 133 should be changed from

```
CALL CSCAL( N, ONE-TAU, C, 1 )
```

to

```
CALL CSCAL( M, ONE-TAU, C, 1 )
```

xQRT1, xQRT2, xQRT3, and xQRT4: The

declaration of the array argument QWORK should be changed to avoid passing a zero leading dimension when M = 0. AUXOPS, BL2OPS, BL3OPS, LAOPS, and LAOPS2: Complex multiplication should count as 6 operations (4 multiplies, 2 adds) rather than 4. The megaflop rates for the complex routines were underestimated as a result. AUXOPS, BL2OPS, BL3OPS, LAOPS, and LAOPS2: The integer variable holding the operation count overflows for test matrices a little larger than those in the recommended test set. For example, the count for the complex Hermitian factorization routine CHETRF overflows for N = 1000. New REAL versions of the operation count routines have been created. LAOPS: The operation counts for the xxxCON routines were underestimated by a factor of about 4. xLACON is called twice for each iteration, and the result must be doubled because the timing routines do not pass in the norm of the original matrix A. LAOPS: The operation count for SLARFG was incorrect. It should be changed from

```
( N+3 )*MULFAC + 2*ADDFAC
```

to

```
(2*N+4)*MULFAC + (N+2)*ADDFAC
```

where a square root is counted as a multiply. AUXOPS: AUXOPS returns 0 when N = 0, but for SLARFT only M and NB matter. Hence

```
AUXOPS( 'SLARFT', M, 0, 0, 0, NB )
```

is legitimate, but returns (incorrectly) zero. IZAMAX: IZAMAX incorrectly returned index 1 for a vector of length 0. This bug was corrected in 1985 in IZAMAX from the BLAS (in *netlib*), but not in ZBLAS from the BLAS. (Timing program) The routines xTIM03, xTIM04, xTIM05, xTIM15, and xTIM16 in the Level 2 BLAS timing program declared the variable UPLO as REAL instead of CHARACTER. (Timing program) The main timing program for the Level 2 BLAS routines had TRANS and UPLO reversed in the format statements numbered 9980 to 9985. In this section, we report preliminary performance results from the first round of testing. Since the leading computer manufacturers were still developing tuned versions of the BLAS at the time of the first release, most of the timing results we received for vector/parallel computers did not measure the full potential of those machines. These numbers are likely to improve as the manufacturers continue to work on their implementation of the BLAS. Further improvements may also come from the higher-level routines as we continue to refine the LAPACK code. Here we give selected results from four computer manufacturers with tuned versions of at least a partial set of the BLAS: Cray, Convex, Alliant, and IBM. LAPACK results are given for four major matrix factorization routines:

SGETRF	LU factorization
SPOTRF	Cholesky factorization
SSYTRF	Bunch-Kaufman factorization (Level 2 BLAS version)
SGEQRF	QR factorization

Note that the version of SSYTRF in the first release was not a blocked routine. We have since implemented a Level 3 BLAS version which will be tested in the next release. BLAS results are given for the Level 2 BLAS routines

SGEMV	general matrix times a vector
SSYMV	symmetric matrix times a vector
STRMV	triangular matrix times a vector
STRSV	solution of a triangular system
SGER	rank-one update of a general matrix
SSYR	rank-one update of a symmetric matrix

and for their Level 3 BLAS equivalents, which multiply a matrix by a matrix, instead of a vector, or solve a system with multiple right-hand sides, or compute a rank-K update of a matrix. In some cases, the results for symmetric matrices differ significantly, depending on whether the upper or lower triangle of the matrix is stored; hence, two lines of results are listed, for example, SSYTRF(U) and SSYTRF(L). In choosing the numbers for these tables, we used the best blocksize for each matrix dimension. In most cases the blocksize that produced the optimum performance was 64; on the Alliant, however, a blocksize of 32 performed better. For the BLAS routines, we give the data for the first set of options in the output file; the performance may vary for a different set of values for UPLO and TRANS.

CRAY Y-MP, 1 processor

LAPACK rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
SGETRF	24	61	140	221	265
SPOTRF	29	84	170	240	272
SSYTRF(U)[†]	28	77	153	219	254
SSYTRF(L)[†]	21	53	111	181	225
SGEQRF	48	126	208	256	272

BLAS rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
SGEMV	175	252	280	289	292
SGEMM	254	284	289	291	292

CRAY 2-S (static memory), 1 processor

LAPACK rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
SGETRF	16	40	102	197	294
SPOTRF	23	71	166	270	331
SSYTRF(U)[†]	17	48	99	149	174
SSYTRF(L)[†]	14	36	77	130	160
SGEQRF	25	70	157	261	336

BLAS rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
SGEMV	153	271	330	349	356
SGEMM	324	417	439	448	454

[†] This routine is based on a Level 2 BLAS implementation.
[†] This routine is based on a Level 2 BLAS implementation.

Details: Unicos 5.0, cft77 3.0
All BLAS have been optimized.

Convex C210, 1 processor

LAPACK rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
SGETRF	6	12	21	30	36
SPOTRF	8	20	33	40	44
SSYTRF†	5	11	16	20	21
SGEQRF	12	21	27	33	38

BLAS rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
SGEMV	34	43	47	47	47
SSYMV	12	18	23	30	34
STRMV	16	26	35	40	42
STRSV	8	14	20	27	35
SGER	17	20	23	23	23
SSYR	11	17	19	21	22
SGEMM	38	44	47	47	47
SSYMM	6	10	14	18	21
STRMM	5	8	12	15	17
STRSM	4	7	11	14	16
SSYRK	6	9	13	16	17

Details: V5.1 of fc -O2
Optimized BLAS: all Level 1, SGEMV, SSYMV,
STRMV, STRSV, SGER, SSYR, SSYR2, and SGEMM

† This routine is based on a Level 2 BLAS implementation.

Alliant FX-80/8, 4 processors

LAPACK rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
DGETRF	1	3	8	17	26
DPOTRF(U)	2	5	12	22	33
DPOTRF(L)	2	5	11	18	20
DSYTRF†	2	5	7	8	7
DGEQRF	3	7	14	25	30

BLAS rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
DGEMV	6	11	13	12	12
DSYMV	5	9	12	12	13
DTRMV	1	2	7	10	12
DTRSV	1	2	4	5	6
DGER	7	12	15	10	8
DSYR	5	8	10	11	6
DGEMM	28	38	45	47	49
DSYMM	2	3	5	8	9
DTRMM	15	25	33	32	30
DTRSM	13	26	34	33	30
DSYRK	3	4	5	6	6

Details: 4 CEs in cluster, other 4 detached
Memory size 96 Mb, cache size 512K, fortran v. 4.1.35
BLAS from Linear Algebra Library, rev. 5.0 (alpha version)

† This routine is based on a Level 2 BLAS implementation.

IBM 3090E/VF, 1 processor

LAPACK rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
DGETRF	5	11	19	31	43
DPOTRF(U)	6	11	22	35	48
DPOTRF(L)	8	16	28	42	52
DSYTRF	4	7	12	16	19
DGEQRF	9	17	30	41	53

BLAS rates in Mflops

Name	Matrix size N				
	32	64	128	256	512
DGEMV	19	36	45	52	56
DSYMV	7	13	19	23	27
DTRMV	5	9	14	18	20
DTRSV	4	8	11	14	16
DGER	12	17	20	21	23
DSYR	6	10	14	18	20
DGEMM	37	56	73	74	75
DSYMM	6	12	18	23	26
DTRMM	6	10	15	18	21
DTRSM	12	23	41	49	55
DSYRK	19	41	59	65	71

Details: ESSL, Release 3
Optimized BLAS: all Level 1 (except DNRM2),
DGEMV, DGEMM, DTRSM, DSYRK

We may be somewhat guilty of optimizing the LAPACK code for the CRAYs, since we have benefited from liberal access to the CRAYs at Cray Research. Our colleague at the IBM ECSEC Center in Rome experimented with DGETRF and DPOTRF and was able to bring the rate for those routines more in line with the performance rates of the BLAS on the IBM 3090. With right-looking variants of LU and Cholesky and a different order of pivoting in LU, the rates for those two factorizations are as follows:

Name	Matrix size N				
	32	64	128	256	512
DGETRF	6	13	26	43	57
DPOTRF	8	17	32	50	62

We are currently evaluating left-looking, right-looking, and hybrid variants of these algorithms on the machines available to us. The right-looking variants appear promising for use on distributed-memory architectures. The people at the IBM ECSEC Center in Rome have also implemented and experimented with a version of a banded Cholesky factorization routine using the Level 3 BLAS. Their results are reported in [4]. Their blocked banded Cholesky factorization routine was included in the first test release as the LAPACK routine xPBTRF. The results reported from Cray Research in Section 5 were for a single processor of the CRAY Y-MP and CRAY 2. These machines now have multiprocessing capabilities, and the Mathematical Sciences Research Group at Cray Research has been working on autotasking the BLAS library. The following table shows the performance rates for 1, 2, 4, and 8 processors of a CRAY Y-MP for three blocked factorization algorithms from the first LAPACK release: SGETRF (LU factorization), SPOTRF (Cholesky factorization), and SGEQRF (QR factorization). The theoretical peak performance of the 8-processor Y-MP is 2667 Mflops. By simply relinking to the appropriate version of the parallel BLAS library, close to this peak performance was achieved on the multiprocessor; no modifications were required in the LAPACK codes.

LAPACK timing results for a CRAY Y-MP/832 (Mflops)

Name	Matrix size N					
	32	64	128	256	512	1024
SGETRF (1 proc)	40	108	195	260	290	304
(2 proc)	32	91	229	408	532	588
(4 proc)	32	90	260	588	914	1097
(8 proc)	32	90	268	736	1450	1980
SPOTRF (1 proc)	34	95	188	259	289	301
(2 proc)	29	84	221	410	539	594
(4 proc)	29	84	252	598	952	1129
(8 proc)	29	84	273	779	1592	2115
SGEQRF (1 proc)	54	139	225	275	294	301
(2 proc)	50	134	256	391	505	562
(4 proc)	50	136	292	612	891	1060
(8 proc)	50	133	328	807	1476	1937

We have not reported results from several other promising machines for which the BLAS were not sufficiently well developed for the LAPACK routines to perform well. Nevertheless, these machines did post some impressive numbers for individual BLAS routines. The Siemans/Fujitsu VP 400-EX at Karlsruhe, using some locally optimized Level 2 and 3 BLAS, attained over 1100 Mflops for certain options of both SGEMV and SGEMM for a matrix of order 512, out of a theoretical peak of 1700 Mflops. The NEC SX2 computers, using the Fortran definitions of the BLAS, reported 570 Mflops for one type of call to SGEMV and 580 Mflops for certain calls to SGEMM and SGER for a matrix of order 512, out of a theoretical peak of 1300 Mflops. IMSL's versions of DGEMV and DGEMM for the NEC SX2 improved these rates to 945 and 983 Mflops for a matrix of order 500. The ETA 10G at the Supercomputing Computational Research Institute at Florida State, also using the Fortran BLAS, ran at just over 300 Mflops for certain options of SGEMV, SGEMM, and others of the Level 3 BLAS for matrices of order 512, out of a theoretical peak of 644 Mflops. The level of response to the initial release of LAPACK was excellent, and our respondents form a good mix of manufacturers, individuals, and research centers. We identified a number of program bugs and also several compiler bugs. The large number of test sites has given us a better idea of the type of computers people are using, which will help us in choosing the problem sizes for testing and timing. The preliminary results have also helped us understand what we need to analyze the timings and compare different variants of some of the algorithms. That analysis will appear in a separate report.

References

1. E. Anderson and J. Dongarra, ''LAPACK Working Note #10: Installing and Testing the Initial Release of LAPACK - UNIX and Non-UNIX Versions,'' Argonne National Laboratory Report, ANL/MCS-TM-130, May 1989.
2. C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, and D. Sorensen, ''LAPACK Working Note #5: Provisional Contents,'' Argonne National Laboratory Report, ANL-88-38, September 1988.
3. A. Carle, K. Cooper, R. Hood, K. Kennedy, L. Torczon, and S. Warren, ''A Practical Environment for Fortran Programming,'' *IEEE Computer*, vol. 20, 11, pp. 75-89, November 1987.
4. P. Mayes and G. Radicati, ''LAPACK Working Note #12: Banded Cholesky Factorization Using Level 3 BLAS,'' Argonne National Laboratory Report, ANL/MCS-TM-134., August 1989.

References