BLAS (Basic Linear Algebra Subprograms)

Menu

Questions/comments? lapack@icl.utk.edu

Contact us get the lastest news

BLAS(Legacy Website) FAQ

Presentation:

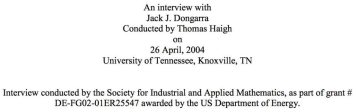
The BLAS (Basic Linear Algebra Subprograms) are routines that provide standard building blocks for performing basic vector and matrix operations. The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. Because the BLAS are efficient, portable, and widely available, they are commonly used in the development of high quality linear algebra software, LAPACK for example.

Acknowledgments:

This material is based upon work supported by the National Science Foundation under Grant No. ASC-9313958 and DOE Grant No. DE-FG03-94ER25219. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF) or the Department of Energy (DOE).

History

Discover the great history behind BLAS. On April 2004 an oral history interview was conducted as part of the SIAM project on the history of software for scientific computing and numerical analysis. This interview is being conducted with Professor Jack Dongarra in his office at the University of Tennessee. The interviewer is Thomas Haigh.
Download Interview
Enjoy!


An interview with
Jack J. Dongarra
Conducted by Thomas Haigh
on
26 April, 2004
University of Tennessee, Knoxville, TN

Interview conducted by the Society for Industrial and Applied Mathematics, as part of grant # DE-FG02-01ER25547 awarded by the US Department of Energy.

Software:

Licensing:

The reference BLAS is a freely-available software package. It is available from netlib via anonymous ftp and the World Wide Web. Thus, it can be included in commercial software packages (and has been). We only ask that proper credit be given to the authors.

Like all software, it is copyrighted. It is not trademarked, but we do ask the following:

- If you modify the source for these routines we ask that you change the name of the routine and comment the changes made to the original.
- We will gladly answer any questions regarding the software. If a modification is done, however, it is the responsibility of the person who modified the routine to provide support.

REFERENCE BLAS Version 3.10.0

- Download blas-3.10.0.tgz
- Updated June 2021

- Quick Reference Guide

Level 1 BLAS

	dim	scalar	vector	vector	scalars	5-element array		prefixes
SUBROUTINE xROTG (					A, B, C, S)		Generate plane rotation	S, D
SUBROUTINE xROTMG(					D1, D2, A, B,	PARAM)	Generate modified plane rotation	S, D
SUBROUTINE xROT (	N,		X, INCX,	Y, INCY,	C, S)		Apply plane rotation	S, D
SUBROUTINE xROTM (	N,		X, INCX,	Y, INCY,		PARAM)	Apply modified plane rotation	S, D
SUBROUTINE xSWAP (	N,		X, INCX,	Y, INCY)			$x \leftrightarrow y$	S, D, C, Z
SUBROUTINE xSCAL (	N,	ALPHA,	X, INCX)				$x \leftarrow \alpha x$	S, D, C, Z, CS, ZD
SUBROUTINE xCOPY (	N,		X, INCX,	Y, INCY)			$y \leftarrow x$	S, D, C, Z
SUBROUTINE xAXPY (	N,	ALPHA,	X, INCX,	Y, INCY)			$y \leftarrow \alpha x + y$	S, D, C, Z
FUNCTION xDOT (	N,		X, INCX,	Y, INCY)			$dot \leftarrow x^T y$	S, D, DS
FUNCTION xDOTU (	N,		X, INCX,	Y, INCY)			$dot \leftarrow x^T y$	C, Z
FUNCTION xDOTC (	N,		X, INCX,	Y, INCY)			$dot \leftarrow x^H y$	C, Z
FUNCTION xxDOT (	N,		X, INCX,	Y, INCY)			$dot \leftarrow \alpha + x^T y$	SDS
FUNCTION xNRM2 (	N,		X, INCX)				$nrm2 \leftarrow \|x\|_2$	S, D, SC, DZ
FUNCTION xASUM (	N,		X, INCX)				$asum \leftarrow \|\|re(x)\|_1 + \|im(x)\|\|_1$	S, D, SC, DZ
FUNCTION IxAMAX(	N,		X, INCX)				$amax \leftarrow 1^{st} k \ni \|re(x_k)\| + \|im(x_k)\| = max(\|re(x_i)\| + \|im(x_i)\|)$	S, D, C, Z

Level 2 BLAS

	options		dim	b-width	scalar	matrix	vector	scalar	vector		
xGEMV (		TRANS,	M, N,		ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y, y \leftarrow \alpha A^T x + \beta y, y \leftarrow \alpha A^H x + \beta y, A - m \times n$	S, D, C, Z
xGBMV (		TRANS,	M, N,	KL, KU,	ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y, y \leftarrow \alpha A^T x + \beta y, y \leftarrow \alpha A^H x + \beta y, A - m \times n$	S, D, C, Z
xHEMV (	UPLO,		N,		ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$	C, Z
xHBMV (	UPLO,		N,	K,	ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$	C, Z
xHPMV (	UPLO,		N,		ALPHA,	AP,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$	C, Z
xSYMV (	UPLO,		N,		ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$	S, D
xSBMV (	UPLO,		N,	K,	ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$	S, D
xSPMV (	UPLO,		N,		ALPHA,	AP,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$	S, D
xTRMV (	UPLO,	TRANS, DIAG,	N,			A, LDA,	X, INCX)			$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$	S, D, C, Z
xTBMV (	UPLO,	TRANS, DIAG,	N,	K,		A, LDA,	X, INCX)			$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$	S, D, C, Z
xTPMV (	UPLO,	TRANS, DIAG,	N,			AP,	X, INCX)			$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$	S, D, C, Z
xTRSV (	UPLO,	TRANS, DIAG,	N,			A, LDA,	X, INCX)			$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$	S, D, C, Z
xTBSV (	UPLO,	TRANS, DIAG,	N,	K,		A, LDA,	X, INCX)			$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$	S, D, C, Z
xTPSV (	UPLO,	TRANS, DIAG,	N,			AP,	X, INCX)			$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$	S, D, C, Z

	options		dim	scalar	vector	vector	matrix			
xGER (			M, N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)			$A \leftarrow \alpha xy^T + A, A - m \times n$	S, D
xGERU (			M, N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)			$A \leftarrow \alpha xy^T + A, A - m \times n$	C, Z
xGERC (			M, N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)			$A \leftarrow \alpha xy^H + A, A - m \times n$	C, Z
xHER (	UPLO,		N,	ALPHA,	X, INCX,		A, LDA)			$A \leftarrow \alpha xx^H + A$	C, Z
xHPR (	UPLO,		N,	ALPHA,	X, INCX,		AP)			$A \leftarrow \alpha xx^H + A$	C, Z
xHER2 (	UPLO,		N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)			$A \leftarrow \alpha xy^H + y(\alpha x)^H + A$	C, Z
xHPR2 (	UPLO,		N,	ALPHA,	X, INCX,	Y, INCY,	AP)			$A \leftarrow \alpha xy^H + y(\alpha x)^H + A$	C, Z
xSYR (	UPLO,		N,	ALPHA,	X, INCX,		A, LDA)			$A \leftarrow \alpha xx^T + A$	S, D
xSPR (	UPLO,		N,	ALPHA,	X, INCX,		AP)			$A \leftarrow \alpha xx^T + A$	S, D
xSYR2 (	UPLO,		N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)			$A \leftarrow \alpha xy^T + \alpha yx^T + A$	S, D
xSPR2 (	UPLO,		N,	ALPHA,	X, INCX,	Y, INCY,	AP)			$A \leftarrow \alpha xy^T + \alpha yx^T + A$	S, D

Level 3 BLAS

	options			dim		scalar	matrix	matrix	scalar	matrix		
xGEMM (		TRANSA,	TRANSB,	M, N,	K,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha op(A) op(B) + \beta C, op(X) = X, X^T, X^H, C - m \times n$	S, D, C, Z
xSYMM (	SIDE,	UPLO,		M, N,		ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB + \beta C, C \leftarrow \alpha BA + \beta C, C - m \times n, A = A^T$	S, D, C, Z
xHEMM (	SIDE,	UPLO,		M, N,		ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB + \beta C, C \leftarrow \alpha BA + \beta C, C \quad m \times n, A = A^H$	C, Z
xSYRK (		UPLO,	TRANS,		N, K,	ALPHA,	A, LDA,		BETA,	C, LDC)	$C \leftarrow \alpha AA^T + \beta C, C \leftarrow \alpha A^T A + \beta C, C - n \times n$	S, D, C, Z
xHERK (		UPLO,	TRANS,		N, K,	ALPHA,	A, LDA,		BETA,	C, LDC)	$C \leftarrow \alpha AA^H + \beta C, C \leftarrow \alpha A^H A + \beta C, C - n \times n$	C, Z
xSYR2K(		UPLO,	TRANS,		N, K,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB^T + \bar{\alpha} BA^T + \beta C, C \leftarrow \alpha A^T B + \bar{\alpha} B^T A + \beta C, C - n \times n$	S, D, C, Z
xHER2K(		UPLO,	TRANS,		N, K,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB^H + \bar{\alpha} BA^H + \beta C, C \leftarrow \alpha A^H B + \bar{\alpha} B^H A + \beta C, C - n \times n$	C, Z
xTRMM (	SIDE,	UPLO,	TRANSA,	DIAG,	M, N,	ALPHA,	A, LDA,	B, LDB)			$B \leftarrow \alpha op(A)B, B \leftarrow \alpha Bop(A), op(A) = A, A^T, A^H, B - m \times n$	S, D, C, Z
xTRSM (	SIDE,	UPLO,	TRANSA,	DIAG,	M, N,	ALPHA,	A, LDA,	B, LDB)			$B \leftarrow \alpha op(A^{-1})B, B \leftarrow \alpha Bop(A^{-1}), op(A) = A, A^T, A^H, B - m \times n$	S, D, C, Z

2

CBLAS

- Download cblas.tgz
- Header file: cblas.h

Level 3 BLAS tuned for single processors with caches

- Download ssgemmbased.tgz
- Written by Kagstrom B., Ling P., and Van Loan C.
- High Performance GEMM-Based Level-3 BLAS Webpage - Fortran (High Performance Computing II, 1991, North-Holland)

Extended precision Level 2 BLAS routines

- Download ecblas2.f

BLAS for windows

The reference BLAS is included inside the LAPACK package. Please refer tools built under Windows using Cmake the cross-platform, open-source build system. The new build system was developed in collaboration with Kitware Inc.

A dedicated website (http://icl.cs.utk.edu/lapack-for-windows/lapack/) is available for Windows users.

- You will find information about your configuration need.
- You will be able to download BLAS pre-built libraries.

GIT Access

The LAPACK GIT (http://github.com/Reference-LAPACK) repositories are to open for read-only for our users. The latest version of BLAS is included in LAPACK package.

Please use our LAPACK development repository to get the latest bug fixed, submit issues or pull requests.

The netlib family and its cousins

Basic Linear Algebra Subprograms (BLAS)	LAPACK
BLAS++	LAPACK++
PLASMA	MAGMA
CLAPACK (no longer maintained)	EISPACK (no longer maintained)

Support

If you have any issue (install, performance), just post your questions on the the LAPACK User Forum. You can also send us an email at lapack@icl.utk.edu

Documentation

Checkout the BLAS Wikipedia page

BLAS Technical Forum

The BLAS Technical Forum standard is a specification of a set of kernel routines for linear algebra, historically called the Basic Linear Algebra Subprograms. http://www.netlib.org/blas/blast-forum/

Optimized BLAS Library

Machine-specific optimized BLAS libraries are available for a variety of computer architectures. These optimized BLAS libraries are provided by the computer vendor or by an independent software vendor (ISV) . For further details, please see our FAQs.

Alternatively, the user can download ATLAS to automatically generate an optimized BLAS library for his architecture. Some prebuilt optimized BLAS libraries are also available from the ATLAS site.

If all else fails, the user can download a Fortran77 reference implementation of the BLAS from netlib. However, keep in mind that this is a reference implementation and is not optimized.

BLAS vendor library List Last updated: July 20, 2005

BLAS Routines

LEVEL 1

- **Single**
  - SROTG - setup Givens rotation
  - SROTMG - setup modified Givens rotation
  - SROT - apply Givens rotation
  - SROTM - apply modified Givens rotation
  - SSWAP - swap x and y
  - SSCAL - x = a*x
  - SCOPY - copy x into y
  - SAXPY - y = a*x + y
  - SDOT - dot product
  - SDSDOT - dot product with extended precision accumulation
  - SNRM2 - Euclidean norm
  - SCNRM2- Euclidean norm
  - SASUM - sum of absolute values
  - ISAMAX - index of max abs value
- **Double**
  - DROTG - setup Givens rotation
  - DROTMG - setup modified Givens rotation
  - DROT - apply Givens rotation

- DROTM - apply modified Givens rotation
- DSWAP - swap x and y
- DSCAL - x = a*x
- DCOPY - copy x into y
- DAXPY - y = a*x + y
- DDOT - dot product
- DSDOT - dot product with extended precision accumulation
- DNRM2 - Euclidean norm
- DZNRM2 - Euclidean norm
- DASUM - sum of absolute values
- IDAMAX - index of max abs value

- **Complex**
  - CROTG - setup Givens rotation
  - CSROT - apply Givens rotation
  - CSWAP - swap x and y
  - CSCAL - x = a*x
  - CSSCAL - x = a*x
  - CCOPY - copy x into y
  - CAXPY - y = a*x + y
  - CDOTU - dot product
  - CDOTC - dot product, conjugating the first vector
  - SCASUM - sum of absolute values
  - ICAMAX - index of max abs value

- **Double Complex**
  - ZROTG - setup Givens rotation
  - ZDROT - apply Givens rotation
  - ZSWAP - swap x and y
  - ZSCAL - x = a*x
  - ZDSCAL - x = a*x
  - ZCOPY - copy x into y
  - ZAXPY - y = a*x + y
  - ZDOTU - dot product
  - ZDOTC - dot product, conjugating the first vector
  - DZASUM - sum of absolute values
  - IZAMAX - index of max abs value

LEVEL 2

- **Single**
  - SGEMV - matrix vector multiply
  - SGBMV - banded matrix vector multiply
  - SSYMV - symmetric matrix vector multiply
  - SSBMV - symmetric banded matrix vector multiply
  - SSPMV - symmetric packed matrix vector multiply
  - STRMV - triangular matrix vector multiply
  - STBMV - triangular banded matrix vector multiply
  - STPMV - triangular packed matrix vector multiply
  - STRSV - solving triangular matrix problems
  - STBSV - solving triangular banded matrix problems
  - STPSV - solving triangular packed matrix problems
  - SGER - performs the rank 1 operation A := alpha*x*y' + A
  - SSYR - performs the symmetric rank 1 operation A := alpha*x*x' + A
  - SSPR - symmetric packed rank 1 operation A := alpha*x*x' + A
  - SSYR2 - performs the symmetric rank 2 operation, A := alpha*x*y' + alpha*y*x' + A
  - SSPR2 - performs the symmetric packed rank 2 operation, A := alpha*x*y' + alpha*y*x' + A

- **Double**
  - DGEMV - matrix vector multiply
  - DGBMV - banded matrix vector multiply
  - DSYMV - symmetric matrix vector multiply
  - DSBMV - symmetric banded matrix vector multiply
  - DSPMV - symmetric packed matrix vector multiply
  - DTRMV - triangular matrix vector multiply
  - DTBMV - triangular banded matrix vector multiply
  - DTPMV - triangular packed matrix vector multiply
  - DTRSV - solving triangular matrix problems
  - DTBSV - solving triangular banded matrix problems
  - DTPSV - solving triangular packed matrix problems
  - DGER - performs the rank 1 operation A := alpha*x*y' + A
  - DSYR - performs the symmetric rank 1 operation A := alpha*x*x' + A
  - DSPR - symmetric packed rank 1 operation A := alpha*x*x' + A
  - DSYR2 - performs the symmetric rank 2 operation, A := alpha*x*y' + alpha*y*x' + A
  - DSPR2 - performs the symmetric packed rank 2 operation, A := alpha*x*y' + alpha*y*x' + A
- **Complex**
  - CGEMV - matrix vector multiply
  - CGBMV - banded matrix vector multiply
  - CHEMV - hermitian matrix vector multiply
  - CHBMV - hermitian banded matrix vector multiply
  - CHPMV - hermitian packed matrix vector multiply
  - CTRMV - triangular matrix vector multiply
  - CTBMV - triangular banded matrix vector multiply
  - CTPMV - triangular packed matrix vector multiply
  - CTRSV - solving triangular matrix problems
  - CTBSV - solving triangular banded matrix problems
  - CTPSV - solving triangular packed matrix problems
  - CGERU - performs the rank 1 operation A := alpha*x*y' + A
  - CGERC - performs the rank 1 operation A := alpha*x*conjg(y') + A
  - CHER - hermitian rank 1 operation A := alpha*x*conjg(x') + A
  - CHPR - hermitian packed rank 1 operation A := alpha*x*conjg(x') + A
  - CHER2 - hermitian rank 2 operation
  - CHPR2 - hermitian packed rank 2 operation
- **Double Complex**
  - ZGEMV - matrix vector multiply
  - ZGBMV - banded matrix vector multiply
  - ZHEMV - hermitian matrix vector multiply
  - ZHBMV - hermitian banded matrix vector multiply
  - ZHPMV - hermitian packed matrix vector multiply
  - ZTRMV - triangular matrix vector multiply
  - ZTBMV - triangular banded matrix vector multiply
  - ZTPMV - triangular packed matrix vector multiply
  - ZTRSV - solving triangular matrix problems
  - ZTBSV - solving triangular banded matrix problems
  - ZTPSV - solving triangular packed matrix problems
  - ZGERU - performs the rank 1 operation A := alpha*x*y' + A
  - ZGERC - performs the rank 1 operation A := alpha*x*conjg(y') + A
  - ZHER - hermitian rank 1 operation A := alpha*x*conjg(x') + A
  - ZHPR - hermitian packed rank 1 operation A := alpha*x*conjg(x') + A
  - ZHER2 - hermitian rank 2 operation
  - ZHPR2 - hermitian packed rank 2 operation

LEVEL 3

- **Single**
  - SGEMM - matrix matrix multiply
  - SSYMM - symmetric matrix matrix multiply
  - SSYRK - symmetric rank-k update to a matrix
  - SSYR2K - symmetric rank-2k update to a matrix
  - STRMM - triangular matrix matrix multiply
  - STRSM - solving triangular matrix with multiple right hand sides
- **Double**
  - DGEMM - matrix matrix multiply
  - DSYMM - symmetric matrix matrix multiply
  - DSYRK - symmetric rank-k update to a matrix
  - DSYR2K - symmetric rank-2k update to a matrix
  - DTRMM - triangular matrix matrix multiply
  - DTRSM - solving triangular matrix with multiple right hand sides
- **Complex**
  - CGEMM - matrix matrix multiply
  - CSYMM - symmetric matrix matrix multiply
  - CHEMM - hermitian matrix matrix multiply
  - CSYRK - symmetric rank-k update to a matrix
  - CHERK - hermitian rank-k update to a matrix
  - CSYR2K - symmetric rank-2k update to a matrix
  - CHER2K - hermitian rank-2k update to a matrix
  - CTRMM - triangular matrix matrix multiply
  - CTRSM - solving triangular matrix with multiple right hand sides
- **Double Complex**
  - ZGEMM - matrix matrix multiply
  - ZSYMM - symmetric matrix matrix multiply
  - ZHEMM - hermitian matrix matrix multiply
  - ZSYRK - symmetric rank-k update to a matrix
  - ZHERK - hermitian rank-k update to a matrix
  - ZSYR2K - symmetric rank-2k update to a matrix
  - ZHER2K - hermitian rank-2k update to a matrix
  - ZTRMM - triangular matrix matrix multiply
  - ZTRSM - solving triangular matrix with multiple right hand sides

Extended precision Level 2 BLAS routines

- SUBROUTINE ECGEMV (TRANS, M, N, ALPHA, A, LDA, X, INCX, BETA, Y, INCY)
- SUBROUTINE ECGBMV (TRANS, M, N, KL, KU, ALPHA, A, LDA, X, INCX, BETA, Y, INCY)
- SUBROUTINE ECHEMV (UPLO, N, ALPHA, A, LDA, X, INCX,BETA, Y, INCY)
- SUBROUTINE ECHBMV (UPLO, N, K, ALPHA, A, LDA, X, INCX,BETA, Y, INCY)
- SUBROUTINE ECHPMV (UPLO, N, ALPHA, AP, X, INCX, BETA, Y, INCY)
- SUBROUTINE ECTRMV (UPLO, TRANS, DIAG, N, A, LDA, X, INCX)
- SUBROUTINE ECTBMV (UPLO, TRANS, DIAG, N, K, A, LDA, X, INCX)
- SUBROUTINE ECTPMV (UPLO, TRANS, DIAG, N, AP, X, INCX)
- SUBROUTINE ECTRSV (UPLO, TRANS, DIAG, N, A, LDA, X, INCX)
- SUBROUTINE ECTBSV (UPLO, TRANS, DIAG, N, K, A, LDA, X, INCX)
- SUBROUTINE ECTPSV (UPLO, TRANS, DIAG, N, AP, X, INCX)
- SUBROUTINE ECGERU (M, N, ALPHA, X, INCX, Y, INCY, A, LDA)
- SUBROUTINE ECGERC (M, N, ALPHA, X, INCX, Y, INCY, A, LDA)
- SUBROUTINE ECHER (UPLO, N, ALPHA, X, INCX, A, LDA)
- SUBROUTINE ECHPR (UPLO, N, ALPHA, X, INCX, AP)
- SUBROUTINE ECHER2 (UPLO, N, ALPHA, X, INCX, Y, INCY, A, LDA)
- SUBROUTINE ECHPR2 (UPLO, N, ALPHA, X, INCX, Y, INCY, AP)

Last updated 2021-06-29 12:30:12 PDT