Linear Algebra Kernel Functionality

The BLAST Forum *

12 August 1997

http://www.netlib.org/utk/papers/blast-forum.html

Abstract

This document lists the perceived functionality of numerical linear algebra library kernels as discussed at meetings of the BLAS Technical Forum. Included are kernels needed for dense matrices, sparse direct methods, sparse iterative methods as well as related kernels needed in non-traditional linear algebra computations such as CFD and finite elements.

The focus is on sequential and shared memory computers, but functionality for the parallel case has been borne in mind as well.

This version of the document gives a brief introduction and then presents tables listing the functionality. Further information on the BLAS technical Forum can be found at the World Wide Web address. Comments on this document and active participation in the Forum are encouraged.

*This document so far edited by Susan Blackford, Jack Dongarra, Sven Hammarling, Linda Kaufman, Antoine Petitet and Barry Smith

1 Introduction

<This section needs re-writing>

A technical forum has recently been established to consider expanding the Basic Linear Algebra Subprograms (BLAS) in a number of directions in the light of modern software, language, and hardware developments. The first two meetings of the BLAS Technical Forum were held in Nashville on February 19-20, 1996 and in Atlanta on May 20-21, 1996.

Working groups have been established to consider the overall functionality, a lightweight interface, and an object based interface, for the BLAS. Other subgroups have also been established, either to advise the current working groups or as placeholders for future working groups on parallel processing issues, sparse operations, data structures, extensions to the existing BLAS and language binding issues.

This document summarizes, in tabular form, the discussions of the Functionality Working Group. Note that this document does not consider issues such as storage formats or data types.

The list is fairly exhaustive and it is not assumed that everything will necessarily be included in the final report. At the very least we expect to provide a priority guide. The list includes the functionality covered by the existing Level 1, 2 and 3 BLAS [6, 1, 3, 2].

This document is expected to develop over time, and some of the topics to be discussed include:

- Motivation
- Functionality
- Justification
  - Algorithm Use
  - Efficiency Issues. E.g. re-use of data
  - Ease of expression
- Comparison with Existing BLAS
  - What's New
- Important Kernels

– (historical considerations and changes)

In this version of the document, the functionality is categorized more or less along traditional BLAS lines, but the Forum discussed other schemes and it may be appropriate to categorize the functionality in more than one way in the final document.

If what you need is not listed here, please comment and if possible join the Forum, to let us know what you need.

2 Justifications for extensions to the BLAS

<Susan added this section from email correspondence with Linda. Perhaps this section should be renamed, re-written, or moved.>

In the original BLAS, each level had its framework. Level 1 covered vector operations, Level 2 covered matrix vector operations, while Level 3 covered matrix-vector operations. The new BLA subroutines have no framework, because it was realized that all the levels were missing some basic operations and because some operations cannot be so neatly categorized. The idea behind specifying a new BLA was proven functionality. No subroutine was to be added because it just neatly fit into a framework and no subroutine was to be discarded because it did not fit into a particular framework.

Some of the new subroutines are auxiliary routines in LAPACK. Only after the LAPACK project was begun was it realized that there were operations like the matrix copy routine, the computation of a norm of a matrix and the generation of Householder transformations that occurred so often that it was wise to make separate subroutines for them.

A second group of these extensions changed the specificity of some of the existing BLAS. The most stunning example of this was the Level 3 routine for rank k update of a symmetric matrix. The present subroutine only allows a positive update which made it useless for the reduction of a symmetric matrix to tridiagonal form to facilitate the computation of the eigensystem of a symmetric matrix. The positivity requirement also meant that the BLAS could not be used for the decomposition of an indefinite symmetric matrix or for updating a quasi-Newton method during function minimization.

Some of the new BLAS might be considered special cases of the original BLAS. The number of Level 2 and 3 BLAS was purposely kept small, but each had a number of parameters that could extend their functionality. Most of these parameters in practice were either 1.0 or 0.0. Adding these parameters, rather than specifying different subroutines increased the opportunities for the user to make mistakes and increased the amount of overhead of a call to the routine.

A large number of the extensions are multiple instances of the Level 1 BLAS. They tend to arise when working with narrow banded and sparse or structured matrices. The canonical example is the decomposition of many narrow banded matrices simultaneously. This type of problem arises when finding several eigenvectors using inverse iteration, when using parallel bisection to determine the eigenvalues of a symmetric matrix or when solving two dimensional separable partial differential equations or preconditioning an iterative method for a nonseparable differential equation. Certainly if one was on a multi-processor machine using a compiler with compiler directives one could specify that various Level 1 BLAS like isamax, sswap and saxpy be done simultaneously. However, when one is not in this situation and one wishes to use portable code, then the existence of a BLA subroutine to cover this situation is helpful.

Other examples of multiple instance routines are the auxiliary routines in LAPACK for generating and applying Givens transformation for reducing a banded matrix to tridiagonal form in order to solve the symmetric eigenvalue problem. Kaufman has shown that simultaneous Givens transformations are also useful when solving the tridiagonal positive definite symmetric generalized eigenvalue problem and when using the QR algorithm to solve the standard tridiagonal symmetric eigenvalue problem. The generation of simultaneous Givens transformations is a particularly good candidate because the best way to implement the conditional to test for the null transformation not only varies with the manufacturer, but is also compiler dependent for a given manufacturer.

For large matrix problems, it might not be feasible or desirable to store the full matrix. In this situation one cannot simply use the matrix-vector multiplication routines in the present BLAS. However, one may be able to form that product with some simultaneous inner product routines because they have some repeated submatrices. This feature is particular true of Toeplitz and Hankel matrices used in signal and image processing.

References

[1] D. S. Dodson, R. G. Grimes, and J. G. Lewis. Sparse extensions to the FORTRAN Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 17:253–272, 1991. (Algorithm 692).

[2] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 16:1–28, 1990. (Algorithm 679).

[3] J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson. An extended set of FORTRAN Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 14:1–32, 1988. (Algorithm 656).

[4] L. Kaufman and D. Warner, High-order, fast direct methods for Separable Elliptical Equations. *SIAM Journal of Numerical Analysis*, 21:672-695, 1984.

[5] L. Kaufman, Implementing the MDM(transpose) Decomposition. *ACM Trans. Math. Software*, 21:476-489, 1995.

[6] C. L. Lawson, R. J. Hanson, D. Kincaid, and F. T. Krogh. Basic Linear Algebra Subprograms for FORTRAN usage. *ACM Trans. Math. Software*, 5:308–323, 1979. (Algorithm 539).

[7] C. Smith and B. Hendrickson and E. Jessup, A Parallel Algorithm for Householder Tridiagonalization. *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, 361-365, 1994.

A Notation

The following notation is used throughout the tables in the appendices.

- x_i - an element of a one-dimensional array
- x_I - subset of entries of x indexed by I
- $y[i]$ - an array from an array of arrays
- x_{C} - compressed (sparse) array, only non-zeros are stored, associated with an index set I
- e - vector whose elements are all unity

The last column of each table gives the justification for a routine's inclusion.

- "LEGACY" refers to the fact that it is a legacy BLA,
- "AUX" means that it has appeared as an auxiliary routine in LAPACK, reference numbers refer to items in the bibliography,
- "DERIV" means that it is a commonly used subset of an existing BLA which deserves its own subroutine, and
- "COMMON" means that it appears so commonly in existing software, that it is surprising that it is not presently a BLA.

A question mark in the last column suggests that it will probably be a new BLA if there is sufficient support and a "-" sign suggests that it will probably never be in the final list. Not all of the auxiliary subroutine in LAPACK are included and some have question marks or - signs next to them because their utility outside of LAPACK has not yet been demonstrated sufficiently.

B Scalar and Vector Operations

This Appendix gives the operations corresponding to the traditional Level 1 BLAS operations. Table 1 lists the scalar and vector reduction operations, Table 2 lists the vector operations, Table 3 lists the sparse vector operations and Table 4 lists those vector operations that involve only data movement. The operations are intended to apply to both real and complex arguments. For the sake of compactness the complex operators are omitted, so that whenever a transpose operation is given the conjugate transpose should also be assumed for the complex case.

Inner product	$r \leftarrow x^T y$,	Legacy
	$r \leftarrow r \pm x^T y$	-
	$r \leftarrow \alpha x^T y$,	-
	$r \leftarrow r + \alpha x^T y$	-
Vector norms	$r \leftarrow \|\|x\|\|_1$,	Legacy
	$r \leftarrow \|\|x\|\|_\infty$	Common
	$r \leftarrow \|\|x\|\|_2$,	Legacy
	$r \leftarrow \|\|x\|\|_p$	-
Sum	$r \leftarrow \sum_i x_i$	Common
Location of max value	$i = \arg\max_i \|x_i\|$	Legacy
Max value & location	$i, x_i, ; i = \arg\max_i x_i$	Common
Max absolute value & location	$i, \|x_i\|, i = \arg\max_i \|x_i\|$	Common
Multiple inner products	$r_i \leftarrow x^T y[i]$	Common
Multiple maximum abs value	$i_j = argmax_i \|x_i j\|$	[4],
		Inverse Iteration
Sum of squares	$(a, b) \leftarrow \sum x_i^2$,	Common
	$a \cdot b^2 = \sum x^2$	-
Generate Givens rotation	$(c, s) \leftarrow \mathrm{rot}(a, b)$	Legacy
Generate muliple Givens rotation	$(c[i], s[i]) \leftarrow \mathrm{rot}(a[i], b[i])$	Aux
Generate Modified Givens rotation	$(z, r) \leftarrow \mathrm{mrot}(a, b)$	Legacy
Generate Jacobi rotation	$(c, s) \leftarrow \mathrm{jrot}(a, b)$	?
Generate Householder transform	$(\alpha, u) \leftarrow \mathrm{house}(x)$,	
	$H = I - \alpha u u^T$	Aux

Table 1: Reduction Operations

Zero	$x \leftarrow 0$	Common
Constant	$x \leftarrow \alpha e$	Common
Conjugate copy	$y \leftarrow \bar{x}$	Common
Scale	$x \leftarrow \alpha x,$	Legacy
	$y \leftarrow \alpha x$	Common
Group scale	$x_{I_i} \leftarrow a_i x_{I_i}$	Common, diagonal scaling
Sum	$y \leftarrow \pm x + y,$	?
	$w \leftarrow \pm x + y$	?
Shift	$x \leftarrow x + \sigma e,$	?
	$y \leftarrow x + \sigma e$	?
AXPY & AXPY like	$y \leftarrow \alpha x + y,$	Legacy
	$w \leftarrow \alpha x + y$	Common
	$y \leftarrow x + \beta y,$	?
	$w \leftarrow x + \beta y$	?
	$y \leftarrow \alpha x + \beta y,$	?
	$w \leftarrow \alpha x + \beta y$	Common, optimization
	$y \leftarrow \alpha x + \beta y + \sigma e,$	?
	$w \leftarrow \alpha x + \beta y + \sigma e$	?
Absolute value	$x_i \leftarrow \|x_i\|$	
	$y_i \leftarrow \|x_i\|$	Common
Elementwise multiply	$x_i \leftarrow x_i y_i,$	Common, diagonal scaling
	$w_i \leftarrow x_i y_i$	?
Elementwise divide	$x_i \leftarrow x_i / y_i,$	Common
	$w_i \leftarrow x_i / y_i$	?
Multiple AXPYs	$y[i] \leftarrow a_i x + y[i]$	[4] inverse iteration
Apply plane rotation	$(\begin{array}{cc} x & y \end{array}) \leftarrow (\begin{array}{cc} x & y \end{array}) R$	Legacy
Apply multiple plane rotation	$(\begin{array}{cc} x[i] & y[i] \end{array}) \leftarrow (\begin{array}{cc} x[i] & y[i] \end{array}) R[i]$	Aux
Apply modified plane rotation	$(\begin{array}{cc} \tilde{x} & \tilde{y} \end{array}) D \leftarrow (\begin{array}{cc} \tilde{x} & \tilde{y} \end{array}) DR$	Legacy

Table 2: Vector Operations

Scatter add	$y_I \leftarrow \pm x_{\rm C} + y_I$	?
Gather add	$x_{\rm C} \leftarrow x_{\rm C} \pm y_I$	?
AXPY (other AXPY and AXPY-like (operations also?)	$y_I \leftarrow \alpha x_{\rm C} + y_I$	Common, sparse decompositions
Inner product	$r \leftarrow x_{\rm C}\, y_I^T$	Common, sparse decompositions
Apply plane rotation	$\begin{pmatrix} x_{\rm C} & y_I \end{pmatrix} \leftarrow \begin{pmatrix} x_{\rm C} & y_I \end{pmatrix} R$	?

Table 3: Sparse Vector Operations

Copy	$y \leftarrow x$	Common
Swap	$y \leftrightarrow x$	Legacy
Sort into order & return index vector	$(iv, y) \leftarrow \mathrm{sort}(x)$	Common
Sort according to index vector	$y \leftarrow \mathrm{sort}(x, iv)$	Common
Scatter	$y_I \leftarrow x_{\rm C}$	Common, sparse decompositions
Gather	$x_{\rm C} \leftarrow y_I$	Common, sparse decompositions

Table 4: Data Movement with Vectors

C Matrix Vector Operations

This Appendix gives the operations corresponding to the traditional Level 2 BLAS operations. Table 5 lists the matrix-vector operations. The operations are intended to apply to both real and complex arguments, and the matrix arguments A, T, L and U are intended to be dense or banded or sparse. In addition, where appropriate, the matrix A can be symmetric (Hermitian) or triangular or general. The matrix T specifically represents a triangular matrix, either upper or lower, L represents a lower triangular matrix and U an upper triangular matrix, all three of which can be unit or non-unit triangular.

D represents at least a diagonal matrix, and maybe also a 2×2 blocked diagonal or a tridiagonal matrix. Q represents an orthogonal (unitary) matrix and it is an open question as to how Q should be stored.

For the sake of compactness the complex operators are omitted, so that whenever a transpose operation is given both the conjugate and conjugate transpose should also be assumed for the complex case.

Matrix vector product	$x \leftarrow Ax,\ x \leftarrow A^T x$	Deriv
	$y \leftarrow \pm Ax,\ y \leftarrow \pm A^T x$	Deriv
	$y \leftarrow \pm Ax + y,\ y \leftarrow \pm A^T x + y$	?
	$y \leftarrow \alpha Ax + \beta y,\ y \leftarrow \alpha A^T x + \beta y$	Legacy
Diagonal extract	$d \leftarrow \mathrm{diag}(A)$	?
Triangular solve	$x \leftarrow T^{-1}x,\ x \leftarrow T^{-T}x$	Legacy
	$x \leftarrow \alpha T^{-1}x,\ x \leftarrow \alpha T^{-T}x$	-
	$y \leftarrow T^{-1}x,\ y \leftarrow T^{-T}x$	?
	$y \leftarrow \alpha T^{-1}x,\ y \leftarrow \alpha T^{-T}x$	-
Double triangular solve,	$x \leftarrow U^{-1}L^{-1}x,\ y \leftarrow U^{-1}L^{-1}x$	-
– including permutations	$x \leftarrow U^{-1}D^{-1}L^{-1}x,\ y \leftarrow U^{-1}D^{-1}L^{-1}x$	-
(Also do these with inverse	$x \leftarrow L^{-1}U^{-1}x,\ y \leftarrow L^{-1}U^{-1}x$	-
transpose, and with TT^T	$x \leftarrow L^{-1}D^{-1}U^{-1}x,\ y \leftarrow L^{-1}D^{-1}U^{-1}x$	-
and TDT^T when SPD)		
Rank one and symmetric ($A = A^T$)	$A \leftarrow \beta xy^T + \alpha A$	?
rank one & two updates	$A \leftarrow \beta xy^T + A$	Legacy
($\alpha = 1$ and $\beta = \pm 1$	$A \leftarrow \beta xx^T + A$	Legacy
as special cases?)	$A \leftarrow \beta xx^T + \alpha A,\ A \leftarrow \beta xDx^T + \alpha A$	?
	$A \leftarrow (\beta x)y^T + y(\beta x)^T + A$	-
	$A \leftarrow (\beta x)y^T + y(\beta x)^T + \alpha A$	-
	$A \leftarrow (\beta xD)y^T + y(\beta xD)^T + \alpha A$	-
Application of Q from	$x \leftarrow Qx,\ x \leftarrow Q^T x$	Aux
QR factorization	$y \leftarrow Q^T x\ y \leftarrow Q^T x$	Aux
Also apply these operations		-
to sparse vectors?		-

Table 5: Matrix Vector Operations

D Matrix Operations

This Appendix gives the operations corresponding to the traditional Level 3 BLAS operations, as well as a number of additional single matrix operations. Table 6 lists single matrix operations and matrix operations that are involve $O(n^2)$ operations, Table 7 lists the $O(n^3)$ matrix-matrix operations and Table 8 lists those matrix operations that involve only data movement. The operations are intended to apply to both real and complex arguments. Where appropriate one or more of the matrices can also be symmetric (Hermitian) or triangular or general. The matrix T specifically represents a triangular matrix, either upper or lower, L represents a lower triangular matrix and U an upper triangular matrix, all three of which can be unit or non-unit triangular.

D represents at least a diagonal matrix, and maybe also a 2×2 blocked diagonal or a tridiagonal matrix. Q represents an orthogonal (unitary) matrix and it is an open question as to how Q should be stored.

For the sake of compactness the complex operators are omitted, so that whenever a transpose operation is given both the conjugate and conjugate transpose should also be assumed for the complex case.

Matrix set	$A \leftarrow \alpha I,$	Common
	$A \leftarrow \alpha I + \beta(e^T e - I)$	-
Matrix norms	$r \leftarrow \|\|A\|\|_1, r \leftarrow \|\|A\|\|_F, r \leftarrow \|\|A\|\|_\infty$	Aux
Maximum absolute value	$r \leftarrow \max \|a_{i,j}\|$	Common
Matrix scaling	$A \leftarrow \alpha A,\ B \leftarrow \alpha A$	Aux
Diagonal scaling	$A \leftarrow DA,\ A \leftarrow AD,\ A \leftarrow D_1 A D_2$	Common
Matrix add	$A \leftarrow \pm A + B,\ C \leftarrow \pm A + B$	Common
Matrix add and scale	$A \leftarrow \alpha A + \beta B,\ C \leftarrow \alpha A + \beta B$	?
Elementwise multiply	$a_{ij} \leftarrow a_{ij} b_{ij},\ c_{ij} \leftarrow a_{ij} b_{ij}$	-
Elementwise divide	$a_{ij} \leftarrow a_{ij}/b_{ij},\ c_{ij} \leftarrow a_{ij}/b_{ij}$	-

Table 6: Matrix Operations – $O(n^2)$

Matrix matrix product	$A \leftarrow AB,\ A \leftarrow A^T B,\ A \leftarrow AB^T$	?
	$C \leftarrow AB,\ C \leftarrow A^T B,\ C \leftarrow AB^T$	Deriv
	$C \leftarrow \pm AB + C,\ C \leftarrow \pm A^T B + C,\ C \leftarrow \pm AB^T + C$	-
	$D \leftarrow \pm AB + C,\ D \leftarrow \pm A^T B + C,\ D \leftarrow \pm AB^T + C$	-
	$C \leftarrow \alpha AB + \beta C,$	Legacy
	$C \leftarrow \alpha AB + \beta C, (C = C^T)$	-
	$D \leftarrow \alpha AB + \beta C$	-
	$C \leftarrow \alpha A^T B + \beta C$	Legacy
	$D \leftarrow \alpha A^T B + \beta C$	-
	$C \leftarrow \alpha AB^T + \beta C$	Legacy
	$D \leftarrow \alpha AB^T + \beta C$	-
Triangular	$B \leftarrow \alpha T^{-1} B$	Legacy
Triangular & double	$B \leftarrow T^{-1} B,\ C \leftarrow T^{-1} B,\ C \leftarrow \alpha T^{-1} B$	-
(with permutations)	$B \leftarrow BT^{-1},\ B \leftarrow \alpha BT^{-1},\ C \leftarrow BT^{-1},\ C \leftarrow \alpha BT^{-1}$	-
triangular solve	$B \leftarrow U^{-1} L^{-1} B,\ C \leftarrow U^{-1} L^{-1} B$	-
(Also with inverse	$B \leftarrow U^{-1} D L^{-1} B,\ C \leftarrow U^{-1} D L^{-1} B$	-
transpose, and for TT^T	$B \leftarrow L^{-1} U^{-1} B,\ C \leftarrow L^{-1} U^{-1} B$	-
& TDT^T when SPD)	$B \leftarrow L^{-1} D U^{-1} B,\ C \leftarrow L^{-1} D U^{-1} B$	-
Symmetric rank k & $k2$	$C \leftarrow \alpha AA^T + \beta C,\ C \leftarrow \alpha A^T A + \beta C$	Legacy
updates ($C = C^T$)	$C \leftarrow \alpha ADA^T + \beta C,\ C \leftarrow \alpha A^T DA + \beta C$	[5] [7]
($\alpha = 1$ and $\beta = \pm 1$	$C \leftarrow (\alpha A) B^T + B (\alpha A)^T + \beta C$	Legacy
as special cases?)	$C \leftarrow (\alpha A)^T B + B^T (\alpha A) + \beta C$	-
	$C \leftarrow (\alpha AD) B^T + B (\alpha AD)^T + \beta C$	?
	$C \leftarrow (\alpha DA)^T B + B^T (\alpha DA) + \beta C$	-
Application of Q from	$X \leftarrow QX,\ X \leftarrow XQ,\ X \leftarrow Q^T X,\ X \leftarrow XQ^T$	?
QR factorization	$Y \leftarrow QX,\ Y \leftarrow XQ,\ Y \leftarrow Q^T X,\ Y \leftarrow XQ^T$	?

Table 7: Matrix Matrix Operations

Matrix copy	$B \leftarrow A$	Aux
Matrix transpose	$A \leftarrow A^T, \ B \leftarrow A^T$	?
Get and put submatrix	$B \leftarrow a_{I,J}$	Aux, Common
(dense and sparse)	$b_{I,J} \leftarrow A, \ b_{I,J} \leftarrow A + b_{I,J}$	Aux, Common
Permute Matrix	$A \leftarrow PA$, $A \leftarrow AP$	Aux, Common
	$A \leftarrow QAP$	Aux, Common

Table 8: Data Movement with Matrices