Mapping Lapack to Matlab, Mathematica and Ada 2005 functions
(DRAFT version, may contain errors)

Nasser M. Abbasi

sometime in 2011 Compiled on January 30, 2024 at 5:52pm

This table lists some of the Lapack functions (only the Single Precision REAL Routines are shown), and Matlab, Mathematica and Ada calls which closely provide that functionality.

Table

lapack	description	Matlab	Mathematica	Ada
SGESV	Solves a general system of linear equations $Ax = b$	A\b f=factorize(A) x=f\b S=inverse(A) x=S*b pinv(A)* b	LinearSolve[A,B] PsedudoInverse[A].B	x:=solve(A,b)
SGBSV	Solves a general banded system of linear equations $Ax = b$	A\b	LinearSolve[A,B]	x:=solve(A,b)

SGTSV	Solves a general tridiagonal system of linear equations $Ax = b$	A\b	LinearSolve[A,B]	x:=solve(A,b)
SPOSV	Solves a symmetric positive definite system of linear $Ax = b$	A\b	LinearSolve[A,B]	x:=solve(A,b)
SPPSV	Solves a symmetric positive definite system of linear equations $Ax = b$, where A is held in packed storage	A\b	LinearSolve[A,B]	x:=solve(A,b)
SPBSV	Solves a symmetric positive definite banded system $Ax = b$	see above. Or R=chol(A) x=R\(R'\B)	see above. Or R=CholeskyDecomposition[A] LinearSolve[Transpose[R],B] LinearSolve[R,%]	x:=solve(A,b)
SPTSV	Solves a symmetric positive definite tridiagonal system $Ax = b$	A\b	LinearSolve[A,B]	x:=solve(A,b)
SSYSV	Solves a real symmetric indefinite system of linear equations $Ax = b$	A\b	LinearSolve[A,B]	x:=solve(A,b)
SSPSV	Solves a real symmetric indefinite system of linear equations $Ax = b$ where A is held in packed storage	A\b	LinearSolve[A,B]	x:=solve(A,b)

SGELS	Computes the least squares solution to an overdetermined system of linear equations, $Ax = b$ or $A^T x = b$, or the minimum norm solution of an underdetermined system, where A is a general rectangular matrix of full rank, using a QR or LQ factorization of A	for overdetermined: A\b for underdetermined: pinv(A)*b or lsqlin(A,b)	for overdetermined: LinearSolve[A,b] for underdetermined: PseudoInverse[A].b or LeastSquares[A,b]	x:=solve(A,b)
SGELSD	Computes the least squares solution to an overdetermined system of linear equations, $Ax = b$ or $A^T x = b$, or the minimum norm solution of an underdetermined system, where A is a general rectangular matrix of full rank, using singular value decomposition (SVD)	Can also use x=A\b [u,s,v]=svd(A) x=v*inv(s)*v'*b	x=LinearSolve[A,b] u,w,v=SingularValueDecomposition[A] invS=DiagonalMatrix[1/Diagonal[s]] x=v.invS.Transpose[v].b	No SVD. Can use x:=solve(A,b)
SGGLSE	Solves the LSE (Constrained Linear Least Squares Problem) using the Generalized RQ factorization	lsqlin()	FindMinimum[]	Missing?

SGGGLM	Solves the GLM (Generalized Linear Regression Model) using the GQR (Generalized QR) factorization	glmfit() requires statistics toolbox	see GeneralizedLinearModelFit[] and LinearModelFit[]	Missing?
SSYEV	Computes all eigenvalues and optionally, eigenvectors of a real symmetric matrix	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()
SSYEVD	Computes all eigenvalues and optionally, eigenvectors of a real symmetric matrix If eigenvectors are desired, it uses a divide and conquer algorithm	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()
SSPEV	Computes all eigenvalues and optionally, eigenvectors of a real symmetric matrix in packed storage	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()
SSPEVD	Computes all eigenvalues and optionally, eigenvectors of a real symmetric matrix in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()

SSBEV	Computes all eigenvalues and optionally, eigenvectors of a real symmetric band matrix	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()
SSBEVD	Computes all eigenvalues and optionally, eigenvectors of a real symmetric band matrix. If eigenvectors are desired, it uses a divide and conquer algorithm	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()
SSTEV	Computes all eigenvalues and optionally, eigenvectors of a real symmetric tridiagonal matrix	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()
SSTEVD	Computes all eigenvalues and optionally, eigenvectors of a real symmetric tridiagonal matrix. If eigenvectors are desired, it uses a divide and conquer algorithm	eig() or eigs()	Eigensystem[] Eigenvalues[] Eigenvectors[]	eigenvalues() eigensystem()

SGEES	Computes all eigenvalues and Schur factorization of a general matrix and orders the factorization so that selected eigenvalues are at the top left of the Schur form	schur()	SchurDecomposition[]	missing?
SGEEV	Computes the eigenvalues and left and right eigenvectors of a general matrix	For right eigenvectors use [V,D] = eig(A) For left eigenvectors of A use [W,D] = eig(A.') W=conj(W)	For right eigenvectors use D,V=Eigensystem[A] v=Transpose[v] For left eigenvectors of A D,W=Eigensystem[Transpose[A]] W=ConjugateTranspose[W]	For right eigenvectors use eigensystem(A,values,vectors) and for left eigenvectors, use transpose() on A and call eigensystem() again then call conjugate(). See annex G for the exact calls.
SGESVD	Computes the singular value decomposition (SVD) a general matrix	svd()	SingularValueDecomposition[]	missing?
SGESDD	Computes the singular value decomposition (SVD) a general matrix using divide-and-conquer	svd()	SingularValueDecomposition[]	missing?

SSYGV	Computes all eigenvalues and the eigenvectors of a generalized symmetric-definite generalized eigenproblem	[V,D]=eig(A,B,'chol')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?
SSYGVD	Computes all eigenvalues and the eigenvectors of a generalized symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$, $BAx = \lambda x$ If eigenvectors are desired, it uses a divide and conquer algorithm	[V,D]=eig(A,B,'chol')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?
SSPGV	Computes all eigenvalues and the eigenvectors of a generalized symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$, $BAx = \lambda x$ where A and B are in packed storage	[V,D]=eig(A,B,'chol')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?

SSPGVD	Computes all eigenvalues and the eigenvectors of a generalized symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$, $BAx = \lambda x$, where A and B are in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm	[V,D]=eig(A,B,'chol')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?
SSBGV	Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetric of the form the form $Ax = \lambda Bx$ A and B are assumed to be symmetric and banded, and B is also positive definite	[V,D]=eig(A,B,'chol')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?

SSBGVD	Computes all eigenvalues and optionally, the eigenvectors of a real generalized symmetric definite banded eigenproblem of the form $Ax = \lambda Bx$ A and B are assumed to be symmetric and banded, and B is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm	[V,D]=eig(A,B,'chol')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?
SGGES	Computes the generalized eigenvalues, Schur form, and left and/or right Schur vectors for a pair of nonsymmetric matrices	schur()	SchurDecomposition[]	missing?
SGGEV	Computes the generalized eigenvalues, and left and/or right generalized eigenvectors for a pair of nonsymmetric matrices	[V,D]=eig(A,B,'qz')	D,V=Eigensystem[A,B] D,V=Eigensystem[A,B,k]	missing?
SGGSVD	Computes the Generalized Singular Value Decomposition	gsvd()	SingularValueList[]	missing?

SGESVX	Solve a general system of linear equations, $Ax = b$, $A^T x = b$, or $A^H x = b$ and provides an estimate of the condition number, and error bounds on the solution	A\b cond(A)	LinearSolve[A,b] LinearAlgebra‘MatrixConditionNumber[A]	Use transpose or conjugate on A first, then call solve(). But missing condition number function.
SGBSVX	Solves a general banded system of linear equations $Ax = b$, $A^T x = b$, or $A^H x = b$,and provides an estimate of the condition number and error bounds on the solution.	A\b cond(A)	LinearSolve[A,b] LinearAlgebra‘MatrixConditionNumber[A]	Use transpose or conjugate on A first, then call solve(). But missing condition number function.
SGTSVX	Solves a general tridiagonal system of linear equations $Ax = b$, $A^T x = b$, or $A^H x = b$, and provides an estimate of the condition ,number and error bounds on the solution	A\b cond(A)	LinearSolve[A,b] LinearAlgebra‘MatrixConditionNumber[A]	Use transpose or conjugate on A first, then call solve(). But missing condition number function.

SPOSVX	Solves a symmetric positive definite system of linear equations $Ax = b$, and provides an estimate of the condition number and error bounds on the solution.	A\b cond(A)	LinearSolve[A,b] LinearAlgebra‘MatrixConditionNumber[A]	x:solve(A,b). But missing condition number function.
SPPSVX	Solves a symmetric positive definite system of linear equations $Ax = b$, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution	A\b cond(A)	LinearSolve[A,b] LinearAlgebra‘MatrixConditionNumber[A]	x:solve(A,b). But missing condition number function.
SPBSVX	Solves a symmetric positive definite banded system of linear equations $Ax = b$, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution.	A\b cond(A)	LinearSolve[A,b] LinearAlgebra‘MatrixConditionNumber[A]	x:solve(A,b). But missing condition number function.

SPTSVX	Solves a symmetric positive definite tridiagonal system of linear equations $Ax = b$, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution.	A\b cond(A)	LinearSolve[A,b] LinearAlgebra`MatrixConditionNumber[A]	x:solve(A,b). But missing condition number function.
SSYSVX	Solves a real symmetric indefinite system of linear equations $Ax = b$, and provides an estimate of the condition number and error bounds on the solution.	A\b cond(A)	LinearSolve[A,b] LinearAlgebra`MatrixConditionNumber[A]	x:solve(A,b). But missing condition number function.
SSPSVX	Solves a real symmetric indefinite system of linear equations $Ax = b$, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution.	A\b cond(A)	LinearSolve[A,b] LinearAlgebra`MatrixConditionNumber[A]	x:solve(A,b). But missing condition number function.

SGELSY	Computes the minimum norm least squares solution to an over-or under-determined system of linear equations $Ax = b$, using a complete orthogonal factorization of A	for overdetermined: A\b for underdetermined: pinv(A)*b or lsqlin(A,b)	for overdetermined: LinearSolve[A,b] for underdetermined: PseudoInverse[A].b or LeastSquares[A,b]	x:=solve(A,b)
SGELSS	Computes the minimum norm least squares solution to an over- or under-determined system of linear equations $Ax = b$, using the singular value decomposition of A.	for overdetermined: A\b for underdetermined: pinv(A)*b or lsqlin(A,b)	for overdetermined: LinearSolve[A,b] for underdetermined: PseudoInverse[A].b or LeastSquares[A,b]	x:=solve(A,b)
SSYEVX	Computes selected eigenvalues and eigenvectors of a symmetric matrix.	use eig() then user selects	Eigenvalues[] then user selects	eigenvalues(A) then user selects

SSYEVR	Computes selected eigenvalues, and optionally, eigenvectors of a real, symmetric matrix. Eigenvalues are computed by the dqds algorithm, and eigenvectors are computed from various "good" LDL^T, representations (also known as Relatively Robust Representations).	No direct support, but can use eig() then user selects	No direct support, but can use Eigensystem() then user selects	No direct support, but can use eigensystem() then user selects
SSYGVX	Computes selected eigenvalues and and optionally, the eigenvectors of a generalized symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$, $BAx = \lambda x$	No direct support, [V,D]=eig(A,B,'chol') then user selects	No direct support, but can use D,V=Eigensystem[A,B] or D,V=Eigensystem[A,B,k] then user selects	missing?
SSPEVX	Computes selected eigenvalues and eigenvectors of a symmetric matrix in packed storage.	No direct support, but can use eig() then user selects	No direct support, but can use Eigensystem() then user selects	No direct support, but can use eigensystem() then user selects

SSPGVX	Computes selected eigenvalues and and optionally, the eigenvectors of a generalized symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$, $BAx = \lambda x$ where A and B are in packed storage.	No direct support, [V,D]=eig(A,B,'chol') then user selects	No direct support, but can use D,V=Eigensystem[A,B] or D,V=Eigensystem[A,B,k] then user selects	missing?
SSBEVX	Computes selected eigenvalues and eigenvectors of a symmetric band matrix.	No direct support, but can use eig() then user selects	No direct support, but can use Eigensystem() then user selects	No direct support, but can use eigensystem() then user selects
SSBGVX	Computes selected eigenvalues, and optionally, the eigenvectors of a real generalized symmetric-definite banded eigenproblem, of the form A*x=(lambda)*B*x. A and B are assumed to be symmetric and banded, and B is also positive definite.	No direct support, [V,D]=eig(A,B,'chol') then user selects	No direct support, but can use D,V=Eigensystem[A,B] or D,V=Eigensystem[A,B,k] then user selects	missing?

SSTEVX	Computes selected eigenvalues and eigenvectors of a real symmetric tridiagonal matrix.	No direct support, but can use eig() then user selects	No direct support, but can use Eigensystem() then user selects	No direct support, but can use eigensystem() then user selects
SSTEVR	Computes selected eigenvalues, and optionally, eigenvectors of a real symmetric tridiagonal matrix. Eigenvalues are computed by the dqds algorithm, and eigenvectors are computed from various "good" LDL^T representations (also known as Relatively Robust Representations).	No direct support, but can use eig() then user selects	No direct support, but can use Eigensystem() then user selects	No direct support, but can use eigensystem() then user selects
SGEESX	Computes the eigenvalues and Schur factorization of a general matrix, orders the factorization so that selected eigenvalues, are at the top left of the Schur form, and computes reciprocal condition numbers for the average of the selected eigenvalues and for the associated right invariant subspace.	No direct support, but can use eig(), shur(), then user selects	No direct support, but can use Eigensystem(), SchurDecomposition[], then user selects	No direct support, but can use eigensystem() then user selects

SGGESX	Computes the generalized eigenvalues, the real Schur form, and optionally, the left and/or right matrices of Schur vectors.	No direct support, but can use eig(), shur(), then user selects	No direct support, but can use Eigensystem[], Schur-Decomposition[], then user selects	No support for generalized eigenvalues. No shur decomposition
SGEEVX	Computes the eigenvalues and left and right eigenvectors of a general matrix, with preliminary balancing of the matrix, and computes reciprocal condition numbers for the eigenvalues and right eigenvectors.	No direct support, but can use eig() and cond()	No direct support, but can use Eigensystem[], and LinearAlgebra‘MatrixConditionNumber[A]	No support but can use eigensystem(), no condition number.
SGGEVX	Computes the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors.	[V,D]=eig(A,B,'chol')	[D,V]=Eigensystem[A,B]	No support for generalized eigenvalues

references

- `http://www.netlib.org/lapack/individualroutines.html`
- Ada 2005 reference manual
- Mathematica and Matlab help