Prefixes
Each routine has a prefix, denoted by a hyphen - in this guide, made of 3 letters xyy, where
x is the data type, and yy is the matrix type.

Data type
gs s single
gc c complex single
gd d double
gz z complex double

Matrix type
full banded packed tridiag generalized problem
---

<table>
<thead>
<tr>
<th>Matrix type</th>
<th>full</th>
<th>banded</th>
<th>packed</th>
<th>tridiag</th>
<th>generalized problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>ge</td>
<td>gb</td>
<td>sp</td>
<td>gt</td>
<td>gg</td>
</tr>
<tr>
<td>symmetric</td>
<td>sy</td>
<td>sb</td>
<td>sp</td>
<td>st</td>
<td></td>
</tr>
<tr>
<td>Hermitian</td>
<td>he</td>
<td>hb</td>
<td>hp</td>
<td>pt</td>
<td></td>
</tr>
<tr>
<td>SPD / HPD</td>
<td>po</td>
<td>pb</td>
<td>pp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triangular</td>
<td>tr</td>
<td>tb</td>
<td>tp</td>
<td></td>
<td>tg</td>
</tr>
<tr>
<td>upper Hessenberg</td>
<td>hs</td>
<td></td>
<td></td>
<td></td>
<td>hg</td>
</tr>
<tr>
<td>trapezoidal</td>
<td>tz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>orthogonal</td>
<td>or</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unitary</td>
<td>un</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diagonal</td>
<td>di</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bidiagonal</td>
<td>bd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For symmetric, Hermitian, and triangular matrices, elements below/above the diagonal
(for upper/lower respectively) are not accessed. Similarly for upper Hessenberg, elements
below the subdiagonal are not accessed.

Packed storage is by columns. For example, a $3 \times 3$ upper triangular is stored as

$$
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  & a_{22} & a_{23} \\
  & & a_{33}
\end{bmatrix}
$$

Banded storage puts columns of the matrix in corresponding columns of the array, and
diagonals in rows of the array, for example:

$$
\begin{bmatrix}
  * & a_{12} & a_{23} & a_{34} & a_{45} \\
  a_{11} & * & a_{22} & a_{33} & a_{44} \\
  a_{21} & a_{32} & * & a_{43} & a_{54} \\
  a_{31} & a_{42} & a_{53} & * & *
\end{bmatrix}
$$

1st diagonal
2nd (main) diagonal
3rd diagonal
4th diagonal

Bi- and tridiagonal matrices are stored as 2 or 3 vectors of length $n$ and $n - 1$.

Updated March 19, 2008.
BLAS and LAPACK guides available from http://www.ews.uiuc.edu/~mrgates2/.
Copyright ©2007–2008 by Mark Gates. This document is free; you can redistribute it
under terms of the GNU General Public License, version 2 or later.
Drivers

Drivers are higher level routines that solve an entire problem.

Linear system, solve $Ax = b$.
- $sv$ — solve
- $svx$ — expert; also $A^T x = b$ or $A^H x = b$, condition number, error bounds, scaling
Matrix types [General ge, gb, gt; SPD po, pp, pb, pt; Symmetric sy, sp, he, hp]

Linear least squares, minimize $\|b - Ax\|_2$.
- ls — full rank, rank$(A) = \min(m,n)$, uses $QR$.
- lsy — rank deficient, uses complete orthogonal factorization.
- lsd — rank deficient, uses SVD.
Matrix types [General ge]

Generalized linear least squares.
Minimize $\|c - Ax\|_2$ subject to $Bx = d$.
- 1se — $B$ full row rank, matrix $\begin{bmatrix} A \\ B \end{bmatrix}$ full col rank.

Minimize $\|y\|_2$ subject to $d = Ax + By$.
- glm — $A$ full col rank, matrix $[A \ B]$ full row rank.
Matrix types [General gg]

Eigenvalues, solve $Ax = \lambda x$.
Symmetric
- $ev$ — all eigenvalues, [eigenvectors]
- $evx$ — expert; also subset
- $evd$ — divide-and-conquer; faster but more memory
- $evr$ — relative robust; fastest and least memory
Matrix types [Symmetric sy, sp, sb, st, he, hp, hb]

Non-symmetric
- $ev$ — eigenvalues, [left, right eigenvectors]
- $evx$ — expert; also balance matrix, condition numbers
- $es$ — Schur factorization
- $esx$ — expert; also condition numbers
Matrix types [General gg]

Generalized eigenvalue, solve $Ax = \lambda Bx$.
Symmetric, $B$ SPD
- $gv$ — all eigenvalues, [eigenvectors]
- $gvx$ — expert; also subset
- $gvd$ — divide-and-conquer, faster but more memory
Matrix types [Symmetric sy, sp, sb, st, he, hp, hb]

Non-symmetric
- $ev$ — eigenvalues, [left, right eigenvectors]
- $evx$ — expert; also balance matrix, condition numbers
- $es$ — Schur factorization
- $esx$ — expert; also condition numbers
Matrix types [General gg]

SVD singular value decomposition, $A = U \Sigma V^H$
- $svd$ — singular values, [left, right vectors]
- $sdd$ — divide-and-conquer; faster but more memory
Matrix types [General ge]

Generalized SVD, $A = U \Sigma_1 Q^T$ and $B = V \Sigma_2 Q^T$
- $svd$ — singular values, [left, right vectors]
Matrix types [General gg]
Computational routines
Computational routines perform one step of solving the problem. Drivers call a sequence of computational routines.

Triangular factorization
-`trf` — factorize: General $LU$, Cholesky $LL^T$, triadi $LDL^T$, sym. indefinite $LDL^T$
-`trs` — solve using factorization
-`con` — condition number estimate
-`rfs` — error bounds, iterative refinement
-`tri` — inverse (not for band)
-`equ` — equilibrate $A$ (not for tridiag, symmetric indefinite, triangular)
Matrix types [General ge, gb, gt; SPD po, pp, pb, pt; Symmetric sy, sp, he, hp, Triangular tr, tb]

Orthogonal factorization
-`qp3` — QR factorization, with pivoting
-`qrf` — QR factorization
-`qrq` — QR factorization
-`qlf` — $QL$ factorization
-`lqf` — $LQ$ factorization
-`tqrf` — $RQ$ factorization
-`tqzf` — $RZ$ trapezoidal factorization
Matrix types [General ge, some Trapezoidal tz]

Eigenvalue
-`trd` — tridiagonal reduction
Matrix types [Symmetric sy, he, sp, hp]
-`gtr` — generate matrix after `-trd`
-`mtr` — multiply matrix after `-trd`
Matrix types [Orthogonal or, op, un, up]
Symmetric tridiagonal eigensolvers
-`eqr` — using implicitly shifted QR
-`pteqr` — using Cholesky and bidiagonal QR
-`erf` — using square-root free QR
-`edc` — using divide-and-conquer
-`egr` — using relatively robust representation
-`ebz` — eigenvalues using bisection
-`ein` — eigenvectors using inverse iteration
Matrix types [Symmetric tridiag st, one SPD pt]

Nonsymmetric
-`hrd` — Hessenberg reduction
-`bal` — balance
-`bak` — back transforming
Matrix types [General ge]
-`ghr` — generate matrix after `-hrd`
-`mhr` — multiply matrix after `-hrd`
Matrix types [Orthogonal or, un]
-`eqr` — Schur factorization
-`ein` — eigenvectors using inverse iteration
Matrix types [upper Hessenberg hs]
-`evc` — eigenvectors
-`exc` — reorder Schur factorization
-`syl` — Sylvester equation
-`ena` — condition numbers
-`sen` — condition numbers of eigenvalue cluster/subspace
Matrix types [Triangular tr]