API Collection

BLAS

Perform common linear algebra operations with Apple's implementation of the Basic Linear Algebra Subprograms (BLAS).

Overview

The vecLib framework contains nine C header files (not counting `vecLib.h`, which merely includes the others).

This document describes the functions declared in the header files `cblas.h` and `vblas.h`, which contain the interfaces for Apple's implementation of the BLAS API.

Note that documentation describing the leading dimension as the first dimension of a matrix refers to column-major ordering. In row-major ordering, the leading dimension is the second dimension of a matrix.

> **Important**
>
> Apple provides the BLAS and LAPACK libraries under the Accelerate framework to be in line with LAPACK 3.9.1. Starting with iOS 19, iPadOS 19, macOS 16, tvOS 19, watchOS 19, and visionOS 3, the libraries are in line with LAPACK 3.12.0. These new interfaces provide additional functionality, as well as a new ILP64 interface. To use the new interfaces, define `ACCELERATE_NEW_LAPACK` before including the Accelerate or vecLib headers. For ILP64 interfaces, also define `ACCELERATE_LAPACK_ILP64`. For Swift projects, specify `ACCELERATE_NEW_LAPACK=1` and `ACCELERATE_LAPACK_ILP64=1` as preprocessor macros in Xcode build settings.

Topics

Specifying the threading model

`struct BLAS`

An enumeration that acts as a namespace for Swift overlays to BLAS.

`func BLASSetThreading(BLAS_THREADING) -> Int32`

Sets the BLAS and LAPACK threading model.

`func BLASGetThreading() -> BLAS_THREADING`

Returns the current BLAS and LAPACK threading model.

`struct BLAS_THREADING`

Constants that describe the BLAS and LAPACK threading model.

General functions

`SetBLASParamErrorProc`

Sets an error handler function.

`cblas_errprn`

Prints an error message.

`cblas_xerbla`

The default error handler for BLAS routines.

`func cblas_icamax(__LAPACK_int, OpaquePointer?, __LAPACK_int) -> __LAPACK_int`

Returns the index of the element with the largest absolute value in a vector (single-precision complex).

`func cblas_idamax(__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int) -> __LAPACK_int`

Returns the index of the element with the largest absolute value in a vector (double-precision).

`func cblas_isamax(__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int) -> __LAPACK_int`

Returns the index of the element with the largest absolute value in a vector (single-precision).

`func cblas_izamax(__LAPACK_int, OpaquePointer?, __LAPACK_int) -> __LAPACK_int`

Returns the index of the element with the largest absolute value in a vector (double-precision complex).

Sparse computation

- Matrix and Vector Operations

  Perform computations with matrices and vectors.

- Pointwise Matrix Operations

  Create, insert values into, and extract values from a pointwise sparse matrix.

- Blockwise Matrix Operations

  Create, insert values into, and extract values from a blockwise sparse matrix.

- General Sparse Matrix Management Operations

  Manage and work with the properties of a sparse matrix.

- Sparse Vector Utility Operations

  Create and work with sparse vector structures.

Data types

`typealias BLASParamErrorProc`

A BLAS error handler callback type.

Constants

Discussion

> **Note**
>
> The types given here are valid for C or C++ and for either PowerPC or Intel processors. The typedefs shown are for C++ and PowerPC processors; for other, conditionally compiled typedefs, see the header files.

`struct CBLAS_ORDER`

`struct CBLAS_TRANSPOSE`

```
struct CBLAS_UPLO
```

```
struct CBLAS_DIAG
```

```
struct CBLAS_SIDE
```

Variables

```
var CblasColMajor: CBLAS_ORDER
```

```
var CblasConjTrans: CBLAS_TRANSPOSE
```

```
var CblasLeft: CBLAS_SIDE
```

```
var CblasLower: CBLAS_UPLO
```

```
var CblasNoTrans: CBLAS_TRANSPOSE
```

```
var CblasNonUnit: CBLAS_DIAG
```

```
var CblasRight: CBLAS_SIDE
```

```
var CblasRowMajor: CBLAS_ORDER
```

```
var CblasTrans: CBLAS_TRANSPOSE
```

```
var CblasUnit: CBLAS_DIAG
```

```
var CblasUpper: CBLAS_UPLO
```

```
var AtlasConj: CBLAS_TRANSPOSE
```

CATLAS and CBLAS vector functions

```
func catlas_caxpby(__LAPACK_int, OpaquePointer, OpaquePointer?, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Computes the product of two vectors, scaling each one separately (single-precision complex).

```
func catlas_cset(__LAPACK_int, OpaquePointer, OpaquePointer, __LAPACK
_int)
```

Modifies a vector (single-precision complex) in place, setting each element to a given value.

```
func catlas_daxpby(__LAPACK_int, Double, UnsafePointer<Double>?, _
_LAPACK_int, Double, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Computes the sum of two vectors, scaling each one separately (double-precision).

```
func catlas_dset(__LAPACK_int, Double, UnsafeMutablePointer<Double>, _
_LAPACK_int)
```

Modifies a vector (double-precision) in place, setting each element to a given value.

```
func catlas_saxpby(__LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK
_int, Float, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Computes the sum of two vectors, scaling each one separately (single-precision).

```
func catlas_sset(__LAPACK_int, Float, UnsafeMutablePointer<Float>, _
_LAPACK_int)
```

Modifies a vector (single-precision) in place, setting each element to a given value.

```
func catlas_zaxpby(__LAPACK_int, OpaquePointer, OpaquePointer?, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Computes the sum of two vectors, scaling each one separately (double-precision complex).

```
func catlas_zset(__LAPACK_int, OpaquePointer, OpaquePointer, __LAPACK
_int)
```

Modifies a vector (double-precision complex) in place, setting each element to a given value.

```
func cblas_sdot(__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
UnsafePointer<Float>?, __LAPACK_int) -> Float
```

Computes the dot product of two vectors (single-precision).

```
func cblas_sdsdot(__LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK
_int, UnsafePointer<Float>?, __LAPACK_int) -> Float
```

Computes the dot product of two single-precision vectors plus an initial single-precision value.

```
func cblas_cdotc_sub(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, OpaquePointer)
```

Calculates the dot product of the complex conjugate of a single-precision complex vector with a second single-precision complex vector.

```
func cblas_cdotu_sub(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, OpaquePointer)
```

Computes the dot product of two single-precision complex vectors.

```
func cblas_ddot(__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafePointer<Double>?, __LAPACK_int) -> Double
```

Computes the dot product of two vectors (double-precision).

```
func cblas_dsdot(__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
UnsafePointer<Float>?, __LAPACK_int) -> Double
```

Computes the double-precision dot product of a pair of single-precision vectors.

```
func cblas_zdotc_sub(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, OpaquePointer)
```

Calculates the dot product of the complex conjugate of a double-precision complex vector with a second double-precision complex vector.

```
func cblas_zdotu_sub(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, OpaquePointer)
```

Computes the dot product of two double-precision complex vectors.

Single-precision float matrix functions

```
func cblas_sasum(__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int) ->
Float
```

Computes the sum of the absolute values of elements in a vector (single-precision).

```
func cblas_saxpy(__LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK
_int, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Computes a constant times a vector plus a vector (single-precision).

```
func cblas_scopy(__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Copies a vector to another vector (single-precision).

```
func cblas_sgbmv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, __LAPACK_int, __LAPACK_int, Float, UnsafePointer<Float>?, _
_LAPACK_int, UnsafePointer<Float>?, __LAPACK_int, Float, UnsafeMutable
Pointer<Float>?, __LAPACK_int)
```

Scales a general band matrix, then multiplies by a vector, then adds a vector (single precision).

```
func cblas_sgemm(CBLAS_ORDER, CBLAS_TRANSPOSE, CBLAS_TRANSPOSE, _
_LAPACK_int, __LAPACK_int, __LAPACK_int, Float, UnsafePointer<Float>?,
__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int, Float, UnsafeMutable
Pointer<Float>?, __LAPACK_int)
```

Multiplies two matrices (single-precision).

```
func cblas_sgemv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, Float, UnsafePointer<Float>?, __LAPACK_int, UnsafePointer<Float
>?, __LAPACK_int, Float, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Multiplies a single-precision matrix by a vector.

```
func cblas_sger(CBLAS_ORDER, __LAPACK_int, __LAPACK_int, Float, Unsafe
Pointer<Float>?, __LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Multiplies vector X by the transpose of vector Y, then adds matrix A (single precison).

```
func cblas_snrm2(__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int) ->
Float
```

Computes the L2 norm (Euclidian length) of a vector (single precision).

```
func cblas_srot(__LAPACK_int, UnsafeMutablePointer<Float>?, __LAPACK
_int, UnsafeMutablePointer<Float>?, __LAPACK_int, Float, Float)
```

Applies a Givens rotation matrix to a pair of vectors.

```
func cblas_srotg(UnsafeMutablePointer<Float>, UnsafeMutablePointer<
Float>, UnsafeMutablePointer<Float>, UnsafeMutablePointer<Float>)
```

Constructs a Givens rotation matrix.

```
func cblas_srotm(__LAPACK_int, UnsafeMutablePointer<Float>?, __LAPACK
_int, UnsafeMutablePointer<Float>?, __LAPACK_int, UnsafePointer<Float>)
```

Applies a modified Givens transformation (single precision).

```
func cblas_srotmg(UnsafeMutablePointer<Float>, UnsafeMutablePointer<
Float>, UnsafeMutablePointer<Float>, Float, UnsafeMutablePointer<Float
>)
```

Generates a modified Givens rotation matrix.

```
func cblas_ssbmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, __LAPACK_int,
Float, UnsafePointer<Float>?, __LAPACK_int, UnsafePointer<Float>?, _
_LAPACK_int, Float, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Scales a symmetric band matrix, then multiplies by a vector, then adds a vector (single-precision).

```
func cblas_sscal(__LAPACK_int, Float, UnsafeMutablePointer<Float>?, _
_LAPACK_int)
```

Multiplies each element of a vector by a constant (single-precision).

```
func cblas_sspmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Unsafe
Pointer<Float>?, UnsafePointer<Float>?, __LAPACK_int, Float, Unsafe
MutablePointer<Float>?, __LAPACK_int)
```

Scales a packed symmetric matrix, then multiplies by a vector, then scales and adds another vector (single precision).

```
func cblas_sspr(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Unsafe
Pointer<Float>?, __LAPACK_int, UnsafeMutablePointer<Float>?)
```

Rank one update: adds a packed symmetric matrix to the product of a scaling factor, a vector, and its transpose (single precision).

```
func cblas_sspr2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Unsafe
Pointer<Float>?, __LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
UnsafeMutablePointer<Float>?)
```

Rank two update of a packed symmetric matrix using two vectors (single precision).

```
func cblas_sswap(__LAPACK_int, UnsafeMutablePointer<Float>?, __LAPACK
_int, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Exchanges the elements of two vectors (single precision).

```
func cblas_ssymm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, __LAPACK_int, _
_LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK_int, UnsafePointer<
Float>?, __LAPACK_int, Float, UnsafeMutablePointer<Float>?, __LAPACK
_int)
```

Multiplies a matrix by a symmetric matrix (single-precision).

```
func cblas_ssymv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Unsafe
Pointer<Float>?, __LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
Float, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Scales a symmetric matrix, multiplies by a vector, then scales and adds another vector (single precision).

```
func cblas_ssyr(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Unsafe
Pointer<Float>?, __LAPACK_int, UnsafeMutablePointer<Float>?, __LAPACK
_int)
```

Rank one update: adds a symmetric matrix to the product of a scaling factor, a vector, and its transpose (single precision).

```
func cblas_ssyr2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Unsafe
Pointer<Float>?, __LAPACK_int, UnsafePointer<Float>?, __LAPACK_int,
UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Rank two update of a symmetric matrix using two vectors (single precision).

```
func cblas_ssyr2k(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK
_int, __LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK_int, Unsafe
Pointer<Float>?, __LAPACK_int, Float, UnsafeMutablePointer<Float>?, _
_LAPACK_int)
```

Performs a rank-2k update of a symmetric matrix (single precision).

```
func cblas_ssyrk(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK_int
, __LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK_int, Float,
UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Rank-k update—multiplies a symmetric matrix by its transpose and adds a second matrix (single precision).

```
func cblas_stbmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, UnsafePointer<Float>?, __LAPACK_int, Unsafe
MutablePointer<Float>?, __LAPACK_int)
```

Scales a triangular band matrix, then multiplies by a vector (single precision).

```
func cblas_stbsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, UnsafePointer<Float>?, __LAPACK_int, Unsafe
MutablePointer<Float>?, __LAPACK_int)
```

Solves a triangular banded system of equations.

```
func cblas_stpmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Float>?, UnsafeMutablePointer<Float>?, _
_LAPACK_int)
```

Multiplies a triangular matrix by a vector, then adds a vector (single precision).

```
func cblas_stpsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Float>?, UnsafeMutablePointer<Float>?, _
_LAPACK_int)
```

Solves a packed triangular system of equations.

```
func cblas_strmm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, Float, UnsafePointer<Float>?, _
_LAPACK_int, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Scales a triangular matrix and multiplies it by a matrix.

```
func cblas_strmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int, UnsafeMutablePointer
<Float>?, __LAPACK_int)
```

Multiplies a triangular matrix by a vector.

```
func cblas_strsm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, Float, UnsafePointer<Float>?, _
_LAPACK_int, UnsafeMutablePointer<Float>?, __LAPACK_int)
```

Solves a triangular system of equations with multiple values for the right side.

```
func cblas_strsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Float>?, __LAPACK_int, UnsafeMutablePointer
<Float>?, __LAPACK_int)
```

Solves a triangular system of equations with a single value for the right side.

```
func appleblas_sgeadd(CBLAS_ORDER, CBLAS_TRANSPOSE, CBLAS_TRANSPOSE, _
_LAPACK_int, __LAPACK_int, Float, UnsafePointer<Float>?, __LAPACK_int,
Float, UnsafePointer<Float>?, __LAPACK_int, UnsafeMutablePointer<Float
>, __LAPACK_int)
```

Single-precision complex matrix functions

```
func cblas_caxpy(__LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK
_int, OpaquePointer?, __LAPACK_int)
```

Computes a constant times a vector plus a vector (single-precision complex).

```
func cblas_ccopy(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Copies a vector to another vector (single-precision complex).

```
func cblas_cgbmv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?, _
_LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer, OpaquePointer
?, __LAPACK_int)
```

Scales a general band matrix, then multiplies by a vector, then adds a vector (single-precision complex).

```
func cblas_cgemm(CBLAS_ORDER, CBLAS_TRANSPOSE, CBLAS_TRANSPOSE, _
_LAPACK_int, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?,
__LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer, Opaque
Pointer?, __LAPACK_int)
```

Multiplies two matrices (single-precision complex).

```
func cblas_cgemv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer?, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Multiplies a matrix by a vector (single-precision complex).

```
func cblas_cgerc(CBLAS_ORDER, __LAPACK_int, __LAPACK_int, OpaquePointer
, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Multiplies vector X by the conjugate transpose of vector Y, then adds matrix A (single-precision complex).

```
func cblas_cgeru(CBLAS_ORDER, __LAPACK_int, __LAPACK_int, OpaquePointer
, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Multiplies vector X by the transpose of vector Y, then adds matrix A (single-precision complex).

```
func cblas_chbmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, __LAPACK_int,
OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK
_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Scales a Hermitian band matrix, then multiplies by a vector, then adds a vector (single-precision complex).

```
func cblas_chemm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, __LAPACK_int, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Multiplies two Hermitian matrices (single-precision complex), then adds a third (with scaling).

```
func cblas_chemv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer, OpaquePointer?, __LAPACK_int)
```

Scales and multiplies a Hermitian matrix by a vector, then adds a second (scaled) vector.

```
func cblas_cher(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Opaque
Pointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int)
```

Hermitian rank 1 update: adds the product of a scaling factor, vector X, and the conjugate transpose of X to matrix A.

```
func cblas_cher2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Hermitian rank 2 update: adds the product of a scaling factor, vector X, and the conjugate transpose of vector Y to the product of the conjugate of the scaling factor, vector Y, and the conjugate transpose of vector X, and adds the result to matrix A.

```
func cblas_cher2k(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK
_int, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, Float, OpaquePointer?, __LAPACK_int)
```

Performs a rank-2k update of a complex Hermitian matrix (single-precision complex).

```
func cblas_cherk(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK_int
, __LAPACK_int, Float, OpaquePointer?, __LAPACK_int, Float, Opaque
Pointer?, __LAPACK_int)
```

Rank-k update—multiplies a Hermitian matrix by its transpose and adds a second matrix (single precision).

```
func cblas_chpmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, OpaquePointer?, __LAPACK_int, OpaquePointer, Opaque
Pointer?, __LAPACK_int)
```

Scales a packed hermitian matrix, multiplies it by a vector, and adds a scaled vector.

```
func cblas_chpr(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Float, Opaque
Pointer?, __LAPACK_int, OpaquePointer?)
```

Scales and multiplies a vector times its conjugate transpose, then adds a matrix.

```
func cblas_chpr2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?)
```

Multiplies a vector times the conjugate transpose of a second vector and vice-versa, sums the results, and adds a matrix.

```
func cblas_crotg(OpaquePointer, OpaquePointer, UnsafeMutablePointer<
Float>, OpaquePointer)
```

Constructs a complex Givens rotation.

```
func cblas_cscal(__LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK
_int)
```

Multiplies each element of a vector by a constant (single-precision complex).

```
func cblas_csrot(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, Float, Float)
```

Applies a Givens rotation matrix to a pair of complex vectors.

```
func cblas_csscal(__LAPACK_int, Float, OpaquePointer?, __LAPACK_int)
```

Multiplies each element of a vector by a constant (single-precision complex).

```
func cblas_cswap(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Exchanges the elements of two vectors (single-precision complex).

```
func cblas_csymm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, __LAPACK_int, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Multiplies a matrix by a symmetric matrix (single-precision complex).

```
func cblas_csyr2k(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK
_int, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Performs a rank-2k update of a symmetric matrix (single-precision complex).

```
func cblas_csyrk(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK_int
, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, Opaque
Pointer, OpaquePointer?, __LAPACK_int)
```

Rank-k update—multiplies a symmetric matrix by its transpose and adds a second matrix (single-precision complex).

```
func cblas_ctbmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int)
```

Scales a triangular band matrix, then multiplies by a vector (single-precision compex).

```
func cblas_ctbsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int)
```

Solves a triangular banded system of equations.

```
func cblas_ctpmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, OpaquePointer?, __LAPACK_int)
```

Multiplies a triangular matrix by a vector, then adds a vector (single-precision complex).

```
func cblas_ctpsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, OpaquePointer?, __LAPACK_int)
```

Solves a packed triangular system of equations.

```
func cblas_ctrmm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?,
__LAPACK_int, OpaquePointer?, __LAPACK_int)
```

Scales a triangular matrix and multiplies it by a matrix.

```
func cblas_ctrmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK
_int)
```

Multiplies a triangular matrix by a vector.

```
func cblas_ctrsm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?,
__LAPACK_int, OpaquePointer?, __LAPACK_int)
```

Solves a triangular system of equations with multiple values for the right side.

```
func cblas_ctrsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK
_int)
```

Solves a triangular system of equations with a single value for the right side.

```
func cblas_scasum(__LAPACK_int, OpaquePointer?, __LAPACK_int) -> Float
```

Computes the sum of the absolute values of real and imaginary parts of elements in a vector (single-precision complex).

```
func cblas_scnrm2(__LAPACK_int, OpaquePointer?, __LAPACK_int) -> Float
```

Computes the unitary norm of a vector (single-precision complex).

Double-precision float matrix functions

```
func cblas_dasum(__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int) ->
Double
```

Computes the sum of the absolute values of elements in a vector (double-precision).

```
func cblas_daxpy(__LAPACK_int, Double, UnsafePointer<Double>?, __LAPACK
_int, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Computes a constant times a vector plus a vector (double-precision).

```
func cblas_dcopy(__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Copies a vector to another vector (double-precision).

```
func cblas_dgbmv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, __LAPACK_int, __LAPACK_int, Double, UnsafePointer<Double>?, _
_LAPACK_int, UnsafePointer<Double>?, __LAPACK_int, Double, Unsafe
MutablePointer<Double>?, __LAPACK_int)
```

Scales a general band matrix, then multiplies by a vector, then adds a vector (double precision).

```
func cblas_dgemm(CBLAS_ORDER, CBLAS_TRANSPOSE, CBLAS_TRANSPOSE, _
_LAPACK_int, __LAPACK_int, __LAPACK_int, Double, UnsafePointer<Double
>?, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int, Double, Unsafe
MutablePointer<Double>?, __LAPACK_int)
```

Multiplies two matrices (double-precision).

```
func cblas_dgemv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, Double, UnsafePointer<Double>?, __LAPACK_int, UnsafePointer<
Double>?, __LAPACK_int, Double, UnsafeMutablePointer<Double>?, __LAPACK
_int)
```

Multiplies a matrix by a vector (double precision).

```
func cblas_dger(CBLAS_ORDER, __LAPACK_int, __LAPACK_int, Double, Unsafe
Pointer<Double>?, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Multiplies vector X by the transpose of vector Y, then adds matrix A (double precison).

```
func cblas_dnrm2(__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int) ->
Double
```

Computes the L2 norm (Euclidian length) of a vector (double precision).

```
func cblas_drot(__LAPACK_int, UnsafeMutablePointer<Double>?, __LAPACK
_int, UnsafeMutablePointer<Double>?, __LAPACK_int, Double, Double)
```

Applies a Givens rotation matrix to a pair of vectors.

```
func cblas_drotg(UnsafeMutablePointer<Double>, UnsafeMutablePointer<
Double>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<Double>)
```

Constructs a Givens rotation matrix.

```
func cblas_drotm(__LAPACK_int, UnsafeMutablePointer<Double>?, __LAPACK
_int, UnsafeMutablePointer<Double>?, __LAPACK_int, UnsafePointer<Double
>)
```

Applies a modified Givens transformation (single precision).

```
func cblas_drotmg(UnsafeMutablePointer<Double>, UnsafeMutablePointer<
Double>, UnsafeMutablePointer<Double>, Double, UnsafeMutablePointer<
Double>)
```

Generates a modified Givens rotation matrix.

```
func cblas_dsbmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, __LAPACK_int,
Double, UnsafePointer<Double>?, __LAPACK_int, UnsafePointer<Double>?, _
_LAPACK_int, Double, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Scales a symmetric band matrix, then multiplies by a vector, then adds a vector (double precision).

```
func cblas_dscal(__LAPACK_int, Double, UnsafeMutablePointer<Double>?, _
_LAPACK_int)
```

Multiplies each element of a vector by a constant (double-precision).

```
func cblas_dspmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Unsafe
Pointer<Double>?, UnsafePointer<Double>?, __LAPACK_int, Double, Unsafe
MutablePointer<Double>?, __LAPACK_int)
```

Scales a packed symmetric matrix, then multiplies by a vector, then scales and adds another vector (double precision).

```
func cblas_dspr(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Unsafe
Pointer<Double>?, __LAPACK_int, UnsafeMutablePointer<Double>?)
```

Rank one update: adds a packed symmetric matrix to the product of a scaling factor, a vector, and its transpose (double precision).

```
func cblas_dspr2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Unsafe
Pointer<Double>?, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafeMutablePointer<Double>?)
```

Rank two update of a packed symmetric matrix using two vectors (single precision).

```
func cblas_dswap(__LAPACK_int, UnsafeMutablePointer<Double>?, __LAPACK
_int, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Exchanges the elements of two vectors (double precision).

```
func cblas_dsymm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, __LAPACK_int, _
_LAPACK_int, Double, UnsafePointer<Double>?, __LAPACK_int, Unsafe
Pointer<Double>?, __LAPACK_int, Double, UnsafeMutablePointer<Double>?,
__LAPACK_int)
```

Multiplies a matrix by a symmetric matrix (double-precision).

```
func cblas_dsymv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Unsafe
Pointer<Double>?, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
Double, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Scales a symmetric matrix, multiplies by a vector, then scales and adds another vector (single precision).

```
func cblas_dsyr(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Unsafe
Pointer<Double>?, __LAPACK_int, UnsafeMutablePointer<Double>?, __LAPACK
_int)
```

Rank one update: adds a symmetric matrix to the product of a scaling factor, a vector, and its transpose (double precision).

```
func cblas_dsyr2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Unsafe
Pointer<Double>?, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Rank two update of a symmetric matrix using two vectors (single precision).

```
func cblas_dsyr2k(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK
_int, __LAPACK_int, Double, UnsafePointer<Double>?, __LAPACK_int,
UnsafePointer<Double>?, __LAPACK_int, Double, UnsafeMutablePointer<
Double>?, __LAPACK_int)
```

Performs a rank-2k update of a symmetric matrix (double precision).

```
func cblas_dsyrk(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK_int
, __LAPACK_int, Double, UnsafePointer<Double>?, __LAPACK_int, Double,
UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Rank-k update—multiplies a symmetric matrix by its transpose and adds a second matrix (double precision).

```
func cblas_dtbmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Scales a triangular band matrix, then multiplies by a vector (double precision).

```
func cblas_dtbsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, UnsafePointer<Double>?, __LAPACK_int,
UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Solves a triangular banded system of equations.

```
func cblas_dtpmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Double>?, UnsafeMutablePointer<Double>?, _
_LAPACK_int)
```

Multiplies a triangular matrix by a vector, then adds a vector (double precision).

```
func cblas_dtpsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Double>?, UnsafeMutablePointer<Double>?, _
_LAPACK_int)
```

Solves a packed triangular system of equations.

```
func cblas_dtrmm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, Double, UnsafePointer<Double>?,
__LAPACK_int, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Scales a triangular matrix and multiplies it by a matrix.

```
func cblas_dtrmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int, UnsafeMutable
Pointer<Double>?, __LAPACK_int)
```

Multiplies a triangular matrix by a vector.

```
func cblas_dtrsm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, Double, UnsafePointer<Double>?,
__LAPACK_int, UnsafeMutablePointer<Double>?, __LAPACK_int)
```

Solves a triangular system of equations with multiple values for the right side.

```
func cblas_dtrsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, UnsafePointer<Double>?, __LAPACK_int, UnsafeMutable
Pointer<Double>?, __LAPACK_int)
```

Solves a triangular system of equations with a single value for the right side.

```
func appleblas_dgeadd(CBLAS_ORDER, CBLAS_TRANSPOSE, CBLAS_TRANSPOSE, _
_LAPACK_int, __LAPACK_int, Double, UnsafePointer<Double>?, __LAPACK_int
, Double, UnsafePointer<Double>?, __LAPACK_int, UnsafeMutablePointer<
Double>, __LAPACK_int)
```

Double-precision complex matrix functions

```
func cblas_dzasum(__LAPACK_int, OpaquePointer?, __LAPACK_int) -> Double
```

Computes the sum of the absolute values of real and imaginary parts of elements in a vector (single-precision complex).

```
func cblas_dznrm2(__LAPACK_int, OpaquePointer?, __LAPACK_int) -> Double
```

Computes the unitary norm of a vector (double-precision complex).

```
func cblas_zaxpy(__LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK
_int, OpaquePointer?, __LAPACK_int)
```

Computes a constant times a vector plus a vector (double-precision complex).

```
func cblas_zcopy(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Copies a vector to another vector (double-precision complex).

```
func cblas_zdrot(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, Double, Double)
```

Applies a Givens rotation matrix to a pair of complex vectors.

```
func cblas_zdscal(__LAPACK_int, Double, OpaquePointer?, __LAPACK_int)
```

Multiplies each element of a vector by a constant (double-precision complex).

```
func cblas_zgbmv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?, _
_LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer, OpaquePointer
?, __LAPACK_int)
```

Scales a general band matrix, then multiplies by a vector, then adds a vector (double-precision complex).

```
func cblas_zgemm(CBLAS_ORDER, CBLAS_TRANSPOSE, CBLAS_TRANSPOSE, _
_LAPACK_int, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?,
__LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer, Opaque
Pointer?, __LAPACK_int)
```

Multiplies two matrices (double-precision complex).

```
func cblas_zgemv(CBLAS_ORDER, CBLAS_TRANSPOSE, __LAPACK_int, __LAPACK
_int, OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer?, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Multiplies a matrix by a vector (double-precision complex).

```
func cblas_zgerc(CBLAS_ORDER, __LAPACK_int, __LAPACK_int, OpaquePointer
, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Multiplies vector X by the conjugate transpose of vector Y, then adds matrix A (double-precision complex).

```
func cblas_zgeru(CBLAS_ORDER, __LAPACK_int, __LAPACK_int, OpaquePointer
, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Multiplies vector X by the transpose of vector Y, then adds matrix A (double-precision complex).

```
func cblas_zhbmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, __LAPACK_int,
OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK
_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Scales a Hermitian band matrix, then multiplies by a vector, then adds a vector (double-precision complex).

```
func cblas_zhemm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, __LAPACK_int, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Multiplies two Hermitian matrices (double-precision complex).

```
func cblas_zhemv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer, OpaquePointer?, __LAPACK_int)
```

Scales and multiplies a Hermitian matrix by a vector, then adds a second (scaled) vector.

```
func cblas_zher(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Opaque
Pointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int)
```

Adds the product of a scaling factor, vector X, and the conjugate transpose of X to matrix A.

```
func cblas_zher2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Hermitian rank 2 update: adds the product of a scaling factor, vector X, and the conjugate transpose of vector Y to the product of the conjugate of the scaling factor, vector Y, and the conjugate transpose of vector X, and adds the result to matrix A.

```
func cblas_zher2k(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK
_int, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, Double, OpaquePointer?, __LAPACK_int)
```

Performs a rank-2k update of a complex Hermitian matrix (double-precision complex).

```
func cblas_zherk(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK_int
, __LAPACK_int, Double, OpaquePointer?, __LAPACK_int, Double, Opaque
Pointer?, __LAPACK_int)
```

Rank-k update—multiplies a Hermitian matrix by its transpose and adds a second matrix (single precision).

```
func cblas_zhpmv(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, OpaquePointer?, __LAPACK_int, OpaquePointer, Opaque
Pointer?, __LAPACK_int)
```

Scales a packed hermitian matrix, multiplies it by a vector, and adds a scaled vector.

```
func cblas_zhpr(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, Double, Opaque
Pointer?, __LAPACK_int, OpaquePointer?)
```

Scales and multiplies a vector times its conjugate transpose, then adds a matrix.

```
func cblas_zhpr2(CBLAS_ORDER, CBLAS_UPLO, __LAPACK_int, OpaquePointer,
OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?)
```

Multiplies a vector times the conjugate transpose of a second vector and vice-versa, sums the results, and adds a matrix.

```
func cblas_zrotg(OpaquePointer, OpaquePointer, UnsafeMutablePointer<
Double>, OpaquePointer)
```

Constructs a complex Givens rotation.

```
func cblas_zscal(__LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK
_int)
```

Multiplies each element of a vector by a constant (double-precision complex).

```
func cblas_zswap(__LAPACK_int, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int)
```

Exchanges the elements of two vectors (double-precision complex).

```
func cblas_zsymm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, __LAPACK_int, _
_LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Multiplies a matrix by a symmetric matrix (double-precision complex).

```
func cblas_zsyr2k(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK
_int, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, Opaque
Pointer?, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int)
```

Performs a rank-2k update of a symmetric matrix (double-precision complex).

```
func cblas_zsyrk(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, __LAPACK_int
, __LAPACK_int, OpaquePointer, OpaquePointer?, __LAPACK_int, Opaque
Pointer, OpaquePointer?, __LAPACK_int)
```

Rank-k update—multiplies a symmetric matrix by its transpose and adds a second matrix (double-precision complex).

```
func cblas_ztbmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int)
```

Scales a triangular band matrix, then multiplies by a vector (double-precision complex).

```
func cblas_ztbsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, __LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer
?, __LAPACK_int)
```

Solves a triangular banded system of equations.

```
func cblas_ztpmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, OpaquePointer?, __LAPACK_int)
```

Multiplies a triangular matrix by a vector, then adds a vector (double-precision compex).

```
func cblas_ztpsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, OpaquePointer?, __LAPACK_int)
```

Solves a packed triangular system of equations.

```
func cblas_ztrmm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?,
__LAPACK_int, OpaquePointer?, __LAPACK_int)
```

Scales a triangular matrix and multiplies it by a matrix.

```
func cblas_ztrmv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK
_int)
```

Multiplies a triangular matrix by a vector.

```
func cblas_ztrsm(CBLAS_ORDER, CBLAS_SIDE, CBLAS_UPLO, CBLAS_TRANSPOSE,
CBLAS_DIAG, __LAPACK_int, __LAPACK_int, OpaquePointer, OpaquePointer?,
__LAPACK_int, OpaquePointer?, __LAPACK_int)
```

Solves a triangular system of equations with multiple values for the right side.

```
func cblas_ztrsv(CBLAS_ORDER, CBLAS_UPLO, CBLAS_TRANSPOSE, CBLAS_DIAG,
__LAPACK_int, OpaquePointer?, __LAPACK_int, OpaquePointer?, __LAPACK
_int)
```

Solves a triangular system of equations with a single value for the right side.

LAPACK functions

LAPACK/BLAS Functions

An updated BLAS interface supporting ILP64 is available.

```
func cgedmd_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<
CChar>, UnsafePointer<CChar>, UnsafePointer<__LAPACK_int>, Unsafe
Pointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, OpaquePointer?,
UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafeMutable
Pointer<__LAPACK_int>, OpaquePointer?, OpaquePointer?, UnsafeMutable
Pointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, OpaquePointer?,
UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int
>, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, Unsafe
Pointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, UnsafePointer<_
_LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>?, UnsafePointer<_
_LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)
```

```
func cgedmdq_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer
<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<
CChar>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>,
UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int
>, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, Unsafe
Pointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float
>, UnsafeMutablePointer<__LAPACK_int>, OpaquePointer?, OpaquePointer?,
UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer<Float>?,
OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, Unsafe
Pointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int>,
OpaquePointer?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Float
>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>?,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)

func cgeqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<Float>, OpaquePointer?,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<Float>, UnsafeMutable
Pointer<__LAPACK_int>?, OpaquePointer?, OpaquePointer, UnsafePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<_
_LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)

func claqp2rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafePointer<Float
>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, OpaquePointer?,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<Float>, UnsafeMutable
Pointer<__LAPACK_int>?, OpaquePointer?, UnsafeMutablePointer<Float>?,
UnsafeMutablePointer<Float>?, OpaquePointer, UnsafeMutablePointer<_
_LAPACK_int>)

func claqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafePointer<Float
>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, OpaquePointer?,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_bool>,
UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer<Float>, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<__LAPACK_int>?, Opaque
Pointer?, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<Float>?,
OpaquePointer?, OpaquePointer?, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)
```

func crscl_(UnsafePointer<__LAPACK_int>, OpaquePointer, OpaquePointer?, UnsafePointer<__LAPACK_int>)

func dgedmd_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)

func dgedmdq_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)

func dgeqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<Double>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)

```
func dlaqp2rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafePointer<
Double>, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, Unsafe
MutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutable
Pointer<__LAPACK_int>, UnsafeMutablePointer<Double>, UnsafeMutable
Pointer<Double>, UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutable
Pointer<Double>?, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<
Double>?, UnsafeMutablePointer<Double>, UnsafeMutablePointer<__LAPACK
_int>)

func dlaqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafePointer<
Double>, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, Unsafe
MutablePointer<Double>?, UnsafePointer<__LAPACK_int>, UnsafeMutable
Pointer<__LAPACK_bool>, UnsafeMutablePointer<__LAPACK_int>, Unsafe
MutablePointer<Double>, UnsafeMutablePointer<Double>, UnsafeMutable
Pointer<__LAPACK_int>?, UnsafeMutablePointer<Double>?, UnsafeMutable
Pointer<Double>?, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<
Double>?, UnsafeMutablePointer<Double>?, UnsafePointer<__LAPACK_int>,
UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int
>)

func sgedmd_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<
CChar>, UnsafePointer<CChar>, UnsafePointer<__LAPACK_int>, Unsafe
Pointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafeMutable
Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<
Float>?, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>,
UnsafePointer<Float>, UnsafeMutablePointer<__LAPACK_int>, UnsafeMutable
Pointer<Float>?, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<
Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Float>?,
UnsafeMutablePointer<Float>?, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutable
Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<
Float>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int
>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)

func sgedmdq_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer
<CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer<
CChar>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, Unsafe
Pointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, UnsafePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>?, UnsafePointer<__LAPACK_int
```

>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafeMutable
Pointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, UnsafeMutable
Pointer<Float>?, UnsafeMutablePointer<Float>?, UnsafePointer<__LAPACK
_int>, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<Float>?,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, Unsafe
Pointer<__LAPACK_int>, UnsafeMutablePointer<Float>?, UnsafePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>, UnsafePointer<__LAPACK_int>,
UnsafeMutablePointer<__LAPACK_int>?, UnsafePointer<__LAPACK_int>,
UnsafeMutablePointer<__LAPACK_int>)

func sgeqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<Float>, UnsafeMutable
Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>, UnsafeMutablePointer<Float>,
UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<Float>?,
UnsafeMutablePointer<Float>, UnsafePointer<__LAPACK_int>, UnsafeMutable
Pointer<__LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)

func slaqp2rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafePointer<Float
>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafeMutable
Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>, UnsafeMutablePointer<Float>,
UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<Float>?,
UnsafeMutablePointer<Float>?, UnsafeMutablePointer<Float>?, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<__LAPACK_int>)

func sgeqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<Float>, UnsafeMutablePointer<Float>, UnsafeMutable
Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>, UnsafeMutablePointer<Float>,
UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<Float>?,
UnsafeMutablePointer<Float>, UnsafePointer<__LAPACK_int>, UnsafeMutable
Pointer<__LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)

func slaqp2rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafePointer<Float
>, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafeMutable
Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<_
_LAPACK_int>, UnsafeMutablePointer<Float>, UnsafeMutablePointer<Float>,

UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<Float>?, Unsafe MutablePointer<Float>, UnsafeMutablePointer<__LAPACK_int>)

func slaqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int >, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe MutablePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafePointer<Float >, UnsafePointer<__LAPACK_int>, UnsafePointer<Float>, UnsafeMutable Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<_ _LAPACK_bool>, UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer <Float>, UnsafeMutablePointer<Float>, UnsafeMutablePointer<__LAPACK_int >?, UnsafeMutablePointer<Float>?, UnsafeMutablePointer<Float>?, Unsafe MutablePointer<Float>?, UnsafeMutablePointer<Float>?, UnsafeMutable Pointer<Float>?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<_ _LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)

func zgedmd_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer< CChar>, UnsafePointer<CChar>, UnsafePointer<__LAPACK_int>, Unsafe Pointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int >, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafeMutable Pointer<__LAPACK_int>, OpaquePointer?, OpaquePointer?, UnsafePointer<_ _LAPACK_int>, UnsafeMutablePointer<Double>?, OpaquePointer?, Unsafe Pointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, Unsafe Pointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafePointer<_ _LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>?, UnsafePointer<_ _LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)

func zgedmdq_(UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer <CChar>, UnsafePointer<CChar>, UnsafePointer<CChar>, UnsafePointer< CChar>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int >, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, Unsafe Pointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, UnsafePointer< Double>, UnsafeMutablePointer<__LAPACK_int>, OpaquePointer?, Opaque Pointer?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<Double>?, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, Unsafe Pointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int>, OpaquePointer?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer< Double>, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int >?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>)

```
func zgeqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<Double>, UnsafeMutablePointer<Double>, OpaquePointer?,
UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK_int>, Unsafe
MutablePointer<Double>, UnsafeMutablePointer<Double>, UnsafeMutable
Pointer<__LAPACK_int>?, OpaquePointer?, OpaquePointer, UnsafePointer<_
_LAPACK_int>, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<_
_LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int>)
```

```
func zlaqp2rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafePointer<
Double>, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, Opaque
Pointer?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK
_int>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<Double>,
UnsafeMutablePointer<__LAPACK_int>?, OpaquePointer?, UnsafeMutable
Pointer<Double>?, UnsafeMutablePointer<Double>?, OpaquePointer, Unsafe
MutablePointer<__LAPACK_int>)
```

```
func zlaqp3rk_(UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int
>, UnsafePointer<__LAPACK_int>, UnsafePointer<__LAPACK_int>, Unsafe
MutablePointer<__LAPACK_int>, UnsafePointer<Double>, UnsafePointer<
Double>, UnsafePointer<__LAPACK_int>, UnsafePointer<Double>, Opaque
Pointer?, UnsafePointer<__LAPACK_int>, UnsafeMutablePointer<__LAPACK
_bool>, UnsafeMutablePointer<__LAPACK_int>, UnsafeMutablePointer<Double
>, UnsafeMutablePointer<Double>, UnsafeMutablePointer<__LAPACK_int>?,
OpaquePointer?, UnsafeMutablePointer<Double>?, UnsafeMutablePointer<
Double>?, OpaquePointer?, OpaquePointer?, UnsafePointer<__LAPACK_int>,
UnsafeMutablePointer<__LAPACK_int>?, UnsafeMutablePointer<__LAPACK_int
>)
```

```
func zrscl_(UnsafePointer<__LAPACK_int>, OpaquePointer, OpaquePointer?,
UnsafePointer<__LAPACK_int>)
```

```
typealias sparse_matrix_double_complex
```

```
typealias sparse_matrix_float_complex
```

```
typealias vFloatPacked
```

See Also

Linear Algebra

{} Solving systems of linear equations with LAPACK

Select the optimal LAPACK routine to solve a system of linear equations.

Finding an interpolating polynomial using the Vandermonde method

Use LAPACK to solve a linear system and find an interpolating polynomial to construct new points between a series of known data points.

{} Compressing an image using linear algebra

Reduce the storage size of an image using singular value decomposition (SVD).