
pyclblas Documentation

Release 0.5.0

Jon Roose

Jun 07, 2017

1	Blas Families	3
1.1	BLAS Level 1 Functions	3
1.1.1	SWAP - Swap elements from 2 vectors	3
1.1.2	SCAL - Scales a vector by a constant	5
1.1.3	SSCAL - Scales a complex vector by a real constant	7
1.1.4	COPY - Copies elements from vector X to vector Y	8
1.1.5	AXPY - Scale X and add to Y	10
1.1.6	DOT - Dot product of two vectors	12
1.1.7	ROTG - Constructs givens plane rotation	15
1.1.8	ROTMG - Constructs the modified givens rotation	17
1.1.9	ROT - Apply givens rotation	19
1.1.10	ROTM - Apply modified givens rotation for points in the plane	21
1.1.11	NRM2 - Euclidean norm of a vector	22
1.1.12	iAMAX - Index of max absolute value	24
1.1.13	ASUM - Sum of absolute values	26
1.2	BLAS Level 2 Functions	28
1.2.1	GEMV - General matrix-Vector multiplication	28
1.2.2	SYMV - Symmetric matrix-Vector multiplication	31
1.2.3	HEMV - Hermitian matrix-vector multiplication	33
1.2.4	TRMV - Triangular matrix vector multiply	34
1.2.5	TRSV - Triangular matrix vector Solve	37
1.2.6	GER - General matrix rank 1 operation	40
1.2.7	GERU - General matrix rank 1 operation	41
1.2.8	GERC - General matrix rank 1 operation	43
1.2.9	SYR - Symmetric rank 1 update	44
1.2.10	HER - Hermitian rank 1 operation	45
1.2.11	SYR2 - Symmetric rank 2 update	46
1.2.12	HER2 - Hermitian rank 2 update	48
1.2.13	TPMV - Triangular packed matrix-vector multiply	49
1.2.14	TPSV - Triangular packed matrix vector solve	52
1.2.15	SPMV - Symmetric packed matrix vector multiply	54
1.2.16	HPMV - Hermitian packed matrix-vector multiplication	55
1.2.17	SPR - Symmetric packed matrix rank 1 update	57
1.2.18	HPR - Hermitian packed matrix rank 1 update	58
1.2.19	SPR2 - Symmetric packed matrix rank 2 update	59
1.2.20	HPR2 - Hermitian packed matrix rank 2 update	60

1.2.21	GBMV - General banded matrix-vector multiplication	62
1.2.22	TBMV - Triangular banded matrix vector multiply	65
1.2.23	SBMV - Symmetric banded matrix-vector multiplication	68
1.2.24	HBMV - Hermitian banded matrix-vector multiplication	70
1.2.25	TBSV - Solving triangular banded matrix	71
1.3	BLAS Level 3 Functions	74
1.3.1	GEMM - General matrix-matrix multiplication	74
1.3.2	TRMM - Triangular matrix-matrix multiplication	78
1.3.3	TRSM - Solving triangular systems of equations	81
1.3.4	SYRK - Symmetric rank-k update of a matrix	85
1.3.5	SYR2K - Symmetric rank-2k update to a matrix	88
1.3.6	SYMM - Symmetric matrix-matrix multiply	91
1.3.7	HEMM - Hermitian matrix-matrix multiplication	95
1.3.8	HERK - Hermitian rank-k update to a matrix	97
1.3.9	HER2K - Hermitian rank-2k update to a matrix	99

Python Module Index	101
----------------------------	------------

Contents:

BLAS Level 1 Functions

SWAP - Swap elements from 2 vectors

`pyclblas.clblasCswap` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasCswap`

interchanges two vectors of complex-float elements.

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*out*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasDswap` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasDswap`

interchanges two vectors of double.

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*out*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.cblasSswap(N, X, offx, incx, Y, offy, incy, commandQueues, eventWaitList)`
wraps: `cblasSswap`

interchanges two vectors of float.

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*out*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.cblasZswap(N, X, offx, incx, Y, offy, incy, commandQueues, eventWaitList)`
wraps: `cblasZswap`

interchanges two vectors of double-complex elements.

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.

- **X** (`pyopencl.Buffer [out]`) – Buffer object storing vector **X**.
- **offx** (`int [in]`) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (`int [in]`) – Increment for the elements of **X**. Must not be zero.
- **Y** (`pyopencl.Buffer [out]`) – Buffer object storing the vector **Y**.
- **offy** (`int [in]`) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (`int [in]`) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (`pyopencl.CommandQueue [in]`) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (`pyopencl.Event [in]`) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

SCAL - Scales a vector by a constant

`pyclblas.clblasCscal` (*N, alpha, X, offx, incx, commandQueues, eventWaitList*)
wraps: `clblasCscal`

Scales a complex-float vector by a complex-float constant.

$$\bullet (X \leftarrow \alpha X)$$

Parameters

- **N** (`int [in]`) – Number of elements in vector **X**.
- **alpha** (`complex [in]`) – The constant factor for vector **X**.
- **X** (`pyopencl.Buffer [out]`) – Buffer object storing vector **X**.
- **offx** (`int [in]`) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (`int [in]`) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (`pyopencl.CommandQueue [in]`) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (`pyopencl.Event [in]`) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasDscal` (*N, alpha, X, offx, incx, commandQueues, eventWaitList*)
wraps: `clblasDscal`

Scales a double vector by a double constant.

$$\bullet (X \leftarrow \alpha X)$$

Parameters

- **N** (`int [in]`) – Number of elements in vector **X**.
- **alpha** (`float [in]`) – The constant factor for vector **X**.

- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.cblasSscal` (*N, alpha, X, offx, incx, commandQueues, eventWaitList*)

wraps: `cblasSscal`

Scales a float vector by a float constant.

$$\bullet (X \leftarrow \alpha X)$$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **alpha** (*float [in]*) – The constant factor for vector **X**.
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.cblasZscal` (*N, alpha, X, offx, incx, commandQueues, eventWaitList*)

wraps: `cblasZscal`

Scales a complex-double vector by a complex-double constant.

$$\bullet (X \leftarrow \alpha X)$$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **alpha** (*complex [in]*) – The constant factor for vector **X**.
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

SSCAL - Scales a complex vector by a real constant

`pyclblas.clblasCsscal` (*N, alpha, X, offx, incx, commandQueues, eventWaitList*)
wraps: `clblasCsscal`

Scales a complex-float vector by a float constant.

$$\bullet (X \leftarrow \alpha X)$$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **alpha** (*float [in]*) – The constant factor for vector **X**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZdscal` (*N, alpha, X, offx, incx, commandQueues, eventWaitList*)
wraps: `clblasZdscal`

Scales a complex-double vector by a double constant.

$$\bullet (X \leftarrow \alpha X)$$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **alpha** (*float [in]*) – The constant factor for vector **X**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

COPY - Copies elements from vector X to vector Y

`pycblas.clblasCcopy` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
 wraps: `clblasCcopy`

Copies complex-float elements from vector X to vector Y.

• ($Y \leftarrow X$)

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasDcopy` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
 wraps: `clblasDcopy`

Copies double elements from vector X to vector Y.

• ($Y \leftarrow X$)

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasScopy` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasScopy`

Copies float elements from vector *X* to vector *Y*.

• ($Y \leftarrow X$)

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasZcopy` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasZcopy`

Copies complex-double elements from vector *X* to vector *Y*.

• ($Y \leftarrow X$)

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

AXPY - Scale X and add to Y

`pycblas.clblasCaxpy` (*N*, *alpha*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasCaxpy`

Scale vector *X* of complex-float elements and add to *Y*.

$$\bullet (Y \leftarrow \alpha X + Y)$$

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **alpha** (*complex* [*in*]) – The constant factor for vector **X**.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasDaxpy` (*N*, *alpha*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasDaxpy`

Scale vector *X* of double elements and add to *Y*.

$$\bullet (Y \leftarrow \alpha X + Y)$$

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X**.
- **alpha** (*float* [*in*]) – The constant factor for vector **X**.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasSaxpy` (*N, alpha, X, offx, incx, Y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasSaxpy`

Scale vector `X` of float elements and add to `Y`.

$$\bullet (Y \leftarrow \alpha X + Y)$$

Parameters

- **N** (*int [in]*) – Number of elements in vector `X`.
- **alpha** (*float [in]*) – The constant factor for vector `X`.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector `X`.
- **offx** (*int [in]*) – Offset of first element of vector `X` in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of `X`. Must not be zero.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector `Y`.
- **offy** (*int [in]*) – Offset of first element of vector `Y` in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of `Y`. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasZaxpy` (*N, alpha, X, offx, incx, Y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasZaxpy`

Scale vector `X` of double-complex elements and add to `Y`.

$$\bullet (Y \leftarrow \alpha X + Y)$$

Parameters

- **N** (*int [in]*) – Number of elements in vector `X`.
- **alpha** (*complex [in]*) – The constant factor for vector `X`.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector `X`.
- **offx** (*int [in]*) – Offset of first element of vector `X` in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of `X`. Must not be zero.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector `Y`.

- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

DOT - Dot product of two vectors

`pycblas.clblasDdot` (*N, dotProduct, offDP, X, offx, incx, Y, offy, incy, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasDdot`

dot product of two vectors containing double elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **dotProduct** (*pyopencl.Buffer [out]*) – Buffer object that will contain the dot-product value.
- **offDP** (*int [in]*) – Offset to dot-product in **dotProduct** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary `cl_mem` scratch buffer object of minimum size `N`.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasSdot` (*N, dotProduct, offDP, X, offx, incx, Y, offy, incy, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasSdot`

dot product of two vectors containing float elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.

- **dotProduct** (*pyopencl.Buffer [out]*) – Buffer object that will contain the dot-product value.
- **offDP** (*int [in]*) – Offset to dot-product in **dotProduct** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object of minimum size N.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasCdotu` (*N, dotProduct, offDP, X, offx, incx, Y, offy, incy, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasCdotu`

dot product of two vectors containing float-complex elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **dotProduct** (*pyopencl.Buffer [out]*) – Buffer object that will contain the dot-product value.
- **offDP** (*int [in]*) – Offset to dot-product in **dotProduct** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object of minimum size N.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasZdotu (*N, dotProduct, offDP, X, offx, incx, Y, offy, incy, scratchBuff, commandQueues, eventWaitList*)

wraps: *clblasZdotu*

dot product of two vectors containing double-complex elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **dotProduct** (*pyopencl.Buffer [out]*) – Buffer object that will contain the dot-product value.
- **offDP** (*int [in]*) – Offset to dot-product in **dotProduct** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary *cl_mem* scratch buffer object of minimum size *N*.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasCdotc (*N, dotProduct, offDP, X, offx, incx, Y, offy, incy, scratchBuff, commandQueues, eventWaitList*)

wraps: *clblasCdotc*

dot product of two vectors containing float-complex elements conjugating the first vector

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **dotProduct** (*pyopencl.Buffer [out]*) – Buffer object that will contain the dot-product value.
- **offDP** (*int [in]*) – Offset to dot-product in **dotProduct** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.

- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object of minimum size N.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of pyopencl.CommandQueue. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of pyopencl.Event. May be None.

Returns A tuple of pyopencl.Event instances, one for each commandQueue supplied.

`pyclblas.clblasZdotc` (*N, dotProduct, offDP, X, offx, incx, Y, offy, incy, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasZdotc`

dot product of two vectors containing double-complex elements conjugating the first vector

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **dotProduct** (*pyopencl.Buffer [out]*) – Buffer object that will contain the dot-product value.
- **offDP** (*int [in]*) – Offset to dot-product in **dotProduct** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object of minimum size N.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of pyopencl.CommandQueue. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of pyopencl.Event. May be None.

Returns A tuple of pyopencl.Event instances, one for each commandQueue supplied.

ROTG - Constructs givens plane rotation

`pyclblas.clblasCrotg` (*CA, offCA, CB, offCB, C, offC, S, offS, commandQueues, eventWaitList*)

wraps: `clblasCrotg`

construct gives plane rotation on float-complex elements

Parameters

- **CA** (*pyopencl.Buffer [out]*) – Buffer object that contains CA.
- **offCA** (*int [in]*) – Offset to CA in **CA** buffer object. Counted in elements.
- **CB** (*pyopencl.Buffer [out]*) – Buffer object that contains CB.
- **offCB** (*int [in]*) – Offset to CB in **CB** buffer object. Counted in elements.
- **C** (*pyopencl.Buffer [out]*) – Buffer object that contains C. C is real.
- **offC** (*int [in]*) – Offset to C in **C** buffer object. Counted in elements.
- **S** (*pyopencl.Buffer [out]*) – Buffer object that contains S.
- **offs** (*int [in]*) – Offset to S in **S** buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasDrotg (*DA, offDA, DB, offDB, C, offC, S, offS, commandQueues, eventWaitList*)
wraps: *clblasDrotg*

construct gives plane rotation on double elements

Parameters

- **DA** (*pyopencl.Buffer [out]*) – Buffer object that contains DA.
- **offDA** (*int [in]*) – Offset to DA in **DA** buffer object. Counted in elements.
- **DB** (*pyopencl.Buffer [out]*) – Buffer object that contains DB.
- **offDB** (*int [in]*) – Offset to DB in **DB** buffer object. Counted in elements.
- **C** (*pyopencl.Buffer [out]*) – Buffer object that contains C.
- **offC** (*int [in]*) – Offset to C in **C** buffer object. Counted in elements.
- **S** (*pyopencl.Buffer [out]*) – Buffer object that contains S.
- **offs** (*int [in]*) – Offset to S in **S** buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasSrotg (*SA, offSA, SB, offSB, C, offC, S, offS, commandQueues, eventWaitList*)
wraps: *clblasSrotg*

construct gives plane rotation on float elements

Parameters

- **SA** (*pyopencl.Buffer [out]*) – Buffer object that contains SA.
- **offSA** (*int [in]*) – Offset to SA in **SA** buffer object. Counted in elements.

- **SB** (*pyopencl.Buffer [out]*) – Buffer object that contains SB.
- **offSB** (*int [in]*) – Offset to SB in **SB** buffer object. Counted in elements.
- **C** (*pyopencl.Buffer [out]*) – Buffer object that contains C.
- **offC** (*int [in]*) – Offset to C in **C** buffer object. Counted in elements.
- **S** (*pyopencl.Buffer [out]*) – Buffer object that contains S.
- **offs** (*int [in]*) – Offset to S in **S** buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZrotg(CA, offCA, CB, offCB, C, offC, S, offs, commandQueues, eventWaitList)`
wraps: `clblasZrotg`

construct gives plane rotation on double-complex elements

Parameters

- **CA** (*pyopencl.Buffer [out]*) – Buffer object that contains CA.
- **offCA** (*int [in]*) – Offset to CA in **CA** buffer object. Counted in elements.
- **CB** (*pyopencl.Buffer [out]*) – Buffer object that contains CB.
- **offCB** (*int [in]*) – Offset to CB in **CB** buffer object. Counted in elements.
- **C** (*pyopencl.Buffer [out]*) – Buffer object that contains C. C is real.
- **offC** (*int [in]*) – Offset to C in **C** buffer object. Counted in elements.
- **S** (*pyopencl.Buffer [out]*) – Buffer object that contains S.
- **offs** (*int [in]*) – Offset to S in **S** buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

ROTMG - Constructs the modified givens rotation

`pyclblas.clblasDrotmg(DD1, offDD1, DD2, offDD2, DX1, offDX1, DY1, offDY1, DPARAM, offD-param, commandQueues, eventWaitList)`
wraps: `clblasDrotmg`

construct the modified givens rotation on double elements

Parameters

- **DD1** (*pyopencl.Buffer [out]*) – Buffer object that contains DD1.
- **offDD1** (*int [in]*) – Offset to DD1 in **DD1** buffer object. Counted in elements.
- **DD2** (*pyopencl.Buffer [out]*) – Buffer object that contains DD2.

- **offDD2** (*int [in]*) – Offset to DD2 in **DD2** buffer object. Counted in elements.
- **DX1** (*pyopencl.Buffer [out]*) – Buffer object that contains DX1.
- **offDX1** (*int [in]*) – Offset to DX1 in **DX1** buffer object. Counted in elements.
- **DY1** (*pyopencl.Buffer [in]*) – Buffer object that contains DY1.
- **offDY1** (*int [in]*) – Offset to DY1 in **DY1** buffer object. Counted in elements.
- **DPARAM** (*pyopencl.Buffer [out]*) – Buffer object that contains DPARAM array of minimum length 5 DPARAM(0) = DFLAG DPARAM(1) = DH11 DPARAM(2) = DH21 DPARAM(3) = DH12 DPARAM(4) = DH22.
- **offDparam** (*int [in]*) – Offset to DPARAM in **DPARAM** buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasSrotmg` (*SD1, offSD1, SD2, offSD2, SX1, offSX1, SY1, offSY1, SPARAM, offSpam, commandQueues, eventWaitList*)

wraps: `clblasSrotmg`

construct the modified givens rotation on float elements

Parameters

- **SD1** (*pyopencl.Buffer [out]*) – Buffer object that contains SD1.
- **offSD1** (*int [in]*) – Offset to SD1 in **SD1** buffer object. Counted in elements.
- **SD2** (*pyopencl.Buffer [out]*) – Buffer object that contains SD2.
- **offSD2** (*int [in]*) – Offset to SD2 in **SD2** buffer object. Counted in elements.
- **SX1** (*pyopencl.Buffer [out]*) – Buffer object that contains SX1.
- **offSX1** (*int [in]*) – Offset to SX1 in **SX1** buffer object. Counted in elements.
- **SY1** (*pyopencl.Buffer [in]*) – Buffer object that contains SY1.
- **offSY1** (*int [in]*) – Offset to SY1 in **SY1** buffer object. Counted in elements.
- **SPARAM** (*pyopencl.Buffer [out]*) – Buffer object that contains SPARAM array of minimum length 5 SPARAM(0) = SFLAG SPARAM(1) = SH11 SPARAM(2) = SH21 SPARAM(3) = SH12 SPARAM(4) = SH22.
- **offSpam** (*int [in]*) – Offset to SPARAM in **SPARAM** buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

ROT - Apply givens rotation

`pyclblas.clblasDrot` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *C*, *S*, *commandQueues*, *eventWaitList*)
wraps: `clblasDrot`

applies a plane rotation for double elements

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X** and **Y**.
- **X** (*pyopencl.Buffer* [*out*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **C** (*float* [*in*]) – C specifies the cosine, cos.
- **S** (*float* [*in*]) – S specifies the sine, sin.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasSrot` (*N*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *C*, *S*, *commandQueues*, *eventWaitList*)
wraps: `clblasSrot`

applies a plane rotation for float elements

Parameters

- **N** (*int* [*in*]) – Number of elements in vector **X** and **Y**.
- **X** (*pyopencl.Buffer* [*out*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*out*]) – Buffer object storing the vector **Y**.
- **offy** (*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **C** (*float* [*in*]) – C specifies the cosine, cos.
- **S** (*float* [*in*]) – S specifies the sine, sin.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.cblasCsrot (*N, X, offx, incx, Y, offy, incy, C, S, commandQueues, eventWaitList*)
wraps: *cblasCsrot*

applies a plane rotation for float-complex elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X** and **Y**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **C** (*float [in]*) – **C** specifies the cosine, *cos*. This number is real.
- **S** (*float [in]*) – **S** specifies the sine, *sin*. This number is real.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.cblasZdrot (*N, X, offx, incx, Y, offy, incy, C, S, commandQueues, eventWaitList*)
wraps: *cblasZdrot*

applies a plane rotation for double-complex elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X** and **Y**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **C** (*float [in]*) – **C** specifies the cosine, *cos*. This number is real.
- **S** (*float [in]*) – **S** specifies the sine, *sin*. This number is real.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

ROTM - Apply modified givens rotation for points in the plane

`pyclblas.clblasDrotm(N, X, offx, incx, Y, offy, incy, DPARAM, offDparam, commandQueues, eventWaitList)`

wraps: `clblasDrotm`

modified givens rotation for double elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X** and **Y**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **DPARAM** (*pyopencl.Buffer [in]*) – Buffer object that contains SPARAM array of minimum length 5 **DPARAM**(1)=DFLAG **DPARAM**(2)=DH11 **DPARAM**(3)=DH21 **DPARAM**(4)=DH12 **DPARAM**(5)=DH22.
- **offDparam** (*int [in]*) – Offset of first element of array **DPARAM** in buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasSrotm(N, X, offx, incx, Y, offy, incy, SPARAM, offSparam, commandQueues, eventWaitList)`

wraps: `clblasSrotm`

modified givens rotation for float elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X** and **Y**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.

- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **SPARAM** (*pyopencl.Buffer [in]*) – Buffer object that contains SPARAM array of minimum length 5 SPARAM(1)=SFLAG SPARAM(2)=SH11 SPARAM(3)=SH21 SPARAM(4)=SH12 SPARAM(5)=SH22.
- **offSparam** (*int [in]*) – Offset of first element of array **SPARAM** in buffer object. Counted in elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

NRM2 - Euclidean norm of a vector

`pyclblas.clblasDnrm2 (N, NRM2, offNRM2, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasDnrm2`

computes the euclidean norm of vector containing double elements $NRM2 = \sqrt{X' * X}$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **NRM2** (*pyopencl.Buffer [out]*) – Buffer object that will contain the NRM2 value.
- **offNRM2** (*int [in]*) – Offset to NRM2 value in **NRM2** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary `cl_mem` scratch buffer object that can hold minimum of $(2*N)$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasSnrm2 (N, NRM2, offNRM2, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasSnrm2`

computes the euclidean norm of vector containing float elements $NRM2 = \sqrt{X' * X}$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.

- **NRM2** (*pyopencl.Buffer [out]*) – Buffer object that will contain the NRM2 value.
- **offNRM2** (*int [in]*) – Offset to NRM2 value in **NRM2** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object that can hold minimum of (2*N) elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasScnrm2(N, NRM2, offNRM2, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`

wraps: `clblasScnrm2`

computes the euclidean norm of vector containing float-complex elements $NRM2 = \sqrt{X^*H * X}$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **NRM2** (*pyopencl.Buffer [out]*) – Buffer object that will contain the NRM2 value. Note that the answer of `Scnrm2` is a real value.
- **offNRM2** (*int [in]*) – Offset to NRM2 value in **NRM2** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object that can hold minimum of (2*N) elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasDznrm2(N, NRM2, offNRM2, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`

wraps: `clblasDznrm2`

computes the euclidean norm of vector containing double-complex elements $NRM2 = \sqrt{X^*H * X}$

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.

- **NRM2** (*pyopencl.Buffer [out]*) – Buffer object that will contain the NRM2 value. Note that the answer of Dznrm2 is a real value.
- **offNRM2** (*int [in]*) – Offset to NRM2 value in **NRM2** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object that can hold minimum of (2*N) elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

iAMAX - Index of max absolute value

`pyclblas.clblasISamax(N, iMax, offiMax, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasISamax`

index of max absolute value in a float array

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **iMax** (*pyopencl.Buffer [out]*) – Buffer object storing the index of first absolute max. The index will be of type unsigned int.
- **offiMax** (*int [in]*) – Offset for storing index in the buffer **iMax** Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem object to store intermediate results It should be able to hold minimum of (2*N) elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasIDamax(N, iMax, offiMax, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasIDamax`

index of max absolute value in a double array

Parameters

- **N**(*int* [in]) – Number of elements in vector **X**.
- **iMax**(*pyopencl.Buffer* [out]) – Buffer object storing the index of first absolute max. The index will be of type unsigned int.
- **offiMax**(*int* [in]) – Offset for storing index in the buffer iMax Counted in elements.
- **X**(*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx**(*int* [in]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx**(*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer* [in]) – Temporary *cl_mem* object to store intermediate results It should be able to hold minimum of (2*N) elements.
- **commandQueues**(*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList**(*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasiCamax(N, iMax, offiMax, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasiCamax`

index of max absolute value in a complex float array

Parameters

- **N**(*int* [in]) – Number of elements in vector **X**.
- **iMax**(*pyopencl.Buffer* [out]) – Buffer object storing the index of first absolute max. The index will be of type unsigned int.
- **offiMax**(*int* [in]) – Offset for storing index in the buffer iMax Counted in elements.
- **X**(*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx**(*int* [in]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx**(*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer* [in]) – Temporary *cl_mem* object to store intermediate results It should be able to hold minimum of (2*N) elements.
- **commandQueues**(*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList**(*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasiZamax(N, iMax, offiMax, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasiZamax`

index of max absolute value in a complex double array

Parameters

- **N**(*int* [in]) – Number of elements in vector **X**.

- **iMax** (*pyopencl.Buffer [out]*) – Buffer object storing the index of first absolute max. The index will be of type unsigned int.
- **offiMax** (*int [in]*) – Offset for storing index in the buffer iMax Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem object to store intermediate results It should be able to hold minimum of (2*N) elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of pyopencl.CommandQueue. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of pyopencl.Event. May be None.

Returns A tuple of pyopencl.Event instances, one for each commandQueue supplied.

ASUM - Sum of absolute values

`pyclblas.clblasDasum(N, asum, offAsum, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasDasum`

absolute sum of values of a vector containing double elements

Parameters

- **N** (*int [in]*) – Number of elements in vector **X**.
- **asum** (*pyopencl.Buffer [out]*) – Buffer object that will contain the absolute sum value.
- **offAsum** (*int [in]*) – Offset to absolute sum in **asum** buffer object. Counted in elements.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object of minimum size N.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of pyopencl.CommandQueue. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of pyopencl.Event. May be None.

Returns A tuple of pyopencl.Event instances, one for each commandQueue supplied.

`pyclblas.clblasSasum(N, asum, offAsum, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasSasum`

absolute sum of values of a vector containing float elements

Parameters

- **N**(*int* [*in*]) – Number of elements in vector **X**.
- **asum**(*pyopencl.Buffer* [*out*]) – Buffer object that will contain the absolute sum value.
- **offAsum**(*int* [*in*]) – Offset to absolute sum in **asum** buffer object. Counted in elements.
- **X**(*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx**(*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx**(*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer* [*in*]) – Temporary *cl_mem* scratch buffer object of minimum size **N**.
- **commandQueues**(*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasScasum(N, asum, offAsum, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasScasum`

absolute sum of values of a vector containing float-complex elements

Parameters

- **N**(*int* [*in*]) – Number of elements in vector **X**.
- **asum**(*pyopencl.Buffer* [*out*]) – Buffer object that will contain the absolute sum value.
- **offAsum**(*int* [*in*]) – Offset to absolute sum in **asum** buffer object. Counted in elements.
- **X**(*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx**(*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx**(*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer* [*in*]) – Temporary *cl_mem* scratch buffer object of minimum size **N**.
- **commandQueues**(*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasDzasum(N, asum, offAsum, X, offx, incx, scratchBuff, commandQueues, eventWaitList)`
wraps: `clblasDzasum`

absolute sum of values of a vector containing double-complex elements

Parameters

- **N**(*int* [*in*]) – Number of elements in vector **X**.

- **asum** (*pyopencl.Buffer [out]*) – Buffer object that will contain the absolute sum value.
- **offAsum** (*int [in]*) – Offset to absolute sum in **asum** buffer object. Counted in elements.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object of minimum size N.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

BLAS Level 2 Functions

GEMV - General matrix-Vector multiplication

pycblas.clblasCgemv (*order, transA, M, N, alpha, A, offA, lda, x, offx, incx, beta, y, offy, incy, commandQueues, eventWaitList*)

wraps: *clblasCgemv*

Matrix-vector product with a general rectangular matrix and float complex elements. Extended version. Matrix-vector products:

- $y \leftarrow \alpha A x + \beta y$
- $y \leftarrow \alpha A^T x + \beta y$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **x**.
- **offx** (*int [in]*) – Offset of first element of vector **x** in buffer object. Counted in elements.

- **incx** (*int [in]*) – Increment for the elements of **x**. It cannot be zero.
- **beta** (*complex [in]*) – The factor of the vector **y**.
- **y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasDgemv` (*order, transA, M, N, alpha, A, offA, lda, x, offx, incx, beta, y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasDgemv`

Matrix-vector product with a general rectangular matrix and double elements. Extended version. Matrix-vector products:

- $(y \leftarrow \alpha A x + \beta y)$
- $(y \leftarrow \alpha A^T x + \beta y)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **x**.
- **offx** (*int [in]*) – Offset of first element of vector **x** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **x**. It cannot be zero.
- **beta** (*float [in]*) – The factor of the vector **y**.
- **y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasSgemv` (*order, transA, M, N, alpha, A, offA, lda, x, offx, incx, beta, y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasSgemv`

Matrix-vector product with a general rectangular matrix and float elements. Extended version. Matrix-vector products:

- $(y \leftarrow \alpha A x + \beta y)$
- $(y \leftarrow \alpha A^T x + \beta y)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when the parameter is set to **clblasColumnMajor**.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **x**.
- **offx** (*int [in]*) – Offset of first element of vector **x** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **x**. It cannot be zero.
- **beta** (*float [in]*) – The factor of the vector **y**.
- **y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasZgemv` (*order, transA, M, N, alpha, A, offA, lda, x, offx, incx, beta, y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasZgemv`

Matrix-vector product with a general rectangular matrix and double complex elements. Extended version. Matrix-vector products:

- $(y \leftarrow \alpha A x + \beta y)$
- $(y \leftarrow \alpha A^T x + \beta y)$

Parameters

- **order** (*clblasOrder* [in]) – Row/column order.
- **transA** (*clblasTranspose* [in]) – How matrix **A** is to be transposed.
- **M** (*int* [in]) – Number of rows in matrix **A**.
- **N** (*int* [in]) – Number of columns in matrix **A**.
- **alpha** (*complex* [in]) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer* [in]) – Buffer object storing matrix **A**.
- **offA** (*int* [in]) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int* [in]) – Leading dimension of matrix **A**.
- **x** (*pyopencl.Buffer* [in]) – Buffer object storing vector **x**.
- **offx** (*int* [in]) – Offset of first element of vector **x** in buffer object. Counted in elements.
- **incx** (*int* [in]) – Increment for the elements of **x**. It cannot be zero.
- **beta** (*complex* [in]) – The factor of the vector **y**.
- **y** (*pyopencl.Buffer* [out]) – Buffer object storing the vector **y**.
- **offy** (*int* [in]) – Offset of first element of vector **y** in buffer object. Counted in elements.
- **incy** (*int* [in]) – Increment for the elements of **y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

SYMV - Symmetric matrix-Vector multiplication

`pyclblas.clblasDsymv` (*order*, *uplo*, *N*, *alpha*, *A*, *offA*, *lda*, *x*, *offx*, *incx*, *beta*, *y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: `clblasDsymv`

Matrix-vector product with a symmetric matrix and double elements. Matrix-vector products:

- $(y \leftarrow \alpha A x + \beta y)$

Parameters

- **order** (*clblasOrder* [in]) – Row/columns order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.

- **N**(*int* [*in*]) – Number of rows and columns in matrix **A**.
- **alpha**(*float* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **A**.
- **offA**(*int* [*in*]) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**. It cannot less than **N**.
- **x**(*pyopencl.Buffer* [*in*]) – Buffer object storing vector **x**.
- **offx**(*int* [*in*]) – Offset of first element of vector **x** in buffer object. Counted in elements.
- **incx**(*int* [*in*]) – Increment for the elements of vector **x**. It cannot be zero.
- **beta**(*float* [*in*]) – The factor of vector **y**.
- **y**(*pyopencl.Buffer* [*out*]) – Buffer object storing vector **y**.
- **offy**(*int* [*in*]) – Offset of first element of vector **y** in buffer object. Counted in elements.
- **incy**(*int* [*in*]) – Increment for the elements of vector **y**. It cannot be zero.
- **commandQueues**(*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pyclblas.clblasSsymv(*order*, *uplo*, *N*, *alpha*, *A*, *offA*, *lda*, *x*, *offx*, *incx*, *beta*, *y*, *offy*, *incy*, *commandQueues*, *eventWaitList*)
wraps: *clblasSsymv*

Matrix-vector product with a symmetric matrix and float elements. Matrix-vector products:

$$\bullet (y \leftarrow \alpha A x + \beta y)$$

Parameters

- **order**(*clblasOrder* [*in*]) – Row/columns order.
- **uplo**(*clblasUplo* [*in*]) – The triangle in matrix being referenced.
- **N**(*int* [*in*]) – Number of rows and columns in matrix **A**.
- **alpha**(*float* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **A**.
- **offA**(*int* [*in*]) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**. It cannot less than **N**.
- **x**(*pyopencl.Buffer* [*in*]) – Buffer object storing vector **x**.
- **offx**(*int* [*in*]) – Offset of first element of vector **x** in buffer object. Counted in elements.
- **incx**(*int* [*in*]) – Increment for the elements of vector **x**. It cannot be zero.
- **beta**(*float* [*in*]) – The factor of vector **y**.

- **y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

HEMV - Hermitian matrix-vector multiplication

`pyclblas.clblasChemv` (*order, uplo, N, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasChemv`

Matrix-vector product with a hermitian matrix and float-complex elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in matrix **A**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*complex [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZhemv`(*order, uplo, N, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasZhemv`

Matrix-vector product with a hermitian matrix and double-complex elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in matrix **A**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot less than **N**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*complex [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

TRMV - Triangular matrix vector multiply

`pyclblas.clblasCtrmv`(*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)
wraps: `clblasCtrmv`

Matrix-vector product with a triangular matrix and float complex elements. Matrix-vector products:

$$\bullet (X \leftarrow A X)$$

$$\bullet (X \leftarrow A^T X)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.

- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pyclblas.clblasDtrmv (*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: *clblasDtrmv*

Matrix-vector product with a triangular matrix and double elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasStrmv(*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: *clblasStrmv*

Matrix-vector product with a triangular matrix and float elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary *cl_mem* scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasZtrmv(*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: *clblasZtrmv*

Matrix-vector product with a triangular matrix and double complex elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.

- **diag**(*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N**(*int [in]*) – Number of rows/columns in matrix **A**.
- **A**(*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa**(*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **x**(*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx**(*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx**(*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer [in]*) – Temporary `cl_mem` scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues**(*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList**(*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

TRSV - Triangular matrix vector Solve

`pyclblas.clblasCtrsv`(*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasCtrsv`

solving triangular matrix problems with float-complex elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order**(*clblasOrder [in]*) – Row/column order.
- **uplo**(*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans**(*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N**(*int [in]*) – Number of rows/columns in matrix **A**.
- **A**(*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa**(*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **x**(*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx**(*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx**(*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues**(*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasDtrsv` (*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasDtrsv`

solving triangular matrix problems with double elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasStrsv` (*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasStrsv`

solving triangular matrix problems with float elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.

- **N**(*int* [in]) – Number of rows/columns in matrix **A**.
- **A**(*pyopencl.Buffer* [in]) – Buffer object storing matrix **A**.
- **offa**(*int* [in]) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int* [in]) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **x**(*pyopencl.Buffer* [out]) – Buffer object storing vector **X**.
- **offx**(*int* [in]) – Offset in number of elements for first element in vector **X**.
- **incx**(*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **commandQueues**(*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList**(*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZtrsv`(*order, uplo, trans, diag, N, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasZtrsv`

solving triangular matrix problems with double-complex elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order**(*clblasOrder* [in]) – Row/column order.
- **uplo**(*clblasUplo* [in]) – The triangle in matrix being referenced.
- **trans**(*clblasTranspose* [in]) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag* [in]) – Specify whether matrix **A** is unit triangular.
- **N**(*int* [in]) – Number of rows/columns in matrix **A**.
- **A**(*pyopencl.Buffer* [in]) – Buffer object storing matrix **A**.
- **offa**(*int* [in]) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int* [in]) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **x**(*pyopencl.Buffer* [out]) – Buffer object storing vector **X**.
- **offx**(*int* [in]) – Offset in number of elements for first element in vector **X**.
- **incx**(*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **commandQueues**(*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList**(*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

GER - General matrix rank 1 operation

`pycblas.cblasDger` (*order*, *M*, *N*, *alpha*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *A*, *offa*, *lda*, *commandQueues*, *eventWaitList*)

wraps: `cblasDger`

vector-vector product with double elements and performs the rank 1 operation A Vector-vector products:

$$\bullet (A \leftarrow \alpha X Y^T + A)$$

Parameters

- **order** (*cblasOrder* [*in*]) – Row/column order.
- **M** (*int* [*in*]) – Number of rows in matrix **A**.
- **N** (*int* [*in*]) – Number of columns in matrix **A**.
- **alpha** (*float* [*in*]) – specifies the scalar alpha.
- **X** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx** (*int* [*in*]) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int* [*in*]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [*in*]) – Buffer object storing vector **Y**.
- **offy** (*int* [*in*]) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int* [*in*]) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer* [*out*]) – Buffer object storing matrix **A**. On exit, A is overwritten by the updated matrix.
- **offa** (*int* [*in*]) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int* [*in*]) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **cblasRowMajor**, or less than **M** when the parameter is set to **cblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.cblasSger` (*order*, *M*, *N*, *alpha*, *X*, *offx*, *incx*, *Y*, *offy*, *incy*, *A*, *offa*, *lda*, *commandQueues*, *eventWaitList*)

wraps: `cblasSger`

vector-vector product with float elements and performs the rank 1 operation A Vector-vector products:

$$\bullet (A \leftarrow \alpha X Y^T + A)$$

Parameters

- **order** (*cblasOrder* [*in*]) – Row/column order.
- **M** (*int* [*in*]) – Number of rows in matrix **A**.
- **N** (*int* [*in*]) – Number of columns in matrix **A**.
- **alpha** (*float* [*in*]) – specifies the scalar alpha.

- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**. On exit, A is overwritten by the updated matrix.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when the parameter is set to **clblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

GERU - General matrix rank 1 operation

`pyclblas.clblasCgeru` (*order, M, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasCgeru`

vector-vector product with float complex elements and performs the rank 1 operation A Vector-vector products:

$$\bullet (A \leftarrow \alpha X Y^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*complex [in]*) – specifies the scalar alpha.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**. On exit, A is overwritten by the updated matrix.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.

- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when the parameter is set to **clblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZgeru` (*order, M, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasZgeru`

vector-vector product with double complex elements and performs the rank 1 operation A Vector-vector products:

$$\bullet (A \leftarrow \alpha X Y^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*complex [in]*) – specifies the scalar alpha.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**. On exit, **A** is overwritten by the updated matrix.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when the parameter is set to **clblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

GERC - General matrix rank 1 operation

`pycblas.cblasCgerc` (*order, M, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)

wraps: `cblasCgerc`

vector-vector product with float complex elements and performs the rank 1 operation A Vector-vector products:

$$\bullet (A \leftarrow \alpha X Y^H + A)$$

Parameters

- **order** (*cblasOrder [in]*) – Row/column order.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*complex [in]*) – specifies the scalar alpha.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**. On exit, A is overwritten by the updated matrix.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **cblasRowMajor**, or less than **M** when the parameter is set to **cblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.cblasZgerc` (*order, M, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)

wraps: `cblasZgerc`

vector-vector product with double complex elements and performs the rank 1 operation A Vector-vector products:

$$\bullet (A \leftarrow \alpha X Y^H + A)$$

Parameters

- **order** (*cblasOrder [in]*) – Row/column order.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **A**.

- **alpha** (*complex [in]*) – specifies the scalar alpha.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**. On exit, A is overwritten by the updated matrix.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when the parameter is set to **clblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

SYR - Symmetric rank 1 update

`pycblas.clblasDsyr` (*order, uplo, N, alpha, X, offx, incx, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasDsyr`

Symmetric rank 1 operation with a general triangular matrix and double elements. Symmetric rank 1 operation:

$$\bullet (A \leftarrow \alpha x x^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of first element of matrix **A** in buffer object.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasSsyx` (*order, uplo, N, alpha, X, offx, incx, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasSsyx`

Symmetric rank 1 operation with a general triangular matrix and float elements. Symmetric rank 1 operation:

$$\bullet (A \leftarrow \alpha x x^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of first element of matrix **A** in buffer object.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

HER - Hermitian rank 1 operation

`pyclblas.clblasCher` (*order, uplo, N, alpha, X, offx, incx, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasCher`

hermitian rank 1 operation with a general triangular matrix and float-complex elements. hermitian rank 1 operation:

$$\bullet (A \leftarrow \alpha X X^H + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A** (a scalar float value).
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.

- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.cblasZher` (*order, uplo, N, alpha, X, offx, incx, A, offa, lda, commandQueues, eventWaitList*)
wraps: `cblasZher`

hermitian rank 1 operation with a general triangular matrix and double-complex elements. hermitian rank 1 operation:

$$\bullet (A \leftarrow \alpha X X^H + A)$$

Parameters

- **order** (*cblasOrder [in]*) – Row/column order.
- **uplo** (*cblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A** (a scalar double value).
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

SYR2 - Symmetric rank 2 update

`pycblas.cblasDsyr2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)
wraps: `cblasDsyr2`

Symmetric rank 2 operation with a general triangular matrix and double elements. Symmetric rank 2 operation:

$$\bullet (A \leftarrow \alpha x y^T + \alpha y x^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of first element of matrix **A** in buffer object.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasSsy2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)

wraps: `clblasSsy2`

Symmetric rank 2 operation with a general triangular matrix and float elements. Symmetric rank 2 operation:

$$\bullet (A \leftarrow \alpha x y^T + \alpha y x^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of first element of matrix **A** in buffer object.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

HER2 - Hermitian rank 2 update

`pycblas.clblasCher2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasCher2`

Hermitian rank 2 operation with a general triangular matrix and float-complex elements. Hermitian rank 2 operation:

$$\bullet (A \leftarrow \alpha X Y^H + \text{conj}\{\alpha\} Y X^H + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZher2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, A, offa, lda, commandQueues, eventWaitList*)
wraps: `clblasZher2`

Hermitian rank 2 operation with a general triangular matrix and double-complex elements. Hermitian rank 2 operation:

$$\bullet (A \leftarrow \alpha X Y^H + \text{conj}\{\alpha\} Y X^H + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **A** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

TPMV - Triangular packed matrix-vector multiply

`pycblas.cblasCtpmv` (*order, uplo, trans, diag, N, AP, offa, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)
wraps: `clblasCtpmv`

Matrix-vector product with a packed triangular matrix and float-complex elements. Matrix-vector products:

- ($X \leftarrow A X$)
- ($X \leftarrow A^T X$)

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **AP** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **AP** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **AP**.
- **AP** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **AP** in packed format.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **AP**.
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.

- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasDtpmv` (*order, uplo, trans, diag, N, AP, offa, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasDtpmv`

Matrix-vector product with a packed triangular matrix and double elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **AP** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **AP** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **AP**.
- **AP** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **AP** in packed format.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **AP**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasStpmv` (*order, uplo, trans, diag, N, AP, offa, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasStpmv`

Matrix-vector product with a packed triangular matrix and float elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **AP** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **AP** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **A**.
- **AP** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **AP** in packed format.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **AP**.
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.cblasZtpmv` (*order, uplo, trans, diag, N, AP, offa, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)
wraps: `clblasZtpmv`

Matrix-vector product with a packed triangular matrix and double-complex elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **AP** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **AP** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **AP**.
- **AP** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **AP** in packed format.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **AP**.
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

TPSV - Triangular packed matrix vector solve

pyclblas.clblasCtpsv (*order, uplo, trans, diag, N, A, offa, X, offx, incx, commandQueues, eventWaitList*)

wraps: *clblasCtpsv*

solving triangular packed matrix problems with float complex elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix in packed format. **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pyclblas.clblasDtpsv (*order, uplo, trans, diag, N, A, offa, X, offx, incx, commandQueues, eventWaitList*)

wraps: *clblasDtpsv*

solving triangular packed matrix problems with double elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.

- **N**(*int* [in]) – Number of rows/columns in matrix **A**.
- **A**(*pyopencl.Buffer* [in]) – Buffer object storing matrix in packed format. **A**.
- **offa**(*int* [in]) – Offset in number of elements for first element in matrix **A**.
- **X**(*pyopencl.Buffer* [out]) – Buffer object storing vector **X**.
- **offx**(*int* [in]) – Offset in number of elements for first element in vector **X**.
- **incx**(*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **commandQueues**(*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList**(*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasStpsv`(*order, uplo, trans, diag, N, A, offa, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasStpsv`

solving triangular packed matrix problems with float elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order**(*clblasOrder* [in]) – Row/column order.
- **uplo**(*clblasUplo* [in]) – The triangle in matrix being referenced.
- **trans**(*clblasTranspose* [in]) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag* [in]) – Specify whether matrix **A** is unit triangular.
- **N**(*int* [in]) – Number of rows/columns in matrix **A**.
- **A**(*pyopencl.Buffer* [in]) – Buffer object storing matrix in packed format. **A**.
- **offa**(*int* [in]) – Offset in number of elements for first element in matrix **A**.
- **X**(*pyopencl.Buffer* [out]) – Buffer object storing vector **X**.
- **offx**(*int* [in]) – Offset in number of elements for first element in vector **X**.
- **incx**(*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **commandQueues**(*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList**(*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZtpsv`(*order, uplo, trans, diag, N, A, offa, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasZtpsv`

solving triangular packed matrix problems with double complex elements. Matrix-vector products:

- $(A X \leftarrow X)$

•($A^T X \leftarrow X$)

Parameters

- **order** (*clblasOrder* [in]) – Row/column order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose* [in]) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag* [in]) – Specify whether matrix **A** is unit triangular.
- **N** (*int* [in]) – Number of rows/columns in matrix **A**.
- **A** (*pyopencl.Buffer* [in]) – Buffer object storing matrix in packed format. **A**.
- **offa** (*int* [in]) – Offset in number of elements for first element in matrix **A**.
- **x** (*pyopencl.Buffer* [out]) – Buffer object storing vector **X**.
- **offx** (*int* [in]) – Offset in number of elements for first element in vector **X**.
- **incx** (*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

SPMV - Symmetric packed matrix vector multiply

`pyclblas.clblasDspmv` (*order, uplo, N, alpha, AP, offa, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasDspmv`

Matrix-vector product with a symmetric packed-matrix and double elements. Matrix-vector products:

•($Y \leftarrow \alpha A X + \beta Y$)

Parameters

- **order** (*clblasOrder* [in]) – Row/columns order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.
- **N** (*int* [in]) – Number of rows and columns in matrix **AP**.
- **alpha** (*float* [in]) – The factor of matrix **AP**.
- **AP** (*pyopencl.Buffer* [in]) – Buffer object storing matrix **AP**.
- **offa** (*int* [in]) – Offset in number of elements for first element in matrix **AP**.
- **x** (*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx** (*int* [in]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [in]) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*float* [in]) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer* [out]) – Buffer object storing vector **Y**.

- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasSspmv` (*order, uplo, N, alpha, AP, offa, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasSspmv`

Matrix-vector product with a symmetric packed-matrix and float elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in matrix **AP**.
- **alpha** (*float [in]*) – The factor of matrix **AP**.
- **AP** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **AP**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **AP**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*float [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

HPMV - Hermitian packed matrix-vector multiplication

`pyclblas.clblasChpmv` (*order, uplo, N, alpha, AP, offa, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasChpmv`

Matrix-vector product with a packed hermitian matrix and float-complex elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder* [in]) – Row/columns order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.
- **N** (*int* [in]) – Number of rows and columns in matrix **AP**.
- **alpha** (*complex* [in]) – The factor of matrix **AP**.
- **AP** (*pyopencl.Buffer* [in]) – Buffer object storing packed matrix **AP**.
- **offa** (*int* [in]) – Offset in number of elements for first element in matrix **AP**.
- **X** (*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx** (*int* [in]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int* [in]) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*complex* [in]) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer* [out]) – Buffer object storing vector **Y**.
- **offy** (*int* [in]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int* [in]) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZhpmv` (*order, uplo, N, alpha, AP, offa, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasZhpmv`

Matrix-vector product with a packed hermitian matrix and double-complex elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder* [in]) – Row/columns order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.
- **N** (*int* [in]) – Number of rows and columns in matrix **AP**.
- **alpha** (*complex* [in]) – The factor of matrix **AP**.
- **AP** (*pyopencl.Buffer* [in]) – Buffer object storing packed matrix **AP**.
- **offa** (*int* [in]) – Offset in number of elements for first element in matrix **AP**.
- **X** (*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx** (*int* [in]) – Offset of first element of vector **X** in buffer object. Counted in elements.

- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*complex [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

SPR - Symmetric packed matrix rank 1 update

`pyclblas.clblasDspr` (*order, uplo, N, alpha, X, offx, incx, AP, offa, commandQueues, eventWaitList*)
wraps: `clblasDspr`

Symmetric rank 1 operation with a general triangular packed-matrix and double elements. Symmetric rank 1 operation:

$$\bullet (A \leftarrow \alpha X X^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **AP** (*pyopencl.Buffer [out]*) – Buffer object storing packed-matrix **AP**.
- **offa** (*int [in]*) – Offset of first element of matrix **AP** in buffer object.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasSspr` (*order, uplo, N, alpha, X, offx, incx, AP, offa, commandQueues, eventWaitList*)
wraps: `clblasSspr`

Symmetric rank 1 operation with a general triangular packed-matrix and float elements. Symmetric rank 1 operation:

$$\bullet (A \leftarrow \alpha X X^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **AP** (*pyopencl.Buffer [out]*) – Buffer object storing packed-matrix **AP**.
- **offa** (*int [in]*) – Offset of first element of matrix **AP** in buffer object.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

HPR - Hermitian packed matrix rank 1 update

`pycblas.clblasChpr (order, uplo, N, alpha, X, offx, incx, AP, offa, commandQueues, eventWaitList)`
wraps: `clblasChpr`

hermitian rank 1 operation with a general triangular packed-matrix and float-complex elements. hermitian rank 1 operation:

$$\bullet (A \leftarrow \alpha X X^H + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A** (a scalar float value).
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **AP** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **AP**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **AP**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasZhpr` (*order, uplo, N, alpha, X, offx, incx, AP, offa, commandQueues, eventWaitList*)
wraps: `clblasZhpr`

hermitian rank 1 operation with a general triangular packed-matrix and double-complex elements. hermitian rank 1 operation:

$$\bullet (A \leftarrow \alpha X X^H + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A** (a scalar float value).
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **AP** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **AP**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **AP**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

SPR2 - Symmetric packed matrix rank 2 update

`pycblas.clblasDspr2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, AP, offa, commandQueues, eventWaitList*)
wraps: `clblasDspr2`

Symmetric rank 2 operation with a general triangular packed-matrix and double elements. Symmetric rank 2 operation:

$$\bullet (A \leftarrow \alpha X Y^T + \alpha Y X^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object.

- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **AP** (*pyopencl.Buffer [out]*) – Buffer object storing packed-matrix **AP**.
- **offa** (*int [in]*) – Offset of first element of matrix **AP** in buffer object.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasSspr2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, AP, offa, commandQueues, eventWaitList*)

wraps: `clblasSspr2`

Symmetric rank 2 operation with a general triangular packed-matrix and float elements. Symmetric rank 2 operation:

$$\bullet (A \leftarrow \alpha X Y^T + \alpha Y X^T + A)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of columns in matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer [in]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **AP** (*pyopencl.Buffer [out]*) – Buffer object storing packed-matrix **AP**.
- **offa** (*int [in]*) – Offset of first element of matrix **AP** in buffer object.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

HPR2 - Hermitian packed matrix rank 2 update

`pycblas.clblasChpr2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, AP, offa, commandQueues, eventWaitList*)

wraps: `clblasChpr2`

Hermitian rank 2 operation with a general triangular packed-matrix and float-complex elements. Hermitian rank 2 operation:

$$\bullet (A \leftarrow \alpha X Y^H + \text{conj}(\alpha) Y X^H + A)$$

Parameters

- **order** (*clblasOrder* [in]) – Row/column order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.
- **N** (*int* [in]) – Number of columns in matrix **A**.
- **alpha** (*complex* [in]) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx** (*int* [in]) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [in]) – Buffer object storing vector **Y**.
- **offy** (*int* [in]) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int* [in]) – Increment for the elements of **Y**. Must not be zero.
- **AP** (*pyopencl.Buffer* [out]) – Buffer object storing packed-matrix **AP**.
- **offa** (*int* [in]) – Offset in number of elements for the first element in matrix **AP**.
- **commandQueues** (*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasZhpr2` (*order, uplo, N, alpha, X, offx, incx, Y, offy, incy, AP, offa, commandQueues, eventWaitList*)

wraps: `clblasZhpr2`

Hermitian rank 2 operation with a general triangular packed-matrix and double-complex elements. Hermitian rank 2 operation:

$$\bullet (A \leftarrow \alpha X Y^H + \text{conj}(\alpha) Y X^H + A)$$

Parameters

- **order** (*clblasOrder* [in]) – Row/column order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix being referenced.
- **N** (*int* [in]) – Number of columns in matrix **A**.
- **alpha** (*complex* [in]) – The factor of matrix **A**.
- **X** (*pyopencl.Buffer* [in]) – Buffer object storing vector **X**.
- **offx** (*int* [in]) – Offset in number of elements for the first element in vector **X**.
- **incx** (*int* [in]) – Increment for the elements of **X**. Must not be zero.
- **Y** (*pyopencl.Buffer* [in]) – Buffer object storing vector **Y**.
- **offy** (*int* [in]) – Offset in number of elements for the first element in vector **Y**.
- **incy** (*int* [in]) – Increment for the elements of **Y**. Must not be zero.
- **AP** (*pyopencl.Buffer* [out]) – Buffer object storing packed-matrix **AP**.

- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **AP**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

GBMV - General banded matrix-vector multiplication

`pycblas.clblasCgbmv` (*order, trans, M, N, KL, KU, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasCgbmv`

Matrix-vector product with a general rectangular banded matrix and float-complex elements. Matrix-vector products:

- $(Y \leftarrow \alpha A X + \beta Y)$
- $(Y \leftarrow \alpha A^T X + \beta Y)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in banded matrix **A**.
- **N** (*int [in]*) – Number of columns in banded matrix **A**.
- **KL** (*int [in]*) – Number of sub-diagonals in banded matrix **A**.
- **KU** (*int [in]*) – Number of super-diagonals in banded matrix **A**.
- **alpha** (*complex [in]*) – The factor of banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing banded matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in banded matrix **A**.
- **lda** (*int [in]*) – Leading dimension of banded matrix **A**. It cannot be less than $(KL + KU + 1)$.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **beta** (*complex [in]*) – The factor of the vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasDgbmv` (*order, trans, M, N, KL, KU, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasDgbmv`

Matrix-vector product with a general rectangular banded matrix and double elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

$$\bullet (Y \leftarrow \alpha A^T X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in banded matrix **A**.
- **N** (*int [in]*) – Number of columns in banded matrix **A**.
- **KL** (*int [in]*) – Number of sub-diagonals in banded matrix **A**.
- **KU** (*int [in]*) – Number of super-diagonals in banded matrix **A**.
- **alpha** (*float [in]*) – The factor of banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing banded matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in banded matrix **A**.
- **lda** (*int [in]*) – Leading dimension of banded matrix **A**. It cannot be less than (**KL** + **KU** + 1).
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **beta** (*float [in]*) – The factor of the vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasSgbmv` (*order, trans, M, N, KL, KU, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasSgbmv`

Matrix-vector product with a general rectangular banded matrix and float elements. Matrix-vector products:

- $(Y \leftarrow \alpha A X + \beta Y)$
- $(Y \leftarrow \alpha A^T X + \beta Y)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in banded matrix **A**.
- **N** (*int [in]*) – Number of columns in banded matrix **A**.
- **KL** (*int [in]*) – Number of sub-diagonals in banded matrix **A**.
- **KU** (*int [in]*) – Number of super-diagonals in banded matrix **A**.
- **alpha** (*float [in]*) – The factor of banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing banded matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in banded matrix **A**.
- **lda** (*int [in]*) – Leading dimension of banded matrix **A**. It cannot be less than $(KL + KU + 1)$.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **beta** (*float [in]*) – The factor of the vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasZgbmv` (*order, trans, M, N, KL, KU, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasZgbmv`

Matrix-vector product with a general rectangular banded matrix and double-complex elements. Matrix-vector products:

- $(Y \leftarrow \alpha A X + \beta Y)$
- $(Y \leftarrow \alpha A^T X + \beta Y)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **M** (*int [in]*) – Number of rows in banded matrix **A**.
- **N** (*int [in]*) – Number of columns in banded matrix **A**.
- **KL** (*int [in]*) – Number of sub-diagonals in banded matrix **A**.
- **KU** (*int [in]*) – Number of super-diagonals in banded matrix **A**.
- **alpha** (*complex [in]*) – The factor of banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing banded matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in banded matrix **A**.
- **lda** (*int [in]*) – Leading dimension of banded matrix **A**. It cannot be less than (**KL** + **KU** + 1).
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **beta** (*complex [in]*) – The factor of the vector **Y**.
- **y** (*pyopencl.Buffer [out]*) – Buffer object storing the vector **y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of **Y**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

TBMV - Triangular banded matrix vector multiply

`pyclblas.clblasCtbnv` (*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)
wraps: `clblasCtbnv`

Matrix-vector product with a triangular banded matrix and float-complex elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.

- **diag**(*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N**(*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K**(*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A**(*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa**(*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int [in]*) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x**(*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx**(*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx**(*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of (1 + (N-1)*abs(incx)) elements.
- **commandQueues**(*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of pyopencl.CommandQueue. Must not be None.
- **eventWaitList**(*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of pyopencl.Event. May be None.

Returns A tuple of pyopencl.Event instances, one for each commandQueue supplied.

`pycblas.clblasDtbmv`(*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)
wraps: `clblasDtbmv`

Matrix-vector product with a triangular banded matrix and double elements. Matrix-vector products:

- ($X \leftarrow A X$)
- ($X \leftarrow A^T X$)

Parameters

- **order**(*clblasOrder [in]*) – Row/column order.
- **uplo**(*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans**(*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N**(*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K**(*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A**(*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa**(*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int [in]*) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x**(*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx**(*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx**(*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff**(*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of (1 + (N-1)*abs(incx)) elements.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasStbmv` (*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasStbmv`

Matrix-vector product with a triangular banded matrix and float elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than $(K + 1)$.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary `cl_mem` scratch buffer object which can hold a minimum of $(1 + (N-1)*abs(incx))$ elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasZtbmv` (*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, scratchBuff, commandQueues, eventWaitList*)

wraps: `clblasZtbmv`

Matrix-vector product with a triangular banded matrix and double-complex elements. Matrix-vector products:

- $(X \leftarrow A X)$
- $(X \leftarrow A^T X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **scratchBuff** (*pyopencl.Buffer [in]*) – Temporary cl_mem scratch buffer object which can hold a minimum of (1 + (N-1)*abs(incx)) elements.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of pyopencl.CommandQueue. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of pyopencl.Event. May be None.

Returns A tuple of pyopencl.Event instances, one for each commandQueue supplied.

SBMV - Symmetric banded matrix-vector multiplication

pycblas.**clblasDsmbv** (*order, uplo, N, K, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)
 wraps: *clblasDsmbv*

Matrix-vector product with a symmetric banded matrix and double elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in banded matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.

- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*float [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasSsbmv` (*order, uplo, N, K, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)
wraps: `clblasSsbmv`

Matrix-vector product with a symmetric banded matrix and float elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in banded matrix **A**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than $(K + 1)$.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*float [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

HBMV - Hermitian banded matrix-vector multiplication

`pycblas.clblasChbmV` (*order, uplo, N, K, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasChbmV`

Matrix-vector product with a hermitian banded matrix and float elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in banded matrix **A**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than $(K + 1)$.
- **X** (*pyopencl.Buffer [in]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx** (*int [in]*) – Increment for the elements of vector **X**. It cannot be zero.
- **beta** (*complex [in]*) – The factor of vector **Y**.
- **Y** (*pyopencl.Buffer [out]*) – Buffer object storing vector **Y**.
- **offy** (*int [in]*) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy** (*int [in]*) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZhbmV` (*order, uplo, N, K, alpha, A, offa, lda, X, offx, incx, beta, Y, offy, incy, commandQueues, eventWaitList*)

wraps: `clblasZhbmV`

Matrix-vector product with a hermitian banded matrix and double elements. Matrix-vector products:

$$\bullet (Y \leftarrow \alpha A X + \beta Y)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/columns order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **N** (*int [in]*) – Number of rows and columns in banded matrix **A**.

- **K**(*int* [*in*]) – Number of sub-diagonals/super-diagonals in banded matrix **A**.
- **alpha**(*complex* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **A**.
- **offa**(*int* [*in*]) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x**(*pyopencl.Buffer* [*in*]) – Buffer object storing vector **X**.
- **offx**(*int* [*in*]) – Offset of first element of vector **X** in buffer object. Counted in elements.
- **incx**(*int* [*in*]) – Increment for the elements of vector **X**. It cannot be zero.
- **beta**(*complex* [*in*]) – The factor of vector **Y**.
- **y**(*pyopencl.Buffer* [*out*]) – Buffer object storing vector **Y**.
- **offy**(*int* [*in*]) – Offset of first element of vector **Y** in buffer object. Counted in elements.
- **incy**(*int* [*in*]) – Increment for the elements of vector **Y**. It cannot be zero.
- **commandQueues**(*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

TBSV - Solving triangular banded matrix

`pyclblas.clblasCtbsv`(*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)
wraps: `clblasCtbsv`

solving triangular banded matrix problems with float-complex elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order**(*clblasOrder* [*in*]) – Row/column order.
- **uplo**(*clblasUplo* [*in*]) – The triangle in matrix being referenced.
- **trans**(*clblasTranspose* [*in*]) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag* [*in*]) – Specify whether matrix **A** is unit triangular.
- **N**(*int* [*in*]) – Number of rows/columns in banded matrix **A**.
- **K**(*int* [*in*]) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **A**.
- **offa**(*int* [*in*]) – Offset in number of elements for first element in matrix **A**.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x**(*pyopencl.Buffer* [*out*]) – Buffer object storing vector **X**.

- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasDtbsv` (*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasDtbsv`

solving triangular banded matrix problems with double elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than $(K + 1)$.
- **X** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasStbsv` (*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)

wraps: `clblasStbsv`

solving triangular banded matrix problems with float elements. Matrix-vector products:

- $(A X \leftarrow X)$
- $(A^T X \leftarrow X)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasZtbsv` (*order, uplo, trans, diag, N, K, A, offa, lda, X, offx, incx, commandQueues, eventWaitList*)
wraps: `clblasZtbsv`

solving triangular banded matrix problems with double-complex elements. Matrix-vector products:

- ($A X \leftarrow X$)
- ($A^T X \leftarrow X$)

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **trans** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix **A** is unit triangular.
- **N** (*int [in]*) – Number of rows/columns in banded matrix **A**.
- **K** (*int [in]*) – Number of sub-diagonals/super-diagonals in triangular banded matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than (**K** + 1).
- **x** (*pyopencl.Buffer [out]*) – Buffer object storing vector **X**.
- **offx** (*int [in]*) – Offset in number of elements for first element in vector **X**.
- **incx** (*int [in]*) – Increment for the elements of **X**. Must not be zero.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

BLAS Level 3 Functions

GEMM - General matrix-matrix multiplication

`pyclblas.clblasCgemm` (*order, transA, transB, M, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)
wraps: `clblasCgemm`

Matrix-matrix product of general rectangular matrices with float complex elements. Extended version. <dl class="section note"> <dt> Note </dt> <dd> This function is not thread-safe. </dd> </dl> Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha A^T B + \beta C)$
- $(C \leftarrow \alpha A B^T + \beta C)$
- $(C \leftarrow \alpha A^T B^T + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **transB** (*clblasTranspose [in]*) – How matrix **B** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **K** (*int [in]*) – Number of columns in matrix **A** and rows in matrix **B**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.

- **ldc** (*int [in]*) – Leading dimension of matrix **C**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasDgemm` (*order, transA, transB, M, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasDgemm`

Matrix-matrix product of general rectangular matrices with double elements. Extended version. <dl class="section note"> <dt> Note </dt> <dd> This function is not thread-safe. </dd> </dl> Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha A^T B + \beta C)$
- $(C \leftarrow \alpha A B^T + \beta C)$
- $(C \leftarrow \alpha A^T B^T + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **transB** (*clblasTranspose [in]*) – How matrix **B** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **K** (*int [in]*) – Number of columns in matrix **A** and rows in matrix **B**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **beta** (*float [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pycblas.clblasSgemm (*order, transA, transB, M, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: *clblasSgemm*

Matrix-matrix product of general rectangular matrices with float elements. Extended version. <dl class="section note"> <dt> Note </dt> <dd> This function is not thread-safe. </dd> </dl> Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha A^T B + \beta C)$
- $(C \leftarrow \alpha A B^T + \beta C)$
- $(C \leftarrow \alpha A^T B^T + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **transB** (*clblasTranspose [in]*) – How matrix **B** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **K** (*int [in]*) – Number of columns in matrix **A** and rows in matrix **B**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **K** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when the parameter is set to **clblasColumnMajor**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **K** when it is set to **clblasColumnMajor**.
- **beta** (*float [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajorOrder**.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasZgemm` (*order, transA, transB, M, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasZgemm`

Matrix-matrix product of general rectangular matrices with double complex elements. Extended version. <dl class="section note"> <dt> Note </dt> <dd> This function is not thread-safe. </dd> </dl> Matrix-matrix products:

$$\bullet (C \leftarrow \alpha A B + \beta C)$$

$$\bullet (C \leftarrow \alpha A^T B + \beta C)$$

$$\bullet (C \leftarrow \alpha A B^T + \beta C)$$

$$\bullet (C \leftarrow \alpha A^T B^T + \beta C)$$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **transB** (*clblasTranspose [in]*) – How matrix **B** is to be transposed.
- **M** (*int [in]*) – Number of rows in matrix **A**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **K** (*int [in]*) – Number of columns in matrix **A** and rows in matrix **B**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

TRMM - Triangular matrix-matrix multiplication

pycblas.clblasCtrmm (*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)

wraps: *clblasCtrmm*

Multiplying a matrix by a triangular matrix with float complex elements. Extended version. Matrix-triangular matrix products:

- $(B \leftarrow \alpha A B)$
- $(B \leftarrow \alpha A^T B)$
- $(B \leftarrow \alpha B A)$
- $(B \leftarrow \alpha B A^T)$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M** (*int [in]*) – Number of rows in matrix **B**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasDtrmm`(*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)

wraps: `clblasDtrmm`

Multiplying a matrix by a triangular matrix with double elements. Extended version. Matrix-triangular matrix products:

- $(B \leftarrow \alpha A B)$
- $(B \leftarrow \alpha A^T B)$
- $(B \leftarrow \alpha B A)$
- $(B \leftarrow \alpha B A^T)$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M** (*int [in]*) – Number of rows in matrix **B**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasStrmm`(*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)

wraps: `clblasStrmm`

Multiplying a matrix by a triangular matrix with float elements. Extended version. Matrix-triangular matrix products:

- $(B \leftarrow \alpha A B)$
- $(B \leftarrow \alpha A^T B)$

- $(B \leftarrow \alpha B A)$
- $(B \leftarrow \alpha B A^T)$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M** (*int [in]*) – Number of rows in matrix **B**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when it is set to **clblasRight**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or not less than **M** when it is set to **clblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasZtrmm` (*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)
wraps: `clblasZtrmm`

Multiplying a matrix by a triangular matrix with double complex elements. Extended version. Matrix-triangular matrix products:

- $(B \leftarrow \alpha A B)$
- $(B \leftarrow \alpha A^T B)$
- $(B \leftarrow \alpha B A)$
- $(B \leftarrow \alpha B A^T)$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M** (*int [in]*) – Number of rows in matrix **B**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be None.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

TRSM - Solving triangular systems of equations

`pycblas.clblasCtrsm` (*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)

wraps: `clblasCtrsm`

Solving triangular systems of equations with multiple right-hand sides and float complex elements. Extended version. Solving triangular systems of equations:

- $(B \leftarrow \alpha A^{-1} B)$
- $(B \leftarrow \alpha A^{\{-T\}} B)$
- $(B \leftarrow \alpha B A^{-1})$
- $(B \leftarrow \alpha B A^{\{-T\}})$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.

- **transA**(*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M**(*int [in]*) – Number of rows in matrix **B**.
- **N**(*int [in]*) – Number of columns in matrix **B**.
- **alpha**(*complex [in]*) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA**(*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda**(*int [in]*) – Leading dimension of matrix **A**.
- **B**(*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB**(*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb**(*int [in]*) – Leading dimension of matrix **B**.
- **commandQueues**(*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasDtrsm`(*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)

wraps: `clblasDtrsm`

Solving triangular systems of equations with multiple right-hand sides and double elements. Extended version.
<dl class="section note"><dt>Note </dt><dd> This function is not thread-safe. </dd></dl> Solving triangular systems of equations:

- $(B \leftarrow \alpha A^{-1}B)$
- $(B \leftarrow \alpha A^{\{-T\}}B)$
- $(B \leftarrow \alpha B A^{-1})$
- $(B \leftarrow \alpha B A^{\{-T\}})$

where **T** is an upper or lower triangular matrix.

Parameters

- **order**(*clblasOrder [in]*) – Row/column order.
- **side**(*clblasSide [in]*) – The side of triangular matrix.
- **uplo**(*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA**(*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag**(*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M**(*int [in]*) – Number of rows in matrix **B**.
- **N**(*int [in]*) – Number of columns in matrix **B**.
- **alpha**(*float [in]*) – The factor of matrix **A**.

- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

pyclblas.clblasStrsm(*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)

wraps: *clblasStrsm*

Solving triangular systems of equations with multiple right-hand sides and float elements. Extended version. <dl class="section note"> <dt> Note </dt> <dd> This function is not thread-safe. </dd> </dl> Solving triangular systems of equations:

- $(B \leftarrow \alpha A^{-1}B)$
- $(B \leftarrow \alpha A^{\{-T\}} B)$
- $(B \leftarrow \alpha B A^{-1})$
- $(B \leftarrow \alpha B A^{\{-T\}})$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M** (*int [in]*) – Number of rows in matrix **B**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when it is set to **clblasRight**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.

- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZtrsm`(*order, side, uplo, transA, diag, M, N, alpha, A, offA, lda, B, offB, ldb, commandQueues, eventWaitList*)
wraps: `clblasZtrsm`

Solving triangular systems of equations with multiple right-hand sides and double complex elements. Extended version. Solving triangular systems of equations:

- $(B \leftarrow \alpha A^{-1}B)$
- $(B \leftarrow \alpha A^{\{-T\}}B)$
- $(B \leftarrow \alpha B A^{-1})$
- $(B \leftarrow \alpha B A^{\{-T\}})$

where **T** is an upper or lower triangular matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **diag** (*clblasDiag [in]*) – Specify whether matrix is unit triangular.
- **M** (*int [in]*) – Number of rows in matrix **B**.
- **N** (*int [in]*) – Number of columns in matrix **B**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.

- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

SYRK - Symmetric rank-k update of a matrix

`pycblas.clblasCsyrk` (*order, uplo, transA, N, K, alpha, A, offA, lda, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasCsyrk`

Rank-k update of a symmetric matrix with complex float elements. Extended version. Rank-k updates:

- $(C \leftarrow \alpha A A^T + \beta C)$
- $(C \leftarrow \alpha A^T A + \beta C)$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing the matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **beta** (*complex [in]*) – The factor of the matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasDsyrik` (*order, uplo, transA, N, K, alpha, A, offA, lda, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasDsyrik`

Rank-k update of a symmetric matrix with double elements. Extended version. Rank-k updates:

$$\bullet (C \leftarrow \alpha A A^T + \beta C)$$

$$\bullet (C \leftarrow \alpha A^T A + \beta C)$$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing the matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **beta** (*float [in]*) – The factor of the matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.cblasSsyrrk` (*order, uplo, transA, N, K, alpha, A, offA, lda, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasSsyrrk`

Rank-k update of a symmetric matrix with float elements. Extended version. Rank-k updates:

$$\bullet (C \leftarrow \alpha A A^T + \beta C)$$

$$\bullet (C \leftarrow \alpha A^T A + \beta C)$$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.

- **K**(*int* [*in*]) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha**(*float* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing the matrix **A**.
- **offA**(*int* [*in*]) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**. It cannot be less than **K** if **A** is in the row-major format, and less than **N** otherwise.
- **beta**(*float* [*in*]) – The factor of the matrix **C**.
- **C**(*pyopencl.Buffer* [*out*]) – Buffer object storing matrix **C**.
- **offC**(*int* [*in*]) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc**(*int* [*in*]) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues**(*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.cblasZsyrk`(*order*, *uplo*, *transA*, *N*, *K*, *alpha*, *A*, *offA*, *lda*, *beta*, *C*, *offC*, *ldc*, *commandQueues*, *eventWaitList*)
 wraps: `cblasZsyrk`

Rank-k update of a symmetric matrix with complex double elements. Extended version. Rank-k updates:

- $(C \leftarrow \alpha A A^T + \beta C)$
- $(C \leftarrow \alpha A^T A + \beta C)$

where **C** is a symmetric matrix.

Parameters

- **order**(*cblasOrder* [*in*]) – Row/column order.
- **uplo**(*cblasUplo* [*in*]) – The triangle in matrix **C** being referenced.
- **transA**(*cblasTranspose* [*in*]) – How matrix **A** is to be transposed.
- **N**(*int* [*in*]) – Number of rows and columns in matrix **C**.
- **K**(*int* [*in*]) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha**(*complex* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing the matrix **A**.
- **offA**(*int* [*in*]) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**.
- **beta**(*complex* [*in*]) – The factor of the matrix **C**.
- **C**(*pyopencl.Buffer* [*out*]) – Buffer object storing matrix **C**.

- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

SYR2K - Symmetric rank-2k update to a matrix

`pycblas.clblasCsyr2k` (*order, uplo, transAB, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)
wraps: `clblasCsyr2k`

Rank-2k update of a symmetric matrix with complex float elements. Extended version. Rank-k updates:

- $(C \leftarrow \alpha A B^T + \alpha B A^T + \beta C)$
- $(C \leftarrow \alpha A^T B + \alpha B^T A + \beta C)$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transAB** (*clblasTranspose [in]*) – How matrices **A** and **B** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrices **A** and **B** if they are not transposed, and number of rows otherwise.
- **alpha** (*complex [in]*) – The factor of matrices **A** and **B**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.

- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pyclblas.clblasDsyrr2k` (*order, uplo, transAB, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasDsyrr2k`

Rank-2k update of a symmetric matrix with double elements. Extended version. Rank-k updates:

- $(C \leftarrow \alpha A B^T + \alpha B A^T + \beta C)$
- $(C \leftarrow \alpha A^T B + \alpha B^T A + \beta C)$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transAB** (*clblasTranspose [in]*) – How matrices **A** and **B** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrices **A** and **B** if they are not transposed, and number of rows otherwise.
- **alpha** (*float [in]*) – The factor of matrices **A** and **B**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **beta** (*float [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasSsyr2k` (*order, uplo, transAB, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasSsyr2k`

Rank-2k update of a symmetric matrix with float elements. Extended version. Rank-k updates:

- $(C \leftarrow \alpha A B^T + \alpha B A^T + \beta C)$
- $(C \leftarrow \alpha A^T B + \alpha B^T A + \beta C)$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transAB** (*clblasTranspose [in]*) – How matrices **A** and **B** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrices **A** and **B** if they are not transposed, and number of rows otherwise.
- **alpha** (*float [in]*) – The factor of matrices **A** and **B**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **K** if **A** is in the row-major format, and less than **N** otherwise.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less less than **K** if **B** matches to the op(**B**) matrix in the row-major format, and less than **N** otherwise.
- **beta** (*float [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZsyr2k` (*order, uplo, transAB, N, K, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)

wraps: `clblasZsyr2k`

Rank-2k update of a symmetric matrix with complex double elements. Extended version. Rank-k updates:

- $(C \leftarrow \alpha A B^T + \alpha B A^T + \beta C)$
- $(C \leftarrow \alpha A^T B + \alpha B^T A + \beta C)$

where **C** is a symmetric matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transAB** (*clblasTranspose [in]*) – How matrices **A** and **B** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrices **A** and **B** if they are not transposed, and number of rows otherwise.
- **alpha** (*complex [in]*) – The factor of matrices **A** and **B**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offA** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offB** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offC** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

SYMM - Symmetric matrix-matrix multiply

`pyclblas.cblasCsymm` (*order, side, uplo, M, N, alpha, A, offA, lda, B, offB, ldb, beta, C, offC, ldc, commandQueues, eventWaitList*)
wraps: `cblasCsymm`

Matrix-matrix product of symmetric rectangular matrices with float-complex elements. Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha B A + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **M** (*int [in]*) – Number of rows in matrices **B** and **C**.
- **N** (*int [in]*) – Number of columns in matrices **B** and **C**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when the parameter is set to **clblasRight**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offb** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajorOrder**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pycblas.clblasDsymbb (order, side, uplo, M, N, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList)`
wraps: `clblasDsymbb`

Matrix-matrix product of symmetric rectangular matrices with double elements. Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha B A + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **M** (*int [in]*) – Number of rows in matrices **B** and **C**.

- **N**(*int* [*in*]) – Number of columns in matrices **B** and **C**.
- **alpha**(*float* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **A**.
- **offa**(*int* [*in*]) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda**(*int* [*in*]) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when the parameter is set to **clblasRight**.
- **B**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **B**.
- **offb**(*int* [*in*]) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb**(*int* [*in*]) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **beta**(*float* [*in*]) – The factor of matrix **C**.
- **C**(*pyopencl.Buffer* [*out*]) – Buffer object storing matrix **C**.
- **offc**(*int* [*in*]) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc**(*int* [*in*]) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **commandQueues**(*pyopencl.CommandQueue* [*in*]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList**(*pyopencl.Event* [*in*]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

`pyclblas.clblasSsymm`(*order, side, uplo, M, N, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList*)
wraps: `clblasSsymm`

Matrix-matrix product of symmetric rectangular matrices with float elements. Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha B A + \beta C)$

Parameters

- **order**(*clblasOrder* [*in*]) – Row/column order.
- **side**(*clblasSide* [*in*]) – The side of triangular matrix.
- **uplo**(*clblasUplo* [*in*]) – The triangle in matrix being referenced.
- **M**(*int* [*in*]) – Number of rows in matrices **B** and **C**.
- **N**(*int* [*in*]) – Number of columns in matrices **B** and **C**.
- **alpha**(*float* [*in*]) – The factor of matrix **A**.
- **A**(*pyopencl.Buffer* [*in*]) – Buffer object storing matrix **A**.

- **offa** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when the parameter is set to **clblasRight**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offb** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **beta** (*float [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajorOrder**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZsymm` (*order, side, uplo, M, N, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList*)
 wraps: `clblasZsymm`

Matrix-matrix product of symmetric rectangular matrices with double-complex elements. Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha B A + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **M** (*int [in]*) – Number of rows in matrices **B** and **C**.
- **N** (*int [in]*) – Number of columns in matrices **B** and **C**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when the parameter is set to **clblasRight**.

- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offb** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajorOrder**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

HEMM - Hermitian matrix-matrix multiplication

`pycblas.cblasChemmm` (*order, side, uplo, M, N, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList*)
 wraps: `clblasChemmm`

Matrix-matrix product of hermitian rectangular matrices with float-complex elements. Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha B A + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **M** (*int [in]*) – Number of rows in matrices **B** and **C**.
- **N** (*int [in]*) – Number of columns in matrices **B** and **C**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when the parameter is set to **clblasRight**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.

- **offb** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajorOrder**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZhemm` (*order, side, uplo, M, N, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList*)
wraps: `clblasZhemm`

Matrix-matrix product of hermitian rectangular matrices with double-complex elements. Matrix-matrix products:

- $(C \leftarrow \alpha A B + \beta C)$
- $(C \leftarrow \alpha B A + \beta C)$

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **side** (*clblasSide [in]*) – The side of triangular matrix.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix being referenced.
- **M** (*int [in]*) – Number of rows in matrices **B** and **C**.
- **N** (*int [in]*) – Number of columns in matrices **B** and **C**.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **A**.
- **offa** (*int [in]*) – Offset of the first element of the matrix **A** in the buffer object. Counted in elements.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **M** when the **side** parameter is set to **clblasLeft**, or less than **N** when the parameter is set to **clblasRight**.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing matrix **B**.
- **offb** (*int [in]*) – Offset of the first element of the matrix **B** in the buffer object. Counted in elements.

- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajor**.
- **beta** (*complex [in]*) – The factor of matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset of the first element of the matrix **C** in the buffer object. Counted in elements.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N** when the **order** parameter is set to **clblasRowMajor**, or less than **M** when it is set to **clblasColumnMajorOrder**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

HERK - Hermitian rank-k update to a matrix

`pycblas.clblasCherk` (*order, uplo, transA, N, K, alpha, A, offa, lda, beta, C, offc, ldc, commandQueues, eventWaitList*)
wraps: `clblasCherk`

Rank-k update of a hermitian matrix with float-complex elements. Rank-k updates:

- $(C \leftarrow \alpha A A^H + \beta C)$
- $(C \leftarrow \alpha A^H A + \beta C)$

where **C** is a hermitian matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing the matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **K** if **A** is in the row-major format, and less than **N** otherwise.
- **beta** (*float [in]*) – The factor of the matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset in number of elements for the first element in matrix **C**.

- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.clblasZherk` (*order, uplo, transA, N, K, alpha, A, offa, lda, beta, C, offc, ldc, commandQueues, eventWaitList*)
 wraps: `clblasZherk`

Rank-k update of a hermitian matrix with double-complex elements. Rank-k updates:

- $(C \leftarrow \alpha A A^H + \beta C)$
- $(C \leftarrow \alpha A^H A + \beta C)$

where **C** is a hermitian matrix.

Parameters

- **order** (*clblasOrder [in]*) – Row/column order.
- **uplo** (*clblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **transA** (*clblasTranspose [in]*) – How matrix **A** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha** (*float [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing the matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **K** if **A** is in the row-major format, and less than **N** otherwise.
- **beta** (*float [in]*) – The factor of the matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset in number of elements for the first element in matrix **C**.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be `None`.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be `None`.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

HER2K - Hermitian rank-2k update to a matrix

`pycblas.cblasCher2k` (*order, uplo, trans, N, K, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList*)
 wraps: `cblasCher2k`

Rank-2k update of a hermitian matrix with float-complex elements. Rank-k updates:

- $(C \leftarrow \alpha A B^H + \text{conj}(\alpha) B A^H + \beta C)$
- $(C \leftarrow \alpha A^H B + \text{conj}(\alpha) B^H A + \beta C)$

where **C** is a hermitian matrix.

Parameters

- **order** (*cblasOrder [in]*) – Row/column order.
- **uplo** (*cblasUplo [in]*) – The triangle in matrix **C** being referenced.
- **trans** (*cblasTranspose [in]*) – How matrix **A** is to be transposed.
- **N** (*int [in]*) – Number of rows and columns in matrix **C**.
- **K** (*int [in]*) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha** (*complex [in]*) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer [in]*) – Buffer object storing the matrix **A**.
- **offa** (*int [in]*) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int [in]*) – Leading dimension of matrix **A**. It cannot be less than **K** if **A** is in the row-major format, and less than **N** otherwise. Vice-versa for transpose case.
- **B** (*pyopencl.Buffer [in]*) – Buffer object storing the matrix **B**.
- **offb** (*int [in]*) – Offset in number of elements for the first element in matrix **B**.
- **ldb** (*int [in]*) – Leading dimension of matrix **B**. It cannot be less than **K** if **B** is in the row-major format, and less than **N** otherwise. Vice-versa for transpose case.
- **beta** (*float [in]*) – The factor of the matrix **C**.
- **C** (*pyopencl.Buffer [out]*) – Buffer object storing matrix **C**.
- **offc** (*int [in]*) – Offset in number of elements for the first element in matrix **C**.
- **ldc** (*int [in]*) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue [in]*) – OpenCL command queues. A list, tuple, or single instance of `pyopencl.CommandQueue`. Must not be None.
- **eventWaitList** (*pyopencl.Event [in]*) – Event wait list. A list, tuple, or single instance of `pyopencl.Event`. May be None.

Returns A tuple of `pyopencl.Event` instances, one for each `commandQueue` supplied.

`pycblas.cblasZher2k` (*order, uplo, trans, N, K, alpha, A, offa, lda, B, offb, ldb, beta, C, offc, ldc, commandQueues, eventWaitList*)
 wraps: `cblasZher2k`

Rank-2k update of a hermitian matrix with double-complex elements. Rank-k updates:

- $(C \leftarrow \alpha A B^H + \text{conj}(\alpha) B A^H + \beta C)$

$$\bullet (C \leftarrow \alpha A^H B + \text{conj}(\alpha) B^H A + \beta C)$$

where **C** is a hermitian matrix.

Parameters

- **order** (*clblasOrder* [in]) – Row/column order.
- **uplo** (*clblasUplo* [in]) – The triangle in matrix **C** being referenced.
- **trans** (*clblasTranspose* [in]) – How matrix **A** is to be transposed.
- **N** (*int* [in]) – Number of rows and columns in matrix **C**.
- **K** (*int* [in]) – Number of columns of the matrix **A** if it is not transposed, and number of rows otherwise.
- **alpha** (*complex* [in]) – The factor of matrix **A**.
- **A** (*pyopencl.Buffer* [in]) – Buffer object storing the matrix **A**.
- **offa** (*int* [in]) – Offset in number of elements for the first element in matrix **A**.
- **lda** (*int* [in]) – Leading dimension of matrix **A**. It cannot be less than **K** if **A** is in the row-major format, and less than **N** otherwise. Vice-versa for transpose case.
- **B** (*pyopencl.Buffer* [in]) – Buffer object storing the matrix **B**.
- **offb** (*int* [in]) – Offset in number of elements for the first element in matrix **B**.
- **ldb** (*int* [in]) – Leading dimension of matrix **B**. It cannot be less than **K** if **B** is in the row-major format, and less than **N** otherwise. Vice-versa for transpose case.
- **beta** (*float* [in]) – The factor of the matrix **C**.
- **C** (*pyopencl.Buffer* [out]) – Buffer object storing matrix **C**.
- **offc** (*int* [in]) – Offset in number of elements for the first element in matrix **C**.
- **ldc** (*int* [in]) – Leading dimension of matrix **C**. It cannot be less than **N**.
- **commandQueues** (*pyopencl.CommandQueue* [in]) – OpenCL command queues. A list, tuple, or single instance of *pyopencl.CommandQueue*. Must not be *None*.
- **eventWaitList** (*pyopencl.Event* [in]) – Event wait list. A list, tuple, or single instance of *pyopencl.Event*. May be *None*.

Returns A tuple of *pyopencl.Event* instances, one for each *commandQueue* supplied.

p

pyclblas, [24](#)

C

- `clblasCaxpy()` (in module `pyclblas`), 10
- `clblasCcopy()` (in module `pyclblas`), 8
- `clblasCdotc()` (in module `pyclblas`), 14
- `clblasCdotu()` (in module `pyclblas`), 13
- `clblasCgbmv()` (in module `pyclblas`), 62
- `clblasCgemm()` (in module `pyclblas`), 74
- `clblasCgemv()` (in module `pyclblas`), 28
- `clblasCgerc()` (in module `pyclblas`), 43
- `clblasCgeru()` (in module `pyclblas`), 41
- `clblasChbmv()` (in module `pyclblas`), 70
- `clblasChemmm()` (in module `pyclblas`), 95
- `clblasChemv()` (in module `pyclblas`), 33
- `clblasCher()` (in module `pyclblas`), 45
- `clblasCher2()` (in module `pyclblas`), 48
- `clblasCher2k()` (in module `pyclblas`), 99
- `clblasCherk()` (in module `pyclblas`), 97
- `clblasChpmv()` (in module `pyclblas`), 55
- `clblasChpr()` (in module `pyclblas`), 58
- `clblasChpr2()` (in module `pyclblas`), 60
- `clblasCrotg()` (in module `pyclblas`), 15
- `clblasCscal()` (in module `pyclblas`), 5
- `clblasCsrot()` (in module `pyclblas`), 20
- `clblasCsscal()` (in module `pyclblas`), 7
- `clblasCswap()` (in module `pyclblas`), 3
- `clblasCsymm()` (in module `pyclblas`), 91
- `clblasCsyr2k()` (in module `pyclblas`), 88
- `clblasCsyrk()` (in module `pyclblas`), 85
- `clblasCtbmv()` (in module `pyclblas`), 65
- `clblasCtbsv()` (in module `pyclblas`), 71
- `clblasCtpmv()` (in module `pyclblas`), 49
- `clblasCtpsv()` (in module `pyclblas`), 52
- `clblasCtrmm()` (in module `pyclblas`), 78
- `clblasCtrmv()` (in module `pyclblas`), 34
- `clblasCtrsm()` (in module `pyclblas`), 81
- `clblasCtrsv()` (in module `pyclblas`), 37
- `clblasDasum()` (in module `pyclblas`), 26
- `clblasDaxpy()` (in module `pyclblas`), 10
- `clblasDcopy()` (in module `pyclblas`), 8
- `clblasDdot()` (in module `pyclblas`), 12
- `clblasDgbmv()` (in module `pyclblas`), 63
- `clblasDgemm()` (in module `pyclblas`), 75
- `clblasDgemv()` (in module `pyclblas`), 29
- `clblasDger()` (in module `pyclblas`), 40
- `clblasDnrm2()` (in module `pyclblas`), 22
- `clblasDrot()` (in module `pyclblas`), 19
- `clblasDrotg()` (in module `pyclblas`), 16
- `clblasDrotm()` (in module `pyclblas`), 21
- `clblasDrotmg()` (in module `pyclblas`), 17
- `clblasDsbbmv()` (in module `pyclblas`), 68
- `clblasDscal()` (in module `pyclblas`), 5
- `clblasDspmv()` (in module `pyclblas`), 54
- `clblasDspr()` (in module `pyclblas`), 57
- `clblasDspr2()` (in module `pyclblas`), 59
- `clblasDswap()` (in module `pyclblas`), 3
- `clblasDsymm()` (in module `pyclblas`), 92
- `clblasDsymv()` (in module `pyclblas`), 31
- `clblasDsyr()` (in module `pyclblas`), 44
- `clblasDsyr2()` (in module `pyclblas`), 46
- `clblasDsyr2k()` (in module `pyclblas`), 89
- `clblasDsyrk()` (in module `pyclblas`), 85
- `clblasDtbbmv()` (in module `pyclblas`), 66
- `clblasDtbsv()` (in module `pyclblas`), 72
- `clblasDtpmv()` (in module `pyclblas`), 50
- `clblasDtpsv()` (in module `pyclblas`), 52
- `clblasDtrmm()` (in module `pyclblas`), 78
- `clblasDtrmv()` (in module `pyclblas`), 35
- `clblasDtrsm()` (in module `pyclblas`), 82
- `clblasDtrsv()` (in module `pyclblas`), 38
- `clblasDzasum()` (in module `pyclblas`), 27
- `clblasDznrm2()` (in module `pyclblas`), 23
- `clblasiCamax()` (in module `pyclblas`), 25
- `clblasiDamax()` (in module `pyclblas`), 24
- `clblasiSamax()` (in module `pyclblas`), 24
- `clblasiZamax()` (in module `pyclblas`), 25
- `clblasSasum()` (in module `pyclblas`), 26
- `clblasSaxpy()` (in module `pyclblas`), 11
- `clblasScasum()` (in module `pyclblas`), 27
- `clblasScnrm2()` (in module `pyclblas`), 23

cblasScopy() (in module pycblas), 9
 cblasSdot() (in module pycblas), 12
 cblasSgbmv() (in module pycblas), 63
 cblasSgemm() (in module pycblas), 76
 cblasSgemv() (in module pycblas), 30
 cblasSger() (in module pycblas), 40
 cblasSnrm2() (in module pycblas), 22
 cblasSrot() (in module pycblas), 19
 cblasSrotg() (in module pycblas), 16
 cblasSrotm() (in module pycblas), 21
 cblasSrotmg() (in module pycblas), 18
 cblasSsbmv() (in module pycblas), 69
 cblasSscal() (in module pycblas), 6
 cblasSspmv() (in module pycblas), 55
 cblasSspr() (in module pycblas), 57
 cblasSspr2() (in module pycblas), 60
 cblasSswap() (in module pycblas), 4
 cblasSsymm() (in module pycblas), 93
 cblasSsymv() (in module pycblas), 32
 cblasSsyr() (in module pycblas), 45
 cblasSsyr2() (in module pycblas), 47
 cblasSsyr2k() (in module pycblas), 89
 cblasSsyrk() (in module pycblas), 86
 cblasStbmv() (in module pycblas), 67
 cblasStbsv() (in module pycblas), 72
 cblasStpmv() (in module pycblas), 50
 cblasStpsv() (in module pycblas), 53
 cblasStrmm() (in module pycblas), 79
 cblasStrmv() (in module pycblas), 36
 cblasStrsm() (in module pycblas), 83
 cblasStrsv() (in module pycblas), 38
 cblasZaxpy() (in module pycblas), 11
 cblasZcopy() (in module pycblas), 9
 cblasZdotc() (in module pycblas), 15
 cblasZdotu() (in module pycblas), 14
 cblasZdrot() (in module pycblas), 20
 cblasZdscal() (in module pycblas), 7
 cblasZgbmv() (in module pycblas), 64
 cblasZgemm() (in module pycblas), 77
 cblasZgemv() (in module pycblas), 30
 cblasZgerc() (in module pycblas), 43
 cblasZgeru() (in module pycblas), 42
 cblasZhbmvm() (in module pycblas), 70
 cblasZhemm() (in module pycblas), 96
 cblasZhemv() (in module pycblas), 33
 cblasZher() (in module pycblas), 46
 cblasZher2() (in module pycblas), 48
 cblasZher2k() (in module pycblas), 99
 cblasZherk() (in module pycblas), 98
 cblasZhpmv() (in module pycblas), 56
 cblasZhpr() (in module pycblas), 58
 cblasZhpr2() (in module pycblas), 61
 cblasZrotg() (in module pycblas), 17
 cblasZscal() (in module pycblas), 6

cblasZswap() (in module pycblas), 4
 cblasZsymm() (in module pycblas), 94
 cblasZsyr2k() (in module pycblas), 90
 cblasZsyrk() (in module pycblas), 87
 cblasZtbmv() (in module pycblas), 67
 cblasZtbsv() (in module pycblas), 73
 cblasZtpmv() (in module pycblas), 51
 cblasZtpsv() (in module pycblas), 53
 cblasZtrmm() (in module pycblas), 80
 cblasZtrmv() (in module pycblas), 36
 cblasZtrsm() (in module pycblas), 84
 cblasZtrsv() (in module pycblas), 39

P

pycblas (module), 3, 5, 7, 8, 10, 12, 15, 17, 19, 21, 22,
 24, 26, 28, 31, 33, 34, 37, 40, 41, 43–46, 48, 49,
 52, 54, 55, 57–60, 62, 65, 68, 70, 71, 74, 78, 81,
 85, 88, 91, 95, 97, 99