Rocblas Deverlopers Guide

Xiangmin Jiao

December 11, 2003

Contents

1 Rocblas Compound Index

1.1 Rocblas Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

2 Rocblas File Index

2.1 Rocblas File List

Here is a list of all files with brief descriptions:

3 Rocblas Class Documentation

3.1 Rocblas Class Reference

```
#include <Rocblas.h>
```

Static Public Member Functions

- void init (const std::string &name)

  Creates window for Rocblas and registers functions.

- void finalize (const std::string &name)

  Delete window for Rocblas.

- void add (const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for addition.

- void sub (const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for subtraction.

- void mul (const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for multiplication.

- void limit1 (const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for limit1.

- void div (const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for division.

- void add_scalar (const Attribute ∗x, const void ∗y, Attribute ∗z, int swap=0)

  Operation wrapper for addition with y as a scalar pointer.

- void sub_scalar (const Attribute ∗x, const void ∗y, Attribute ∗z, int swap=0)

  Operation wrapper for subtraction with y as a scalar pointer.

- void mul_scalar (const Attribute ∗x, const void ∗y, Attribute ∗z, int swap=0)

  Operation wrapper for multiplication with y as a scalar pointer.

- void div_scalar (const Attribute ∗x, const void ∗y, Attribute ∗z, int swap=0)

  Operation wrapper for division with y as a scalar pointer.

- void dot (const Attribute ∗x, const Attribute ∗y, Attribute ∗z, const Attribute ∗mults=NULL)

  Wrapper for dot product.

- void dot_scalar (const Attribute ∗x, const Attribute ∗y, void ∗z, const Attribute ∗mults=NULL)

  Wrapper for 2-norm with z as a scalar pointer.

- void dot_MPI (const Attribute ∗x, const Attribute ∗y, Attribute ∗z, const int ∗comm, const Attribute ∗mults=NULL)

  Wrapper for dot product.

- void dot_scalar_MPI (const Attribute ∗x, const Attribute ∗y, void ∗z, const int ∗comm, const Attribute ∗mults=NULL)

  Wrapper for 2-norm with z as a scalar pointer.

- void nrm2 (const Attribute ∗x, Attribute ∗y, const Attribute ∗mults=NULL)

  Wrapper for 2-norm.

- void nrm2_scalar (const Attribute ∗x, void ∗y, const Attribute ∗mults=NULL)

  Wrapper for 2-norm with y as a scalar pointer.

- void nrm2_MPI (const Attribute ∗x, Attribute ∗y, const int ∗comm, const Attribute ∗mults=NULL)

  Wrapper for 2-norm with MPI.

- void nrm2_scalar_MPI (const Attribute ∗x, void ∗y, const int ∗comm, const Attribute ∗mults=NULL)

  Wrapper for 2-norm with y as a scalar pointer with MPI.

- void swap (Attribute ∗x, Attribute ∗y)

  Wrapper for swap.

- void copy (const Attribute ∗x, Attribute ∗y)

  Wrapper for copy.

- void copy_scalar (const void ∗x, Attribute ∗y)

  Operation wrapper for copy (x is a scalar pointer).

- void neg (const Attribute ∗x, Attribute ∗y)

  Wrapper for neg (y=-x).

- void max_MPI (const Attribute ∗x, Attribute ∗y, int ∗comm)

Wrapper for max.

- void max_scalar_MPI (const Attribute ∗x, void ∗y, int ∗comm)

  Operation wrapper for max (y is a scalar pointer).

- void min_MPI (const Attribute ∗x, Attribute ∗y, int ∗comm)

  Wrapper for min.

- void min_scalar_MPI (const Attribute ∗x, void ∗y, int ∗comm)

  Operation wrapper for min (y is a scalar pointer).

- void sum_MPI (const Attribute ∗x, Attribute ∗y, int ∗comm)

  Wrapper for sum.

- void sum_scalar_MPI (const Attribute ∗x, void ∗y, int ∗comm)

  Operation wrapper for sum (y is a scalar pointer).

- void axpy (const Attribute ∗a, const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for z = a ∗ x + y.

- void axpy_scalar (const void ∗a, const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Operation wrapper for z = a ∗ x + y (a is a scalar pointer).

Protected Types

- enum {
  BLAS_VOID, BLAS_SCALAR, BLAS_VEC, BLAS_SCNE,
  BLAS_VECNE }

Static Protected Member Functions

- template<class FuncType, int ytype> void calc (Attribute ∗z, const Attribute ∗x, const void ∗yin, FuncType opp, bool swap=false)

  Performs the operation: z = x op y.

- template<class data_type, int ztype> void calcDot (void ∗zout, const Attribute ∗x, const Attribute ∗y, const int ∗comm=NULL, const Attribute ∗mults=NULL)

  Performs the operation: z = <x, y>.

- template<class FuncType, int ytype> void gen2arg (Attribute ∗z, void ∗yin, FuncType opp)

  Performs the operation opp(x, y).

- template<class data_type, int atype> void axpy_gen (const void ∗a, const Attribute ∗x, const Attribute ∗y, Attribute ∗z)

  Performs the operation: z = a∗x + y.

- template<class FuncType> void calcChoose (const Attribute ∗x, const Attribute ∗y, Attribute ∗z, FuncType opp)

  Chooses which calc function to call based on type of y.

- template<int attr_type> int get_stride (const Attribute ∗attr)
- template<class data_type, int attr_type, bool is_staggered> data_type & getref (data_type ∗base, const int r, const int c, const int nc)
- template<class data_type, int attr_type, bool is_staggered> const data_type & getref (const data_type ∗base, const int r, const int c, const int nc)
- template<class OPint, class OPdbl, int OPMPI> void reduce_MPI (const Attribute ∗x, Attribute ∗z, int ∗comm, int, double)
- template<class OPint, class OPdbl, int OPMPI> void reduce_scalar_MPI (const Attribute ∗x, void ∗y, int ∗comm, int, double)

3.1.1 Member Enumeration Documentation

3.1.1.1 anonymous enum `[protected]`

Enumeration values:

BLAS_VOID

BLAS_SCALAR

BLAS_VEC

BLAS_SCNE

BLAS_VECNE

3.1.2 Member Function Documentation

3.1.2.1 void Rocblas::add (const Attribute ∗ *x*, const Attribute ∗ *y*, Attribute ∗ *z*) `[static]`

Operation wrapper for addition.

3.1.2.2 void Rocblas::add_scalar (const Attribute ∗ *x*, const void ∗ *y*, Attribute ∗ *z*, int *swap* = 0) `[static]`

Operation wrapper for addition with y as a scalar pointer.

3.1.2.3 void Rocblas::axpy (const Attribute ∗ *a*, const Attribute ∗ *x*, const Attribute ∗ *y*, Attribute ∗ *z*) `[static]`

Operation wrapper for z = a ∗ x + y.

3.1.2.4 template<class data_type, int atype> void Rocblas::axpy_gen (const void ∗ *a*, const Attribute ∗ *x*, const Attribute ∗ *y*, Attribute ∗ *z*) `[static, protected]`

Performs the operation: z = a∗x + y.

3.1.2.5 void Rocblas::axpy_scalar (const void ∗ *a*, const Attribute ∗ *x*, const Attribute ∗ *y*, Attribute ∗ *z*) `[static]`

Operation wrapper for z = a ∗ x + y (a is a scalar pointer).

3.1.2.6 template<class FuncType, int ytype> void Rocblas::calc (Attribute ∗ *z*, const Attribute ∗ *x*, const void ∗ *yin*, FuncType *opp*, bool *swap* = false) `[static, protected]`

Performs the operation: z = x op y.

3.1.2.7 template<class FuncType> void Rocblas::calcChoose (const Attribute ∗ *x*, const Attribute ∗ *y*, Attribute ∗ *z*, FuncType *opp*) `[static, protected]`

Chooses which calc function to call based on type of y.

3.1.2.8 template<class data_type, int ytype> void Rocblas::calcDot (void ∗ *zout*, const Attribute ∗ *x*, const Attribute ∗ *y*, const int ∗ *comm* = NULL, const Attribute ∗ *mults* = NULL) `[static, protected]`

Performs the operation: z = <x, y>.

3.1.2.9 void Rocblas::copy (const Attribute ∗ *x*, Attribute ∗ *y*) `[static]`

Wrapper for copy.

3.1.2.10 void Rocblas::copy_scalar (const void * x, Attribute * y) [static]

Operation wrapper for copy (x is a scalar pointer).

3.1.2.11 void Rocblas::div (const Attribute * x, const Attribute * y, Attribute * z) [static]

Operation wrapper for division.

3.1.2.12 void Rocblas::div_scalar (const Attribute * x, const void * y, Attribute * z, int swap = 0) [static]

Operation wrapper for division with y as a scalar pointer.

3.1.2.13 void Rocblas::dot (const Attribute * x, const Attribute * y, Attribute * z, const Attribute * mults = NULL) [static]

Wrapper for dot product.

3.1.2.14 void Rocblas::dot_MPI (const Attribute * x, const Attribute * y, Attribute * z, const int * comm, const Attribute * mults = NULL) [static]

Wrapper for dot product.

3.1.2.15 void Rocblas::dot_scalar (const Attribute * x, const Attribute * y, void * z, const Attribute * mults = NULL) [static]

Wrapper for 2-norm with z as a scalar pointer.

3.1.2.16 void Rocblas::dot_scalar_MPI (const Attribute * x, const Attribute * y, void * z, const int * comm, const Attribute * mults = NULL) [static]

Wrapper for 2-norm with z as a scalar pointer.

3.1.2.17 void Rocblas::finalize (const std::string & name) [static]

Delete window for Rocblas.

3.1.2.18 template<class FuncType, int ytype> void Rocblas::gen2arg (Attribute * z, void * yin, FuncType opp) [static, protected]

Performs the operation opp(x, y).

3.1.2.19 template<int attr_type> int Rocblas::get_stride (const Attribute * attr) [inline, static, protected]

3.1.2.20 template<class data_type, int attr_type, bool is_staggered> const data_type & Rocblas::getref (const data_type * *base*, const int *r*, const int *c*, const int *nc*) `[inline, static, protected]`

3.1.2.21 template<class data_type, int attr_type, bool is_staggered> data_type & Rocblas::getref (data_type * *base*, const int *r*, const int *c*, const int *nc*) `[inline, static, protected]`

3.1.2.22 void Rocblas::init (const std::string & *name*) `[static]`

Creates window for Rocblas and registers functions.

3.1.2.23 void Rocblas::limit1 (const Attribute * *x*, const Attribute * *y*, Attribute * *z*) `[static]`

Operation wrapper for limit1.

3.1.2.24 void Rocblas::max_MPI (const Attribute * *x*, Attribute * *y*, int * *comm*) `[static]`

Wrapper for max.

3.1.2.25 void Rocblas::max_scalar_MPI (const Attribute * *x*, void * *y*, int * *comm*) `[static]`

Operation wrapper for max (y is a scalar pointer).

3.1.2.26 void Rocblas::min_MPI (const Attribute * *x*, Attribute * *y*, int * *comm*) `[static]`

Wrapper for min.

3.1.2.27 void Rocblas::min_scalar_MPI (const Attribute * *x*, void * *y*, int * *comm*) `[static]`

Operation wrapper for min (y is a scalar pointer).

3.1.2.28 void Rocblas::mul (const Attribute * *x*, const Attribute * *y*, Attribute * *z*) `[static]`

Operation wrapper for multiplication.

3.1.2.29 void Rocblas::mul_scalar (const Attribute * x, const void * y, Attribute * z, int swap = 0) [static]

Operation wrapper for multiplication with y as a scalar pointer.

3.1.2.30 void Rocblas::neg (const Attribute * x, Attribute * y) [static]

Wrapper for neg (y=-x).

3.1.2.31 void Rocblas::nrm2 (const Attribute * x, Attribute * y, const Attribute * mults = NULL) [static]

Wrapper for 2-norm.

3.1.2.32 void Rocblas::nrm2_MPI (const Attribute * x, Attribute * y, const int * comm, const Attribute * mults = NULL) [static]

Wrapper for 2-norm with MPI.

3.1.2.33 void Rocblas::nrm2_scalar (const Attribute * x, void * y, const Attribute * mults = NULL) [static]

Wrapper for 2-norm with y as a scalar pointer.

3.1.2.34 void Rocblas::nrm2_scalar_MPI (const Attribute * x, void * y, const int * comm, const Attribute * mults = NULL) [static]

Wrapper for 2-norm with y as a scalar pointer with MPI.

3.1.2.35 template<class OPint, class OPdbl, int OPMPI> void Rocblas::reduce_MPI (const Attribute * x, Attribute * z, int * comm, int, double) [inline, static, protected]

3.1.2.36 template<class OPint, class OPdbl, int OPMPI> void Rocblas::reduce_scalar_MPI (const Attribute * x, void * y, int * comm, int, double) [inline, static, protected]

3.1.2.37 void Rocblas::sub (const Attribute * x, const Attribute * y, Attribute * z) [static]

Operation wrapper for subtraction.

3.1.2.38 void Rocblas::sub_scalar (const Attribute ∗ *x*, const void ∗ *y*, Attribute ∗ *z*, int *swap* = 0) `[static]`

Operation wrapper for subtraction with y as a scalar pointer.

3.1.2.39 void Rocblas::sum_MPI (const Attribute ∗ *x*, Attribute ∗ *y*, int ∗ *comm*) `[static]`

Wrapper for sum.

3.1.2.40 void Rocblas::sum_scalar_MPI (const Attribute ∗ *x*, void ∗ *y*, int ∗ *comm*) `[static]`

Operation wrapper for sum (y is a scalar pointer).

3.1.2.41 void Rocblas::swap (Attribute ∗ *x*, Attribute ∗ *y*) `[static]`

Wrapper for swap.

The documentation for this class was generated from the following files:

- Rocblas.h
- Rocblas.C

3.2 Rocblas::assn< T > Struct Template Reference

Public Member Functions

- void operator() (T &x, const T &y)

template<class T> struct Rocblas::assn< T >

3.2.1 Member Function Documentation

3.2.1.1 template<class T> void Rocblas::assn< T >::operator() (T & *x*, const T & *y*) `[inline]`

The documentation for this struct was generated from the following file:

- Rocblas.C

3.3 Rocblas::nega< T > Struct Template Reference

Public Member Functions

- void operator() (T &x, const T &y)

template<class T> struct Rocblas::nega< T >

3.3.1 Member Function Documentation

3.3.1.1 template<class T> void Rocblas::nega< T >::operator() (T & *x*, const T & *y*) `[inline]`

The documentation for this struct was generated from the following file:

- Rocblas.C

3.4 Rocblas::swapp< T > Struct Template Reference

Public Member Functions

- void operator() (T &x, T &y)

template<class T> struct Rocblas::swapp< T >

3.4.1 Member Function Documentation

3.4.1.1 template<class T> void Rocblas::swapp< T >::operator() (T & *x*, T & *y*) `[inline]`

The documentation for this struct was generated from the following file:

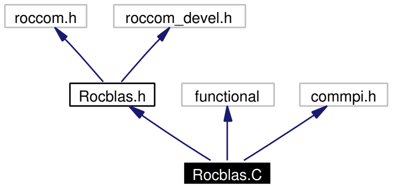
- Rocblas.C

4 Rocblas File Documentation

4.1 Rocblas.C File Reference

```
#include "Rocblas.h"
#include <functional>
#include "commpi.h"
```

Include dependency graph for Rocblas.C:



Compounds

- struct Rocblas::assn
- struct Rocblas::limit1v
- struct Rocblas::maxv
- struct Rocblas::minv
- struct Rocblas::nega
- struct Rocblas::sumv
- struct Rocblas::swapp

Defines

- #define ROCBLAS_LOAD_MODULE COM_F_FUNC(rocblas_load_module)
- #define ROCBLAS_UNLOAD_MODULE COM_F_FUNC(rocblas_unload_module)

Functions

- void Rocblas_load_module (const char ∗name)

  Calls Rocblas initialization function.

- void Rocblas_unload_module (const char ∗name)
- void ROCBLAS_LOAD_MODULE (const char ∗name, int length)
- void ROCBLAS_UNLOAD_MODULE (const char ∗name, int length)

4.1.1 Detailed Description

Implementation of Rocblas.

4.1.2 Define Documentation

4.1.2.1 #define ROCBLAS_LOAD_MODULE COM_F_FUNC(rocblas_load_module)

4.1.2.2 #define ROCBLAS_UNLOAD_MODULE COM_F_FUNC(rocblas_unload_module)

4.1.3 Function Documentation

4.1.3.1 void ROCBLAS_LOAD_MODULE (const char ∗ *name*, int *length*)

4.1.3.2 void Rocblas_load_module (const char ∗ *name*)

Calls Rocblas initialization function.

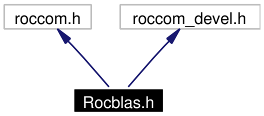
4.1.3.3 void ROCBLAS_UNLOAD_MODULE (const char ∗ *name*, int *length*)

4.1.3.4 void Rocblas_unload_module (const char ∗ *name*)

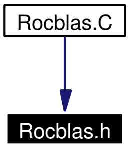
4.2 Rocblas.h File Reference

```
#include "roccom.h"
#include "roccom_devel.h"
```

Include dependency graph for Rocblas.h:



This graph shows which files directly or indirectly include this file:



Compounds

- class Rocblas

Functions

- void Rocblas_load_module (const char ∗name)

  Calls Rocblas initialization function.

- void Rocblas_unload_module (const char ∗name)

4.2.1 Detailed Description

Definition for Rocblas API.

4.2.2 Function Documentation

4.2.2.1 void Rocblas_load_module (const char ∗ *name*)

Calls Rocblas initialization function.

4.2.2.2 void Rocblas_unload_module (const char ∗ *name*)

Index