SISL Reference Manual

0.1.0

Generated by Doxygen 1.4.0

Tue May 30 12:14:43 2006

Contents

Chapter 1

SISL Directory Hierarchy

1.1 SISL Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Chapter 2

SISL File Index

2.1 SISL File List

Here is a list of all documented files with brief descriptions:

Chapter 3

SISL Directory Documentation

3.1 /home/ensmjc/Codes/Codes/SISL/src/ Directory Reference

Files

- file **array-util.c**
- file **array-util.h**
- file **asc2mtx.c**
- file **compare.c**
- file **compare.h**
- file iter_solver.c

  Various iterative solvers for linear systems.

- file iter_solver.h

  Declarations for SISL iterative solvers.

- file **iter_solver_c.c**
- file matrix.c

  Public functions for matrix manipulation and arithmetic.

- file **matrix.h**
- file **matrix_arith.c**
- file **matrix_complex.c**
- file **matrix_complex.h**
- file **matrix_private.c**
- file **matrix_private.h**
- file **mpi_wrapper.c**
- file **mpi_wrapper.h**
- file sisl-logging.c

  Logging functions for use with GLIB logging facilities.

- file **sisl-logging.h**
- file **sisl-test.c**
- file **sisl.h**
- file **sislconfig.h**

- file **vector.c**
- file **vector.h**
- file **vector_arith.c**
- file **vector_complex.c**
- file **vector_complex.h**
- file **vector_inner.c**
- file **vector_inner.h**
- file **vector_private.c**
- file **vector_private.h**

Chapter 4

SISL File Documentation

4.1 iter_solver.c File Reference

Various iterative solvers for linear systems.

Functions

- sisl_solver_workspace_t ∗ sisl_solver_workspace_new (guint n, sisl_complex_t rc, sisl_dist_t d)
- gint sisl_solve (sisl_solver_t solver, sisl_matrix_t ∗A, sisl_vector_t ∗x, sisl_vector_t ∗b, gdouble tol, guint niter, sisl_solver_workspace_t ∗w, sisl_solver_performance_t ∗perf)

4.1.1 Detailed Description

Various iterative solvers for linear systems.

Author:
Michael Carley

Date:
Fri Mar 17 11:49:21 2006

Functions implementing (some of) the iterative solvers in Barrett, R. et al, 'Templates for the solution of linear systems: Building blocks for iterative methods', SIAM, 1994.

4.1.2 Function Documentation

4.1.2.1 gint sisl_solve (sisl_solver_t *solver*, sisl_matrix_t ∗ *A*, sisl_vector_t ∗ *x*, sisl_vector_t ∗ *b*, gdouble *tol*, guint *niter*, sisl_solver_workspace_t ∗ *w*, sisl_solver_performance_t ∗ *perf*)

Iterative solution of a linear system

Parameters:

solver iterative solver to use (sisl_solver_t)

A left hand side matrix

x solution

b right hand side

tol tolerance for solution

niter maximum number of iterations

w workspace allocated with sisl_solver_workspace_new

perf solution performance data (convergence tolerance, etc.)

Returns:

0 on success

4.1.2.2 sisl_solver_workspace_t∗ sisl_solver_workspace_new (guint *n*, sisl_complex_t *rc*, sisl_dist_t *d*)

Allocate a workspace for iterative solution of linear systems

Parameters:

n maximum size of problem

rc SISL_REAL or SISL_COMPLEX

d SISL_SINGLE or SISL_MULTI (single or multi-processor)

Returns:

pointer to newly allocated workspace

4.2 iter_solver.h File Reference

Declarations for SISL iterative solvers.

Enumerations

- enum sisl_solver_t { , SISL_ITER_CG , SISL_ITER_BICG , SISL_ITER_CGS, SISL_ITER_BICGSTAB }

Functions

- sisl_solver_workspace_t ∗ sisl_solver_workspace_new (guint n, sisl_complex_t rc, sisl_dist_t d)
- gint sisl_solve (sisl_solver_t solver, sisl_matrix_t ∗A, sisl_vector_t ∗x, sisl_vector_t ∗b, gdouble tol, guint niter, sisl_solver_workspace_t ∗w, sisl_solver_performance_t ∗perf)

4.2.1 Detailed Description

Declarations for SISL iterative solvers.

Author:
Michael Carley

Date:
Fri Mar 17 11:44:05 2006

4.2.2 Enumeration Type Documentation

4.2.2.1 enum sisl_solver_t

Enumeration values:

SISL_ITER_CG Conjugate gradient

SISL_ITER_BICG Biconjugate gradient

SISL_ITER_CGS Conjugate gradient squared

SISL_ITER_BICGSTAB Stabilized biconjugate gradient

4.2.3 Function Documentation

4.2.3.1 gint sisl_solve (sisl_solver_t *solver*, sisl_matrix_t ∗ *A*, sisl_vector_t ∗ *x*, sisl_vector_t ∗ *b*, gdouble *tol*, guint *niter*, sisl_solver_workspace_t ∗ *w*, sisl_solver_performance_t ∗ *perf*)

Iterative solution of a linear system

Parameters:

solver iterative solver to use (sisl_solver_t)

A left hand side matrix

x solution

b right hand side

tol tolerance for solution

niter maximum number of iterations

w workspace allocated with sisl_solver_workspace_new

perf solution performance data (convergence tolerance, etc.)

Returns:

0 on success

4.2.3.2 sisl_solver_workspace_t* sisl_solver_workspace_new (guint *n*, sisl_complex_t *rc*, sisl_dist_t *d*)

Allocate a workspace for iterative solution of linear systems

Parameters:

n maximum size of problem

rc SISL_REAL or SISL_COMPLEX

d SISL_SINGLE or SISL_MULTI (single or multi-processor)

Returns:

pointer to newly allocated workspace

4.3 matrix.c File Reference

Public functions for matrix manipulation and arithmetic.

Functions

- sisl_matrix_t ∗ sisl_mat_new (guint nrow, guint ncol, sisl_mat_layout_t layout, sisl_vector_density_t density, sisl_complex_t rc, sisl_dist_t dist)
- gint sisl_mat_clear (sisl_matrix_t ∗m)
- gint sisl_mat_set_size (sisl_matrix_t ∗m, guint rows, guint cols)
- gint sisl_mat_write (sisl_matrix_t ∗m, FILE ∗f)
- gint sisl_mat_write_sparse (sisl_matrix_t ∗m, FILE ∗f)
- gint sisl_mat_add_element (sisl_matrix_t ∗m, guint i, guint j, gdouble x)
- gint sisl_mat_addto_element (sisl_matrix_t ∗m, guint i, guint j, gdouble x)
- gint sisl_mat_vector_multiply (sisl_matrix_t ∗m, sisl_vector_t ∗v, sisl_vector_t ∗w)
- gint sisl_mat_trans_vector_multiply (sisl_matrix_t ∗m, sisl_vector_t ∗v, sisl_vector_t ∗w)
- gint sisl_mat_set_element (sisl_matrix_t ∗m, guint i, guint j, gdouble x)
- gint sisl_mat_size (sisl_matrix_t ∗m, guint ∗rows, guint ∗cols)
- gdouble sisl_mat_get_element (sisl_matrix_t ∗m, guint i, guint j)
- gint sisl_mat_compact (sisl_matrix_t ∗m)
- gint sisl_mat_set_distribution (sisl_matrix_t ∗m, sisl_dist_t dist)
- gboolean sisl_mat_has_row (sisl_matrix_t ∗m, guint i)
- sisl_dist_t sisl_mat_distribution (sisl_matrix_t ∗m)
- sisl_vector_t ∗ sisl_mat_get_row (sisl_matrix_t ∗m, guint i)
- gint sisl_mat_set_all (sisl_matrix_t ∗m, gdouble x)
- gint sisl_mat_split_chunks (sisl_matrix_t ∗m)
- gchar ∗ sisl_mat_file_header_string (gint rows, gint cols, gchar fmt)

4.3.1 Detailed Description

Public functions for matrix manipulation and arithmetic.

Author:
Michael Carley

Date:
Tue May 30 12:12:07 2006

Various functions for handling matrices and doing matrix arithmetic. This includes real and complex matrices and those on distributed systems using MPI. All internals are hidden from the user who can switch between serial and parallel systems at will and real and complex problems almost at will.

4.3.2 Function Documentation

4.3.2.1 gint sisl_mat_add_element (sisl_matrix_t ∗ *m*, guint *i*, guint *j*, gdouble *x*)

Add an element to a matrix. For sparse matrices, this inserts an extra element; for dense matrices, the behaviour is the same as sisl_mat_set_element. For complex matrices, this function sets the real part.

Parameters:

m matrix;

i row index of element;

j column index of element;

x value to set.

Returns:

0 on success.

4.3.2.2 gint sisl_mat_addto_element (sisl_matrix_t $*$ *m*, guint *i*, guint *j*, gdouble *x*)

Add a value to a matrix entry $A_{ij} = A_{ij} + x$.

Parameters:

m matrix;

i row index of entry;

j column index of entry;

x value to add to entry.

Returns:

0 on success.

4.3.2.3 gint sisl_mat_clear (sisl_matrix_t $*$ *m*)

Clear a matrix

Parameters:

m matrix to be cleared.

Returns:

0 on success.

4.3.2.4 gint sisl_mat_compact (sisl_matrix_t $*$ *m*)

Compact a matrix, removing zero entries from sparse rows. This function has no effect on dense rows.

Parameters:

m matrix to compact.

Returns:

0 on success.

4.3.2.5 sisl_dist_t sisl_mat_distribution (sisl_matrix_t $*$ *m*)

Check distribution of matrix.

Parameters:

m matrix.

Returns:

SISL_SINGLE or SISL_MULTI.

4.3.2.6 gchar$*$ sisl_mat_file_header_string (gint *rows*, gint *cols*, gchar *fmt*)

Generate a header string for output file format, to be used in writing from non-SISL programs. The string format is:

[MTXFILE][block length][version number][A|B][matrix size]

Parameters:

- ***rows*** number of rows in matrix;
- ***cols*** number of columns in matrix;
- ***fmt*** format 'A' for ASCII, 'B' for binary.

Returns:

pointer to header string.

4.3.2.7 gdouble sisl_mat_get_element (sisl_matrix_t $*$ *m*, guint *i*, guint *j*)

Extract the value of a matrix element A_{ij}.

Parameters:

- ***m*** matrix;
- ***i*** row index;
- ***j*** column index.

Returns:

value A_{ij}.

4.3.2.8 sisl_vector_t$*$ sisl_mat_get_row (sisl_matrix_t $*$ *m*, guint *i*)

Extract a pointer to a row of a matrix.

Parameters:

- ***m*** matrix;
- ***i*** row index.

Returns:

vector containing row i of matrix.

4.3.2.9 gboolean sisl_mat_has_row (sisl_matrix_t $*$ *m*, guint *i*)

Check if matrix has row i, checking row indices explicitly for sparse matrices.

Parameters:

- ***m*** matrix;
- ***i*** row index

Returns:

TRUE if matrix has row i, FALSE otherwise.

4.3.2.10 sisl_matrix_t$*$ sisl_mat_new (guint *nrow*, guint *ncol*, sisl_mat_layout_t *layout*, sisl_vector_density_t *density*, sisl_complex_t *rc*, sisl_dist_t *dist*)

Allocate space for a new matrix. Note that the maximum number of rows or columns refers to the number of entries allocated and not to the physical size of the matrix (to allow for sparse matrices and for distributed matrices on parallel systems).

Parameters:

nrow maximum number of rows;

ncol maximum number of columns;

layout sparse or dense layout of rows;

density (SISL_SPARSE, SISL_DENSE_ROWS, SISL_DENSE_BLOCK);

rc real or complex;

dist single- or multi-processor.

Returns:

the new matrix.

4.3.2.11 gint sisl_mat_set_all (sisl_matrix_t $*$ *m*, gdouble *x*)

Set all entries of a matrix to a given value. For complex matrices, this sets the real part.

Parameters:

m matrix;

x value to set.

Returns:

0 on success.

4.3.2.12 gint sisl_mat_set_distribution (sisl_matrix_t $*$ *m*, sisl_dist_t *dist*)

Set the distribution of a matrix.

Parameters:

m matrix;

dist distribution (SISL_SINGLE for single processor; SISL_MULTI for distributed matrix).

Returns:

0 on success.

4.3.2.13 gint sisl_mat_set_element (sisl_matrix_t $*$ *m*, guint *i*, guint *j*, gdouble *x*)

Set an element of a matrix $A_{ij} = x$. For complex matrices, this sets the real part.

Parameters:

m matrix;

i row index;

j column index;

x value.

Returns:
0 on success.

4.3.2.14 gint sisl_mat_set_size (sisl_matrix_t ∗ *m*, guint *rows*, guint *cols*)

Set the size of a matrix. Note that the size is the real size of the matrix and not that part of it on a given processor.

Parameters:

m matrix;

rows number of rows;

cols numbers of columns.

Returns:
0 on success.

4.3.2.15 gint sisl_mat_size (sisl_matrix_t ∗ *m*, guint ∗ *rows*, guint ∗ *cols*)

Get the (real) size of a matrix.

Parameters:

m matrix;

rows number of rows in whole matrix;

cols number of columns in whole matrix;

Returns:
0 on success.

4.3.2.16 gint sisl_mat_split_chunks (sisl_matrix_t ∗ *m*)

Split a matrix across multiple processors in blocks of rows.

Parameters:

m matrix to split.

Returns:
0 on success.

4.3.2.17 gint sisl_mat_trans_vector_multiply (sisl_matrix_t $*$ *m*, sisl_vector_t $*$ *v*, sisl_vector_t $*$ *w*)

Transpose matrix-vector multiply $w = A^T v$. This function works for real and complex matrices and vectors and for those distributed across multiple processors.

Parameters:

m matrix;

v vector to multiply;

w vector for result.

Returns:

0 on success.

4.3.2.18 gint sisl_mat_vector_multiply (sisl_matrix_t $*$ *m*, sisl_vector_t $*$ *v*, sisl_vector_t $*$ *w*)

Matrix-vector multiply $w = Av$. This function works for real and complex matrices and vectors and for those distributed across multiple processors.

Parameters:

m matrix;

v vector to multiply;

w vector for result.

Returns:

0 on success.

4.3.2.19 gint sisl_mat_write (sisl_matrix_t $*$ *m*, FILE $*$ *f*)

Write a matrix to file in dense matrix format.

Parameters:

m matrix;

f file pointer.

Returns:

0 on success.

4.3.2.20 gint sisl_mat_write_sparse (sisl_matrix_t $*$ *m*, FILE $*$ *f*)

Write a matrix to file in sparse matrix format.

Parameters:

m matrix;

f file pointer.

Returns:

0 on success.

4.4 sisl-logging.c File Reference

Logging functions for use with GLIB logging facilities.

Functions

- gint sisl_logging_init (FILE ∗f, gchar ∗p, GLogLevelFlags log_level, gpointer exit_func)

4.4.1 Detailed Description

Logging functions for use with GLIB logging facilities.

Author:

Michael Carley

Date:

Fri Mar 17 09:33:52 2006

4.4.2 Function Documentation

4.4.2.1 gint sisl_logging_init (FILE ∗ *f*, gchar ∗ *p*, GLogLevelFlags *log_level*, gpointer *exit_func*)

Initialize SISL logging

Parameters:

f file stream for messages

p string to prepend to messages

log_level maximum logging level to handle (see gts_log)

exit_func function to call if exiting on an error

Returns:

0 on success

Index