ADI for Reaction-Diffusion Systems

Generated by Doxygen 1.8.6

Wed May 6 2015 14:49:04

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Module Documentation

4.1 Tridiagonal matrices.

4.1.1 Detailed Description

These matrices don't depend on the time step or the diffusing quantities and are thus constant.

4.2 Return the diagonals.

Functions

- std::vector< double > TriDiagMatrix::getL () const
- std::vector< double > TriDiagMatrix::getM () const
- std::vector< double > TriDiagMatrix::getU () const

4.2.1 Detailed Description

4.2.2 Function Documentation

4.2.2.1 std::vector<double> TriDiagMatrix::getL ( ) const [inline]

Return the lower diagonal

4.2.2.2 std::vector<double> TriDiagMatrix::getM ( ) const [inline]

Return the middle diagonal

4.2.2.3 std::vector<double> TriDiagMatrix::getU ( ) const [inline]

Return the upper diagonal

4.3 Diagonals of the tridiagonal matrix.

4.3.1 Detailed Description

The diagonals all have the same length (n_), but have 0 values where they are outside of the matrix. The indices determine the row.

| m0 u0 0 | | l1 m1 u1 | | 0 l2 m2 |

i.e.: l_[0] and u_[2] are 0.

Chapter 5

Namespace Documentation

5.1 PeriodicTriDiagMatrixSolver Namespace Reference

Functions

- void solve (int n, const TriDiagMatrix &mat, const std::vector< double > &rhs, double *result, unsigned int inc)

5.1.1 Detailed Description

Solver for a tridiagonal matrix system.

5.1.2 Function Documentation

5.1.2.1 void PeriodicTriDiagMatrixSolver::solve (int *n*, const TriDiagMatrix & *mat*, const std::vector< double > & *rhs*, double * *result*, unsigned int *inc*)

Solve a tridiagonal matrix system.

Parameters

n	number of elements in the result
mat	tridiagonal matrix
rhs	right-hand side of the system
result	vector for the result, pointer to the first element
inc	increment for the elements of the result

5.2 TriDiagMatrixSolver Namespace Reference

Functions

- void solve (int n, const TriDiagMatrix &mat, const std::vector< double > &rhs, double *result, unsigned int inc)

5.2.1 Detailed Description

Solver for a tridiagonal matrix system.

5.2.2 Function Documentation

5.2.2.1 void TriDiagMatrixSolver::solve (int *n*, const TriDiagMatrix & *mat*, const std::vector< double > & *rhs*, double * *result*, unsigned int *inc*)

Solve a tridiagonal matrix system using the Thomas algorithm.

Parameters

n	number of elements in the result
mat	tridiagonal matrix
rhs	right-hand side of the system
result	vector for the result, pointer to the first element
inc	increment for the elements of the result

Chapter 6

Class Documentation

6.1 Coin Struct Reference

Public Attributes

- uivector **symbols**
- float **weight**

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

- lodepng.cpp

6.2 ColorProfile Struct Reference

Public Attributes

- unsigned char **sixteenbit**
- unsigned char **sixteenbit_done**
- unsigned char **colored**
- unsigned char **colored_done**
- unsigned char **key**
- unsigned short **key_r**
- unsigned short **key_g**
- unsigned short **key_b**
- unsigned char **alpha**
- unsigned char **alpha_done**
- unsigned **numcolors**
- ColorTree **tree**
- unsigned char * **palette**
- unsigned **maxnumcolors**
- unsigned char **numcolors_done**
- unsigned **greybits**
- unsigned char **greybits_done**

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

- lodepng.cpp

6.3 ColorTree Struct Reference

Public Attributes

- ColorTree * **children** [16]
- int **index**

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

- lodepng.cpp

6.4 GrayScott Class Reference

Public Member Functions

- GrayScott (int N, double L, double dt, double Du, double Dv, double F, double k, int nSteps, std::string pngname)
- ~GrayScott ()
- void run ()
- void step ()
- int size () const
- std::vector< double > getU () const
- int getCurrStep () const
- double getDt () const
- double getTime () const

6.4.1 Constructor & Destructor Documentation

6.4.1.1 GrayScott::GrayScott (int *N*, double *L*, double *dt*, double *Du*, double *Dv*, double *F*, double *k*, int *nSteps*, std::string *pngname*)

Construct a new simulation object.

Parameters

N	number of grid cells in one dimension
L	length of the domain in one dimension
dt	time step
Du	diffusion coefficient for u
Dv	diffusion coefficient for v
F	model parameter
k	model parameter
nSteps	number of steps in the simulation

6.4.1.2 GrayScott::~GrayScott ( )

Destructor

6.4.2 Member Function Documentation

6.4.2.1 int GrayScott::getCurrStep ( ) const [inline]

Get the current step in the simulation.

Returns

the current step.

6.4.2.2 double GrayScott::getDt ( ) const [inline]

Get the time step for the simulation.

Returns

the timestep

6.4.2.3 double GrayScott::getTime ( ) const [inline]

Get the time that has passed.

Returns

time

6.4.2.4 std::vector<double> GrayScott::getU ( ) const [inline]

Get the field U of the simulation.

Returns

the field U

6.4.2.5 void GrayScott::run ( )

Run the simulation.

6.4.2.6 int GrayScott::size ( ) const [inline]

Return the size of the system in one dimension.

Returns

number of grid cells in one dimension.

6.4.2.7 void GrayScott::step ( )

Perform one simulation step.

Public because of the visualization

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

- grayscott.hpp
- grayscott.cpp

6.5 GSViewer Class Reference

`#include <gsviewer.hpp>`

Public Member Functions

- GSViewer (int argc, char ∗argv[])
- void visualize (GrayScott ∗simulation)

6.5.1 Detailed Description

Viewer for the Gray-Scott Reaction Diffusion simulation.

This class implements a viewer for the Gray-Scott Reaction Diffusion simulation. It visualizes the field U as the simulation is running using OpenGL.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 GSViewer::GSViewer (int *argc*, char ∗ *argv[]*)

Construct a Gray-Scott viewer.

The constructor initializes glut.

Parameters

argc	argument from the main function
argv	argument from the main function

6.5.3 Member Function Documentation

6.5.3.1 void GSViewer::visualize (GrayScott ∗ *simulation*)

Run the visualization.

The visualization function will run the the simulation and visualize the field U. It initializes the window and will run the glutMainLoop.

Parameters

simulation	the simulation that has to be run and visualized

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

- gsviewer.hpp
- gsviewer.cpp

6.6 Hash Struct Reference

Public Attributes

- int ∗ **head**
- unsigned short ∗ **chain**
- int ∗ **val**
- int ∗ **headz**

- unsigned short * **chainz**
- unsigned short * **zeros**

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

- lodepng.cpp

6.7 HuffmanTree Struct Reference

Public Attributes

- unsigned * **tree2d**
- unsigned * **tree1d**
- unsigned * **lengths**
- unsigned **maxbitlen**
- unsigned **numcodes**

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

- lodepng.cpp

6.8 LodePNGColorMode Struct Reference

Public Attributes

- LodePNGColorType **colortype**
- unsigned **bitdepth**
- unsigned char * **palette**
- size_t **palettesize**
- unsigned **key_defined**
- unsigned **key_r**
- unsigned **key_g**
- unsigned **key_b**

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

- lodepng.h

6.9 LodePNGCompressSettings Struct Reference

Public Attributes

- unsigned **btype**
- unsigned **use_lz77**
- unsigned **windowsize**
- unsigned **minmatch**
- unsigned **nicematch**
- unsigned **lazymatching**
- unsigned(* **custom_zlib**)(unsigned char **, size_t *, const unsigned char *, size_t, const LodePNGCompressSettings *)

- unsigned(∗ **custom_deflate**)(unsigned char ∗∗, size_t ∗, const unsigned char ∗, size_t, const LodePNGCompressSettings ∗)
- const void ∗ **custom_context**

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

- lodepng.h

6.10 LodePNGDecoderSettings Struct Reference

Public Attributes

- LodePNGDecompressSettings **zlibsettings**
- unsigned **ignore_crc**
- unsigned **fix_png**
- unsigned **color_convert**
- unsigned **read_text_chunks**
- unsigned **remember_unknown_chunks**

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

- lodepng.h

6.11 LodePNGDecompressSettings Struct Reference

Public Attributes

- unsigned **ignore_adler32**
- unsigned(∗ **custom_zlib**)(unsigned char ∗∗, size_t ∗, const unsigned char ∗, size_t, const LodePNGDecompressSettings ∗)
- unsigned(∗ **custom_inflate**)(unsigned char ∗∗, size_t ∗, const unsigned char ∗, size_t, const LodePNGDecompressSettings ∗)
- const void ∗ **custom_context**

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

- lodepng.h

6.12 LodePNGEncoderSettings Struct Reference

Public Attributes

- LodePNGCompressSettings **zlibsettings**
- LodePNGAutoConvert **auto_convert**
- unsigned **filter_palette_zero**
- LodePNGFilterStrategy **filter_strategy**
- const unsigned char ∗ **predefined_filters**
- unsigned **force_palette**
- unsigned **add_id**
- unsigned **text_compression**

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

- lodepng.h

6.13 LodePNGInfo Struct Reference

Public Attributes

- unsigned **compression_method**
- unsigned **filter_method**
- unsigned **interlace_method**
- LodePNGColorMode **color**
- unsigned **background_defined**
- unsigned **background_r**
- unsigned **background_g**
- unsigned **background_b**
- size_t **text_num**
- char ** **text_keys**
- char ** **text_strings**
- size_t **itext_num**
- char ** **itext_keys**
- char ** **itext_langtags**
- char ** **itext_transkeys**
- char ** **itext_strings**
- unsigned **time_defined**
- LodePNGTime **time**
- unsigned **phys_defined**
- unsigned **phys_x**
- unsigned **phys_y**
- unsigned **phys_unit**
- unsigned char * **unknown_chunks_data** [3]
- size_t **unknown_chunks_size** [3]

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

- lodepng.h

6.14 LodePNGState Struct Reference

Public Attributes

- LodePNGDecoderSettings **decoder**
- LodePNGEncoderSettings **encoder**
- LodePNGColorMode **info_raw**
- LodePNGInfo **info_png**
- unsigned **error**

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

- lodepng.h

6.15 LodePNGTime Struct Reference

Public Attributes

- unsigned **year**
- unsigned **month**
- unsigned **day**
- unsigned **hour**
- unsigned **minute**
- unsigned **second**

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

- lodepng.h

6.16 TriDiagMatrix Class Reference

Public Member Functions

- TriDiagMatrix ()
- TriDiagMatrix (int N, double l, double m, double u)
- int size () const
- std::vector< double > getL () const
- std::vector< double > getM () const
- std::vector< double > getU () const

Friends

- std::ostream & operator<< (std::ostream &os, const TriDiagMatrix &matrix)

6.16.1 Constructor & Destructor Documentation

6.16.1.1 TriDiagMatrix::TriDiagMatrix ( )

Default constructor

6.16.1.2 TriDiagMatrix::TriDiagMatrix (int *N*, double *l*, double *m*, double *u*)

Construct an object of the type TriDiagMatrix

Parameters

N	size of the matrix
l	value on the lower diagonal
m	value on the middle diagonal
u	value on the upper diagonal

6.16.2 Member Function Documentation

6.16.2.1 int TriDiagMatrix::size ( ) const [inline]

Return the size of the matrix in one dimension.

Returns

the matrix size in one dimension

6.16.3 Friends And Related Function Documentation

6.16.3.1 std::ostream& operator<< (std::ostream & *os,* const TriDiagMatrix & *matrix*) [friend]

Print the matrix to the stream os.

Parameters

os	stream to print the matrix to
matrix	matrix

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

- tridiagmatrix.hpp
- tridiagmatrix.cpp

6.17 ucvector Struct Reference

Public Attributes

- unsigned char * **data**
- size_t **size**
- size_t **allocsize**

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

- lodepng.cpp

6.18 uivector Struct Reference

Public Attributes

- unsigned * **data**
- size_t **size**
- size_t **allocsize**

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

- lodepng.cpp

Index