

bnlib-3d-kolmogorov

1.2

Generated by Doxygen 1.5.5

Tue Jun 3 12:23:08 2008

Contents

1	Main Page	1
2	Namespace Index	3
2.1	Namespace List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	BNLib Namespace Reference	7
5	File Documentation	9
5.1	kolmogorov_3d.hxx File Reference	9

Chapter 1

Main Page

This is the documentation for the part of [BNLib](#) that can be used to generate three dimensional fields with statistics consistent with Kolmogorov hypothesis. This document is an extract from the complete documentation of [BNLib](#). It is automatically generated from the source code using Doxygen.

The parts of the code which are designed end-user accessibility are in the [BNLib](#) namespace. The details of the implementation are in the `BNLib::Kolmog` namespace. Not all details are documented – please see the source code for the definitive story.

The main entry point is the function [BNLib::Kolmogorov3D](#). You should start with the documentation of this function.

Chapter 2

Namespace Index

2.1 Namespace List

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

BNLib (General utilities namespace)	7
---	---

Chapter 3

File Index

3.1 File List

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

kolmogorov_3d.hxx	9
---	---

Chapter 4

Namespace Documentation

4.1 BNLib Namespace Reference

General utilities namespace.

Enumerations

- enum `Kolmogorov3DOptions` {
 `KInitialEFB` = 0, `KInitialFBB` = 1, `KWeightedInterp` = 2, `KBalancedIters` = 4,
 `KEdgeBalanced` = 8 }

Functions

- `size_t Kolmogorov3D` (double *cube, `size_t` N, `RDist` &rfn)
- `size_t Kolmogorov3D` (double *cube, `size_t` Nx, `size_t` Ny, `size_t` Nz, `RDist` &rfn, `Kolmogorov3DOptions` opt=`KInitialEFB`) throw (const char *)
- `size_t Kolmogorov3DF` (float *cube, `size_t` Nx, `size_t` Ny, `size_t` Nz, `RDist` &rfn)

4.1.1 Detailed Description

General utilities namespace.

4.1.2 Enumeration Type Documentation

4.1.2.1 enum BNLib::Kolmogorov3DOptions

Options for the 3d turbulence generation routines. Has the structure of a bit field.

Enumerator:

- KInitialEFB*** Generate the initial corners using the exact algorithm considering edge, face and body diagonals
- KInitialFBB*** Generate the initial corners using approximate algorithm (not recommended)
- KWeightedInterp*** If this bit is set, weigh interpolation by inverse distance to parent point

KBalancedIters If this bit is set, only interpolate parents which have a valid symmetric parent. This applies to face interpolation

KEdgeBalanced As *KBalancedIters*, but for edge interpolation.

4.1.3 Function Documentation

4.1.3.1 `size_t BNLib::Kolmogorov3D (double * cube, size_t Nx, size_t Ny, size_t Nz, RDist & rfn, Kolmogorov3DOptions opt = KInitialEFB) throw (const char *)`

Generated 3d Kolmogorov turbulence on a *Nx* by *Ny* by *Nz* grid. This is the main entry point to the three dimensional Kolmogorov field generation.

Parameters:

cube is the pre-allocated output array of size *Nx* times *Ny* times *Nz*.

Nx,Ny,Nz lengths in the *Nx,Ny,Nz* directions. Direction *Nx* must not be smaller than then two other directions.

rfn the gaussian random number generator to be used by the routine

opt Options for adjusting the generation algorithm. See [BNLib::Kolmogorov3DOptions](#) .

Returns:

The maximum order used + 1

4.1.3.2 `size_t BNLib::Kolmogorov3D (double * cube, size_t N, RDist & rfn)`

Generate 3d komogorov turbulence on *N*cubed grid

4.1.3.3 `size_t BNLib::Kolmogorov3DF (float * cube, size_t Nx, size_t Ny, size_t Nz, RDist & rfn)`

As *Kolmogorov3D*, but generate on single precision ("float") grid.

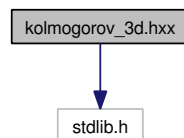
Chapter 5

File Documentation

5.1 kolmogorov_3d.hxx File Reference

```
#include <stdlib.h>
```

Include dependency graph for kolmogorov_3d.hxx:



Namespaces

- namespace `BNLib`
- namespace `BNLib::Kolmog`

Enumerations

- enum `BNLib::Kolmogorov3DOptions` {
 `BNLib::KInitialEFB` = 0, `BNLib::KInitialFBB` = 1, `BNLib::KWeightedInterp` = 2, `BNLib::KBalancedIters` = 4,
 `BNLib::KEdgeBalanced` = 8 }

Functions

- `size_t BNLib::Kolmogorov3D` (double *cube, size_t N, RDist &rfn)
- `size_t BNLib::Kolmogorov3D` (double *cube, size_t Nx, size_t Ny, size_t Nz, RDist &rfn, Kolmogorov3DOptions opt=KInitialEFB) throw (const char *)
- `size_t BNLib::Kolmogorov3DF` (float *cube, size_t Nx, size_t Ny, size_t Nz, RDist &rfn)
- void `BNLib::Kolmog::KolmogorovCorners3D` (double *cube, size_t N, RDist &rfn)
- double `BNLib::Kolmog::KMidPointVar_CI` (size_t np, size_t o)
- double `BNLib::Kolmog::KMidPointVar_FI` (size_t np, size_t o)

- double `BNLib::Kolmog::KMidPointVar_EI` (size_t np, size_t o)
- void `BNLib::Kolmog::KMagnifyGrid` (double *og, const size_t N, double *dd, const size_t Nx, const size_t Ny, const size_t Nz)
- bool `BNLib::Kolmog::pTwoNPlustOne` (unsigned n)

5.1.1 Detailed Description

Bojan Nikolic <bn204@mrao.cam.ac.uk>, <bojan@bnikolic.co.uk>

June 2007

The 3-d generalisation of kolmogorov turbulence generation.

Index

- BNLib, [7](#)
 - KBalancedIters, [7](#)
 - KEdgeBalanced, [8](#)
 - KInitialEFB, [7](#)
 - KInitialFBB, [7](#)
 - Kolmogorov3DOptions, [7](#)
 - Kolmogorov3D, [8](#)
 - Kolmogorov3DF, [8](#)
 - KWeightedInterp, [7](#)

- KBalancedIters
 - BNLib, [7](#)
- KEdgeBalanced
 - BNLib, [8](#)
- KInitialEFB
 - BNLib, [7](#)
- KInitialFBB
 - BNLib, [7](#)
- Kolmogorov3DOptions
 - BNLib, [7](#)
- Kolmogorov3D
 - BNLib, [8](#)
- Kolmogorov3DF
 - BNLib, [8](#)
- kolmogorov_3d.hxx, [9](#)
- KWeightedInterp
 - BNLib, [7](#)