LinePicking

Generated by Doxygen 1.8.2

Tue Aug 28 2012 00:22:20

Contents

Chapter 1

Todo List

Global CubeDistanceCDF (double a, double ∗b)

Write up how to derive the CDF.

Global CubeDistanceVar (double ∗parameters)

Try to derive a value algebaricaly

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

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

Chapter 4

Data Structure Documentation

4.1 LinePickingRec Struct Reference

```
#include <LinePicking.h>
```

Data Fields

- double(∗ **PDF**)(double, double ∗)
- double(∗ **CDF**)(double, double ∗)
- double(∗ **MEAN**)(double ∗)
- double(∗ **VAR**)(double ∗)
- void(∗ **SUPPORT**)(double ∗, double ∗)
- void(∗ **CHECK_PAR**)(double ∗, int ∗, char ∗)
- int ∗ **Npar**
- char ∗∗ **name**
- char ∗∗ **description**

4.1.1 Detailed Description

structure thingo test

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

- LinePicking.h

Chapter 5

File Documentation

5.1 beta.h File Reference

Implements the beta function and the regularized incomplete beta function.

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

Functions

- double beta (double, double)
- double beta_inc (double, double, double)

5.1.1 Detailed Description

Implements the beta function and the regularized incomplete beta function.

Author

Matthew Roughan matthew.roughan@adelaide.edu.au

Date

22/09/2012

5.1.2 Function Documentation

5.1.2.1 double beta (double *x*, double *y*)

Implements the beta function. i.e., the Euler integral of the first kind, defined by $\mathrm{B}(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\,dt$.

Parameters

$x	in $\mathrm{B}(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\,dt$.
$y	in $\mathrm{B}(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\,dt$.

Returns

The result of evaluating Eulers integral of the first kind with the given parameters.

5.1.2.2 double beta_inc (double *a*, double *b*, double *x*)

Implements the regularized incomplete beta function. Defined as $I_x(a,b) = \frac{B(x;\,a,b)}{B(a,b)}$. Where $\mathrm{B}(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt$ and $B(x;\,a,b) = \int_0^x t^{a-1}\,(1-t)^{b-1}\,dt$.

See Also

http://doi.acm.org/10.1145/131766.131776
http://www.boost.org/doc/libs/1_38_0/libs/math/doc/sf_and_dist/html/math_toolkit/special/sf_beta/ibeta_function.html
http://dlmf.nist.gov/8.17

Parameters

$a	in $I_x(a,b) = \frac{B(x;\,a,b)}{B(a,b)}$.
$b	in $I_x(a,b) = \frac{B(x;\,a,b)}{B(a,b)}$.
$x	in $I_x(a,b) = \frac{B(x;\,a,b)}{B(a,b)}$.

Returns

The result of evaluating the incomplete beta function with the given parameters.

5.2 Cube.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points within a cube.

Functions

- double CubeDistancePDF (double t, double ∗parameters)
- double CubeDistanceCDF (double a, double ∗b)
- double CubeDistanceMean (double ∗parameters)
- double CubeDistanceVar (double ∗parameters)
- void CubeDistanceSupport (double ∗t, double ∗parameters)
- void CubeDistanceCheckParameters (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **CubeDistanceName**
- char ∗ **CubeDistanceDescription**
- int **CubeDistanceNpar**

5.2.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points within a cube.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.2.2 Function Documentation

5.2.2.1 double CubeDistanceCDF (double *t*, double * *parameters*)

Implements the CDF of the distance between two random points within a cube.

Derived by Eric Parsonage eric.parsonage@adelaide.edu.au

Parameters

$t	The distance calcuate the culmultive density for.
$parameters[0]	The size of the cube (i.e., the length of any side).

Todo Write up how to derive the CDF.

Returns

The culmultive density at $t.

5.2.2.2 void CubeDistanceCheckParameters (double * *parameters*, int * *result*, char * *error_str*)

Intended to determine if the parameters supplied are valid input to the other functions implemented in this file. However as there is only one parameter and the calling function checks that it is positive this is merely a place holder to allow for a complete implementation in geometries that have more complex relationships between parameters.

Parameters

$parameters[0]	Contains the length of the side of the cube under consideration.
$result	Pointer to storage for an integer indicating any errors in the supplied parameters.
$error_str	Pointer to storage for a message explaining any errors in the supplied parameters.

Returns

Any error conditions are indicated by placing a value other than 0 in the location pointed to by $result and a message explaining the error is copied in to the location pointed to $error_str

5.2.2.3 double CubeDistanceMean (double * *parameters*)

Implements the mean of the distance between two random points within a cube.

From Mathai, A. M.; Moschopoulos, P.; and Pederzoli, G. "Distance between Random Points in a Cube." J. Statistica 59, 61-81, 1999. but with 'corrected typos'

Parameters

$parameters[0]	The size of the cube (i.e., the length of any side).

Returns

The mean distance between two points in a unit cube

See Also

http://mathworld.wolfram.com/CubeLinePicking.html

5.2.2.4 double CubeDistancePDF (double *t*, double ∗ *parameters*)

Implements the PDF of the distance between two random points within a cube.

From Mathai, A. M.; Moschopoulos, P.; and Pederzoli, G. "Distance between Random Points in a Cube." J. Statistica 59, 61-81, 1999. but with 'corrected typos'

Parameters

$t	The distance to calcuate the density for.
$parameters[0]	The size of the cube (i.e., the length of any side).

Returns

The density at $t.

See Also

http://mathworld.wolfram.com/CubeLinePicking.html

5.2.2.5 void CubeDistanceSupport (double ∗ *t*, double ∗ *parameters*)

Calculates the support for the PDF and CDF of the distance between two random points within a cube.

Parameters

$t	Pointer to storage for lower and upper ends of the support for the PDF and CDF of the distance between two random points within a cube.
$parameters[0]	The size of the cube (i.e., the length of any side).

Returns

The lower end of the interval is returned in $t[0] and the upper end of the interval is returned in $t[1].

5.2.2.6 double CubeDistanceVar (double ∗ *parameters*)

Implements the variance of the distance between two random points within a cube.

This was calculated numerically.

Parameters

$parameters[0]	The size of the cube (i.e., the length of any side).

Todo Try to derive a value algebaricaly

Returns

The variance of the distances between two points in a unit cube

5.3 Disk.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a disk.

Functions

- double **DiskDistancePDF** (double t, double ∗parameters)
- double **DiskDistanceCDF** (double a, double ∗b)
- double **DiskDistanceMean** (double ∗parameters)
- double **DiskDistanceVar** (double ∗parameters)
- void **DiskDistanceSupport** (double ∗t, double ∗parameters)
- void **DiskDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **DiskDistanceName**
- char ∗ **DiskDistanceDescription**
- int **DiskDistanceNpar**

5.3.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a disk.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.4 Hyperball.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points within a hyper-ball.

Functions

- double **HyperballDistancePDF** (double t, double ∗parameters)
- double **HyperballDistanceCDF** (double a, double ∗b)
- double **HyperballDistanceMean** (double ∗parameters)
- double **HyperballDistanceVar** (double ∗parameters)
- void **HyperballDistanceSupport** (double ∗t, double ∗parameters)
- void **HyperballDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **HyperballDistanceName**
- char ∗ **HyperballDistanceDescription**
- int **HyperballDistanceNpar**

5.4.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points within a hyper-ball.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.5 Line.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a line.

Functions

- double **LineDistancePDF** (double t, double ∗parameters)
- double **LineDistanceCDF** (double a, double ∗b)
- double **LineDistanceMean** (double ∗parameters)
- double **LineDistanceVar** (double ∗parameters)
- void **LineDistanceSupport** (double ∗t, double ∗parameters)
- void **LineDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **LineDistanceName**
- char ∗ **LineDistanceDescription**
- int **LineDistanceNpar**

5.5.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a line.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.6 LinePicking.h File Reference

Exposes to matlab and R a set of functions that implement PDF, CDF, mean and variance of the distance between two random points in various geometries.

```
#include <math.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <stdio.h>
#include "Square.h"
#include "Disk.h"
#include "Hyperball.h"
#include "Rectangle.h"
#include "Line.h"
#include "Cube.h"
#include "Sphere.h"
#include "SphereGeodesic.h"
#include "PrismGeodesic.h"
```

Data Structures

- struct LinePickingRec

Macros

- #define **_LINEPICKING_H**
- #define **ExpandFields**(_x)
- #define **elements**(x) (sizeof(x) / sizeof(x[0]))
- #define **NUMBER_OF_MODES** elements(LinePickingFields)

Functions

- void **LinePickingModeLookup** (int ∗, char ∗∗, char ∗∗)
- void **LinePickingAllmodes** ()
- void LinePickingCheckParameters (int ∗, double ∗, int ∗, int ∗, char ∗∗)
- void **LinePickingSupport** (double ∗, int ∗, double ∗, int ∗, int ∗, char ∗∗)
- void **LinePickingPDF** (double ∗, double ∗, int ∗, int ∗, double ∗, int ∗, int ∗, char ∗∗)
- void **LinePickingCDF** (double ∗, double ∗, int ∗, int ∗, double ∗, int ∗, int ∗, char ∗∗)
- void **LinePickingMean** (double ∗, int ∗, double ∗, int ∗, int ∗, char ∗∗)
- void **LinePickingVar** (double ∗, int ∗, double ∗, int ∗, int ∗, char ∗∗)

Variables

- LinePickingRec **LinePickingFields** []

5.6.1 Detailed Description

Exposes to matlab and R a set of functions that implement PDF, CDF, mean and variance of the distance between two random points in various geometries.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au
Matthew Roughan matthew.roughan@adelaide.edu.au

Date

22/09/2012

5.6.2 Macro Definition Documentation

5.6.2.1 #define ExpandFields(_x)

Value:

```
&_x##DistancePDF,&_x##DistanceCDF,\
&_x##DistanceMean,&_x##DistanceVar,\
&_x##DistanceSupport,&_x##DistanceCheckParameters,\
&_x##DistanceNpar,&_x##DistanceName,\
&_x##DistanceDescription
```

5.6.3 Function Documentation

5.6.3.1 void LinePickingCheckParameters (int ∗ *mode*, double ∗ *parameters*, int ∗ *Npar*, int ∗ *result*, char ∗∗ *error_str*)

Summary here; one sentence on one line (should not, but can exceed 80 chars).

A more detailed description goes here.

A blank line forms a paragraph. There should be no trailing white-space anywhere.

Parameters

$first	"@param" is a Doxygen directive to describe a function parameter. Like some other directives, it takes a term/summary on the same line and a description (this text) indented by 2 spaces on the next line. All descriptive text should wrap at 80 chars, without going over. Newlines are NOT supported within directives; if a newline would be before this text, it would be appended to the general description above.
$second	There should be no newline between multiple directives (of the same type).
$third	(optional) TRUE if Third should be done. Defaults to FALSE. Only optional parameters are explicitly stated as such. The description should clarify the default value if omitted.

Returns

"@return" is a different Doxygen directive to describe the return value of a function, if there is any.

5.6.4 Variable Documentation

5.6.4.1 LinePickingRec LinePickingFields[ ]

Initial value:

```
=
{
    {ExpandFields(Square)},
    {ExpandFields(Disk)},
    {ExpandFields(Hyperball)},
    {ExpandFields(Rectangle)},
    {ExpandFields(Line)},
    {ExpandFields(Cube)},
    {ExpandFields(Sphere)},
    {ExpandFields(SphereGeodesic)},
    {ExpandFields(PrismGeodesic)}
}
```

5.7 PrismGeodesic.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on the surface (not including the ends) of an upright prism of any cross section. The distance is measured around the surface i.e., it is a geodesic.

Functions

- double PrismGeodesicDistancePDF (double t, double ∗parameters)
- double **PrismGeodesicDistanceCDF** (double t, double ∗parameters)
- double **PrismGeodesicDistanceMean** (double ∗parameters)
- double **PrismGeodesicDistanceVar** (double ∗parameters)
- void **PrismGeodesicDistanceSupport** (double ∗t, double ∗parameters)
- void **PrismGeodesicDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **PrismGeodesicDistanceName**
- char ∗ **PrismGeodesicDistanceDescription**
- int **PrismGeodesicDistanceNpar**

5.7.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on the surface (not including the ends) of an upright prism of any cross section. The distance is measured around the surface i.e., it is a geodesic.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.7.2 Function Documentation

5.7.2.1 double PrismGeodesicDistancePDF (double *w,* double ∗ *parameters*)

distance density (at w) between two points on the surface of an upright prism of length and perimeter given in parameters[0] and parameters[1] respectively. The distance is measured around the the surface of the prism (i.e., a geodesic) TODO Derived by Eric Parsonage eric.parsonage@adelaide.edu.au soon to be written up somewhere

5.8 Rectangle.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a rectangle.

Functions

- double **RectangleDistancePDF** (double t, double ∗parameters)
- double **RectangleDistanceCDF** (double t, double ∗parameters)
- double **RectangleDistanceMean** (double ∗parameters)
- double **RectangleDistanceVar** (double ∗parameters)
- void **RectangleDistanceSupport** (double ∗t, double ∗parameters)
- void **RectangleDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **RectangleDistanceName**
- char ∗ **RectangleDistanceDescription**
- int **RectangleDistanceNpar**

5.8.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a rectangle.

Author

Eric Parsonage `eric.parsonage@adelaide.edu.au`

Date

22/09/2012

5.9 Sphere.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on the surface of a sphere.

Functions

- double **SphereDistancePDF** (double t, double ∗parameters)
- double **SphereDistanceCDF** (double a, double ∗b)
- double **SphereDistanceMean** (double ∗parameters)
- double **SphereDistanceVar** (double ∗parameters)
- void **SphereDistanceSupport** (double ∗t, double ∗parameters)
- void **SphereDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **SphereDistanceName**
- char ∗ **SphereDistanceDescription**
- int **SphereDistanceNpar**

5.9.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on the surface of a sphere.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.10 SphereGeodesic.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on the surface of a sphere. The distance is measured around the surface of the sphere i.e., it is a geodesic.

Functions

- double **SphereGeodesicDistancePDF** (double t, double ∗parameters)
- double **SphereGeodesicDistanceCDF** (double a, double ∗b)
- double **SphereGeodesicDistanceMean** (double ∗parameters)
- double **SphereGeodesicDistanceVar** (double ∗parameters)
- void **SphereGeodesicDistanceSupport** (double ∗t, double ∗parameters)
- void **SphereGeodesicDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **SphereGeodesicDistanceName**
- char ∗ **SphereGeodesicDistanceDescription**
- int **SphereGeodesicDistanceNpar**

5.10.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on the surface of a sphere. The distance is measured around the surface of the sphere i.e., it is a geodesic.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

5.11 Square.h File Reference

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a square.

Functions

- double **SquareDistancePDF** (double t, double ∗parameters)
- double **SquareDistanceCDF** (double a, double ∗b)
- double **SquareDistanceMean** (double ∗parameters)
- double **SquareDistanceVar** (double ∗parameters)
- void **SquareDistanceSupport** (double ∗t, double ∗parameters)
- void **SquareDistanceCheckParameters** (double ∗parameters, int ∗result, char ∗error_str)

Variables

- char ∗ **SquareDistanceName**
- char ∗ **SquareDistanceDescription**
- int **SquareDistanceNpar**

5.11.1 Detailed Description

Functions to provide PDF, CDF, mean and variance of the distance between two random points on a square.

Author

Eric Parsonage eric.parsonage@adelaide.edu.au

Date

22/09/2012

Index