Reference Manual

Generated by Doxygen 1.5.8

Sun May 17 10:14:41 2009

Contents

Chapter 1

Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

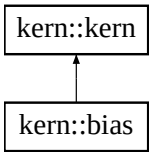
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 kern::bias Class Reference

Inheritance diagram for kern::bias::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def **extractParamNames**
- def gradX
- def gradient

Public Attributes

- **type**
- **variance**
- **nParams**
- **stationary**

3.1.1 Detailed Description

```
% The white noise kernel arises from assuming independent Gaussian
% noise for each point in the function. The variance of the noise is
% given by the kern.variance parameter.
%
% This kernel is not intended to be used independently, it is provided
% so that it may be combined with other kernels in a compound kernel.
```

3.1.2 Member Function Documentation

3.1.2.1 def kern::bias::compute (*self*, *x*, *x2* = None)

```
% BIASKERNCOMPUTE Compute the BIAS kernel given the parameters and X.
% FORMAT
% DESC computes the kernel parameters for the bias noise
% kernel given inputs associated with rows and columns.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : the input matrix associated with the rows of the kernel.
% ARG x2 : the inpute matrix associated with the columns of the kernel.
% RETURN k : the kernel matrix computed at the given points.
%
% FORMAT
% DESC computes the kernel matrix for the bias noise
% kernel given a design matrix of inputs.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : the kernel matrix computed at the given points.
%
% SEEALSO : biasKernParamInit, kernCompute, create, biasKernDiagCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.1.2.2 def kern::bias::diagCompute (*self*, *x*)

```
% BIASKERNDIAGCOMPUTE Compute diagonal of BIAS kernel.
% FORMAT
% DESC computes the diagonal of the kernel matrix for the bias noise kernel given a design matrix of input
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : a vector containing the diagonal of the kernel matrix
% computed at the given points.
%
% SEEALSO : biasKernParamInit, kernDiagCompute, create, biasKernCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.1.2.3 def kern::bias::diagGradient (*self*, *X*, *covDiag*)

```
% BIASKERNDIAGGRADIENT Compute the gradient of the BIAS kernel's diagonal wrt parameters.
% FORMAT
% DESC computes the gradient of functions of the diagonal of the
% bias noise kernel matrix with respect to the parameters of the kernel. The
% parameters' gradients are returned in the order given by the
```

```
% biasKernExtractParam command.
% ARG kern : the kernel structure for which the gradients are
% computed.
% ARG x : the input data for which the gradient is being computed.
% ARG factors : partial derivatives of the function of interest with
% respect to the diagonal elements of the kernel.
% RETURN g : gradients of the relevant function with respect to each
% of the parameters. Ordering should match the ordering given in
% biasKernExtractParam.
%
% SEEALSO : biasKernParamInit, kernDiagGradient, biasKernExtractParam, biasKernGradient
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.1.2.4 def kern::bias::diagGradX (*self*, *X*)

```
% BIASKERNDIAGGRADX Gradient of BIAS kernel's diagonal with respect to X.
% FORMAT
% DESC computes the gradient of the diagonal of the bias noise kernel matrix with
% respect to the elements of the design matrix given in X.
% ARG kern : the kernel structure for which gradients are being computed.
% ARG X : the input data in the form of a design matrix.
% RETURN gX : the gradients of the diagonal with respect to each element
% of X. The returned matrix has the same dimensions as X.
%
% SEEALSO : biasKernParamInit, kernDiagGradX, biaskernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.1.2.5 def kern::bias::display (*self*, *numSpaces* = 0)

```
% BIASKERNDISPLAY Display parameters of the BIAS kernel.
% FORMAT
% DESC displays the parameters of the bias noise
% kernel and the kernel type to the console.
% ARG kern : the kernel to display.
%
% FORMAT does the same as above, but indents the display according
% to the amount specified.
% ARG kern : the kernel to display.
% ARG spacing : how many spaces to indent the display of the kernel by.
%
% SEEALSO : biasKernParamInit, modelDisplay, kernDisplay
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.1.2.6 def kern::bias::expandParam (*self*, *params*)

```
% BIASKERNEXPANDPARAM Create kernel structure from BIAS kernel's parameters.
% FORMAT
% DESC returns a bias noise kernel structure filled with the
```

```
% parameters in the given vector. This is used as a helper function to
% enable parameters to be optimised in, for example, the NETLAB
% optimisation functions.
% ARG kern : the kernel structure in which the parameters are to be
% placed.
% ARG param : vector of parameters which are to be placed in the
% kernel structure.
% RETURN kern : kernel structure with the given parameters in the
% relevant locations.
%
% SEEALSO : biasKernParamInit, biasKernExtractParam, kernExpandParam
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

3.1.2.7 def kern::bias::extractParam (*self*)

```
% BIASKERNEXTRACTPARAM Extract parameters from the BIAS kernel structure.
% FORMAT
% DESC Extract parameters from the bias noise kernel structure into a
% vector of parameters for optimisation.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the kernel. If
% the field 'transforms' is not empty in the kernel matrix, the
% parameters will be transformed before optimisation (for example
% positive only parameters could be logged before being returned).
%
% FORMAT
% DESC Extract parameters and parameter names from the bias noise
% kernel structure.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the kernel. If
% the field 'transforms' is not empty in the kernel matrix, the
% parameters will be transformed before optimisation (for example
% positive only parameters could be logged before being returned).
% RETURN names : cell array of strings giving paramter names.
%
% SEEALSO biasKernParamInit, biasKernExpandParam, kernExtractParam, scg, conjgrad
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

3.1.2.8 def kern::bias::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
% BIASKERNGRADIENT Gradient of BIAS kernel's parameters.
% FORMAT
% DESC computes the gradient of functions with respect to the
% bias noise
% kernel's parameters. As well as the kernel structure and the
% input positions, the user provides a matrix PARTIAL which gives
% the partial derivatives of the function with respect to the
% relevant elements of the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x : the input locations for which the gradients are being
% computed.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The argument takes
% the form of a square matrix of dimension  numData, where numData is
% the number of rows in X.
```

```
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters. The ordering of the vector should match
% that provided by the function kernExtractParam.
%
% FORMAT
% DESC computes the derivatives as above, but input locations are
% now provided in two matrices associated with rows and columns of
% the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x1 : the input locations associated with the rows of the
% kernel matrix.
% ARG x2 : the input locations associated with the columns of the
% kernel matrix.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The matrix should
% have the same number of rows as X1 and the same number of columns
% as X2 has rows.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters.
%
% SEEALSO biasKernParamInit, kernGradient, biasKernDiagGradient, kernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.1.2.9 def kern::bias::gradX (*self*, *X*, *X2* = None)

```
% BIASKERNGRADX Gradient of BIAS kernel with respect to input locations.
% FORMAT
% DESC computes the gradident of the bias noise
% kernel with respect to the input positions where both the row
% positions and column positions are provided separately.
% ARG kern : kernel structure for which gradients are being
% computed.
% ARG x1 : row locations against which gradients are being computed.
% ARG x2 : column locations against which gradients are being computed.
% RETURN g : the returned gradients. The gradients are returned in
% a matrix which is numData2 x numInputs x numData1. Where numData1 is
% the number of data points in X1, numData2 is the number of data
% points in X2 and numInputs is the number of input
% dimensions in X.
%
% SEEALSO biasKernParamInit, kernGradX, biasKernDiagGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

Reimplemented from kern::kern.

3.1.2.10 def kern::bias::paramInit (*self*, *inDim* = None, *X* = None)

```
% BIASKERNPARAMINIT BIAS kernel parameter initialisation.
%
% SEEALSO : cmpndKernParamInit
%
% FORMAT
% DESC initialises the bias noise
%  kernel structure with some default parameters.
% ARG kern : the kernel structure which requires initialisation.
% RETURN kern : the kernel structure with the default parameters placed in.
```

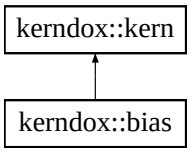
```
%
% SEEALSO : create, kernParamInit
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

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

- kern.py

3.2 kerndox::bias Class Reference

Inheritance diagram for kerndox::bias::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def **extractParamNames**
- def gradX
- def gradient

Public Attributes

- **type**
- **variance**
- **nParams**
- **stationary**

3.2.1 Detailed Description

```
The white noise kernel arises from assuming independent Gaussian
noise for each point in the function. The variance of the noise is
given by the kern.variance parameter.

This kernel is not intended to be used independently, it is provided
so that it may be combined with other kernels in a compound kernel.
```

3.2.2 Member Function Documentation

3.2.2.1 def kerndox::bias::compute (*self*, *x*, *x2* = None)

```
BIASKERNCOMPUTE Compute the BIAS kernel given the parameters and X.
FORMAT
DESC computes the kernel parameters for the bias noise
kernel given inputs associated with rows and columns.
\param kern : the kernel structure for which the matrix is computed.
\param x : the input matrix associated with the rows of the kernel.
```

```
\param x2 : the inpute matrix associated with the columns of the kernel.
\return  k : the kernel matrix computed at the given points.
%
FORMAT
DESC computes the kernel matrix for the bias noise
kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : the kernel matrix computed at the given points.
%
SEEALSO : biasKernParamInit, kernCompute, create, biasKernDiagCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.2.2.2 def kerndox::bias::diagCompute (*self*, *x*)

```
BIASKERNDIAGCOMPUTE Compute diagonal of BIAS kernel.
FORMAT
DESC computes the diagonal of the kernel matrix for the bias noise kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : a vector containing the diagonal of the kernel matrix
computed at the given points.
%
SEEALSO : biasKernParamInit, kernDiagCompute, create, biasKernCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.2.2.3 def kerndox::bias::diagGradient (*self*, *X*, *covDiag*)

```
BIASKERNDIAGGRADIENT Compute the gradient of the BIAS kernel's diagonal wrt parameters.
FORMAT
DESC computes the gradient of functions of the diagonal of the
bias noise kernel matrix with respect to the parameters of the kernel. The
parameters' gradients are returned in the order given by the
biasKernExtractParam command.
\param kern : the kernel structure for which the gradients are
computed.
\param x : the input data for which the gradient is being computed.
\param factors : partial derivatives of the function of interest with
respect to the diagonal elements of the kernel.
\return  g : gradients of the relevant function with respect to each
of the parameters. Ordering should match the ordering given in
biasKernExtractParam.
%
SEEALSO : biasKernParamInit, kernDiagGradient, biasKernExtractParam, biasKernGradient
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.2.2.4 def kerndox::bias::diagGradX (*self*, *X*)

```
BIASKERNDIAGGRADX Gradient of BIAS kernel's diagonal with respect to X.
```

```
FORMAT
DESC computes the gradient of the diagonal of the bias noise kernel matrix with
respect to the elements of the design matrix given in X.
\param kern : the kernel structure for which gradients are being computed.
\param X : the input data in the form of a design matrix.
\return  gX : the gradients of the diagonal with respect to each element
of X. The returned matrix has the same dimensions as X.
%
SEEALSO : biasKernParamInit, kernDiagGradX, biaskernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.2.2.5 def kerndox::bias::display (*self*, *numSpaces* = 0)

```
BIASKERNDISPLAY Display parameters of the BIAS kernel.
FORMAT
DESC displays the parameters of the bias noise
kernel and the kernel type to the console.
\param kern : the kernel to display.
%
FORMAT does the same as above, but indents the display according
to the amount specified.
\param kern : the kernel to display.
\param spacing : how many spaces to indent the display of the kernel by.
%
SEEALSO : biasKernParamInit, modelDisplay, kernDisplay
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.2.2.6 def kerndox::bias::expandParam (*self*, *params*)

```
BIASKERNEXPANDPARAM Create kernel structure from BIAS kernel's parameters.
FORMAT
DESC returns a bias noise kernel structure filled with the
parameters in the given vector. This is used as a helper function to
enable parameters to be optimised in, for example, the NETLAB
optimisation functions.
\param kern : the kernel structure in which the parameters are to be
placed.
\param param : vector of parameters which are to be placed in the
kernel structure.
\return  kern : kernel structure with the given parameters in the
relevant locations.
%
SEEALSO : biasKernParamInit, biasKernExtractParam, kernExpandParam
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

3.2.2.7 def kerndox::bias::extractParam (*self*)

```
BIASKERNEXTRACTPARAM Extract parameters from the BIAS kernel structure.
FORMAT
DESC Extract parameters from the bias noise kernel structure into a
```

```
vector of parameters for optimisation.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the kernel. If
the field 'transforms' is not empty in the kernel matrix, the
parameters will be transformed before optimisation (for example
positive only parameters could be logged before being returned).
%
FORMAT
DESC Extract parameters and parameter names from the bias noise
kernel structure.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the kernel. If
the field 'transforms' is not empty in the kernel matrix, the
parameters will be transformed before optimisation (for example
positive only parameters could be logged before being returned).
\return  names : cell array of strings giving paramter names.
%
SEEALSO biasKernParamInit, biasKernExpandParam, kernExtractParam, scg, conjgrad
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

3.2.2.8 def kerndox::bias::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
BIASKERNGRADIENT Gradient of BIAS kernel's parameters.
FORMAT
DESC computes the gradient of functions with respect to the
bias noise
kernel's parameters. As well as the kernel structure and the
input positions, the user provides a matrix PARTIAL which gives
the partial derivatives of the function with respect to the
relevant elements of the kernel matrix.

\param kern : the kernel structure for which the gradients are being
computed.

\param x : the input locations for which the gradients are being
computed.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The argument takes
the form of a square matrix of dimension  numData, where numData is
the number of rows in X.

\return  g : gradients of the function of interest with respect to
the kernel parameters. The ordering of the vector should match
that provided by the function kernExtractParam.
%
FORMAT
DESC computes the derivatives as above, but input locations are
now provided in two matrices associated with rows and columns of
the kernel matrix.

\param kern : the kernel structure for which the gradients are being
computed.

\param x1 : the input locations associated with the rows of the
kernel matrix.

\param x2 : the input locations associated with the columns of the
kernel matrix.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The matrix should
have the same number of rows as X1 and the same number of columns
```

```
as X2 has rows.

\return  g : gradients of the function of interest with respect to
the kernel parameters.
%
SEEALSO biasKernParamInit, kernGradient, biasKernDiagGradient, kernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.2.2.9 def kerndox::bias::gradX (*self*, *X*, *X2* = None)

```
BIASKERNGRADX Gradient of BIAS kernel with respect to input locations.
FORMAT
DESC computes the gradident of the bias noise
kernel with respect to the input positions where both the row
positions and column positions are provided separately.
\param kern : kernel structure for which gradients are being
computed.
\param x1 : row locations against which gradients are being computed.
\param x2 : column locations against which gradients are being computed.
\return  g : the returned gradients. The gradients are returned in
a matrix which is numData2 x numInputs x numData1. Where numData1 is
the number of data points in X1, numData2 is the number of data
points in X2 and numInputs is the number of input
dimensions in X.
%
SEEALSO biasKernParamInit, kernGradX, biasKernDiagGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

Reimplemented from kerndox::kern.

3.2.2.10 def kerndox::bias::paramInit (*self*, *inDim* = None, *X* = None)

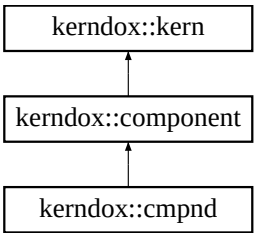
```
BIASKERNPARAMINIT BIAS kernel parameter initialisation.
%
SEEALSO : cmpndKernParamInit
%
FORMAT
DESC initialises the bias noise
 kernel structure with some default parameters.
\param kern : the kernel structure which requires initialisation.
\return  kern : the kernel structure with the default parameters placed in.
%
SEEALSO : create, kernParamInit
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

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

- kerndox.py

3.3 kerndox::cmpnd Class Reference

Inheritance diagram for kerndox::cmpnd::



Public Member Functions

- def **__init__**
- def **paramInit**
- def **compute**
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def gradX
- def gradient
- def expandParam
- def extractParam
- def extractParamNames

Public Attributes

- **type**
- **stationary**
- **whiteVariance**

3.3.1 Detailed Description

```
The is short for compound kernel and it is developed
by summing several kernels together.
```

3.3.2 Member Function Documentation

3.3.2.1 def kerndox::cmpnd::diagCompute (*self*, *x*)

```
DIAGCOMPUTE Compute diagonal of CMPND kernel.
FORMAT
DESC computes the diagonal of the kernel matrix for the
compound kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : a vector containing the diagonal of the kernel matrix
computed at the given points.
```

```
%
SEEALSO : cmpndKernParamInit, kernDiagCompute, create, cmpndKernCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.3.2.2 def kerndox::cmpnd::diagGradient (*self*, *x*, *covDiag*)

```
DIAGGRADIENT Compute the gradient of the CMPND kernel's diagonal wrt parameters.
FORMAT
DESC computes the gradient of functions of the diagonal of the
compound kernel matrix with respect to the parameters of the kernel. The
parameters' gradients are returned in the order given by the
cmpndKernExtractParam command.
\param kern : the kernel structure for which the gradients are
computed.
\param x : the input data for which the gradient is being computed.
\param factors : partial derivatives of the function of interest with
respect to the diagonal elements of the kernel.
\return  g : gradients of the relevant function with respect to each
of the parameters. Ordering should match the ordering given in
cmpndKernExtractParam.
%
SEEALSO : cmpndKernParamInit, kernDiagGradient, cmpndKernExtractParam, cmpndKernGradient
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.3.2.3 def kerndox::cmpnd::diagGradX (*self*, *X*)

```
DIAGGRADX Gradient of CMPND kernel's diagonal with respect to X.
FORMAT
DESC computes the gradient of the diagonal of the compound kernel matrix with
respect to the elements of the design matrix given in X.
\param kern : the kernel structure for which gradients are being computed.
\param X : the input data in the form of a design matrix.
\return  gX : the gradients of the diagonal with respect to each element
of X. The returned matrix has the same dimensions as X.
%
SEEALSO : cmpndKernParamInit, kernDiagGradX, cmpndkernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.3.2.4 def kerndox::cmpnd::display (*self*, *numSpaces* = 0)

```
DISPLAY Display parameters of the CMPND kernel.
FORMAT
DESC displays the parameters of the compound
kernel and the kernel type to the console.
\param kern : the kernel to display.
%
FORMAT does the same as above, but indents the display according
to the amount specified.
```

```
\param kern : the kernel to display.
\param spacing : how many spaces to indent the display of the kernel by.
%
SEEALSO : cmpndKernParamInit, modelDisplay, kernDisplay
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.3.2.5 def kerndox::cmpnd::expandParam (*self*, *params*)

```
EXPANDPARAM Create kernel structure from CMPND kernel's parameters.
FORMAT
DESC returns a compound kernel structure filled with the
parameters in the given vector. This is used as a helper function to
enable parameters to be optimised in, for example, the NETLAB
optimisation functions.
\param kern : the kernel structure in which the parameters are to be
placed.
\param param : vector of parameters which are to be placed in the
kernel structure.
\return  kern : kernel structure with the given parameters in the
relevant locations.
%
SEEALSO : cmpndKernParamInit, cmpndKernExtractParam, kernExpandParam
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

3.3.2.6 def kerndox::cmpnd::extractParam (*self*)

```
EXTRACTPARAM Extract parameters from the CMPND kernel structure.
FORMAT
DESC Extract parameters from the compound kernel matrix into a vector of
parameters for optimisation.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the
kernel. The vector of 'transforms' is assumed to be empty
here. Any transformations of parameters should be done in
component kernels.
%
SEEALSO cmpndKernParamInit, cmpndKernExpandParam, kernExtractParam, scg, conjgrad
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

3.3.2.7 def kerndox::cmpnd::extractParamNames (*self*)

```
EXTRACTPARAMNAMES Extract parameter names from the CMPND kernel structure.
FORMAT
DESC Extract parameters from the compound kernel matrix into a vector of
parameters for optimisation.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  names : a list of parameter names.
%
SEEALSO cmpndKernParamInit, cmpndKernExpandParam, kernExtractParam, scg, conjgrad
%
```

```
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

3.3.2.8 def kerndox::cmpnd::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
GRADIENT Gradient of CMPND kernel's parameters.
FORMAT
DESC computes the gradient of functions with respect to the
compound
kernel's parameters. As well as the kernel structure and the
input positions, the user provides a matrix PARTIAL which gives
the partial derivatives of the function with respect to the
relevant elements of the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x : the input locations for which the gradients are being
computed.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The argument takes
the form of a square matrix of dimension  numData, where numData is
the number of rows in X.
\return  g : gradients of the function of interest with respect to
the kernel parameters. The ordering of the vector should match
that provided by the function kernExtractParam.
%
FORMAT
DESC computes the derivatives as above, but input locations are
now provided in two matrices associated with rows and columns of
the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x1 : the input locations associated with the rows of the
kernel matrix.
\param x2 : the input locations associated with the columns of the
kernel matrix.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The matrix should
have the same number of rows as X1 and the same number of columns
as X2 has rows.
\return  g : gradients of the function of interest with respect to
the kernel parameters.
%
SEEALSO cmpndKernParamInit, kernGradient, cmpndKernDiagGradient, kernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.3.2.9 def kerndox::cmpnd::gradX (*self*, *X*, *X2*)

```
GRADX Gradient of CMPND kernel with respect to a point x.
FORMAT
DESC computes the gradient of the compound
kernel with respect to the input positions.
\param kern : kernel structure for which gradients are being
computed.
\param x : locations against which gradients are being computed.
\return  g : the returned gradients. The gradients are returned in
a matrix which is numData x numInputs x numData. Where numData is
the number of data points and numInputs is the number of input
dimensions in X.
```

```
%
FORMAT
DESC computes the gradident of the compound
kernel with respect to the input positions where both the row
positions and column positions are provided separately.
\param kern : kernel structure for which gradients are being
computed.
\param x1 : row locations against which gradients are being computed.
\param x2 : column locations against which gradients are being computed.
\return  g : the returned gradients. The gradients are returned in
a matrix which is numData2 x numInputs x numData1. Where numData1 is
the number of data points in X1, numData2 is the number of data
points in X2 and numInputs is the number of input
dimensions in X.
%
SEEALSO cmpndKernParamInit, kernGradX, cmpndKernDiagGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

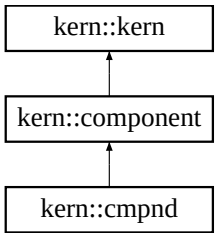
Reimplemented from kerndox::kern.

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

- kerndox.py

3.4 kern::cmpnd Class Reference

Inheritance diagram for kern::cmpnd::



Public Member Functions

- def **__init__**
- def **paramInit**
- def **compute**
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def gradX
- def gradient
- def expandParam
- def extractParam
- def extractParamNames

Public Attributes

- **type**
- **stationary**
- **whiteVariance**

3.4.1 Detailed Description

```
The is short for compound kernel and it is developed
by summing several kernels together.
```

3.4.2 Member Function Documentation

3.4.2.1 def kern::cmpnd::diagCompute (*self*, *x*)

```
% DIAGCOMPUTE Compute diagonal of CMPND kernel.
% FORMAT
% DESC computes the diagonal of the kernel matrix for the
% compound kernel given a design matrix of inputs.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : a vector containing the diagonal of the kernel matrix
% computed at the given points.
```

```
%
% SEEALSO : cmpndKernParamInit, kernDiagCompute, create, cmpndKernCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.4.2.2 def kern::cmpnd::diagGradient (*self*, *x*, *covDiag*)

```
% DIAGGRADIENT Compute the gradient of the CMPND kernel's diagonal wrt parameters.
% FORMAT
% DESC computes the gradient of functions of the diagonal of the
% compound kernel matrix with respect to the parameters of the kernel. The
% parameters' gradients are returned in the order given by the
% cmpndKernExtractParam command.
% ARG kern : the kernel structure for which the gradients are
% computed.
% ARG x : the input data for which the gradient is being computed.
% ARG factors : partial derivatives of the function of interest with
% respect to the diagonal elements of the kernel.
% RETURN g : gradients of the relevant function with respect to each
% of the parameters. Ordering should match the ordering given in
% cmpndKernExtractParam.
%
% SEEALSO : cmpndKernParamInit, kernDiagGradient, cmpndKernExtractParam, cmpndKernGradient
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.4.2.3 def kern::cmpnd::diagGradX (*self*, *X*)

```
% DIAGGRADX Gradient of CMPND kernel's diagonal with respect to X.
% FORMAT
% DESC computes the gradient of the diagonal of the compound kernel matrix with
% respect to the elements of the design matrix given in X.
% ARG kern : the kernel structure for which gradients are being computed.
% ARG X : the input data in the form of a design matrix.
% RETURN gX : the gradients of the diagonal with respect to each element
% of X. The returned matrix has the same dimensions as X.
%
% SEEALSO : cmpndKernParamInit, kernDiagGradX, cmpndkernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.4.2.4 def kern::cmpnd::display (*self*, *numSpaces* = 0)

```
% DISPLAY Display parameters of the CMPND kernel.
% FORMAT
% DESC displays the parameters of the compound
% kernel and the kernel type to the console.
% ARG kern : the kernel to display.
%
% FORMAT does the same as above, but indents the display according
% to the amount specified.
```

```
% ARG kern : the kernel to display.
% ARG spacing : how many spaces to indent the display of the kernel by.
%
% SEEALSO : cmpndKernParamInit, modelDisplay, kernDisplay
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.4.2.5 def kern::cmpnd::expandParam (*self*, *params*)

```
% EXPANDPARAM Create kernel structure from CMPND kernel's parameters.
% FORMAT
% DESC returns a compound kernel structure filled with the
% parameters in the given vector. This is used as a helper function to
% enable parameters to be optimised in, for example, the NETLAB
% optimisation functions.
% ARG kern : the kernel structure in which the parameters are to be
% placed.
% ARG param : vector of parameters which are to be placed in the
% kernel structure.
% RETURN kern : kernel structure with the given parameters in the
% relevant locations.
%
% SEEALSO : cmpndKernParamInit, cmpndKernExtractParam, kernExpandParam
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

3.4.2.6 def kern::cmpnd::extractParam (*self*)

```
% EXTRACTPARAM Extract parameters from the CMPND kernel structure.
% FORMAT
% DESC Extract parameters from the compound kernel matrix into a vector of
% parameters for optimisation.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the
% kernel. The vector of 'transforms' is assumed to be empty
% here. Any transformations of parameters should be done in
% component kernels.
%
% SEEALSO cmpndKernParamInit, cmpndKernExpandParam, kernExtractParam, scg, conjgrad
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

3.4.2.7 def kern::cmpnd::extractParamNames (*self*)

```
% EXTRACTPARAMNAMES Extract parameter names from the CMPND kernel structure.
% FORMAT
% DESC Extract parameters from the compound kernel matrix into a vector of
% parameters for optimisation.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN names : a list of parameter names.
%
% SEEALSO cmpndKernParamInit, cmpndKernExpandParam, kernExtractParam, scg, conjgrad
%
```

```
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

3.4.2.8 def kern::cmpnd::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
% GRADIENT Gradient of CMPND kernel's parameters.
% FORMAT
% DESC computes the gradient of functions with respect to the
% compound
% kernel's parameters. As well as the kernel structure and the
% input positions, the user provides a matrix PARTIAL which gives
% the partial derivatives of the function with respect to the
% relevant elements of the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x : the input locations for which the gradients are being
% computed.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The argument takes
% the form of a square matrix of dimension  numData, where numData is
% the number of rows in X.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters. The ordering of the vector should match
% that provided by the function kernExtractParam.
%
% FORMAT
% DESC computes the derivatives as above, but input locations are
% now provided in two matrices associated with rows and columns of
% the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x1 : the input locations associated with the rows of the
% kernel matrix.
% ARG x2 : the input locations associated with the columns of the
% kernel matrix.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The matrix should
% have the same number of rows as X1 and the same number of columns
% as X2 has rows.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters.
%
% SEEALSO cmpndKernParamInit, kernGradient, cmpndKernDiagGradient, kernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.4.2.9 def kern::cmpnd::gradX (*self*, *X*, *X2*)

```
% GRADX Gradient of CMPND kernel with respect to a point x.
% FORMAT
% DESC computes the gradient of the compound
% kernel with respect to the input positions.
% ARG kern : kernel structure for which gradients are being
% computed.
% ARG x : locations against which gradients are being computed.
% RETURN g : the returned gradients. The gradients are returned in
% a matrix which is numData x numInputs x numData. Where numData is
% the number of data points and numInputs is the number of input
% dimensions in X.
```

```
%
% FORMAT
% DESC computes the gradident of the compound
% kernel with respect to the input positions where both the row
% positions and column positions are provided separately.
% ARG kern : kernel structure for which gradients are being
% computed.
% ARG x1 : row locations against which gradients are being computed.
% ARG x2 : column locations against which gradients are being computed.
% RETURN g : the returned gradients. The gradients are returned in
% a matrix which is numData2 x numInputs x numData1. Where numData1 is
% the number of data points in X1, numData2 is the number of data
% points in X2 and numInputs is the number of input
% dimensions in X.
%
% SEEALSO cmpndKernParamInit, kernGradX, cmpndKernDiagGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

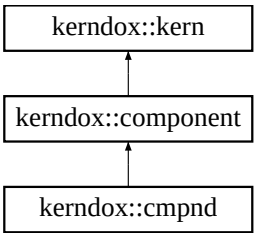
Reimplemented from kern::kern.

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

- kern.py

3.5 kerndox::component Class Reference

Inheritance diagram for kerndox::component::



Public Member Functions

- def **__init__**
- def **addKern**
- def setIndex

Public Attributes

- **paramGroups**
- **comp**
- **index**
- **nParams**
- **stationary**

3.5.1 Detailed Description

```
The base class for kernels which have several components to
them (such as compound or tensor kernels).
```

3.5.2 Member Function Documentation

3.5.2.1 def kerndox::component::setIndex (*self*, *component*, *indices*)

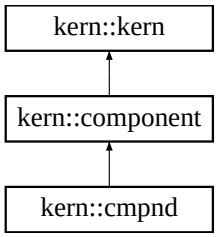
```
SETINDEX Set the indices in the compound kernel.
```

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

- kerndox.py

3.6 kern::component Class Reference

Inheritance diagram for kern::component::



Public Member Functions

- def **__init__**
- def **addKern**
- def setIndex

Public Attributes

- **paramGroups**
- **comp**
- **index**
- **nParams**
- **stationary**

3.6.1 Detailed Description

```
The base class for kernels which have several components to
them (such as compound or tensor kernels).
```

3.6.2 Member Function Documentation

3.6.2.1 def kern::component::setIndex (*self*, *component*, *indices*)

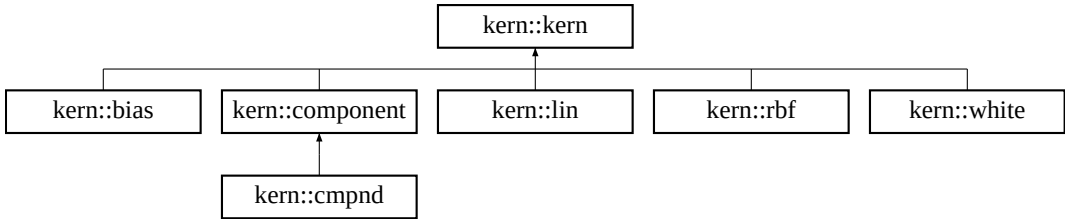
```
SETINDEX Set the indices in the compound kernel.
```

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

- kern.py

3.7 kern::kern Class Reference

Inheritance diagram for kern::kern::



Public Member Functions

- def **__init__**
- def **paramInit**
- def **compute**
- def **diagCompute**
- def **diagGradX**
- def **diagGradient**
- def **display**
- def **gradX**
- def **gradient**
- def gradTransParam

Public Attributes

- **whiteVariance**
- **positiveTime**
- **inputDimension**

3.7.1 Detailed Description

`The base kernel class from which all kernels are derived.`

3.7.2 Member Function Documentation

3.7.2.1 def kern::kern::gradTransParam (*self*, *X*, *X2* = `None`, *covGrad* = `None`)

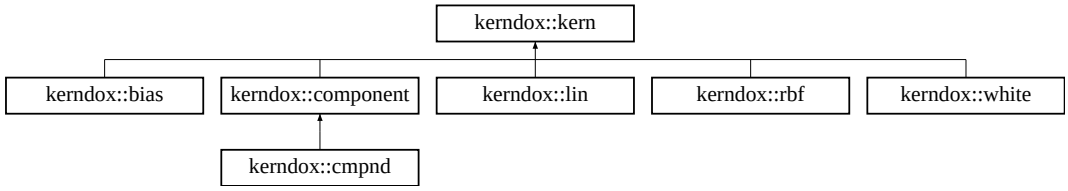
`Return the gradient of the transformed parameters.`

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

- kern.py

3.8 kerndox::kern Class Reference

Inheritance diagram for kerndox::kern::



Public Member Functions

- def **__init__**
- def **paramInit**
- def **compute**
- def **diagCompute**
- def **diagGradX**
- def **diagGradient**
- def **display**
- def **gradX**
- def **gradient**
- def gradTransParam

Public Attributes

- **whiteVariance**
- **positiveTime**
- **inputDimension**

3.8.1 Detailed Description

```
The base kernel class from which all kernels are derived.
```

3.8.2 Member Function Documentation

3.8.2.1 def kerndox::kern::gradTransParam (*self*, *X*, *X2* = `None`, *covGrad* = `None`)

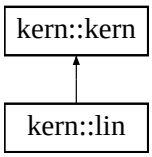
```
Return the gradient of the transformed parameters.
```

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

- kerndox.py

3.9 kern::lin Class Reference

Inheritance diagram for kern::lin::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def extractParamNames
- def gradX
- def gradient

Public Attributes

- **type**
- **variance**
- **nParams**
- **stationary**
- **normalised**

3.9.1 Detailed Description

```
% The linear kernel (LIN) is the simple inner product
% kernel. Sampling from this kernel produces linear functions.
%
% k(x_i, x_j) = sigma2 * x_i'*x_j
%
% There is one parameter, sigma2, which is stored in the field
% kern.variance.
```

3.9.2 Member Function Documentation

3.9.2.1 def kern::lin::compute (*self*, *x*, *x2* = None)

```
% LINKERNCOMPUTE Compute the LIN kernel given the parameters and X.
% FORMAT
```

```
% DESC computes the kernel parameters for the linear
% kernel given inputs associated with rows and columns.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : the input matrix associated with the rows of the kernel.
% ARG x2 : the input matrix associated with the columns of the kernel.
% RETURN k : the kernel matrix computed at the given points.
%
% FORMAT
% DESC computes the kernel matrix for the linear
% kernel given a design matrix of inputs.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : the kernel matrix computed at the given points.
%
% SEEALSO : linKernParamInit, kernCompute, kernCreate, linKernDiagCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.9.2.2 def kern::lin::diagCompute (*self*, *x*)

```
% LINKERNDIAGCOMPUTE Compute diagonal of LIN kernel.
% FORMAT
% DESC computes the diagonal of the kernel matrix for the linear kernel given a design matrix of inputs.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : a vector containing the diagonal of the kernel matrix
% computed at the given points.
%
% SEEALSO : linKernParamInit, kernDiagCompute, kernCreate, linKernCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.9.2.3 def kern::lin::diagGradient (*self*, *X*, *covDiag*)

```
% DIAGGRADIENT Compute the gradient of the RBF kernel's diagonal wrt parameters.
% FORMAT
% DESC computes the gradient of functions of the diagonal of the
% radial basis function kernel matrix with respect to the parameters of the kernel. The
% parameters' gradients are returned in the order given by the
% rbfKernExtractParam command.
% ARG kern : the kernel structure for which the gradients are
% computed.
% ARG x : the input data for which the gradient is being computed.
% ARG factors : partial derivatives of the function of interest with
% respect to the diagonal elements of the kernel.
% RETURN g : gradients of the relevant function with respect to each
% of the parameters. Ordering should match the ordering given in
% rbfKernExtractParam.
%
% SEEALSO : gradient
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kern::kern.

3.9.2.4 def kern::lin::diagGradX (*self*, *X*)

```
% LINKERNDIAGGRADX Gradient of LIN kernel's diagonal with respect to X.
% FORMAT
% DESC computes the gradient of the diagonal of the linear kernel matrix with
% respect to the elements of the design matrix given in X.
% ARG kern : the kernel structure for which gradients are being computed.
% ARG X : the input data in the form of a design matrix.
% RETURN gX : the gradients of the diagonal with respect to each element
% of X. The returned matrix has the same dimensions as X.
%
% SEEALSO : linKernParamInit, kernDiagGradX, linkernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.9.2.5 def kern::lin::display (*self*, *numSpaces* = 0)

```
% DISPLAY Display parameters of the LIN kernel.
% FORMAT
% DESC displays the parameters of the radial basis function
% kernel and the kernel type to the console.
% ARG kern : the kernel to display.
%
% FORMAT does the same as above, but indents the display according
% to the amount specified.
% ARG kern : the kernel to display.
% ARG spacing : how many spaces to indent the display of the kernel by.
%
% SEEALSO :
%
% COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

Reimplemented from kern::kern.

3.9.2.6 def kern::lin::expandParam (*self*, *params*)

```
% EXPANDPARAM Create kernel structure from LIN kernel's parameters.
% FORMAT
% DESC returns a linear kernel structure filled with the
% parameters in the given vector. This is used as a helper function to
% enable parameters to be optimised in, for example, the NETLAB
% optimisation functions.
% ARG kern : the kernel structure in which the parameters are to be
% placed.
% ARG param : vector of parameters which are to be placed in the
% kernel structure.
% RETURN kern : kernel structure with the given parameters in the
% relevant locations.
%
% SEEALSO : extractParam
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

3.9.2.7 def kern::lin::extractParam (*self*)

```
% EXTRACTPARAM Extract parameters from the LIN kernel structure.
```

```
% FORMAT
% DESC Extract parameters from the linear kernel
% structure into a vector of parameters for optimisation.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the kernel. If
% the field 'transforms' is not empty in the kernel matrix, the
% parameters will be transformed before optimisation (for example
% positive only parameters could be logged before being returned).
%
% SEEALSO expandParam, netlab.scg, netlab.conjgrad
%
% COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

3.9.2.8 def kern::lin::extractParamNames (*self*)

```
% EXTRACTPARAMNAMES Extract parameter names from the LIN kernel structure.
% FORMAT
% DESC Extract parameter names from the linear kernel structure.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN names : cell array of strings giving names to the parameters.
```

3.9.2.9 def kern::lin::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
% LINKERNGRADIENT Gradient of LIN kernel's parameters.
% FORMAT
% DESC computes the gradient of functions with respect to the
% linear
% kernel's parameters. As well as the kernel structure and the
% input positions, the user provides a matrix PARTIAL which gives
% the partial derivatives of the function with respect to the
% relevant elements of the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x : the input locations for which the gradients are being
% computed.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The argument takes
% the form of a square matrix of dimension  numData, where numData is
% the number of rows in X.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters. The ordering of the vector should match
% that provided by the function kernExtractParam.
%
% FORMAT
% DESC computes the derivatives as above, but input locations are
% now provided in two matrices associated with rows and columns of
% the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x1 : the input locations associated with the rows of the
% kernel matrix.
% ARG x2 : the input locations associated with the columns of the
% kernel matrix.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The matrix should
% have the same number of rows as X1 and the same number of columns
% as X2 has rows.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters.
```

```
%
% SEEALSO linKernParamInit, kernGradient, linKernDiagGradient, kernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.9.2.10 def kern::lin::gradX (*self*, *X*, *X2*)

```
% GRADX Gradient of LIN kernel with respect to input locations.
% FORMAT
% DESC computes the gradient of the linear
% kernel with respect to the input positions where both the row
% positions and column positions are provided separately.
% ARG kern : kernel structure for which gradients are being
% computed.
% ARG x1 : row locations against which gradients are being computed.
% ARG x2 : column locations against which gradients are being computed.
% RETURN g : the returned gradients. The gradients are returned in
% a matrix which is numData2 x numInputs x numData1. Where numData1 is
% the number of data points in X1, numData2 is the number of data
% points in X2 and numInputs is the number of input
% dimensions in X.
%
% SEEALSO : diagGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kern::kern.

3.9.2.11 def kern::lin::paramInit (*self*, *inDim* = None, *X* = None)

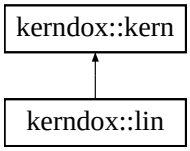
```
% LINKERNPARAMINIT LIN kernel parameter initialisation.
% The linear kernel (LIN) is the simple inner product
% kernel. Sampling from this kernel produces linear functions.
%
% k(x_i, x_j) = sigma2 * x_i'*x_j
%
% There is one parameter, sigma2, which is stored in the field
% kern.variance.
%
% SEEALSO : linardKernParamInit
%
% FORMAT
% DESC initialises the linear
%  kernel structure with some default parameters.
% ARG kern : the kernel structure which requires initialisation.
% RETURN kern : the kernel structure with the default parameters placed in.
%
% SEEALSO : kernCreate, kernParamInit
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

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

- kern.py

3.10 kerndox::lin Class Reference

Inheritance diagram for kerndox::lin::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def extractParamNames
- def gradX
- def gradient

Public Attributes

- **type**
- **variance**
- **nParams**
- **stationary**
- **normalised**

3.10.1 Detailed Description

```
The linear kernel (LIN) is the simple inner product
kernel. Sampling from this kernel produces linear functions.
%
k(x_i, x_j) = sigma2 * x_i'*x_j
%
There is one parameter, sigma2, which is stored in the field
kern.variance.
```

3.10.2 Member Function Documentation

3.10.2.1 def kerndox::lin::compute (*self*, *x*, *x2* = None)

```
LINKERNCOMPUTE Compute the LIN kernel given the parameters and X.
FORMAT
```

```
DESC computes the kernel parameters for the linear
kernel given inputs associated with rows and columns.
\param kern : the kernel structure for which the matrix is computed.
\param x : the input matrix associated with the rows of the kernel.
\param x2 : the input matrix associated with the columns of the kernel.
\return  k : the kernel matrix computed at the given points.
%
FORMAT
DESC computes the kernel matrix for the linear
kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : the kernel matrix computed at the given points.
%
SEEALSO : linKernParamInit, kernCompute, kernCreate, linKernDiagCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.2 def kerndox::lin::diagCompute (*self*, *x*)

```
LINKERNDIAGCOMPUTE Compute diagonal of LIN kernel.
FORMAT
DESC computes the diagonal of the kernel matrix for the linear kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : a vector containing the diagonal of the kernel matrix
computed at the given points.
%
SEEALSO : linKernParamInit, kernDiagCompute, kernCreate, linKernCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.3 def kerndox::lin::diagGradient (*self*, *X*, *covDiag*)

```
DIAGGRADIENT Compute the gradient of the RBF kernel's diagonal wrt parameters.
FORMAT
DESC computes the gradient of functions of the diagonal of the
radial basis function kernel matrix with respect to the parameters of the kernel. The
parameters' gradients are returned in the order given by the
rbfKernExtractParam command.
\param kern : the kernel structure for which the gradients are
computed.
\param x : the input data for which the gradient is being computed.
\param factors : partial derivatives of the function of interest with
respect to the diagonal elements of the kernel.
\return  g : gradients of the relevant function with respect to each
of the parameters. Ordering should match the ordering given in
rbfKernExtractParam.
%
SEEALSO : gradient
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.4 def kerndox::lin::diagGradX (*self*, *X*)

```
LINKERNDIAGGRADX Gradient of LIN kernel's diagonal with respect to X.
FORMAT
DESC computes the gradient of the diagonal of the linear kernel matrix with
respect to the elements of the design matrix given in X.
\param kern : the kernel structure for which gradients are being computed.
\param X : the input data in the form of a design matrix.
\return  gX : the gradients of the diagonal with respect to each element
of X. The returned matrix has the same dimensions as X.
%
SEEALSO : linKernParamInit, kernDiagGradX, linkernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.5 def kerndox::lin::display (*self*, *numSpaces* = 0)

```
DISPLAY Display parameters of the LIN kernel.
FORMAT
DESC displays the parameters of the radial basis function
kernel and the kernel type to the console.
\param kern : the kernel to display.
%
FORMAT does the same as above, but indents the display according
to the amount specified.
\param kern : the kernel to display.
\param spacing : how many spaces to indent the display of the kernel by.
%
SEEALSO :
%
COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.6 def kerndox::lin::expandParam (*self*, *params*)

```
EXPANDPARAM Create kernel structure from LIN kernel's parameters.
FORMAT
DESC returns a linear kernel structure filled with the
parameters in the given vector. This is used as a helper function to
enable parameters to be optimised in, for example, the NETLAB
optimisation functions.
\param kern : the kernel structure in which the parameters are to be
placed.
\param param : vector of parameters which are to be placed in the
kernel structure.
\return  kern : kernel structure with the given parameters in the
relevant locations.
%
SEEALSO : extractParam
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

3.10.2.7 def kerndox::lin::extractParam (*self*)

```
EXTRACTPARAM Extract parameters from the LIN kernel structure.
```

```
FORMAT
DESC Extract parameters from the linear kernel
structure into a vector of parameters for optimisation.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the kernel. If
the field 'transforms' is not empty in the kernel matrix, the
parameters will be transformed before optimisation (for example
positive only parameters could be logged before being returned).
%
SEEALSO expandParam, netlab.scg, netlab.conjgrad
%
COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

3.10.2.8 def kerndox::lin::extractParamNames (*self*)

```
EXTRACTPARAMNAMES Extract parameter names from the LIN kernel structure.
FORMAT
DESC Extract parameter names from the linear kernel structure.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  names : cell array of strings giving names to the parameters.
```

3.10.2.9 def kerndox::lin::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
LINKERNGRADIENT Gradient of LIN kernel's parameters.
FORMAT
DESC computes the gradient of functions with respect to the
linear
kernel's parameters. As well as the kernel structure and the
input positions, the user provides a matrix PARTIAL which gives
the partial derivatives of the function with respect to the
relevant elements of the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x : the input locations for which the gradients are being
computed.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The argument takes
the form of a square matrix of dimension  numData, where numData is
the number of rows in X.
\return  g : gradients of the function of interest with respect to
the kernel parameters. The ordering of the vector should match
that provided by the function kernExtractParam.
%
FORMAT
DESC computes the derivatives as above, but input locations are
now provided in two matrices associated with rows and columns of
the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x1 : the input locations associated with the rows of the
kernel matrix.
\param x2 : the input locations associated with the columns of the
kernel matrix.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The matrix should
have the same number of rows as X1 and the same number of columns
as X2 has rows.
\return  g : gradients of the function of interest with respect to
the kernel parameters.
```

```
%
SEEALSO linKernParamInit, kernGradient, linKernDiagGradient, kernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.10 def kerndox::lin::gradX (*self*, *X*, *X2*)

```
GRADX Gradient of LIN kernel with respect to input locations.
FORMAT
DESC computes the gradient of the linear
kernel with respect to the input positions where both the row
positions and column positions are provided separately.
\param kern : kernel structure for which gradients are being
computed.
\param x1 : row locations against which gradients are being computed.
\param x2 : column locations against which gradients are being computed.
\return  g : the returned gradients. The gradients are returned in
a matrix which is numData2 x numInputs x numData1. Where numData1 is
the number of data points in X1, numData2 is the number of data
points in X2 and numInputs is the number of input
dimensions in X.
%
SEEALSO : diagGradX
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kerndox::kern.

3.10.2.11 def kerndox::lin::paramInit (*self*, *inDim* = None, *X* = None)

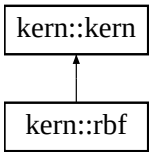
```
LINKERNPARAMINIT LIN kernel parameter initialisation.
The linear kernel (LIN) is the simple inner product
kernel. Sampling from this kernel produces linear functions.
%
k(x_i, x_j) = sigma2 * x_i'*x_j
%
There is one parameter, sigma2, which is stored in the field
kern.variance.
%
SEEALSO : linardKernParamInit
%
FORMAT
DESC initialises the linear
 kernel structure with some default parameters.
\param kern : the kernel structure which requires initialisation.
\return  kern : the kernel structure with the default parameters placed in.
%
SEEALSO : kernCreate, kernParamInit
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

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

- kerndox.py

3.11 kern::rbf Class Reference

Inheritance diagram for kern::rbf::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def extractParamNames
- def gradX
- def gradXpoint
- def gradient

Public Attributes

- **type**
- **inverseWidth**
- **variance**
- **nParams**
- **stationary**
- **normalised**

3.11.1 Detailed Description

```
% The radial basis function kernel (RBF) is sometimes also known as
% the squared exponential kernel. It is a very smooth non-linear
% kernel and is a popular choice for generic use.
%
% k(x_i, x_j) = sigma2 * exp(-gamma/2 *(x_i - x_j)'*(x_i - x_j))
%
% The parameters are sigma2, the process variance (kern.variance)
% and gamma, the inverse width (kern.inverseWidth). The inverse
% width controls how wide the basis functions are, the larger
% gamma, the smaller the basis functions are.
%
% There is also an automatic relevance determination version of
% this kernel provided.
```

3.11.2 Member Function Documentation

3.11.2.1 def kern::rbf::compute (*self*, *x*, *x2* = None)

```
% DESC computes the kernel parameters for the radial basis function
kernel given inputs associated with rows and columns.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : the input matrix associated with the rows of the kernel.
% ARG x2 : the input matrix associated with the columns of the kernel.
% RETURN k : the kernel matrix computed at the given points.

% FORMAT
% DESC computes the kernel matrix for the radial basis function
kernel given a design matrix of inputs.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : the kernel matrix computed at the given points.
%
% SEEALSO : diagCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kern::kern.

3.11.2.2 def kern::rbf::diagCompute (*self*, *x*)

```
% DIAGCOMPUTE Compute diagonal of RBF kernel.
% FORMAT
% DESC computes the diagonal of the kernel matrix for the radial basis function kernel given a design matr
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : a vector containing the diagonal of the kernel matrix
% computed at the given points.
%
% SEEALSO : compute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

Reimplemented from kern::kern.

3.11.2.3 def kern::rbf::diagGradient (*self*, *X*, *covDiag*)

```
% DIAGGRADIENT Compute the gradient of the RBF kernel's diagonal wrt parameters.
% FORMAT
% DESC computes the gradient of functions of the diagonal of the
% radial basis function kernel matrix with respect to the parameters of the kernel. The
% parameters' gradients are returned in the order given by the
% rbfKernExtractParam command.
% ARG kern : the kernel structure for which the gradients are
% computed.
% ARG x : the input data for which the gradient is being computed.
% ARG factors : partial derivatives of the function of interest with
% respect to the diagonal elements of the kernel.
% RETURN g : gradients of the relevant function with respect to each
% of the parameters. Ordering should match the ordering given in
% rbfKernExtractParam.
%
% SEEALSO : gradient
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kern::kern.

3.11.2.4 def kern::rbf::diagGradX (*self*, *X*)

```
% DIAGGRADX Gradient of RBF kernel's diagonal with respect to X.
% FORMAT
% DESC computes the gradient of the diagonal of the radial basis function kernel matrix with
% respect to the elements of the design matrix given in X.
% ARG kern : the kernel structure for which gradients are being computed.
% ARG X : the input data in the form of a design matrix.
% RETURN gX : the gradients of the diagonal with respect to each element
% of X. The returned matrix has the same dimensions as X.
%
% SEEALSO : gradX
%
% COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

Reimplemented from kern::kern.

3.11.2.5 def kern::rbf::display (*self*, *numSpaces* = 0)

```
% DISPLAY Display parameters of the RBF kernel.
% FORMAT
% DESC displays the parameters of the radial basis function
% kernel and the kernel type to the console.
% ARG kern : the kernel to display.
%
% FORMAT does the same as above, but indents the display according
% to the amount specified.
% ARG kern : the kernel to display.
% ARG spacing : how many spaces to indent the display of the kernel by.
%
% SEEALSO :
%
% COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

Reimplemented from kern::kern.

3.11.2.6 def kern::rbf::expandParam (*self*, *params*)

```
% EXPANDPARAM Create kernel structure from RBF kernel's parameters.
% FORMAT
% DESC returns a radial basis function kernel structure filled with the
% parameters in the given vector. This is used as a helper function to
% enable parameters to be optimised in, for example, the NETLAB
% optimisation functions.
% ARG kern : the kernel structure in which the parameters are to be
% placed.
% ARG param : vector of parameters which are to be placed in the
% kernel structure.
% RETURN kern : kernel structure with the given parameters in the
% relevant locations.
%
% SEEALSO : extractParam
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

3.11.2.7 def kern::rbf::extractParam (*self*)

```
% EXTRACTPARAM Extract parameters from the RBF kernel structure.
% FORMAT
% DESC Extract parameters from the radial basis function kernel
% structure into a vector of parameters for optimisation.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the kernel. If
% the field 'transforms' is not empty in the kernel matrix, the
% parameters will be transformed before optimisation (for example
% positive only parameters could be logged before being returned).
%
% SEEALSO expandParam, netlab.scg, netlab.conjgrad
%
% COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

3.11.2.8 def kern::rbf::extractParamNames (*self*)

```
% EXTRACTPARAMNAMES Extract parameter names from the RBF kernel structure.
% FORMAT
% DESC Extract parameter names from the radial basis
% function kernel structure.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN names : cell array of strings giving names to the parameters.
```

3.11.2.9 def kern::rbf::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
% GRADIENT Gradient of RBF kernel's parameters.
% FORMAT
% DESC computes the gradient of functions with respect to the
% radial basis function
% kernel's parameters. As well as the kernel structure and the
% input positions, the user provides a matrix PARTIAL which gives
% the partial derivatives of the function with respect to the
% relevant elements of the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x : the input locations for which the gradients are being
% computed.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The argument takes
% the form of a square matrix of dimension  numData, where numData is
% the number of rows in X.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters. The ordering of the vector should match
% that provided by the function extractParam.
%
% FORMAT
% DESC computes the derivatives as above, but input locations are
% now provided in two matrices associated with rows and columns of
% the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x1 : the input locations associated with the rows of the
% kernel matrix.
% ARG x2 : the input locations associated with the columns of the
% kernel matrix.
% ARG partial : matrix of partial derivatives of the function of
```

```
% interest with respect to the kernel matrix. The matrix should
% have the same number of rows as X1 and the same number of columns
% as X2 has rows.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters.
%
% SEEALSO diagGradient, gradX
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kern::kern.

3.11.2.10 def kern::rbf::gradX (*self*, *X*, *X2*)

```
% GRADX Gradient of RBF kernel with respect to input locations.
% FORMAT
% DESC computes the gradident of the radial basis function
% kernel with respect to the input positions where both the row
% positions and column positions are provided separately.
% ARG kern : kernel structure for which gradients are being
% computed.
% ARG x1 : row locations against which gradients are being computed.
% ARG x2 : column locations against which gradients are being computed.
% RETURN g : the returned gradients. The gradients are returned in
% a matrix which is numData2 x numInputs x numData1. Where numData1 is
% the number of data points in X1, numData2 is the number of data
% points in X2 and numInputs is the number of input
% dimensions in X.
%
% SEEALSO : diagGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kern::kern.

3.11.2.11 def kern::rbf::gradXpoint (*self*, *x*, *X2*)

```
% GRADXPOINT Gradient with respect to one point of x.
```

3.11.2.12 def kern::rbf::paramInit (*self*, *inDim* = None, *X* = None)

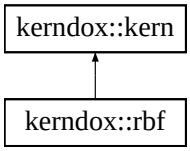
```
% RBFKERNPARAMINIT RBF kernel parameter initialisation.
%
% SEEALSO : rbfardKernParamInit
%
% FORMAT
% DESC initialises the radial basis function
%  kernel structure with some default parameters.
% ARG kern : the kernel structure which requires initialisation.
% RETURN kern : the kernel structure with the default parameters placed in.
%
% SEEALSO : create, kernParamInit
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

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

- kern.py

3.12 kerndox::rbf Class Reference

Inheritance diagram for kerndox::rbf::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def extractParamNames
- def gradX
- def gradXpoint
- def gradient

Public Attributes

- **type**
- **inverseWidth**
- **variance**
- **nParams**
- **stationary**
- **normalised**

3.12.1 Detailed Description

```
The radial basis function kernel (RBF) is sometimes also known as
the squared exponential kernel. It is a very smooth non-linear
kernel and is a popular choice for generic use.
%
k(x_i, x_j) = sigma2 * exp(-gamma/2 *(x_i - x_j)'*(x_i - x_j))
%
The parameters are sigma2, the process variance (kern.variance)
and gamma, the inverse width (kern.inverseWidth). The inverse
width controls how wide the basis functions are, the larger
gamma, the smaller the basis functions are.
%
There is also an automatic relevance determination version of
this kernel provided.
```

3.12.2 Member Function Documentation

3.12.2.1 def kerndox::rbf::compute (*self*, *x*, *x2* = None)

```
DESC computes the kernel parameters for the radial basis function
kernel given inputs associated with rows and columns.
\param kern : the kernel structure for which the matrix is computed.
\param x : the input matrix associated with the rows of the kernel.
\param x2 : the input matrix associated with the columns of the kernel.
\return  k : the kernel matrix computed at the given points.

FORMAT
DESC computes the kernel matrix for the radial basis function
kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : the kernel matrix computed at the given points.
%
SEEALSO : diagCompute
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kerndox::kern.

3.12.2.2 def kerndox::rbf::diagCompute (*self*, *x*)

```
DIAGCOMPUTE Compute diagonal of RBF kernel.
FORMAT
DESC computes the diagonal of the kernel matrix for the radial basis function kernel given a design matrix
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : a vector containing the diagonal of the kernel matrix
computed at the given points.
%
SEEALSO : compute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

Reimplemented from kerndox::kern.

3.12.2.3 def kerndox::rbf::diagGradient (*self*, *X*, *covDiag*)

```
DIAGGRADIENT Compute the gradient of the RBF kernel's diagonal wrt parameters.
FORMAT
DESC computes the gradient of functions of the diagonal of the
radial basis function kernel matrix with respect to the parameters of the kernel. The
parameters' gradients are returned in the order given by the
rbfKernExtractParam command.
\param kern : the kernel structure for which the gradients are
computed.
\param x : the input data for which the gradient is being computed.
\param factors : partial derivatives of the function of interest with
respect to the diagonal elements of the kernel.
\return  g : gradients of the relevant function with respect to each
of the parameters. Ordering should match the ordering given in
rbfKernExtractParam.
%
SEEALSO : gradient
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kerndox::kern.

3.12.2.4 def kerndox::rbf::diagGradX (*self*, *X*)

```
DIAGGRADX Gradient of RBF kernel's diagonal with respect to X.
FORMAT
DESC computes the gradient of the diagonal of the radial basis function kernel matrix with
respect to the elements of the design matrix given in X.
\param kern : the kernel structure for which gradients are being computed.
\param X : the input data in the form of a design matrix.
\return  gX : the gradients of the diagonal with respect to each element
of X. The returned matrix has the same dimensions as X.
%
SEEALSO : gradX
%
COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

Reimplemented from kerndox::kern.

3.12.2.5 def kerndox::rbf::display (*self*, *numSpaces* = 0)

```
DISPLAY Display parameters of the RBF kernel.
FORMAT
DESC displays the parameters of the radial basis function
kernel and the kernel type to the console.
\param kern : the kernel to display.
%
FORMAT does the same as above, but indents the display according
to the amount specified.
\param kern : the kernel to display.
\param spacing : how many spaces to indent the display of the kernel by.
%
SEEALSO :
%
COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

Reimplemented from kerndox::kern.

3.12.2.6 def kerndox::rbf::expandParam (*self*, *params*)

```
EXPANDPARAM Create kernel structure from RBF kernel's parameters.
FORMAT
DESC returns a radial basis function kernel structure filled with the
parameters in the given vector. This is used as a helper function to
enable parameters to be optimised in, for example, the NETLAB
optimisation functions.
\param kern : the kernel structure in which the parameters are to be
placed.
\param param : vector of parameters which are to be placed in the
kernel structure.
\return  kern : kernel structure with the given parameters in the
relevant locations.
%
SEEALSO : extractParam
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

3.12.2.7 def kerndox::rbf::extractParam (*self*)

```
EXTRACTPARAM Extract parameters from the RBF kernel structure.
FORMAT
DESC Extract parameters from the radial basis function kernel
structure into a vector of parameters for optimisation.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the kernel. If
the field 'transforms' is not empty in the kernel matrix, the
parameters will be transformed before optimisation (for example
positive only parameters could be logged before being returned).
%
SEEALSO expandParam, netlab.scg, netlab.conjgrad
%
COPYRIGHT : Neil D. Lawrence, 2004--2006, 2009
```

3.12.2.8 def kerndox::rbf::extractParamNames (*self*)

```
EXTRACTPARAMNAMES Extract parameter names from the RBF kernel structure.
FORMAT
DESC Extract parameter names from the radial basis
function kernel structure.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  names : cell array of strings giving names to the parameters.
```

3.12.2.9 def kerndox::rbf::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
GRADIENT Gradient of RBF kernel's parameters.
FORMAT
DESC computes the gradient of functions with respect to the
radial basis function
kernel's parameters. As well as the kernel structure and the
input positions, the user provides a matrix PARTIAL which gives
the partial derivatives of the function with respect to the
relevant elements of the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x : the input locations for which the gradients are being
computed.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The argument takes
the form of a square matrix of dimension  numData, where numData is
the number of rows in X.
\return  g : gradients of the function of interest with respect to
the kernel parameters. The ordering of the vector should match
that provided by the function extractParam.
%
FORMAT
DESC computes the derivatives as above, but input locations are
now provided in two matrices associated with rows and columns of
the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x1 : the input locations associated with the rows of the
kernel matrix.
\param x2 : the input locations associated with the columns of the
kernel matrix.
\param partial : matrix of partial derivatives of the function of
```

```
interest with respect to the kernel matrix. The matrix should
have the same number of rows as X1 and the same number of columns
as X2 has rows.
\return  g : gradients of the function of interest with respect to
the kernel parameters.
%
SEEALSO diagGradient, gradX
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kerndox::kern.

3.12.2.10 def kerndox::rbf::gradX (*self*, *X*, *X2*)

```
GRADX Gradient of RBF kernel with respect to input locations.
FORMAT
DESC computes the gradident of the radial basis function
kernel with respect to the input positions where both the row
positions and column positions are provided separately.
\param kern : kernel structure for which gradients are being
computed.
\param x1 : row locations against which gradients are being computed.
\param x2 : column locations against which gradients are being computed.
\return  g : the returned gradients. The gradients are returned in
a matrix which is numData2 x numInputs x numData1. Where numData1 is
the number of data points in X1, numData2 is the number of data
points in X2 and numInputs is the number of input
dimensions in X.
%
SEEALSO : diagGradX
%
COPYRIGHT : Neil D. Lawrence, 2004-2006, 2009
```

Reimplemented from kerndox::kern.

3.12.2.11 def kerndox::rbf::gradXpoint (*self*, *x*, *X2*)

```
GRADXPOINT Gradient with respect to one point of x.
```

3.12.2.12 def kerndox::rbf::paramInit (*self*, *inDim* = None, *X* = None)

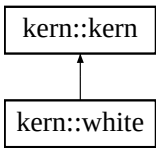
```
RBFKERNPARAMINIT RBF kernel parameter initialisation.
%
SEEALSO : rbfardKernParamInit
%
FORMAT
DESC initialises the radial basis function
 kernel structure with some default parameters.
\param kern : the kernel structure which requires initialisation.
\return  kern : the kernel structure with the default parameters placed in.
%
SEEALSO : create, kernParamInit
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

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

- kerndox.py

3.13 kern::white Class Reference

Inheritance diagram for kern::white::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def **extractParamNames**
- def gradX
- def gradient

Public Attributes

- **type**
- **variance**
- **nParams**
- **stationary**

3.13.1 Detailed Description

```
% The white noise kernel arises from assuming independent Gaussian
% noise for each point in the function. The variance of the noise is
% given by the kern.variance parameter.
%
% This kernel is not intended to be used independently, it is provided
% so that it may be combined with other kernels in a compound kernel.
```

3.13.2 Member Function Documentation

3.13.2.1 def kern::white::compute (*self*, *x*, *x2* = None)

```
% WHITEKERNCOMPUTE Compute the WHITE kernel given the parameters and X.
% FORMAT
% DESC computes the kernel parameters for the white noise
% kernel given inputs associated with rows and columns.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : the input matrix associated with the rows of the kernel.
```

```
% ARG x2 : the inpute matrix associated with the columns of the kernel.
% RETURN k : the kernel matrix computed at the given points.
%
% FORMAT
% DESC computes the kernel matrix for the white noise
% kernel given a design matrix of inputs.
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : the kernel matrix computed at the given points.
%
% SEEALSO : whiteKernParamInit, kernCompute, create, whiteKernDiagCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.13.2.2 def kern::white::diagCompute (*self*, *x*)

```
% WHITEKERNDIAGCOMPUTE Compute diagonal of WHITE kernel.
% FORMAT
% DESC computes the diagonal of the kernel matrix for the white noise kernel given a design matrix of inpu
% ARG kern : the kernel structure for which the matrix is computed.
% ARG x : input data matrix in the form of a design matrix.
% RETURN k : a vector containing the diagonal of the kernel matrix
% computed at the given points.
%
% SEEALSO : whiteKernParamInit, kernDiagCompute, create, whiteKernCompute
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.13.2.3 def kern::white::diagGradient (*self*, *X*, *covDiag*)

```
% WHITEKERNDIAGGRADIENT Compute the gradient of the WHITE kernel's diagonal wrt parameters.
% FORMAT
% DESC computes the gradient of functions of the diagonal of the
% white noise kernel matrix with respect to the parameters of the kernel. The
% parameters' gradients are returned in the order given by the
% whiteKernExtractParam command.
% ARG kern : the kernel structure for which the gradients are
% computed.
% ARG x : the input data for which the gradient is being computed.
% ARG factors : partial derivatives of the function of interest with
% respect to the diagonal elements of the kernel.
% RETURN g : gradients of the relevant function with respect to each
% of the parameters. Ordering should match the ordering given in
% whiteKernExtractParam.
%
% SEEALSO : whiteKernParamInit, kernDiagGradient, whiteKernExtractParam, whiteKernGradient
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.13.2.4 def kern::white::diagGradX (*self*, *X*)

```
% WHITEKERNDIAGGRADX Gradient of WHITE kernel's diagonal with respect to X.
```

```
% FORMAT
% DESC computes the gradient of the diagonal of the white noise kernel matrix with
% respect to the elements of the design matrix given in X.
% ARG kern : the kernel structure for which gradients are being computed.
% ARG X : the input data in the form of a design matrix.
% RETURN gX : the gradients of the diagonal with respect to each element
% of X. The returned matrix has the same dimensions as X.
%
% SEEALSO : whiteKernParamInit, kernDiagGradX, whitekernGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.13.2.5 def kern::white::display (*self*, *numSpaces* = 0)

```
% WHITEKERNDISPLAY Display parameters of the WHITE kernel.
% FORMAT
% DESC displays the parameters of the white noise
% kernel and the kernel type to the console.
% ARG kern : the kernel to display.
%
% FORMAT does the same as above, but indents the display according
% to the amount specified.
% ARG kern : the kernel to display.
% ARG spacing : how many spaces to indent the display of the kernel by.
%
% SEEALSO : whiteKernParamInit, modelDisplay, kernDisplay
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.13.2.6 def kern::white::expandParam (*self*, *params*)

```
% WHITEKERNEXPANDPARAM Create kernel structure from WHITE kernel's parameters.
% FORMAT
% DESC returns a white noise kernel structure filled with the
% parameters in the given vector. This is used as a helper function to
% enable parameters to be optimised in, for example, the NETLAB
% optimisation functions.
% ARG kern : the kernel structure in which the parameters are to be
% placed.
% ARG param : vector of parameters which are to be placed in the
% kernel structure.
% RETURN kern : kernel structure with the given parameters in the
% relevant locations.
%
% SEEALSO : whiteKernParamInit, whiteKernExtractParam, kernExpandParam
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

3.13.2.7 def kern::white::extractParam (*self*)

```
% WHITEKERNEXTRACTPARAM Extract parameters from the WHITE kernel structure.
% FORMAT
% DESC Extract parameters from the white noise kernel structure into a
```

```
% vector of parameters for optimisation.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the kernel. If
% the field 'transforms' is not empty in the kernel matrix, the
% parameters will be transformed before optimisation (for example
% positive only parameters could be logged before being returned).
%
% FORMAT
% DESC Extract parameters and parameter names from the white noise
% kernel structure.
% ARG kern : the kernel structure containing the parameters to be
% extracted.
% RETURN param : vector of parameters extracted from the kernel. If
% the field 'transforms' is not empty in the kernel matrix, the
% parameters will be transformed before optimisation (for example
% positive only parameters could be logged before being returned).
% RETURN names : cell array of strings giving paramter names.
%
% SEEALSO whiteKernParamInit, whiteKernExpandParam, kernExtractParam, scg, conjgrad
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
%
```

3.13.2.8 def kern::white::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
% WHITEKERNGRADIENT Gradient of WHITE kernel's parameters.
% FORMAT
% DESC computes the gradient of functions with respect to the
% white noise
% kernel's parameters. As well as the kernel structure and the
% input positions, the user provides a matrix PARTIAL which gives
% the partial derivatives of the function with respect to the
% relevant elements of the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x : the input locations for which the gradients are being
% computed.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The argument takes
% the form of a square matrix of dimension  numData, where numData is
% the number of rows in X.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters. The ordering of the vector should match
% that provided by the function kernExtractParam.
%
% FORMAT
% DESC computes the derivatives as above, but input locations are
% now provided in two matrices associated with rows and columns of
% the kernel matrix.
% ARG kern : the kernel structure for which the gradients are being
% computed.
% ARG x1 : the input locations associated with the rows of the
% kernel matrix.
% ARG x2 : the input locations associated with the columns of the
% kernel matrix.
% ARG partial : matrix of partial derivatives of the function of
% interest with respect to the kernel matrix. The matrix should
% have the same number of rows as X1 and the same number of columns
% as X2 has rows.
% RETURN g : gradients of the function of interest with respect to
% the kernel parameters.
%
% SEEALSO whiteKernParamInit, kernGradient, whiteKernDiagGradient, kernGradX
%
```

```
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kern::kern.

3.13.2.9 def kern::white::gradX (*self*, *X*, *X2* = None)

```
% WHITEKERNGRADX Gradient of WHITE kernel with respect to input locations.
% FORMAT
% DESC computes the gradident of the white noise
% kernel with respect to the input positions where both the row
% positions and column positions are provided separately.
% ARG kern : kernel structure for which gradients are being
% computed.
% ARG x1 : row locations against which gradients are being computed.
% ARG x2 : column locations against which gradients are being computed.
% RETURN g : the returned gradients. The gradients are returned in
% a matrix which is numData2 x numInputs x numData1. Where numData1 is
% the number of data points in X1, numData2 is the number of data
% points in X2 and numInputs is the number of input
% dimensions in X.
%
% SEEALSO whiteKernParamInit, kernGradX, whiteKernDiagGradX
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

Reimplemented from kern::kern.

3.13.2.10 def kern::white::paramInit (*self*, *inDim* = None, *X* = None)

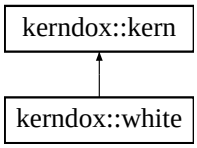
```
% WHITEKERNPARAMINIT WHITE kernel parameter initialisation.
%
% SEEALSO : cmpndKernParamInit
%
% FORMAT
% DESC initialises the white noise
%  kernel structure with some default parameters.
% ARG kern : the kernel structure which requires initialisation.
% RETURN kern : the kernel structure with the default parameters placed in.
%
% SEEALSO : create, kernParamInit
%
% COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

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

- kern.py

3.14 kerndox::white Class Reference

Inheritance diagram for kerndox::white::



Public Member Functions

- def **__init__**
- def paramInit
- def compute
- def diagCompute
- def diagGradX
- def diagGradient
- def display
- def expandParam
- def extractParam
- def **extractParamNames**
- def gradX
- def gradient

Public Attributes

- **type**
- **variance**
- **nParams**
- **stationary**

3.14.1 Detailed Description

```
The white noise kernel arises from assuming independent Gaussian
noise for each point in the function. The variance of the noise is
given by the kern.variance parameter.

This kernel is not intended to be used independently, it is provided
so that it may be combined with other kernels in a compound kernel.
```

3.14.2 Member Function Documentation

3.14.2.1 def kerndox::white::compute (*self*, *x*, *x2* = None)

```
WHITEKERNCOMPUTE Compute the WHITE kernel given the parameters and X.
FORMAT
DESC computes the kernel parameters for the white noise
kernel given inputs associated with rows and columns.
\param kern : the kernel structure for which the matrix is computed.
\param x : the input matrix associated with the rows of the kernel.
```

```
\param x2 : the inpute matrix associated with the columns of the kernel.
\return  k : the kernel matrix computed at the given points.
%
FORMAT
DESC computes the kernel matrix for the white noise
kernel given a design matrix of inputs.
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : the kernel matrix computed at the given points.
%
SEEALSO : whiteKernParamInit, kernCompute, create, whiteKernDiagCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.14.2.2 def kerndox::white::diagCompute (*self*, *x*)

```
WHITEKERNDIAGCOMPUTE Compute diagonal of WHITE kernel.
FORMAT
DESC computes the diagonal of the kernel matrix for the white noise kernel given a design matrix of inputs
\param kern : the kernel structure for which the matrix is computed.
\param x : input data matrix in the form of a design matrix.
\return  k : a vector containing the diagonal of the kernel matrix
computed at the given points.
%
SEEALSO : whiteKernParamInit, kernDiagCompute, create, whiteKernCompute
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.14.2.3 def kerndox::white::diagGradient (*self*, *X*, *covDiag*)

```
WHITEKERNDIAGGRADIENT Compute the gradient of the WHITE kernel's diagonal wrt parameters.
FORMAT
DESC computes the gradient of functions of the diagonal of the
white noise kernel matrix with respect to the parameters of the kernel. The
parameters' gradients are returned in the order given by the
whiteKernExtractParam command.
\param kern : the kernel structure for which the gradients are
computed.
\param x : the input data for which the gradient is being computed.
\param factors : partial derivatives of the function of interest with
respect to the diagonal elements of the kernel.
\return  g : gradients of the relevant function with respect to each
of the parameters. Ordering should match the ordering given in
whiteKernExtractParam.
%
SEEALSO : whiteKernParamInit, kernDiagGradient, whiteKernExtractParam, whiteKernGradient
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.14.2.4 def kerndox::white::diagGradX (*self*, *X*)

```
WHITEKERNDIAGGRADX Gradient of WHITE kernel's diagonal with respect to X.
```

```
FORMAT
DESC computes the gradient of the diagonal of the white noise kernel matrix with
respect to the elements of the design matrix given in X.
\param kern : the kernel structure for which gradients are being computed.
\param X : the input data in the form of a design matrix.
\return  gX : the gradients of the diagonal with respect to each element
of X. The returned matrix has the same dimensions as X.
%
SEEALSO : whiteKernParamInit, kernDiagGradX, whitekernGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.14.2.5 def kerndox::white::display (*self*, *numSpaces* = 0)

```
WHITEKERNDISPLAY Display parameters of the WHITE kernel.
FORMAT
DESC displays the parameters of the white noise
kernel and the kernel type to the console.
\param kern : the kernel to display.
%
FORMAT does the same as above, but indents the display according
to the amount specified.
\param kern : the kernel to display.
\param spacing : how many spaces to indent the display of the kernel by.
%
SEEALSO : whiteKernParamInit, modelDisplay, kernDisplay
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.14.2.6 def kerndox::white::expandParam (*self*, *params*)

```
WHITEKERNEXPANDPARAM Create kernel structure from WHITE kernel's parameters.
FORMAT
DESC returns a white noise kernel structure filled with the
parameters in the given vector. This is used as a helper function to
enable parameters to be optimised in, for example, the NETLAB
optimisation functions.
\param kern : the kernel structure in which the parameters are to be
placed.
\param param : vector of parameters which are to be placed in the
kernel structure.
\return  kern : kernel structure with the given parameters in the
relevant locations.
%
SEEALSO : whiteKernParamInit, whiteKernExtractParam, kernExpandParam
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

3.14.2.7 def kerndox::white::extractParam (*self*)

```
WHITEKERNEXTRACTPARAM Extract parameters from the WHITE kernel structure.
FORMAT
DESC Extract parameters from the white noise kernel structure into a
```

```
vector of parameters for optimisation.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the kernel. If
the field 'transforms' is not empty in the kernel matrix, the
parameters will be transformed before optimisation (for example
positive only parameters could be logged before being returned).
%
FORMAT
DESC Extract parameters and parameter names from the white noise
kernel structure.
\param kern : the kernel structure containing the parameters to be
extracted.
\return  param : vector of parameters extracted from the kernel. If
the field 'transforms' is not empty in the kernel matrix, the
parameters will be transformed before optimisation (for example
positive only parameters could be logged before being returned).
\return  names : cell array of strings giving paramter names.
%
SEEALSO whiteKernParamInit, whiteKernExpandParam, kernExtractParam, scg, conjgrad
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
%
```

3.14.2.8 def kerndox::white::gradient (*self*, *X*, *X2* = None, *covGrad* = None)

```
WHITEKERNGRADIENT Gradient of WHITE kernel's parameters.
FORMAT
DESC computes the gradient of functions with respect to the
white noise
kernel's parameters. As well as the kernel structure and the
input positions, the user provides a matrix PARTIAL which gives
the partial derivatives of the function with respect to the
relevant elements of the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x : the input locations for which the gradients are being
computed.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The argument takes
the form of a square matrix of dimension  numData, where numData is
the number of rows in X.
\return  g : gradients of the function of interest with respect to
the kernel parameters. The ordering of the vector should match
that provided by the function kernExtractParam.
%
FORMAT
DESC computes the derivatives as above, but input locations are
now provided in two matrices associated with rows and columns of
the kernel matrix.
\param kern : the kernel structure for which the gradients are being
computed.
\param x1 : the input locations associated with the rows of the
kernel matrix.
\param x2 : the input locations associated with the columns of the
kernel matrix.
\param partial : matrix of partial derivatives of the function of
interest with respect to the kernel matrix. The matrix should
have the same number of rows as X1 and the same number of columns
as X2 has rows.
\return  g : gradients of the function of interest with respect to
the kernel parameters.
%
SEEALSO whiteKernParamInit, kernGradient, whiteKernDiagGradient, kernGradX
%
```

```
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

Reimplemented from kerndox::kern.

3.14.2.9 def kerndox::white::gradX (*self*, *X*, *X2* = None)

```
WHITEKERNGRADX Gradient of WHITE kernel with respect to input locations.
FORMAT
DESC computes the gradident of the white noise
kernel with respect to the input positions where both the row
positions and column positions are provided separately.
\param kern : kernel structure for which gradients are being
computed.
\param x1 : row locations against which gradients are being computed.
\param x2 : column locations against which gradients are being computed.
\return  g : the returned gradients. The gradients are returned in
a matrix which is numData2 x numInputs x numData1. Where numData1 is
the number of data points in X1, numData2 is the number of data
points in X2 and numInputs is the number of input
dimensions in X.
%
SEEALSO whiteKernParamInit, kernGradX, whiteKernDiagGradX
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006
```

Reimplemented from kerndox::kern.

3.14.2.10 def kerndox::white::paramInit (*self*, *inDim* = None, *X* = None)

```
WHITEKERNPARAMINIT WHITE kernel parameter initialisation.
%
SEEALSO : cmpndKernParamInit
%
FORMAT
DESC initialises the white noise
 kernel structure with some default parameters.
\param kern : the kernel structure which requires initialisation.
\return  kern : the kernel structure with the default parameters placed in.
%
SEEALSO : create, kernParamInit
%
COPYRIGHT : Neil D. Lawrence, 2004, 2005, 2006, 2009
```

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

- kerndox.py

Index