

ooDACE toolbox

1.4

Generated by Doxygen 1.8.2

Mon Jun 3 2013 11:49:07

## Contents

<b>1</b>	<b>The ooDACE toolbox documentation</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Download . . . . .	1
1.3	Quick start guide . . . . .	1
1.4	DACE toolbox interface . . . . .	2
1.5	Contribute . . . . .	2
<b>2</b>	<b>Test List</b>	<b>2</b>
<b>3</b>	<b>Todo List</b>	<b>2</b>
<b>4</b>	<b>Hierarchical Index</b>	<b>3</b>
4.1	Class Hierarchy . . . . .	3
<b>5</b>	<b>Class Index</b>	<b>3</b>
5.1	Class List . . . . .	3
<b>6</b>	<b>File Index</b>	<b>4</b>
6.1	File List . . . . .	4
<b>7</b>	<b>Class Documentation</b>	<b>7</b>
7.1	BasicGaussianProcess Class Reference . . . . .	7
7.1.1	Detailed Description . . . . .	10
7.1.2	Constructor & Destructor Documentation . . . . .	11
7.1.3	Member Function Documentation . . . . .	11
7.1.4	Member Data Documentation . . . . .	19
7.2	BlindKriging Class Reference . . . . .	21
7.2.1	Detailed Description . . . . .	24
7.2.2	Constructor & Destructor Documentation . . . . .	25
7.2.3	Member Function Documentation . . . . .	25
7.2.4	Member Data Documentation . . . . .	33
7.3	CoKriging Class Reference . . . . .	35
7.3.1	Detailed Description . . . . .	38
7.3.2	Constructor & Destructor Documentation . . . . .	38
7.3.3	Member Function Documentation . . . . .	38
7.3.4	Member Data Documentation . . . . .	47
7.4	Kriging Class Reference . . . . .	49
7.4.1	Detailed Description . . . . .	52
7.4.2	Constructor & Destructor Documentation . . . . .	52
7.4.3	Member Function Documentation . . . . .	52
7.4.4	Member Data Documentation . . . . .	60

7.5	MatlabGA Class Reference . . . . .	62
7.5.1	Detailed Description . . . . .	63
7.5.2	Constructor & Destructor Documentation . . . . .	63
7.5.3	Member Function Documentation . . . . .	63
7.6	MatlabOptimizer Class Reference . . . . .	65
7.6.1	Detailed Description . . . . .	67
7.6.2	Constructor & Destructor Documentation . . . . .	67
7.6.3	Member Function Documentation . . . . .	67
7.7	Optimizer Class Reference . . . . .	69
7.7.1	Detailed Description . . . . .	71
7.7.2	Constructor & Destructor Documentation . . . . .	71
7.7.3	Member Function Documentation . . . . .	71
7.8	SQPLabOptimizer Class Reference . . . . .	74
7.8.1	Detailed Description . . . . .	75
7.8.2	Constructor & Destructor Documentation . . . . .	75
7.8.3	Member Function Documentation . . . . .	75
<b>8</b>	<b>File Documentation</b>	<b>78</b>
8.1	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/BasicGaussianProcess.m File Reference . . . . .	78
8.1.1	Detailed Description . . . . .	78
8.2	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/correlationFunction.m File Reference . . . . .	78
8.2.1	Detailed Description . . . . .	78
8.3	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/correlationFunction.m File Reference . . . . .	79
8.3.1	Detailed Description . . . . .	79
8.4	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/cvpe.m File Reference . . . . .	80
8.4.1	Detailed Description . . . . .	80
8.5	/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/cvpe.m File Reference . . . . .	80
8.5.1	Detailed Description . . . . .	80
8.6	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/extrinsicCorrelationMatrix.m File Reference . . . . .	81
8.6.1	Detailed Description . . . . .	81
8.7	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/extrinsicCorrelationMatrix.m File Reference . . . . .	81
8.7.1	Detailed Description . . . . .	81
8.8	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/fit.m File Reference . . . . .	82
8.8.1	Detailed Description . . . . .	82
8.9	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/fit.m File Reference . . . . .	83
8.9.1	Detailed Description . . . . .	83
8.10	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/fit.m File Reference . . . . .	83
8.10.1	Detailed Description . . . . .	83

8.11	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/generateDegrees.m File Reference	84
8.11.1	Detailed Description	84
8.12	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/getExpression.m File Reference	84
8.12.1	Detailed Description	84
8.13	/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/getExpression.m File Reference	85
8.13.1	Detailed Description	85
8.14	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/imse.m File Reference	86
8.14.1	Detailed Description	86
8.15	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/intrinsicCovarianceMatrix.m File Reference	86
8.15.1	Detailed Description	86
8.16	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/intrinsicCovarianceMatrix.m File Reference	87
8.16.1	Detailed Description	87
8.17	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/likelihood.m File Reference	87
8.17.1	Detailed Description	87
8.18	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/marginalLikelihood.m File Reference	88
8.18.1	Detailed Description	88
8.19	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/mseTestset.m File Reference	89
8.19.1	Detailed Description	89
8.20	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/plotLikelihood.m File Reference	89
8.20.1	Detailed Description	89
8.21	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/plotVariogram.m File Reference	90
8.21.1	Detailed Description	90
8.22	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/predict.m File Reference	90
8.22.1	Detailed Description	90
8.23	/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/predict.m File Reference	91
8.23.1	Detailed Description	91
8.24	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/predict_derivatives.m File Reference	92
8.24.1	Detailed Description	92
8.25	/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/predict_derivatives.m File Reference	92
8.25.1	Detailed Description	92
8.26	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/pseudoLikelihood.m File Reference	93
8.26.1	Detailed Description	93
8.27	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/rcValues.m File Reference	93
8.27.1	Detailed Description	93
8.28	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/regressionFunction.m File Reference	94
8.28.1	Detailed Description	94

8.29	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/regressionFunction.m File Reference . . .	95
8.29.1	Detailed Description . . . . .	95
8.30	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/regressionFunction.m File Reference . . .	95
8.30.1	Detailed Description . . . . .	95
8.31	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/regressionMatrix.m File Reference . . . . .	96
8.31.1	Detailed Description . . . . .	96
8.32	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/regressionMatrix.m File Reference . . . . .	96
8.32.1	Detailed Description . . . . .	96
8.33	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/regressionMatrix.m File Reference . . . . .	97
8.33.1	Detailed Description . . . . .	97
8.34	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/tuneParameters.m File Reference . . . . .	98
8.34.1	Detailed Description . . . . .	98
8.35	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateModel.m File Reference . . . . .	98
8.35.1	Detailed Description . . . . .	98
8.36	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateRegression.m File Reference . . . . .	99
8.36.1	Detailed Description . . . . .	99
8.37	/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateStochasticProcess.m File Reference . . . . .	99
8.37.1	Detailed Description . . . . .	99
8.38	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/BlindKriging.m File Reference . . . . .	100
8.38.1	Detailed Description . . . . .	100
8.39	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/polynomialCoding.m File Reference . . . . .	101
8.39.1	Detailed Description . . . . .	101
8.40	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/posteriorBeta.m File Reference . . . . .	101
8.40.1	Detailed Description . . . . .	101
8.41	/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/Rmatrix.m File Reference . . . . .	102
8.41.1	Detailed Description . . . . .	102
8.42	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/CoKriging.m File Reference . . . . .	103
8.42.1	Detailed Description . . . . .	103
8.43	/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/setData.m File Reference . . . . .	103
8.43.1	Detailed Description . . . . .	103
8.44	/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/setData.m File Reference . . . . .	104
8.44.1	Detailed Description . . . . .	104
8.45	/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/Kriging.m File Reference . . . . .	105
8.45.1	Detailed Description . . . . .	105
8.46	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrcubic.m File Reference . . . . .	105
8.46.1	Detailed Description . . . . .	105
8.46.2	Function Documentation . . . . .	106

8.47	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/correx.m File Reference	106
8.47.1	Detailed Description	106
8.47.2	Function Documentation	107
8.48	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrgauss.m File Reference	107
8.48.1	Detailed Description	107
8.48.2	Function Documentation	108
8.49	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrgaussp.m File Reference	108
8.49.1	Detailed Description	108
8.49.2	Function Documentation	108
8.50	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrlin.m File Reference	109
8.50.1	Detailed Description	109
8.50.2	Function Documentation	109
8.51	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32.m File Reference	109
8.51.1	Detailed Description	110
8.51.2	Function Documentation	110
8.52	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32_iso.m File Reference	110
8.52.1	Detailed Description	110
8.52.2	Function Documentation	111
8.53	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32_variance.m File Reference	111
8.53.1	Detailed Description	111
8.53.2	Function Documentation	112
8.54	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m File Reference	112
8.54.1	Detailed Description	112
8.54.2	Function Documentation	113
8.55	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern52.m File Reference	113
8.55.1	Detailed Description	113
8.55.2	Function Documentation	114
8.56	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrspherical.m File Reference	114
8.56.1	Detailed Description	114
8.56.2	Function Documentation	115
8.57	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrspline.m File Reference	115
8.57.1	Detailed Description	115
8.57.2	Function Documentation	116
8.58	/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/covmatern32.m File Reference	116
8.58.1	Detailed Description	116
8.58.2	Function Documentation	117
8.59	/home/ilm/projecten/meta/scripts/ooDACE/dacefit.m File Reference	117
8.59.1	Detailed Description	117
8.59.2	Function Documentation	117
8.60	/home/ilm/projecten/meta/scripts/ooDACE/datasets/generateDatasets.m File Reference	118

8.60.1 Detailed Description . . . . .	118
8.60.2 Function Documentation . . . . .	119
8.61 /home/ilm/projecten/meta/scripts/ooDACE/demo.m File Reference . . . . .	119
8.61.1 Detailed Description . . . . .	119
8.61.2 Function Documentation . . . . .	119
8.62 /home/ilm/projecten/meta/scripts/ooDACE/doc/mainpage.m File Reference . . . . .	120
8.63 /home/ilm/projecten/meta/scripts/ooDACE/oodacefit.m File Reference . . . . .	120
8.63.1 Detailed Description . . . . .	120
8.63.2 Function Documentation . . . . .	121
8.64 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/getPopulationSize.m File Reference . . . . .	121
8.64.1 Detailed Description . . . . .	121
8.65 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getPopulationSize.m File Reference . . . . .	122
8.65.1 Detailed Description . . . . .	122
8.66 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/MatlabGA.m File Reference . . . . .	122
8.66.1 Detailed Description . . . . .	122
8.67 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/optimize.m File Reference . . . . .	123
8.67.1 Detailed Description . . . . .	123
8.68 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/optimize.m File Reference . . . . .	124
8.68.1 Detailed Description . . . . .	124
8.69 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/optimize.m File Reference . . . . .	124
8.69.1 Detailed Description . . . . .	124
8.70 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQLabOptimizer/optimize.m File Reference . . . . .	125
8.70.1 Detailed Description . . . . .	125
8.71 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/setInputConstraints.m File Reference . . . . .	126
8.71.1 Detailed Description . . . . .	126
8.72 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/setInputConstraints.m File Reference . . . . .	126
8.72.1 Detailed Description . . . . .	126
8.73 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInputConstraints.m File Reference . . . . .	127
8.73.1 Detailed Description . . . . .	127
8.74 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/MatlabOptimizer.m File Reference . . . . .	127
8.74.1 Detailed Description . . . . .	128
8.75 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getBounds.m File Reference . . . . .	128
8.75.1 Detailed Description . . . . .	128
8.76 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getHint.m File Reference . . . . .	129
8.76.1 Detailed Description . . . . .	129

8.77	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInitialPopulation.m File Reference	129
8.77.1	Detailed Description	129
8.78	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInputDimension.m File Reference	130
8.78.1	Detailed Description	130
8.79	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getOutputDimension.m File Reference	131
8.79.1	Detailed Description	131
8.80	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/Optimizer.m File Reference	131
8.80.1	Detailed Description	131
8.81	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setBounds.m File Reference	132
8.81.1	Detailed Description	132
8.82	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setDimensions.m File Reference	133
8.82.1	Detailed Description	133
8.83	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setHint.m File Reference	133
8.83.1	Detailed Description	133
8.84	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInitialPopulation.m File Reference	134
8.84.1	Detailed Description	134
8.85	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/simulator.m File Reference	134
8.85.1	Detailed Description	134
8.86	/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/SQPLabOptimizer.m File Reference	135
8.86.1	Detailed Description	135
8.87	/home/ilm/projecten/meta/scripts/ooDACE/plotKrigingModel.m File Reference	136
8.87.1	Detailed Description	136
8.87.2	Function Documentation	136
8.88	/home/ilm/projecten/meta/scripts/ooDACE/predictor.m File Reference	137
8.88.1	Detailed Description	137
8.88.2	Function Documentation	137
8.89	/home/ilm/projecten/meta/scripts/ooDACE/runBlindKrigingExamples.m File Reference	138
8.89.1	Detailed Description	138
8.89.2	Function Documentation	138
8.90	/home/ilm/projecten/meta/scripts/ooDACE/runRegressionTests.m File Reference	138
8.90.1	Detailed Description	139
8.90.2	Function Documentation	139
8.91	/home/ilm/projecten/meta/scripts/ooDACE/startup.m File Reference	139
8.91.1	Detailed Description	140
8.91.2	Function Documentation	140
8.92	/home/ilm/projecten/meta/scripts/ooDACE/tools/averageEuclideanError.m File Reference	140



8.92.1 Detailed Description . . . . .	141
8.92.2 Function Documentation . . . . .	141
8.93 /home/ilm/projecten/meta/scripts/ooDACE/tools/buildVandermondeMatrix.m File Reference . . . . .	141
8.93.1 Detailed Description . . . . .	141
8.93.2 Function Documentation . . . . .	142
8.94 /home/ilm/projecten/meta/scripts/ooDACE/tools/cfix.m File Reference . . . . .	142
8.94.1 Detailed Description . . . . .	142
8.94.2 Function Documentation . . . . .	143
8.95 /home/ilm/projecten/meta/scripts/ooDACE/tools/makeEvalGrid.m File Reference . . . . .	143
8.95.1 Detailed Description . . . . .	143
8.95.2 Function Documentation . . . . .	144
8.96 /home/ilm/projecten/meta/scripts/ooDACE/tools/makeGrid.m File Reference . . . . .	144
8.96.1 Detailed Description . . . . .	144
8.96.2 Function Documentation . . . . .	145
8.97 /home/ilm/projecten/meta/scripts/ooDACE/tools/mergeStruct.m File Reference . . . . .	145
8.97.1 Detailed Description . . . . .	145
8.97.2 Function Documentation . . . . .	145
8.98 /home/ilm/projecten/meta/scripts/ooDACE/tools/plotScatteredData.m File Reference . . . . .	146
8.98.1 Detailed Description . . . . .	146
8.98.2 Function Documentation . . . . .	146
8.99 /home/ilm/projecten/meta/scripts/ooDACE/tools/powerBase.m File Reference . . . . .	147
8.99.1 Detailed Description . . . . .	147
8.99.2 Function Documentation . . . . .	147

## Index

147

## 1 The ooDACE toolbox documentation

### 1.1 Introduction

The **ooDACE** toolbox is a versatile Matlab toolbox that implements the popular Gaussian Process based kriging surrogate models. [Kriging](#) is in particular popular for approximating (and optimizing) deterministic computer experiments. Given a dataset the toolbox automatically fits a kriging surrogate model to it. Afterwards the kriging surrogate can be fully exploited instead of the (probably more expensive) simulation code.

The toolbox is aimed for solving complex applications (expensive simulation codes, physical experiments, ...) and for researching new kriging extensions and techniques.

### 1.2 Download

See [download page](#)

### 1.3 Quick start guide

**Note**

Before the toolbox can be used you have to include the toolbox in Matlab's search path. You can do this manually by running `startup`, or, if Matlab is started in the root toolbox directory, then `startup` will be run automatically.

`startup`

Now the toolbox is ready to be used. The ooDACE toolbox is designed in an object oriented (OO) fashion. It is strongly recommended to exploit the OO design directly, i.e., use the [Kriging](#) and [Optimizer](#) matlab classes. However, for convenience wrapper scripts (`dacefit`, `predictor`) are provided that emulate the DACE toolbox interface (see wrapper scripts for more information).

Lets define **n** as the number of observations and **d** as the number of input parameters. Then the n-by-d input sample matrix is denoted by **samples** (each row is one observation) and the corresponding output values are stored in the n-by-1 matrix **values**.

The **ooDACE** toolbox provides a script, `oodacefit.m`, that just takes your dataset (a **samples** and **values** matrix) and returns a fitted kriging object, all other parameters are set to some sensible defaults. For instance,

```
% ordinary kriging
k = ooDACEfit( samples, values );
y = k.predict(x);

% or, regression kriging
opts.lambda0 = 0;
opts.lambdaBounds = [-5 ; 5]; % log scale
k = ooDACEfit( samples, values, opts );
y = k.predict(x);
```

For more flexibility the user can utilize the kriging classes directly. **lb** and **ub** are 1-by-d arrays defining the lower bounds and upper bounds, respectively, needed to optimize the **hyperparameters**. In addition, a set of starting values has to be specified, namely, **hyperparameters0** is also an 1-by-d array. Example code to create a kriging model follows:

```
...
% Generate kriging options structure
opts = Kriging.getDefaultOptions();
opts.hpBounds = [lb ; ub]; % hyperparameter optimization bounds

% configure the optimization algorithm (only one optimizer is included)
% the Matlab Optimization toolbox is REQUIRED
optimopts.GradObj = 'on';
optimopts.DerivativeCheck = 'off';
optimopts.Diagnostics = 'off';
optimopts.Algorithm = 'active-set';
opts.hpOptimizer = MatlabOptimizer( dim, 1, optimopts );

% create and fit the kriging model
k = Kriging( opts, hyperparameters0, 'regpoly0', @corrGauss );
k = k.fit( samples, values );

% k represents the approximation and can now be used, e.g.,
[y mse] = k.predict( [1 2] )
...
```

See the included `demo.m` and `oodacefit.m` scripts for more example code on how to use the **ooDACE** toolbox (including more advanced features such as using blind kriging ([BlindKriging](#)) or how to use regression instead of interpolation). For more information on the classes and their methods please refer to the doxygen documentation and the source files.

## 1.4 DACE toolbox interface

The **ooDACE** toolbox provides two scripts `dacefit.m` and `predictor.m` that emulate the behavior of the **D-ACE** toolbox ([1]). Note, that full compatibility between ooDACE and the DACE toolbox is not provided. The scripts merely aim to ease the transition from the DACE toolbox to the ooDACE toolbox.

Example code:

```
krige = dacefit(samples, values, 'regpoly0', 'corrgauss',
               hyperparameters0, lb, ub )
y = predictor([1 2], krige)
```

Obviously, a lot less code is used to copy the setup described above. However, less code means less flexibility (e.g., blind kriging and regression kriging are not available using the wrapper scripts). Hence, it is suggested to learn the object oriented interface of **ooDACE** and use it instead.

## 1.5 Contribute

Suggestions on how to improve the ooDACE toolbox are always welcome. For more information please see the [feedback page](#).

## 2 Test List

### Member **demo** (var id)

Test case 1: Fits an ordinary kriging model on the branin function Covers: ordinary kriging interpolation, marginalLikelihood

Test case 2: Fits an ordinary kriging model on the bird function Covers: ordinary kriging regression, re-interpolation of variance, pseudoLikelihood, likelihood debug plot

Test case 3: Fits a blind kriging model on the branin function. Covers: blind kriging

Test case 4: Fits a cokriging model on a mathematical 1D function. Covers: cokriging

Test case 5: Fits a stochastic kriging model on the branin function plus some stochastic noise. Covers: stochastic kriging, sigma2 optimization

## 3 Todo List

### Class **BasicGaussianProcess**

Refactor correlation functions into proper basis function class hierarchy.

solve the correlation matrix vs covariance matrix issue

### Member **BasicGaussianProcess::imse** ()

Implement generic monte carlo integration

### Member **BasicGaussianProcess::marginalLikelihood** (var dps, var dsigma2)

Adjoint derivatives work, but are very slow due to naive implementation

### Member **BasicGaussianProcess::updateRegression** (var F, var hp)

Rho is only used by co-kriging, can we abstract this somehow ?

### Class **CoKriging**

Generalize to an arbitrary number of (multi-fidelity) datasets

### Member **Kriging::getDefaultOptions** ()

Apparently Matlab 2008b doesn't auto forward the (static) call to **BasicGaussianProcess::getDefaultOptions()**

## 4 Hierarchical Index

### 4.1 Class Hierarchy

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

**BasicGaussianProcess**

7

CoKriging	35
Kriging	49
BlindKriging	21
Optimizer	69
MatlabGA	62
MatlabOptimizer	65
SQPLabOptimizer	74

## 5 Class Index

### 5.1 Class List

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

<b>BasicGaussianProcess</b>	
A kriging surrogate model (also known as a Gaussian Process)	7
<b>BlindKriging</b>	
A blind kriging surrogate model	21
<b>CoKriging</b>	
A cokriging surrogate model	35
<b>Kriging</b>	
A kriging surrogate model	49
<b>MatlabGA</b>	
Wrapper around the matlab optimizers	62
<b>MatlabOptimizer</b>	
Wrapper around the matlab optimizers	65
<b>Optimizer</b>	
Abstract base class for an optimizer	69
<b>SQPLabOptimizer</b>	
Wrapper around the SQPLab optimization package	74

## 6 File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

/home/ilm/projecten/meta/scripts/ooDACE/dacefit.m	117
/home/ilm/projecten/meta/scripts/ooDACE/demo.m	119
/home/ilm/projecten/meta/scripts/ooDACE/oodacefit.m	120
/home/ilm/projecten/meta/scripts/ooDACE/plotKrigingModel.m	136

/home/ilm/projecten/meta/scripts/ooDACE/ <a href="#">predictor.m</a>	137
/home/ilm/projecten/meta/scripts/ooDACE/ <a href="#">runBlindKrigingExamples.m</a>	138
/home/ilm/projecten/meta/scripts/ooDACE/ <a href="#">runRegressionTests.m</a>	138
/home/ilm/projecten/meta/scripts/ooDACE/ <a href="#">startup.m</a>	139
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">BasicGaussianProcess.m</a>	78
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">correlationFunction.m</a>	78
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">cvpe.m</a>	80
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">extrinsicCorrelationMatrix.m</a>	81
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">fit.m</a>	82
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">generateDegrees.m</a>	84
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">getExpression.m</a>	84
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">imse.m</a>	86
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">intrinsicCovarianceMatrix.m</a>	86
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">likelihood.m</a>	87
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">marginalLikelihood.m</a>	88
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">mseTestset.m</a>	89
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">plotLikelihood.m</a>	89
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">plotVariogram.m</a>	90
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">predict.m</a>	90
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">predict_derivatives.m</a>	92
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">pseudoLikelihood.m</a>	93
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">rcValues.m</a>	93
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">regressionFunction.m</a>	94
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">regressionMatrix.m</a>	96
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">tuneParameters.m</a>	98
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">updateModel.m</a>	98
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">updateRegression.m</a>	99
/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/ <a href="#">updateStochasticProcess.m</a>	99
/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">BlindKriging.m</a>	100
/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">fit.m</a>	83
/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">polynomialCoding.m</a>	101
/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">posteriorBeta.m</a>	101

/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">regressionFunction.m</a>	95
/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">regressionMatrix.m</a>	96
/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/ <a href="#">Rmatrix.m</a>	102
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">CoKriging.m</a>	103
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">correlationFunction.m</a>	79
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">extrinsicCorrelationMatrix.m</a>	81
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">fit.m</a>	83
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">intrinsicCovarianceMatrix.m</a>	87
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">regressionFunction.m</a>	95
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">regressionMatrix.m</a>	97
/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/ <a href="#">setData.m</a>	103
/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/ <a href="#">cvpe.m</a>	80
/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/ <a href="#">getExpression.m</a>	85
/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/ <a href="#">Kriging.m</a>	105
/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/ <a href="#">predict.m</a>	91
/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/ <a href="#">predict_derivatives.m</a>	92
/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/ <a href="#">setData.m</a>	104
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrccubic.m</a>	105
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">correxpm.m</a>	106
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrgauss.m</a>	107
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrgaussp.m</a>	108
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrllin.m</a>	109
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrmatern32.m</a>	109
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrmatern32_iso.m</a>	110
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrmatern32_variance.m</a>	111
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrmatern32iso.m</a>	112
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrmatern52.m</a>	113
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrnspherical.m</a>	114
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">corrnspline.m</a>	115
/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/ <a href="#">covmatern32.m</a>	116
/home/ilm/projecten/meta/scripts/ooDACE/datasets/ <a href="#">generateDatasets.m</a>	118
/home/ilm/projecten/meta/scripts/ooDACE/doc/ <a href="#">mainpage.m</a>	120

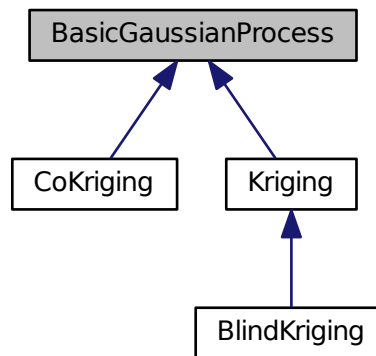
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/getPopulationSize.m	121
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/MatlabGA.m	122
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/optimize.m	123
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/setInputConstraints.m	126
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/MatlabOptimizer.m	127
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/optimize.m	124
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/setInputConstraints.m	126
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getBounds.m	128
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getHint.m	129
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInitialPopulation.m	129
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInputDimension.m	130
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getOutputDimension.m	131
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getPopulationSize.m	122
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/optimize.m	124
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/Optimizer.m	131
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setBounds.m	132
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setDimensions.m	133
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setHint.m	133
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInitialPopulation.m	134
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInputConstraints.m	127
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/optimize.m	125
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/simulator.m	134
/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/SQPLabOptimizer.m	135
/home/ilm/projecten/meta/scripts/ooDACE/tools/averageEuclideanError.m	140
/home/ilm/projecten/meta/scripts/ooDACE/tools/buildVandermondeMatrix.m	141
/home/ilm/projecten/meta/scripts/ooDACE/tools/cfix.m	142
/home/ilm/projecten/meta/scripts/ooDACE/tools/makeEvalGrid.m	143
/home/ilm/projecten/meta/scripts/ooDACE/tools/makeGrid.m	144
/home/ilm/projecten/meta/scripts/ooDACE/tools/mergeStruct.m	145
/home/ilm/projecten/meta/scripts/ooDACE/tools/plotScatteredData.m	146
/home/ilm/projecten/meta/scripts/ooDACE/tools/powerBase.m	147

## 7 Class Documentation

### 7.1 BasicGaussianProcess Class Reference

A kriging surrogate model (also known as a Gaussian Process)

Inheritance diagram for BasicGaussianProcess:



#### Public Member Functions

- function [BasicGaussianProcess](#) (var varargin)  
*Class constructor.*
- function [getHyperparameters](#) ()  
*Returns the hyperparameters.*
- function [getRho](#) ()  
*Returns the rho parameter (only valid for [CoKriging](#))*
- function [getProcessVariance](#) ()  
*Returns the process variance (sigma2)*
- function [getCorrelationMatrix](#) ()  
*Returns the full extrinsic correlation matrix.*
- function [getSigma](#) ()  
*Returns the intrinsic covariance matrix.*
- function [getRegressionMatrix](#) ()  
*Returns the model matrix (Vandermonde matrix)*
- function [getSamples](#) ()  
*Returns the input sample matrix.*
- function [getValues](#) ()  
*Returns the output value matrix.*
- function [setOption](#) (var key, var value)  
*Sets a value in the options structure.*
- function [cleanup](#) ()  
*Clears some unused variables.*
- function [display](#) ()  
*Returns user-friendly description of the class instance.*



- function [fit](#) (var samples, var values)  
*Fits a gaussian process for a dataset.*
- function [predict](#) (var points)  
*Predict the mean and/or variance for one or more points x.*
- function [predict\\_derivatives](#) (var point)  
*Predict the derivatives of the mean and/or variance for a points.*
- function [predict\\_limit](#) (var points)  
*Limit predictor of kriging (EXPERIMENTAL)*
- function [regressionFunction](#) (var options)  
*Returns the regression function.*
- function [correlationFunction](#) (var options)  
*Returns the internal correlation function handle and a symbolic expression of the the correlation part.*
- function [getExpression](#) (var options)  
*Returns the Matlab expression of this Gaussian Process model.*
- function [cvpe](#) ()  
*Calculates the log of the leave-one-out cross validation error (LOO-CV)*
- function [marginalLikelihood](#) (var dpsi, var dsigma2)  
*Marginal likelihood function.*
- function [pseudoLikelihood](#) (var dpsi, var dsigma2)  
*Leave-one-out predictive log probability (pseudo-likelihood)*
- function [mseTestset](#) (var testx, var testy)  
*Calculates error on holdout set.*
- function [imse](#) ()  
*Calculates the log of the integrated mean squared error.*
- function [rcValues](#) ()  
*Quantifies magnification of noise (lower is better)*
- function [plotVariogram](#) ()  
*Variogram plot (EXPERIMENTAL)*

#### Static Public Member Functions

- static function [getDefaultOptions](#) ()  
*Returns a default options structure.*

#### Public Attributes

- const var [RHO](#)  
*index of the rho parameter*
- const var [LAMBDA](#)  
*index of the lambda parameter*
- const var [SIGMA2](#)  
*index of the sigma2 parameter*
- const var [HP](#)  
*index of the correlation function parameters*

## Protected Member Functions

- function [setData](#) (var samples, var values)  
*Sets samples and values matrix.*
- function [updateModel](#) (var F, var hp)  
*Constructs model.*
- function [updateRegression](#) (var F, var hp)  
*Constructs regression part.*
- function [updateStochasticProcess](#) (var hp)  
*Constructs correlation part.*
- function [extrinsicCorrelationMatrix](#) (var points1, var points2)  
*Constructs extrinsic correlation matrix.*
- function [intrinsicCovarianceMatrix](#) (var points1, var points2)  
*Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)*
- function [regressionMatrix](#) (var points)  
*Constructs regression matrix.*
- function [tuneParameters](#) (var F)  
*Hyperparameter optimization.*
- function [generateDegrees](#) (var dj)  
*Generate degrees matrix from individual ones.*

## Protected Attributes

- var [options](#)
- var [regressionFcn](#)  
*degrees matrix (strings are converted)*
- var [correlationFcn](#)  
*string -> function handle*
- var [hyperparameters0](#)  
*initial hp values OR the real ones (if optimization is done outside the class)*
- var [dist](#)  
*sample inter-distance*
- var [distIdxPsi](#)  
*indexing needed to calculate psi from D*
- var [optimIdx](#)  
*logical indices to parameters that are optimized*
- var [optimNrParameters](#)  
*number of optimization parameter (vector; one entry per type of parameter)*
- var [alpha](#)  
*Regression coefficients.*
- var [gamma](#)  
*'Correlation part' coefficients*
- var [hyperparameters](#)  
*correlation parameters*
- var [rho](#)
- var [C](#)  
*Choleski Decomposition of extrinsic + intrinsic matrices.*
- var [sigma2](#)  
*process variance of the GP (extrinsic variance )*
- var [tau2](#)  
*intrinsic variance*

- var [Sigma](#)  
*intrinsic covariance matrix (amount of regression of stochastic part)*
- var [Ft](#)  
*decorrelated model matrix*
- var [R](#)  
*from QR decomposition of regression part*
- var [sigma2\\_reinterp](#)  
*Reinterpolation version of `sigma2`.*
- var [C\\_reinterp](#)  
*Reinterpolation version of `C`.*
- var [Ft\\_reinterp](#)  
*Reinterpolation version of `Ft`.*
- var [R\\_reinterp](#)  
*Reinterpolation version of `R`.*

### 7.1.1 Detailed Description

A kriging surrogate model (also known as a Gaussian Process)

Papers:

- "Design and Analysis of Computer Experiments", J. Sacks, W. Welch, T. Mitchell, H. Wynn, 1989
- "Design and Analysis of Simulation Experiments", J.P.C. Kleijnen, Springer, 2008
- "Engineering Design Via Surrogate Modelling: A Practical Guide", A. Forrester and A. Sobester and A. Keane, Wiley, 2008
- "Gaussian Processes for Machine Learning", C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

**Todo** Refactor correlation functions into proper basis function class hierarchy.  
solve the correlation matrix vs covariance matrix issue

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 function `BasicGaussianProcess ( var varargin )` `[inline]`

Class constructor.

Parameters

<i>options</i>	Options structure
<i>hyperparameters0</i>	Initial theta values
<i>regressionFcn</i>	The type of trend function to use ('regpoly0', 'regpoly1', ...)
<i>correlationFcn</i>	Function handle to the correlation function (@corrGauss, ...)

Returns

instance of the basicGaussianProcess class.

### 7.1.3 Member Function Documentation

#### 7.1.3.1 function `cleanup ( )` `[inline]`

Clears some unused variables.

7.1.3.2 function correlationFunction ( var *options* )

Returns the internal correlation function handle and a symbolic expression of the the correlation part.

Example: [correlationFcn expression] = correlationFunction( this,struct( 'latex', true, 'includeCoefficients', false) )

The expression is based on the scaled data

## Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>correlationFcn</i>	String of correlation function
<i>expression</i>	Symbolic expression

## 7.1.3.3 function cvpe ( )

Calculates the log of the leave-one-out cross validation error (LOO-CV)

Error function used is the Mean Squared Error (MSE)

## Return values

<i>out</i>	log(looscv) score
------------	-------------------

## Papers:

- "Blind Kriging: A New Method for Developing Metamodels", V.R. Joseph and Y. Hung and A. Sudjianto, ASME Journal of Mechanical Design, 2008
- "Predictive Approaches for Choosing Hyperparameters in Gaussian Process", S. Sundararajan, S.S. Keerthu, 1999

7.1.3.4 function display ( ) [*inline*]

Returns user-friendly description of the class instance.

7.1.3.5 function extrinsicCorrelationMatrix ( var *points1*, var *points2* ) [*protected*]

Constructs extrinsic correlation matrix.

Generate correlation matrix of points1 vs points2 using the current hyperparameters as followed:

## Parameters

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

## Return values

<i>psi</i>	Correlation matrix
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

- extrinsicCorrelationMatrix(this)
  - Assume points1 = points2 = samples
- extrinsicCorrelationMatrix(this, points1)
  - Assume points2 = samples

- `extrinsicCorrelationMatrix(this, points1, points2)`

NOTE: The first case returns derivatives w.r.t. X The latter two cases returns derivatives w.r.t. the hyperparameters

#### 7.1.3.6 function fit ( var samples, var values )

Fits a gaussian process for a dataset.

Need to be invoked before calling any of the prediction methods.

##### Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix

##### Note

[Kriging](#) can't do a cleanup automatically, if needed `cleanup()` can be called manually.

#### 7.1.3.7 function generateDegrees ( var dj ) [protected]

Generate degrees matrix from individual ones.

Generates a (full) degree matrix based on one or more degree matrices for variable j The option 'regressionMaxLevelInteractions' determines the maximum level of interactions ([BasicGaussianProcess::getDefaultOptions](#))

##### Parameters

<i>dj</i>	Cell array of degree matrices (for each dimension)
-----------	--

##### Return values

<i>degrees</i>	Full degrees matrix
<i>idx</i>	Cell array of indices (used by blind kriging)

#### 7.1.3.8 function getCorrelationMatrix ( ) [inline]

Returns the full extrinsic correlation matrix.

#### 7.1.3.9 static function getDefaultOptions ( ) [inline],[static]

Returns a default options structure.

Available options:

```
options = struct( ...
    'generateHyperparameters0', false, ...
    'hpBounds', [], ... % hyperparameter bounds
    'hpOptimizer', [], ... % optimizer class
    'hpLikelihood', @marginalLikelihood, ...
    'sigma20', NaN, ... % initial value for sigma2
    'sigma2Bounds', [0.001 ; 10], ... % sigma2 parameter bounds
    'lambda0', -Inf, ... % initial lambda values
    'lambdaBounds', [-10 ; 0], ... % lambda parameter bounds (in log scale)
    'Sigma', [], ... % intrinsic covariance matrix (stochastic kriging)
    'reinterpolation', false, ... % reinterpolate error (replaces standard
    error)
    'lowRankApproximation', false, ... % enable low rank approximation of
    correlation matrix
    'rankTol', 1e-12, ... % tolerance for lowRankApprox.
    'rankMax', Inf, ... % maximum rank to achieve for lowRankApprox.
    'regressionMaxLevelInteractions', 2, ... % consider maximal two-level
    interactions
    'debug', false, ... % enables debug plot of the likelihood function
... % Cokriging specific
    'rho0', -Inf, ... % initial scaling factor between datasets
    'rhoBounds', [1 ; 5] ... % scaling factor bounds
```

```
);
```

## Return values

<i>options</i>	Options structure
----------------	-------------------

7.1.3.10 function `getExpression ( var options )`

Returns the Matlab expression of this Gaussian Process model.

Example: `expression = getExpression( this )`

## Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>expression</i>	Symbolic expression
-------------------	---------------------

7.1.3.11 function `getHyperparameters ( ) [inline]`

Returns the hyperparameters.

7.1.3.12 function `getProcessVariance ( ) [inline]`

Returns the process variance (sigma2)

7.1.3.13 function `getRegressionMatrix ( ) [inline]`

Returns the model matrix (Vandermonde matrix)

7.1.3.14 function `getRho ( ) [inline]`

Returns the rho parameter (only valid for [CoKriging](#))

7.1.3.15 function `getSamples ( ) [inline]`

Returns the input sample matrix.

7.1.3.16 function `getSigma ( ) [inline]`

Returns the intrinsic covariance matrix.

7.1.3.17 function `getValues ( ) [inline]`

Returns the output value matrix.

7.1.3.18 function `imse ( )`

Calculates the log of the integrated mean squared error.

Only supported for 1D and 2D problems.

## Return values

<i>out</i>	log(imse) score
------------	-----------------

## Papers:

- See Sacks 1989

- See pechiny paper...

**Todo** Implement generic monte carlo integration

#### 7.1.3.19 function intrinsicCovarianceMatrix ( var *points1*, var *points2* ) [protected]

Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)

Generate covariance matrix matrix of points1 vs points2 using the current hyperparameters as followed:

##### Parameters

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

##### Return values

<i>psi</i>	Covariance matrix
<i>dpsi</i>	Derivative of covariance matrix w.r.t. the hyperparameters OR the input points

- intrinsicCovarianceMatrix(this)
  - Assume points1 = points2 = samples
- intrinsicCovarianceMatrix(this, points1)
  - Not used (and not implemented yet)
- intrinsicCovarianceMatrix(this, points1, points2)
  - Not used (and not implemented yet)

#### 7.1.3.20 function marginalLikelihood ( var *dpsi*, var *dsigma2* )

Marginal likelihood function.

Used for Maximum Likelihood Estimation (MLE)

##### Parameters

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

##### Return values

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

Papers:

- "Gaussian Processes for Machine Learning" (Chapter 5), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006
- "An adjoint for likelihood maximization" D.J.J. Toal, A.I.J. Forrester, N.W. Bressloff, A.J. Keane, C. Holden, Proc. of the Royal Society, 2009

**Todo** Adjoint derivatives work, but are very slow due to naive implementation

#### 7.1.3.21 function mseTestset ( var *testx*, var *testy* )

Calculates error on holdout set.

Error function used is the Mean Squared Error (MSE) function.

#### Parameters

<i>testx</i>	input samples of the test set
<i>testy</i>	output samples of the test set

#### Return values

<i>out</i>	mse error on test set
------------	-----------------------

#### 7.1.3.22 function plotVariogram ( )

Variogram plot (EXPERIMENTAL)

Plots the experimental (semi-)variogram (based on the data) as well as the theoretical kriging (semi-)variogram (defined by the correlation function).

#### Return values

<i>h</i>	Figure handle
----------	---------------

Empirical variogram:

- Methods of moments estimator Matheron (Cressie, 1993)
- More robust estimators: Cressie-Hawkins (1980), Genton (1998) In general: use median instead of mean

#### 7.1.3.23 function predict ( var points )

Predict the mean and/or variance for one or more points x.

#### Parameters

<i>points</i>	Matrix of input points to be predicted
---------------	--

#### Return values

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

#### Precondition

The kriging object should be fitted using a dataset ([BasicGaussianProcess::fit](#))

#### 7.1.3.24 function predict\_derivatives ( var point )

Predict the derivatives of the mean and/or variance for a points.

NOTE:

#### Parameters

<i>point</i>	input point to calculate the derivative of
--------------	--

#### Return values

<i>dvalues</i>	Derivative w.r.t. the output
<i>sigma2</i>	Derivative w.r.t. the output variance (optional)



- limited to one point at a time (x is a vector)

**Precondition**

The kriging object should be fitted using a dataset ([BasicGaussianProcess::fit](#))

**7.1.3.25 function predict\_limit ( var points )**

Limit predictor of kriging (EXPERIMENTAL)

**Parameters**

<i>points</i>	Matrix of input points to be predicted
---------------	--

**Return values**

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

**7.1.3.26 function pseudoLikelihood ( var dpsl, var dsigma2 )**

Leave-one-out predictive log probability (pseudo-likelihood)

Calculates the leave-one-out predictive log probability.

**Parameters**

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

**Return values**

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

**Papers:**

- "Predictive Approaches for Choosing Hyperparameters in Gaussian Process", S. Sundararajan, S.S. Keerthu, 1999

**7.1.3.27 function rcValues ( )**

Quantifies magnification of noise (lower is better)

Robustness-criterion (In theory useful only for ordinary kriging).

**Return values**

<i>rc</i>	robustness-criterion
-----------	----------------------

Returns a 2xn matrix:

- the first row contains the absolute robustness
- the second row contains the relative robustness Papers:
- "Kriging models that are robust w.r.t. simulation errors" A.Y.D. Siem, D. den Hertog (tech report)

**7.1.3.28 function regressionFunction ( var options )**

Returns the regression function.

Example: [regressionFcn expression terms] = regressionFunction( this, struct('latex', true, 'precision', '%.5g') )

The expression is based on the scaled data

#### Parameters

<i>options</i>	Options struct
----------------	----------------

#### Return values

<i>regressionFcn</i>	Degree matrix representing the regression function
<i>expression</i>	Symbolic expression
<i>terms</i>	Cell array of the individual terms

#### 7.1.3.29 function regressionMatrix ( var *points* ) [protected]

Constructs regression matrix.

Regression matrix (model matrix, Vandermonde matrix, ...) for a set of points Based on this.regressionFcn.

#### Parameters

<i>points</i>	input point matrix (optional)
---------------	-------------------------------

#### Return values

<i>F</i>	Model matrix
<i>dF</i>	Derivative of model matrix w.r.t. the hyperparameters OR the input points

#### 7.1.3.30 function setData ( var *samples*, var *values* ) [inline],[protected]

Sets samples and values matrix.

#### Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix

#### 7.1.3.31 function setOption ( var *key*, var *value* ) [inline]

Sets a value in the options structure.

#### Parameters

<i>key</i>	option name
<i>value</i>	option value

#### 7.1.3.32 function tuneParameters ( var *F* ) [protected]

Hyperparameter optimization.

Setups and invokes the optimizer.

#### Parameters

<i>F</i>	model matrix
----------	--------------

## Return values

<i>optimHp</i>	optimized hyperparameters
<i>perf</i>	Performance score (likelihood score)

7.1.3.33 function updateModel ( var *F*, var *hp* ) [protected]

Constructs model.

Full update of the model (regression + correlation part)

## Parameters

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

7.1.3.34 function updateRegression ( var *F*, var *hp* ) [protected]

Constructs regression part.

Updates regression part of the model.

## Parameters

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

## Return values

<i>err</i>	error string (if any)
------------	-----------------------

**Todo** Rho is only used by co-kriging, can we abstract this somehow ?

7.1.3.35 function updateStochasticProcess ( var *hp* ) [protected]

Constructs correlation part.

Updates correlation part of the model.

## Parameters

<i>hp</i>	hyperparameters
-----------	-----------------

## Return values

<i>err</i>	error string (if any)
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

## 7.1.4 Member Data Documentation

7.1.4.1 var *alpha* [protected]

Regression coefficients.

7.1.4.2 var *C* [protected]

Choleski Decomposition of extrinsic + intrinsic matrices.

**7.1.4.3 var C\_reinterp** [protected]

Reinterpolation version of C.

**7.1.4.4 var correlationFcn** [protected]

string -> function handle

**7.1.4.5 var dist** [protected]

sample inter-distance

**7.1.4.6 var distIdxPsi** [protected]

indexing needed to calculate psi from D

**7.1.4.7 var Ft** [protected]

decorrelated model matrix

**7.1.4.8 var Ft\_reinterp** [protected]

Reinterpolation version of Ft.

**7.1.4.9 var gamma** [protected]

'Correlation part' coefficients

**7.1.4.10 const var HP**

index of the correlation function parameters

**7.1.4.11 var hyperparameters** [protected]

correlation parameters

**7.1.4.12 var hyperparameters0** [protected]

initial hp values OR the real ones (if optimization is done outside the class)

**7.1.4.13 const var LAMBDA**

index of the lambda parameter

**7.1.4.14 var optimIdx** [protected]

logical indices to parameters that are optimized

**7.1.4.15 var optimNrParameters** [protected]

number of optimization parameter (vector; one entry per type of parameter)

**7.1.4.16 var options** [protected]**7.1.4.17 var R** [protected]

from QR decomposition of regression part

**7.1.4.18 var R\_reinterp** [protected]

Reinterpolation version of R.

7.1.4.19 `var regressionFcn` [protected]

degrees matrix (strings are converted)

7.1.4.20 `const var RHO`

index of the rho parameter

7.1.4.21 `var rho` [protected]7.1.4.22 `var Sigma` [protected]

intrinsic covariance matrix (amount of regression of stochastic part)

7.1.4.23 `const var SIGMA2`

index of the sigma2 parameter

7.1.4.24 `var sigma2` [protected]

process variance of the GP (extrinsic variance )

7.1.4.25 `var sigma2_reinterp` [protected]

Reinterpolation version of `sigma2`.

## Note

reinterpolation: might be nicer to just construct and keep a sub-GP... takes more space, some calculations are done twice but performance shouldn't take a very big hit...

7.1.4.26 `var tau2` [protected]

intrinsic variance

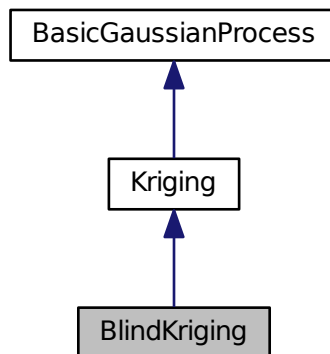
The documentation for this class was generated from the following files:

- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/BasicGaussianProcess.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/correlationFunction.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/cvpe.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/extrinsicCorrelationMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/fit.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/generateDegrees.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/getExpression.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/imse.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/intrinsicCovarianceMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/likelihood.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/marginalLikelihood.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/mseTestset.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/plotLikelihood.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/plotVariogram.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/predict.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/predict\\_derivatives.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/pseudoLikelihood.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/rcValues.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/regressionFunction.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/regressionMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/tuneParameters.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateModel.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateRegression.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateStochasticProcess.m](#)

## 7.2 BlindKriging Class Reference

A blind kriging surrogate model.

Inheritance diagram for BlindKriging:



### Public Member Functions

- function [BlindKriging](#) (var varargin)  
*Class constructor.*
- function [getStatistics](#) ()  
*Returns some useful statistics.*
- function [fit](#) (var samples, var values)  
*Fits a blind kriging model.*
- function [regressionFunction](#) (var varargin)  
*Returns the regression function.*
- function [getProcessVariance](#) ()  
*Returns the process variance.*
- function [getSamples](#) ()  
*Returns the input sample matrix.*
- function [getValues](#) ()  
*Returns the output value matrix.*
- function [predict](#) (var points)  
*Predict the mean and/or variance for one or more points x.*
- function [predict\\_derivatives](#) (var point)  
*Predict the derivatives of the mean and/or variance for a points x.*
- function [getExpression](#) (var outputIndex)  
*Returns the Matlab expression of this Gaussian Process model.*
- function [cvpe](#) ()  
*Calculates cross validated prediction error (cvpe)*
- function [getHyperparameters](#) ()  
*Returns the hyperparameters.*
- function [getRho](#) ()  
*Returns the rho parameter (only valid for [CoKriging](#))*

- function [getCorrelationMatrix](#) ()  
*Returns the full extrinsic correlation matrix.*
- function [getSigma](#) ()  
*Returns the intrinsic covariance matrix.*
- function [getRegressionMatrix](#) ()  
*Returns the model matrix (Vandermonde matrix)*
- function [setOption](#) (var key, var value)  
*Sets a value in the options structure.*
- function [cleanup](#) ()  
*Clears some unused variables.*
- function [display](#) ()  
*Returns user-friendly description of the class instance.*
- function [predict\\_limit](#) (var points)  
*Limit predictor of kriging (EXPERIMENTAL)*
- function [correlationFunction](#) (var options)  
*Returns the internal correlation function handle and a symbolic expression of the the correlation part.*
- function [marginalLikelihood](#) (var dps, var dsigma2)  
*Marginal likelihood function.*
- function [pseudoLikelihood](#) (var dps, var dsigma2)  
*Leave-one-out predictive log probability (pseudo-likelihood)*
- function [mseTestset](#) (var testx, var testy)  
*Calculates error on holdout set.*
- function [imse](#) ()  
*Calculates the log of the integrated mean squared error.*
- function [rcValues](#) ()  
*Quantifies magnification of noise (lower is better)*
- function [plotVariogram](#) ()  
*Variogram plot (EXPERIMENTAL)*

#### Static Public Member Functions

- static function [getDefaultOptions](#) ()  
*Returns a default options structure.*

#### Public Attributes

- const var [RHO](#)  
*index of the rho parameter*
- const var [LAMBDA](#)  
*index of the lambda parameter*
- const var [SIGMA2](#)  
*index of the sigma2 parameter*
- const var [HP](#)  
*index of the correlation function parameters*

## Protected Member Functions

- function `regressionMatrix` (var points)  
*Constructs regression matrix.*
- function `setData` (var samples, var values)  
*Sets samples and values matrix.*
- function `updateModel` (var F, var hp)  
*Constructs model.*
- function `updateRegression` (var F, var hp)  
*Constructs regression part.*
- function `updateStochasticProcess` (var hp)  
*Constructs correlation part.*
- function `extrinsicCorrelationMatrix` (var points1, var points2)  
*Constructs extrinsic correlation matrix.*
- function `intrinsicCovarianceMatrix` (var points1, var points2)  
*Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)*
- function `tuneParameters` (var F)  
*Hyperparameter optimization.*
- function `generateDegrees` (var dj)  
*Generate degrees matrix from individual ones.*

## Protected Attributes

- var `options`
- var `regressionFcn`  
*degrees matrix (strings are converted)*
- var `correlationFcn`  
*string -> function handle*
- var `hyperparameters0`  
*initial hp values OR the real ones (if optimization is done outside the class)*
- var `dist`  
*sample inter-distance*
- var `distIdxPsi`  
*indexing needed to calculate psi from D*
- var `optimIdx`  
*logical indices to parameters that are optimized*
- var `optimNrParameters`  
*number of optimization parameter (vector; one entry per type of parameter)*
- var `alpha`  
*Regression coefficients.*
- var `gamma`  
*'Correlation part' coefficients*
- var `hyperparameters`  
*correlation parameters*
- var `rho`
- var `C`  
*Choleski Decomposition of extrinsic + intrinsic matrices.*
- var `sigma2`  
*process variance of the GP (extrinsic variance )*
- var `tau2`  
*intrinsic variance*



- var [Sigma](#)  
*intrinsic covariance matrix (amount of regression of stochastic part)*
- var [Ft](#)  
*decorrelated model matrix*
- var [R](#)  
*from QR decomposition of regression part*
- var [sigma2\\_reinterp](#)  
*Reinterpolation version of `sigma2`.*
- var [C\\_reinterp](#)  
*Reinterpolation version of `C`.*
- var [Ft\\_reinterp](#)  
*Reinterpolation version of `Ft`.*
- var [R\\_reinterp](#)  
*Reinterpolation version of `R`.*

### 7.2.1 Detailed Description

A blind kriging surrogate model.

Papers:

- "Blind Kriging: A New Method for Developing Metamodels", V.R. Joseph and Y. Hung and A. Sudjianto, ASME Journal of Mechanical Design, 2008
- "Functionally Induced Priors for the Analysis of Experiments", V.R. Joseph and J.D. Delaney, Technometrics, 2007

Limitations:

- quartic is the highest order supported

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 function `BlindKriging ( var varargin )` `[inline]`

Class constructor.

Initializes the Blind kriging model. Takes the same parameters as [Kriging](#)

Returns

instance of the blind kriging class

### 7.2.3 Member Function Documentation

#### 7.2.3.1 function `cleanup ( )` `[inline]`, `[inherited]`

Clears some unused variables.

#### 7.2.3.2 function `correlationFunction ( var options )` `[inherited]`

Returns the internal correlation function handle and a symbolic expression of the the correlation part.

Example: `[correlationFcn expression] = correlationFunction( this,struct( 'latex', true, 'includeCoefficients', false) )`

The expression is based on the scaled data

## Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>correlationFcn</i>	String of correlation function
<i>expression</i>	Symbolic expression

7.2.3.3 function `cvpe ( )` [inherited]

Calculates cross validated prediction error (cvpe)

Error function used is the Mean Squared Error (MSE)

## Return values

<i>out</i>	cvpe score
------------	------------

7.2.3.4 function `display ( )` [inline],[inherited]

Returns user-friendly description of the class instance.

7.2.3.5 function `extrinsicCorrelationMatrix ( var points1, var points2 )` [protected],[inherited]

Constructs extrinsic correlation matrix.

Generate correlation matrix of points1 vs points2 using the current hyperparameters as followed:

## Parameters

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

## Return values

<i>psi</i>	Correlation matrix
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

- `extrinsicCorrelationMatrix(this)`
  - Assume points1 = points2 = samples
- `extrinsicCorrelationMatrix(this, points1)`
  - Assume points2 = samples
- `extrinsicCorrelationMatrix(this, points1, points2)`

NOTE: The first case returns derivatives w.r.t. X The latter two cases returns derivatives w.r.t. the hyperparameters

7.2.3.6 function `fit ( var samples, var values )`

Fits a blind kriging model.

Need to be invoked before calling any of the prediction methods.

## Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix

## Return values

<i>IK</i>	The initial kriging model
-----------	---------------------------

7.2.3.7 function generateDegrees ( var *dj* ) [protected],[inherited]

Generate degrees matrix from individual ones.

Generates a (full) degree matrix based on one or more degree matrices for variable *j*. The option 'regressionMaxLevelInteractions' determines the maximum level of interactions ([BasicGaussianProcess::getDefaultOptions](#))

## Parameters

<i>dj</i>	Cell array of degree matrices (for each dimension)
-----------	--

## Return values

<i>degrees</i>	Full degrees matrix
<i>idx</i>	Cell array of indices (used by blind kriging)

## 7.2.3.8 function getCorrelationMatrix ( ) [inline],[inherited]

Returns the full extrinsic correlation matrix.

## 7.2.3.9 static function getDefaultOptions ( ) [inline],[static]

Returns a default options structure.

Blind kriging specific options:

```
options.regressionMetric = 'cvpe'; % metric to guide the feature
    selection phase
options.retuneParameters = false; % retune parameters after every BK
    step
options.RmatrixThreshold = 250; % when to use full matrix R or when to
    approximate by using sparse diagonal of R
options.regressionMaxOrder = 2; % maximum order of candidate feature to
    consider (quadratic)
```

## Return values

<i>options</i>	Options structure
----------------	-------------------

7.2.3.10 function getExpression ( var *outputIndex* ) [inherited]

Returns the Matlab expression of this Gaussian Process model.

Example: expression = getExpression( this )

## Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>expression</i>	Symbolic expression
-------------------	---------------------

## 7.2.3.11 function getHyperparameters ( ) [inline],[inherited]

Returns the hyperparameters.

**7.2.3.12** `function getProcessVariance ( ) [inline],[inherited]`

Returns the process variance.

Return values

<i>sigma2</i>	process variance
---------------	------------------

**7.2.3.13** `function getRegressionMatrix ( ) [inline],[inherited]`

Returns the model matrix (Vandermonde matrix)

**7.2.3.14** `function getRho ( ) [inline],[inherited]`

Returns the rho parameter (only valid for [CoKriging](#))

**7.2.3.15** `function getSamples ( ) [inline],[inherited]`

Returns the input sample matrix.

Return values

<i>samples</i>	unscaled samples (original)
<i>scaledSamples</i>	scaled samples

**7.2.3.16** `function getSigma ( ) [inline],[inherited]`

Returns the intrinsic covariance matrix.

**7.2.3.17** `function getStatistics ( ) [inline]`

Returns some useful statistics.

Return values

<i>stats</i>	Statistics structure
--------------	----------------------

**7.2.3.18** `function getValues ( ) [inline],[inherited]`

Returns the output value matrix.

Return values

<i>values</i>	unscaled values (original)
<i>scaledValues</i>	scaled values

**7.2.3.19** `function imse ( ) [inherited]`

Calculates the log of the integrated mean squared error.

Only supported for 1D and 2D problems.

Return values

<i>out</i>	log(imse) score
------------	-----------------

Papers:

- See Sacks 1989

- See pechiny paper...

**Todo** Implement generic monte carlo integration

### 7.2.3.20 function intrinsicCovarianceMatrix ( var *points1*, var *points2* ) [protected],[inherited]

Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)

Generate covariance matrix matrix of points1 vs points2 using the current hyperparameters as followed:

#### Parameters

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

#### Return values

<i>psi</i>	Covariance matrix
<i>dpsi</i>	Derivative of covariance matrix w.r.t. the hyperparameters OR the input points

- intrinsicCovarianceMatrix(this)
  - Assume points1 = points2 = samples
- intrinsicCovarianceMatrix(this, points1)
  - Not used (and not implemented yet)
- intrinsicCovarianceMatrix(this, points1, points2)
  - Not used (and not implemented yet)

### 7.2.3.21 function marginalLikelihood ( var *dpsi*, var *dsigma2* ) [inherited]

Marginal likelihood function.

Used for Maximum Likelihood Estimation (MLE)

#### Parameters

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

#### Return values

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

Papers:

- "Gaussian Processes for Machine Learning" (Chapter 5), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006
- "An adjoint for likelihood maximization" D.J.J. Toal, A.I.J. Forrester, N.W. Bressloff, A.J. Keane, C. Holden, Proc. of the Royal Society, 2009

**Todo** Adjoint derivatives work, but are very slow due to naive implementation

### 7.2.3.22 function mseTestset ( var *testx*, var *testy* ) [inherited]

Calculates error on holdout set.

Error function used is the Mean Squared Error (MSE) function.

#### Parameters

<i>testx</i>	input samples of the test set
<i>testy</i>	output samples of the test set

#### Return values

<i>out</i>	mse error on test set
------------	-----------------------

#### 7.2.3.23 function plotVariogram ( ) [inherited]

Variogram plot (EXPERIMENTAL)

Plots the experimental (semi-)variogram (based on the data) as well as the theoretical kriging (semi-)variogram (defined by the correlation function).

#### Return values

<i>h</i>	Figure handle
----------	---------------

Empirical variogram:

- Methods of moments estimator Matheron (Cressie, 1993)
- More robust estimators: Cressie-Hawkins (1980), Genton (1998) In general: use median instead of mean

#### 7.2.3.24 function predict ( var points ) [inherited]

Predict the mean and/or variance for one or more points x.

#### Parameters

<i>points</i>	Matrix of input points to be predicted
---------------	--

#### Return values

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

#### Precondition

The kriging object should be fitted using a dataset (*fit*)

#### 7.2.3.25 function predict\_derivatives ( var point ) [inherited]

Predict the derivatives of the mean and/or variance for a points x.

NOTE:

#### Parameters

<i>point</i>	input point to calculate the derivative of
--------------	--

#### Return values

<i>dvalues</i>	Derivative w.r.t. the output
<i>sigma2</i>	Derivative w.r.t. the output variance (optional)

- limited to one point at a time (x is a row vector)

### 7.2.3.26 function predictLimit ( var points ) [inherited]

Limit predictor of kriging (EXPERIMENTAL)

#### Parameters

<i>points</i>	Matrix of input points to be predicted
---------------	--

#### Return values

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

### 7.2.3.27 function pseudoLikelihood ( var dpsi, var dsigma2 ) [inherited]

Leave-one-out predictive log probability (pseudo-likelihood)

Calculates the leave-one-out predictive log probability.

#### Parameters

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

#### Return values

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

Papers:

- "Predictive Approaches for Choosing Hyperparameters in Gaussian Process", S. Sundararajan, S.S. Keerthu, 1999

### 7.2.3.28 function rcValues ( ) [inherited]

Quantifies magnification of noise (lower is better)

Robustness-criterion (In theory useful only for ordinary kriging).

#### Return values

<i>rc</i>	robustness-criterion
-----------	----------------------

Returns a 2xn matrix:

- the first row contains the absolute robustness
- the second row contains the relative robustness Papers:
- "Kriging models that are robust w.r.t. simulation errors" A.Y.D. Siem, D. den Hertog (tech report)

### 7.2.3.29 function regressionFunction ( var varargin )

Returns the regression function.

Example: [regressionFcn expression terms] = regressionFunction( this, struct('latex', true, 'precision', '%.5g') )

## Parameters

<i>varargin</i>	Options
-----------------	---------

## Return values

<i>regressionFcn</i>	Degree matrix representing the regression function
<i>expression</i>	Symbolic expression
<i>terms</i>	Cell array of the individual terms

## Note

Symbolic expression does not support maxorder greater than two

7.2.3.30 function `regressionMatrix ( var points )` [protected]

Constructs regression matrix.

Regression matrix (model matrix, Vandermonde matrix, ...) for a set of points Based on this.`regressionFcn`.

## Parameters

<i>points</i>	points matrix (optional)
---------------	--------------------------

## Return values

<i>F</i>	model matrix
<i>dF</i>	derivative of the model matrix w.r.t. points or the hyperparameters

Uses coded sample matrix!

7.2.3.31 function `setData ( var samples, var values )` [protected],[inherited]

Sets samples and values matrix.

Scales samples and values, passing the scaled dataset to the underlying base method.

## Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix

7.2.3.32 function `setOption ( var key, var value )` [inline],[inherited]

Sets a value in the options structure.

## Parameters

<i>key</i>	option name
<i>value</i>	option value

7.2.3.33 function `tuneParameters ( var F )` [protected],[inherited]

Hyperparameter optimization.

Setups and invokes the optimizer.



## Parameters

<i>F</i>	model matrix
----------	--------------

## Return values

<i>optimHp</i>	optimized hyperparameters
<i>perf</i>	Performance score (likelihood score)

7.2.3.34 function updateModel ( var *F*, var *hp* ) [protected],[inherited]

Constructs model.

Full update of the model (regression + correlation part)

## Parameters

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

7.2.3.35 function updateRegression ( var *F*, var *hp* ) [protected],[inherited]

Constructs regression part.

Updates regression part of the model.

## Parameters

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

## Return values

<i>err</i>	error string (if any)
------------	-----------------------

**Todo** Rho is only used by co-kriging, can we abstract this somehow ?

7.2.3.36 function updateStochasticProcess ( var *hp* ) [protected],[inherited]

Constructs correlation part.

Updates correlation part of the model.

## Parameters

<i>hp</i>	hyperparameters
-----------	-----------------

## Return values

<i>err</i>	error string (if any)
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

## 7.2.4 Member Data Documentation

## 7.2.4.1 var alpha [protected],[inherited]

Regression coefficients.

**7.2.4.2** `var C` [protected],[inherited]

Choleski Decomposition of extrinsic + intrinsic matrices.

**7.2.4.3** `var C_reinterp` [protected],[inherited]

Reinterpolation version of C.

**7.2.4.4** `var correlationFcn` [protected],[inherited]

string -> function handle

**7.2.4.5** `var dist` [protected],[inherited]

sample inter-distance

**7.2.4.6** `var distIdxPsi` [protected],[inherited]

indexing needed to calculate psi from D

**7.2.4.7** `var Ft` [protected],[inherited]

decorrelated model matrix

**7.2.4.8** `var Ft_reinterp` [protected],[inherited]

Reinterpolation version of Ft.

**7.2.4.9** `var gamma` [protected],[inherited]

'Correlation part' coefficients

**7.2.4.10** `const var HP` [inherited]

index of the correlation function parameters

**7.2.4.11** `var hyperparameters` [protected],[inherited]

correlation parameters

**7.2.4.12** `var hyperparameters0` [protected],[inherited]

initial hp values OR the real ones (if optimization is done outside the class)

**7.2.4.13** `const var LAMBDA` [inherited]

index of the lambda parameter

**7.2.4.14** `var optimIdx` [protected],[inherited]

logical indices to parameters that are optimized

**7.2.4.15** `var optimNrParameters` [protected],[inherited]

number of optimization parameter (vector; one entry per type of parameter)

**7.2.4.16** `var options` [protected],[inherited]

**7.2.4.17** `var R` [protected],[inherited]

from QR decomposition of regression part

7.2.4.18 `var R_reinterp` [protected],[inherited]

Reinterpolation version of R.

7.2.4.19 `var regressionFcn` [protected],[inherited]

degrees matrix (strings are converted)

7.2.4.20 `const var RHO` [inherited]

index of the rho parameter

7.2.4.21 `var rho` [protected],[inherited]

7.2.4.22 `var Sigma` [protected],[inherited]

intrinsic covariance matrix (amount of regression of stochastic part)

7.2.4.23 `const var SIGMA2` [inherited]

index of the sigma2 parameter

7.2.4.24 `var sigma2` [protected],[inherited]

process variance of the GP (extrinsic variance )

7.2.4.25 `var sigma2_reinterp` [protected],[inherited]

Reinterpolation version of `sigma2`.

#### Note

reinterpolation: might be nicer to just construct and keep a sub-GP... takes more space, some calculations are done twice but performance shouldn't take a very big hit...

7.2.4.26 `var tau2` [protected],[inherited]

intrinsic variance

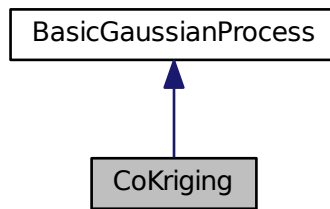
The documentation for this class was generated from the following files:

- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/BlindKriging.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/fit.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/polynomialCoding.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/posteriorBeta.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/regressionFunction.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/regressionMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/Rmatrix.m](#)

## 7.3 CoKriging Class Reference

A cokriging surrogate model.

Inheritance diagram for CoKriging:



#### Public Member Functions

- function [CoKriging](#) (var varargin)
- function [getSamplesIdx](#) (var t)  
*Returns samples of dataset t.*
- function [getValuesIdx](#) (var t)  
*Returns values of dataset t.*
- function [fit](#) (var samples, var values)  
*Fits a gaussian process for multi-fidelity datasets.*
- function [regressionFunction](#) (var varargin)  
*Returns the regression function.*
- function [correlationFunction](#) (var varargin)  
*Returns the internal correlation function handle.*
- function [getHyperparameters](#) ()  
*Returns the hyperparameters.*
- function [getRho](#) ()  
*Returns the rho parameter (only valid for [CoKriging](#))*
- function [getProcessVariance](#) ()  
*Returns the process variance (sigma2)*
- function [getCorrelationMatrix](#) ()  
*Returns the full extrinsic correlation matrix.*
- function [getSigma](#) ()  
*Returns the intrinsic covariance matrix.*
- function [getRegressionMatrix](#) ()  
*Returns the model matrix (Vandermonde matrix)*
- function [getSamples](#) ()  
*Returns the input sample matrix.*
- function [getValues](#) ()  
*Returns the output value matrix.*
- function [setOption](#) (var key, var value)  
*Sets a value in the options structure.*
- function [cleanup](#) ()  
*Clears some unused variables.*
- function [display](#) ()  
*Returns user-friendly description of the class instance.*
- function [predict](#) (var points)

- Predict the mean and/or variance for one or more points  $x$ .*
  - function [predict\\_derivatives](#) (var point)
    - Predict the derivatives of the mean and/or variance for a points.*
  - function [predict\\_limit](#) (var points)
    - Limit predictor of kriging (EXPERIMENTAL)*
  - function [getExpression](#) (var options)
    - Returns the Matlab expression of this Gaussian Process model.*
  - function [cvpe](#) ()
    - Calculates the log of the leave-one-out cross validation error (LOO-CV)*
  - function [marginalLikelihood](#) (var dps, var dsigma2)
    - Marginal likelihood function.*
  - function [pseudoLikelihood](#) (var dps, var dsigma2)
    - Leave-one-out predictive log probability (pseudo-likelihood)*
  - function [mseTestset](#) (var testx, var testy)
    - Calculates error on holdout set.*
  - function [imse](#) ()
    - Calculates the log of the integrated mean squared error.*
  - function [rcValues](#) ()
    - Quantifies magnification of noise (lower is better)*
  - function [plotVariogram](#) ()
    - Variogram plot (EXPERIMENTAL)*

#### Static Public Member Functions

- static function [getDefaultOptions](#) ()
  - Returns a default options structure.*

#### Public Attributes

- const var [RHO](#)
  - index of the rho parameter*
- const var [LAMBDA](#)
  - index of the lambda parameter*
- const var [SIGMA2](#)
  - index of the sigma2 parameter*
- const var [HP](#)
  - index of the correlation function parameters*

#### Protected Member Functions

- function [setData](#) (var samples, var values)
  - Sets samples and values matrix.*
- function [extrinsicCorrelationMatrix](#) (var points1, var points2)
  - Constructs extrinsic correlation matrix.*
- function [intrinsicCovarianceMatrix](#) (var points1, var points2)
  - Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)*
- function [regressionMatrix](#) (var points)
  - Constructs regression matrix.*
- function [updateModel](#) (var F, var hp)
  - Constructs model.*

- function `updateRegression` (var F, var hp)  
*Constructs regression part.*
- function `updateStochasticProcess` (var hp)  
*Constructs correlation part.*
- function `tuneParameters` (var F)  
*Hyperparameter optimization.*
- function `generateDegrees` (var dj)  
*Generate degrees matrix from individual ones.*

#### Protected Attributes

- var `options`
- var `regressionFcn`  
*degrees matrix (strings are converted)*
- var `correlationFcn`  
*string -> function handle*
- var `hyperparameters0`  
*initial hp values OR the real ones (if optimization is done outside the class)*
- var `dist`  
*sample inter-distance*
- var `distIdxPsi`  
*indexing needed to calculate psi from D*
- var `optimIdx`  
*logical indices to parameters that are optimized*
- var `optimNrParameters`  
*number of optimization parameter (vector; one entry per type of parameter)*
- var `alpha`  
*Regression coefficients.*
- var `gamma`  
*'Correlation part' coefficients*
- var `hyperparameters`  
*correlation parameters*
- var `rho`
- var `C`  
*Choleski Decomposition of extrinsic + intrinsic matrices.*
- var `sigma2`  
*process variance of the GP (extrinsic variance )*
- var `tau2`  
*intrinsic variance*
- var `Sigma`  
*intrinsic covariance matrix (amount of regression of stochastic part)*
- var `Ft`  
*decorrelated model matrix*
- var `R`  
*from QR decomposition of regression part*
- var `sigma2_reinterp`  
*Reinterpolation version of `sigma2`.*
- var `C_reinterp`  
*Reinterpolation version of `C`.*
- var `Ft_reinterp`  
*Reinterpolation version of `Ft`.*
- var `R_reinterp`  
*Reinterpolation version of `R`.*

### 7.3.1 Detailed Description

A cokriging surrogate model.

Papers:

- "Bayesian Analysis of Computer Code Outputs" M. Kennedy, A. O'Hagan, N. Higgins (2001)
- "Multi-fidelity optimization via surrogate modelling" A.I.J Forrester, A. Sobester, A.J. Keane (2007)

Limitations:

- Supports 2 fidelity datasets
- Likelihood function doesn't support gradients

**Todo** Generalize to an arbitrary number of (multi-fidelity) datasets

### 7.3.2 Constructor & Destructor Documentation

7.3.2.1 `function CoKriging ( var varargin ) [inline]`

### 7.3.3 Member Function Documentation

7.3.3.1 `function cleanup ( ) [inline],[inherited]`

Clears some unused variables.

7.3.3.2 `function correlationFunction ( var varargin )`

Returns the internal correlation function handle.

Symbolic expression not supported for cokriging.

Parameters

<i>varargin</i>	Options
-----------------	---------

Return values

<i>correlationFcn</i>	String of correlation function
-----------------------	--------------------------------

7.3.3.3 `function cvpe ( ) [inherited]`

Calculates the log of the leave-one-out cross validation error (LOO-CV)

Error function used is the Mean Squared Error (MSE)

Return values

<i>out</i>	log(loocv) score
------------	------------------

Papers:

- "Blind Kriging: A New Method for Developing Metamodels", V.R. Joseph and Y. Hung and A. Sudjianto, ASME Journal of Mechanical Design, 2008
- "Predictive Approaches for Choosing Hyperparameters in Gaussian Process", S. Sundararajan, S.S. Keerthu, 1999

**7.3.3.4** function `display ( )` [`inline`],[`inherited`]

Returns user-friendly description of the class instance.

**7.3.3.5** function `extrinsicCorrelationMatrix ( var points1, var points2 )` [`protected`]

Constructs extrinsic correlation matrix.

Generate correlation matrix of *points1* vs *points2* using the current hyperparameters as followed:

**Parameters**

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

**Return values**

<i>psi</i>	Correlation matrix
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

- `extrinsicCorrelationMatrix(this)`
  - Assume *points1* = *points2* = samples
- `extrinsicCorrelationMatrix(this, points1)`
  - Assume *points2* = samples
- `extrinsicCorrelationMatrix(this, points1, points2)`

NOTE: The first case returns derivatives w.r.t. X The latter two cases returns derivatives w.r.t. the hyperparameters

Issues:

- Actually returns a covariance matrix instead of a correlation matrix shouldn't be a problem as long as an `extrinsicCorrelationMatrix` method doesn't mix both types...

**7.3.3.6** function `fit ( var samples, var values )`

Fits a gaussian process for multi-fidelity datasets.

Need to be invoked before calling any of the prediction methods.

**Parameters**

<i>samples</i>	input sample matrix (cell array)
<i>values</i>	output value matrix (cell array)

*samples/values* are (columnwise!) cell arrays. {1} is cheap data ... more expensive ... {end} is most expensive  
NOTE: though only 2 datasets are supported atm (length of *samples/values* must be 2)

**Note**

Allow to update the [CoKriging](#) model with new data:

- this means no scaling as the scaling changes when new data arrives (we inherit from [BasicGaussian-Process](#))
- no refitting with the same hyperparameters (no *xval* calculation)

**7.3.3.7** function `generateDegrees ( var dj )` [`protected`],[`inherited`]

Generate degrees matrix from individual ones.



Generates a (full) degree matrix based on one or more degree matrices for variable *j*. The option 'regressionMaxLevelInteractions' determines the maximum level of interactions ([BasicGaussianProcess::getDefaultOptions](#))

#### Parameters

<i>dj</i>	Cell array of degree matrices (for each dimension)
-----------	--

#### Return values

<i>degrees</i>	Full degrees matrix
<i>idx</i>	Cell array of indices (used by blind kriging)

#### 7.3.3.8 function getCorrelationMatrix ( ) [inline],[inherited]

Returns the full extrinsic correlation matrix.

#### 7.3.3.9 static function getDefaultOptions ( ) [inline],[static]

Returns a default options structure.

#### Return values

<i>options</i>	Options structure
----------------	-------------------

#### 7.3.3.10 function getExpression ( var options ) [inherited]

Returns the Matlab expression of this Gaussian Process model.

Example: expression = getExpression( this )

#### Parameters

<i>options</i>	Options struct
----------------	----------------

#### Return values

<i>expression</i>	Symbolic expression
-------------------	---------------------

#### 7.3.3.11 function getHyperparameters ( ) [inline],[inherited]

Returns the hyperparameters.

#### 7.3.3.12 function getProcessVariance ( ) [inline],[inherited]

Returns the process variance (sigma2)

#### 7.3.3.13 function getRegressionMatrix ( ) [inline],[inherited]

Returns the model matrix (Vandermonde matrix)

#### 7.3.3.14 function getRho ( ) [inline],[inherited]

Returns the rho parameter (only valid for [CoKriging](#))

#### 7.3.3.15 function getSamples ( ) [inline],[inherited]

Returns the input sample matrix.

**7.3.3.16** `function getSamplesIdx ( var t ) [inline]`

Returns samples of dataset *t*.

**Parameters**

<i>t</i>	index of dataset to retrieve
----------	------------------------------

**Return values**

<i>samples</i>	samples of dataset <i>t</i>
----------------	-----------------------------

**7.3.3.17** `function getSigma ( ) [inline],[inherited]`

Returns the intrinsic covariance matrix.

**7.3.3.18** `function getValues ( ) [inline],[inherited]`

Returns the output value matrix.

**7.3.3.19** `function getValuesIdx ( var t ) [inline]`

Returns values of dataset *t*.

**Parameters**

<i>t</i>	index of dataset to retrieve
----------	------------------------------

**Return values**

<i>values</i>	values of dataset <i>t</i>
---------------	----------------------------

**7.3.3.20** `function imse ( ) [inherited]`

Calculates the log of the integrated mean squared error.

Only supported for 1D and 2D problems.

**Return values**

<i>out</i>	log(imse) score
------------	-----------------

**Papers:**

- See Sacks 1989
- See pechiny paper...

**Todo** Implement generic monte carlo integration

**7.3.3.21** `function intrinsicCovarianceMatrix ( var points1, var points2 ) [protected]`

Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)

Generate covariance matrix matrix of *points1* vs *points2* using the current hyperparameters as followed:

**Parameters**

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

## Return values

<i>psi</i>	Covariance matrix
<i>dpsi</i>	Derivative of covariance matrix w.r.t. the hyperparameters OR the input points

- `intrinsicCovarianceMatrix(this)`
  - Assume `points1 = points2 = samples`
- `intrinsicCovarianceMatrix(this, points1)`
  - Not used (and not implemented yet)
- `intrinsicCovarianceMatrix(this, points1, points2)`
  - Not used (and not implemented yet)

7.3.3.22 function `marginalLikelihood ( var dpsi, var dsigma2 )` [inherited]

Marginal likelihood function.

Used for Maximum Likelihood Estimation (MLE)

## Parameters

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

## Return values

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

Papers:

- "Gaussian Processes for Machine Learning" (Chapter 5), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006
- "An adjoint for likelihood maximization" D.J.J. Toal, A.I.J. Forrester, N.W. Bressloff, A.J. Keane, C. Holden, Proc. of the Royal Society, 2009

**Todo** Adjoint derivatives work, but are very slow due to naive implementation

7.3.3.23 function `mseTestset ( var testx, var testy )` [inherited]

Calculates error on holdout set.

Error function used is the Mean Squared Error (MSE) function.

## Parameters

<i>testx</i>	input samples of the test set
<i>testy</i>	output samples of the test set

## Return values

<i>out</i>	mse error on test set
------------	-----------------------

7.3.3.24 function `plotVariogram ( )` [inherited]

Variogram plot (EXPERIMENTAL)

Plots the experimental (semi-)variogram (based on the data) as well as the theoretical kriging (semi-)variogram (defined by the correlation function).

#### Return values

<i>h</i>	Figure handle
----------	---------------

Empirical variogram:

- Methods of moments estimator Matheron (Cressie, 1993)
- More robust estimators: Cressie-Hawkins (1980), Genton (1998) In general: use median instead of mean

#### 7.3.3.25 function predict ( var *points* ) [inherited]

Predict the mean and/or variance for one or more points x.

#### Parameters

<i>points</i>	Matrix of input points to be predicted
---------------	--

#### Return values

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

#### Precondition

The kriging object should be fitted using a dataset ([BasicGaussianProcess::fit](#))

#### 7.3.3.26 function predict\_derivatives ( var *point* ) [inherited]

Predict the derivatives of the mean and/or variance for a points.

NOTE:

#### Parameters

<i>point</i>	input point to calculate the derivative of
--------------	--

#### Return values

<i>dvalues</i>	Derivative w.r.t. the output
<i>sigma2</i>	Derivative w.r.t. the output variance (optional)

- limited to one point at a time (x is a vector)

#### Precondition

The kriging object should be fitted using a dataset ([BasicGaussianProcess::fit](#))

#### 7.3.3.27 function predict\_limit ( var *points* ) [inherited]

Limit predictor of kriging (EXPERIMENTAL)

#### Parameters

<i>points</i>	Matrix of input points to be predicted
---------------	--

## Return values

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

7.3.3.28 function pseudoLikelihood ( var *dpsi*, var *dsigma2* ) [inherited]

Leave-one-out predictive log probability (pseudo-likelihood)

Calculates the leave-one-out predictive log probability.

## Parameters

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

## Return values

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

## Papers:

- "Predictive Approaches for Choosing Hyperparameters in Gaussian Process", S. Sundararajan, S.S. Keerthu, 1999

## 7.3.3.29 function rcValues ( ) [inherited]

Quantifies magnification of noise (lower is better)

Robustness-criterion (In theory useful only for ordinary kriging).

## Return values

<i>rc</i>	robustness-criterion
-----------	----------------------

Returns a 2xn matrix:

- the first row contains the absolute robustness
- the second row contains the relative robustness Papers:
- "Kriging models that are robust w.r.t. simulation errors" A.Y.D. Siem, D. den Hertog (tech report)

7.3.3.30 function regressionFunction ( var *varargin* )

Returns the regression function.

Symbolic expression and terms not supported for cokriging.

## Parameters

<i>varargin</i>	Options
-----------------	---------

## Return values

<i>regressionFcn</i>	Degree matrix representing the regression function
<i>expression</i>	Symbolic expression
<i>terms</i>	Cell array of the individual terms

**7.3.3.31 function regressionMatrix ( var *points* )** [protected]

Constructs regression matrix.

Regression matrix (model matrix, Vandermonde matrix, ...) for a set of points Based on this.regressionFcn.

**Parameters**

<i>points</i>	input point matrix (optional)
---------------	-------------------------------

**Return values**

<i>F</i>	Model matrix
<i>dF</i>	Derivative of model matrix w.r.t. the hyperparameters OR the input points

**7.3.3.32 function setData ( var *samples*, var *values* )** [protected]

Sets samples and values matrix.

Concatenate sample/values cell array to numeric array passing the resulting dataset to the underlying base method.

**Parameters**

<i>samples</i>	input sample matrix (cell array)
<i>values</i>	output value matrix (cell array)

**7.3.3.33 function setOption ( var *key*, var *value* )** [inline],[inherited]

Sets a value in the options structure.

**Parameters**

<i>key</i>	option name
<i>value</i>	option value

**7.3.3.34 function tuneParameters ( var *F* )** [protected],[inherited]

Hyperparameter optimization.

Setups and invokes the optimizer.

**Parameters**

<i>F</i>	model matrix
----------	--------------

**Return values**

<i>optimHp</i>	optimized hyperparameters
<i>perf</i>	Performance score (likelihood score)

**7.3.3.35 function updateModel ( var *F*, var *hp* )** [protected],[inherited]

Constructs model.

Full update of the model (regression + correlation part)

**Parameters**

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

**7.3.3.36** `function updateRegression ( var F, var hp )` `[protected]`, `[inherited]`

Constructs regression part.

Updates regression part of the model.

#### Parameters

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

#### Return values

<i>err</i>	error string (if any)
------------	-----------------------

**Todo** Rho is only used by co-kriging, can we abstract this somehow ?

**7.3.3.37** `function updateStochasticProcess ( var hp )` `[protected]`, `[inherited]`

Constructs correlation part.

Updates correlation part of the model.

#### Parameters

<i>hp</i>	hyperparameters
-----------	-----------------

#### Return values

<i>err</i>	error string (if any)
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

### 7.3.4 Member Data Documentation

**7.3.4.1** `var alpha` `[protected]`, `[inherited]`

Regression coefficients.

**7.3.4.2** `var C` `[protected]`, `[inherited]`

Choleski Decomposition of extrinsic + intrinsic matrices.

**7.3.4.3** `var C_reinterp` `[protected]`, `[inherited]`

Reinterpolation version of C.

**7.3.4.4** `var correlationFcn` `[protected]`, `[inherited]`

string -> function handle

**7.3.4.5** `var dist` `[protected]`, `[inherited]`

sample inter-distance

**7.3.4.6** `var distIdxPsi` `[protected]`, `[inherited]`

indexing needed to calculate psi from D

7.3.4.7 **var Ft** [protected],[inherited]

decorrelated model matrix

7.3.4.8 **var Ft.reinterp** [protected],[inherited]

Reinterpolation version of Ft.

7.3.4.9 **var gamma** [protected],[inherited]

'Correlation part' coefficients

7.3.4.10 **const var HP** [inherited]

index of the correlation function parameters

7.3.4.11 **var hyperparameters** [protected],[inherited]

correlation parameters

7.3.4.12 **var hyperparameters0** [protected],[inherited]

initial hp values OR the real ones (if optimization is done outside the class)

7.3.4.13 **const var LAMBDA** [inherited]

index of the lambda parameter

7.3.4.14 **var optimIdx** [protected],[inherited]

logical indices to parameters that are optimized

7.3.4.15 **var optimNrParameters** [protected],[inherited]

number of optimization parameter (vector; one entry per type of parameter)

7.3.4.16 **var options** [protected],[inherited]

7.3.4.17 **var R** [protected],[inherited]

from QR decomposition of regression part

7.3.4.18 **var R.reinterp** [protected],[inherited]

Reinterpolation version of R.

7.3.4.19 **var regressionFcn** [protected],[inherited]

degrees matrix (strings are converted)

7.3.4.20 **const var RHO** [inherited]

index of the rho parameter

7.3.4.21 **var rho** [protected],[inherited]

7.3.4.22 **var Sigma** [protected],[inherited]

intrinsic covariance matrix (amount of regression of stochastic part)



**7.3.4.23** `const var SIGMA2` [inherited]

index of the sigma2 parameter

**7.3.4.24** `var sigma2` [protected], [inherited]

process variance of the GP (extrinsic variance )

**7.3.4.25** `var sigma2_reinterp` [protected], [inherited]

Reinterpolation version of `sigma2`.

**Note**

reinterpolation: might be nicer to just construct and keep a sub-GP... takes more space, some calculations are done twice but performance shouldn't take a very big hit...

**7.3.4.26** `var tau2` [protected], [inherited]

intrinsic variance

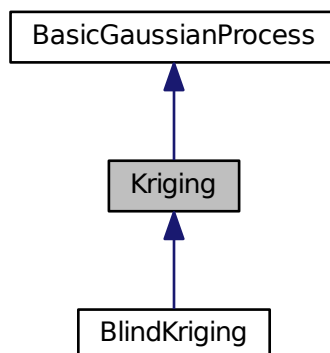
The documentation for this class was generated from the following files:

- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/CoKriging.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/correlationFunction.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/extrinsicCorrelationMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/fit.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/intrinsicCovarianceMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/regressionFunction.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/regressionMatrix.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/setData.m](#)

**7.4 Kriging Class Reference**

A kriging surrogate model.

Inheritance diagram for Kriging:



## Public Member Functions

- function [Kriging](#) (var varargin)  
*Class constructor.*
- function [getProcessVariance](#) ()  
*Returns the process variance.*
- function [getSamples](#) ()  
*Returns the input sample matrix.*
- function [getValues](#) ()  
*Returns the output value matrix.*
- function [predict](#) (var points)  
*Predict the mean and/or variance for one or more points x.*
- function [predict\\_derivatives](#) (var point)  
*Predict the derivatives of the mean and/or variance for a points x.*
- function [getExpression](#) (var outputIndex)  
*Returns the Matlab expression of this Gaussian Process model.*
- function [cvpe](#) ()  
*Calculates cross validated prediction error (cvpe)*
- function [getHyperparameters](#) ()  
*Returns the hyperparameters.*
- function [getRho](#) ()  
*Returns the rho parameter (only valid for [CoKriging](#))*
- function [getCorrelationMatrix](#) ()  
*Returns the full extrinsic correlation matrix.*
- function [getSigma](#) ()  
*Returns the intrinsic covariance matrix.*
- function [getRegressionMatrix](#) ()  
*Returns the model matrix (Vandermonde matrix)*
- function [setOption](#) (var key, var value)  
*Sets a value in the options structure.*
- function [cleanup](#) ()  
*Clears some unused variables.*
- function [display](#) ()  
*Returns user-friendly description of the class instance.*
- function [fit](#) (var samples, var values)  
*Fits a gaussian process for a dataset.*
- function [predict\\_limit](#) (var points)  
*Limit predictor of kriging (EXPERIMENTAL)*
- function [regressionFunction](#) (var options)  
*Returns the regression function.*
- function [correlationFunction](#) (var options)  
*Returns the internal correlation function handle and a symbolic expression of the the correlation part.*
- function [marginalLikelihood](#) (var dpsi, var dsigma2)  
*Marginal likelihood function.*
- function [pseudoLikelihood](#) (var dpsi, var dsigma2)  
*Leave-one-out predictive log probability (pseudo-likelihood)*
- function [mseTestset](#) (var testx, var testy)  
*Calculates error on holdout set.*
- function [imse](#) ()  
*Calculates the log of the integrated mean squared error.*
- function [rcValues](#) ()  
*Quantifies magnification of noise (lower is better)*
- function [plotVariogram](#) ()  
*Variogram plot (EXPERIMENTAL)*

### Static Public Member Functions

- static function [getDefaultOptions](#) ()  
*Returns a default options structure.*

### Public Attributes

- const var [RHO](#)  
*index of the rho parameter*
- const var [LAMBDA](#)  
*index of the lambda parameter*
- const var [SIGMA2](#)  
*index of the sigma2 parameter*
- const var [HP](#)  
*index of the correlation function parameters*

### Protected Member Functions

- function [setData](#) (var samples, var values)  
*Sets samples and values matrix.*
- function [updateModel](#) (var F, var hp)  
*Constructs model.*
- function [updateRegression](#) (var F, var hp)  
*Constructs regression part.*
- function [updateStochasticProcess](#) (var hp)  
*Constructs correlation part.*
- function [extrinsicCorrelationMatrix](#) (var points1, var points2)  
*Constructs extrinsic correlation matrix.*
- function [intrinsicCovarianceMatrix](#) (var points1, var points2)  
*Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)*
- function [regressionMatrix](#) (var points)  
*Constructs regression matrix.*
- function [tuneParameters](#) (var F)  
*Hyperparameter optimization.*
- function [generateDegrees](#) (var dj)  
*Generate degrees matrix from individual ones.*

### Protected Attributes

- var [options](#)
- var [regressionFcn](#)  
*degrees matrix (strings are converted)*
- var [correlationFcn](#)  
*string -> function handle*
- var [hyperparameters0](#)  
*initial hp values OR the real ones (if optimization is done outside the class)*
- var [dist](#)  
*sample inter-distance*
- var [distIdxPsi](#)  
*indexing needed to calculate psi from D*
- var [optimIdx](#)

- logical indices to parameters that are optimized*
- var [optimNrParameters](#)  
*number of optimization parameter (vector; one entry per type of parameter)*
- var [alpha](#)  
*Regression coefficients.*
- var [gamma](#)  
*'Correlation part' coefficients*
- var [hyperparameters](#)  
*correlation parameters*
- var [rho](#)
- var [C](#)  
*Choleski Decomposition of extrinsic + intrinsic matrices.*
- var [sigma2](#)  
*process variance of the GP (extrinsic variance )*
- var [tau2](#)  
*intrinsic variance*
- var [Sigma](#)  
*intrinsic covariance matrix (amount of regression of stochastic part)*
- var [Ft](#)  
*decorrelated model matrix*
- var [R](#)  
*from QR decomposition of regression part*
- var [sigma2\\_reinterp](#)  
*Reinterpolation version of `sigma2`.*
- var [C\\_reinterp](#)  
*Reinterpolation version of `C`.*
- var [Ft\\_reinterp](#)  
*Reinterpolation version of `Ft`.*
- var [R\\_reinterp](#)  
*Reinterpolation version of `R`.*

#### 7.4.1 Detailed Description

A kriging surrogate model.

Is basically the same as the base class [BasicGaussianProcess](#) besides the scaling of the samples/values

This class is derived by KrigingModel, a class in the SUMO-Toolbox that works like an interface between SUMO and the actual kriging implementation

#### 7.4.2 Constructor & Destructor Documentation

##### 7.4.2.1 `function Kriging ( var varargin ) [inline]`

Class constructor.

Initializes the kriging model. Takes the same parameters as [BasicGaussianProcess](#)

##### Returns

instance of the kriging class

## 7.4.3 Member Function Documentation

## 7.4.3.1 function cleanup ( ) [inline],[inherited]

Clears some unused variables.

## 7.4.3.2 function correlationFunction ( var options ) [inherited]

Returns the internal correlation function handle and a symbolic expression of the the correlation part.

Example: [correlationFcn expression] = correlationFunction( this,struct( 'latex', true, 'includeCoefficients', false) )

The expression is based on the scaled data

## Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>correlationFcn</i>	String of correlation function
<i>expression</i>	Symbolic expression

## 7.4.3.3 function cvpe ( )

Calculates cross validated prediction error (cvpe)

Error function used is the Mean Squared Error (MSE)

## Return values

<i>out</i>	cvpe score
------------	------------

## 7.4.3.4 function display ( ) [inline],[inherited]

Returns user-friendly description of the class instance.

## 7.4.3.5 function extrinsicCorrelationMatrix ( var points1, var points2 ) [protected],[inherited]

Constructs extrinsic correlation matrix.

Generate correlation matrix of points1 vs points2 using the current hyperparameters as followed:

## Parameters

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

## Return values

<i>psi</i>	Correlation matrix
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

- extrinsicCorrelationMatrix(this)
  - Assume points1 = points2 = samples
- extrinsicCorrelationMatrix(this, points1)
  - Assume points2 = samples
- extrinsicCorrelationMatrix(this, points1, points2)

NOTE: The first case returns derivatives w.r.t. X The latter two cases returns derivatives w.r.t. the hyperparameters

#### 7.4.3.6 function fit ( var *samples*, var *values* ) [inherited]

Fits a gaussian process for a dataset.

Need to be invoked before calling any of the prediction methods.

##### Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix

##### Note

[Kriging](#) can't do a cleanup automatically, if needed [cleanup\(\)](#) can be called manually.

#### 7.4.3.7 function generateDegrees ( var *dj* ) [protected],[inherited]

Generate degrees matrix from individual ones.

Generates a (full) degree matrix based on one or more degree matrices for variable j The option 'regressionMaxLevelInteractions' determines the maximum level of interactions ([BasicGaussianProcess::getDefaultOptions](#))

##### Parameters

<i>dj</i>	Cell array of degree matrices (for each dimension)
-----------	--

##### Return values

<i>degrees</i>	Full degrees matrix
<i>idx</i>	Cell array of indices (used by blind kriging)

#### 7.4.3.8 function getCorrelationMatrix ( ) [inline],[inherited]

Returns the full extrinsic correlation matrix.

#### 7.4.3.9 static function getDefaultOptions ( ) [inline],[static]

Returns a default options structure.

##### Return values

<i>options</i>	Options structure
----------------	-------------------

**Todo** Apparently Matlab 2008b doesn't auto forward the (static) call to [BasicGaussianProcess::getDefaultOptions\(\)](#)

#### 7.4.3.10 function getExpression ( var *outputIndex* )

Returns the Matlab expression of this Gaussian Process model.

Example: expression = getExpression( this )

##### Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>expression</i>	Symbolic expression
-------------------	---------------------

7.4.3.11 function `getHyperparameters ( )` [inline],[inherited]

Returns the hyperparameters.

7.4.3.12 function `getProcessVariance ( )` [inline]

Returns the process variance.

## Return values

<i>sigma2</i>	process variance
---------------	------------------

7.4.3.13 function `getRegressionMatrix ( )` [inline],[inherited]

Returns the model matrix (Vandermonde matrix)

7.4.3.14 function `getRho ( )` [inline],[inherited]

Returns the rho parameter (only valid for [CoKriging](#))

7.4.3.15 function `getSamples ( )` [inline]

Returns the input sample matrix.

## Return values

<i>samples</i>	unscaled samples (original)
<i>scaledSamples</i>	scaled samples

7.4.3.16 function `getSigma ( )` [inline],[inherited]

Returns the intrinsic covariance matrix.

7.4.3.17 function `getValues ( )` [inline]

Returns the output value matrix.

## Return values

<i>values</i>	unscaled values (original)
<i>scaledValues</i>	scaled values

7.4.3.18 function `imse ( )` [inherited]

Calculates the log of the integrated mean squared error.

Only supported for 1D and 2D problems.

## Return values

<i>out</i>	log(imse) score
------------	-----------------

Papers:

- See Sacks 1989
- See pechiniy paper...

**Todo** Implement generic monte carlo integration

**7.4.3.19** `function intrinsicCovarianceMatrix ( var points1, var points2 )` `[protected], [inherited]`

Constructs intrinsic covariance matrix (stochastic kriging/regression kriging)

Generate covariance matrix matrix of points1 vs points2 using the current hyperparameters as followed:

#### Parameters

<i>points1</i>	input point matrix (optional)
<i>points2</i>	input point matrix (optional)

#### Return values

<i>psi</i>	Covariance matrix
<i>dpsi</i>	Derivative of covariance matrix w.r.t. the hyperparameters OR the input points

- `intrinsicCovarianceMatrix(this)`
  - Assume `points1 = points2 = samples`
- `intrinsicCovarianceMatrix(this, points1)`
  - Not used (and not implemented yet)
- `intrinsicCovarianceMatrix(this, points1, points2)`
  - Not used (and not implemented yet)

**7.4.3.20** `function marginalLikelihood ( var dpsi, var dsigma2 )` `[inherited]`

Marginal likelihood function.

Used for Maximum Likelihood Estimation (MLE)

#### Parameters

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

#### Return values

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

Papers:

- "Gaussian Processes for Machine Learning" (Chapter 5), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006
- "An adjoint for likelihood maximization" D.J.J. Toal, A.I.J. Forrester, N.W. Bressloff, A.J. Keane, C. Holden, Proc. of the Royal Society, 2009

**Todo** Adjoint derivatives work, but are very slow due to naive implementation

**7.4.3.21** `function mseTestset ( var testx, var testy )` `[inherited]`

Calculates error on holdout set.

Error function used is the Mean Squared Error (MSE) function.



## Parameters

<i>testx</i>	input samples of the test set
<i>testy</i>	output samples of the test set

## Return values

<i>out</i>	mse error on test set
------------	-----------------------

## 7.4.3.22 function plotVariogram ( ) [inherited]

## Variogram plot (EXPERIMENTAL)

Plots the experimental (semi-)variogram (based on the data) as well as the theoretical kriging (semi-)variogram (defined by the correlation function).

## Return values

<i>h</i>	Figure handle
----------	---------------

Empirical variogram:

- Methods of moments estimator Matheron (Cressie, 1993)
- More robust estimators: Cressie-Hawkins (1980), Genton (1998) In general: use median instead of mean

## 7.4.3.23 function predict ( var points )

Predict the mean and/or variance for one or more points x.

## Parameters

<i>points</i>	Matrix of input points to be predicted
---------------	--

## Return values

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

## Precondition

The kriging object should be fitted using a dataset (*fit*)

## 7.4.3.24 function predict\_derivatives ( var point )

Predict the derivatives of the mean and/or variance for a points x.

NOTE:

## Parameters

<i>point</i>	input point to calculate the derivative of
--------------	--

## Return values

<i>dvalues</i>	Derivative w.r.t. the output
<i>sigma2</i>	Derivative w.r.t. the output variance (optional)

- limited to one point at a time (x is a row vector)

**7.4.3.25** `function predict_limit ( var points )` `[inherited]`

Limit predictor of kriging (EXPERIMENTAL)

**Parameters**

<i>points</i>	Matrix of input points to be predicted
---------------	--

**Return values**

<i>values</i>	predicted output values
<i>sigma2</i>	predicted variance of the output values (optional)

**7.4.3.26** `function pseudoLikelihood ( var dpsi, var dsigma2 )` `[inherited]`

Leave-one-out predictive log probability (pseudo-likelihood)

Calculates the leave-one-out predictive log probability.

**Parameters**

<i>dpsi</i>	cell array of derivative matrices (optional; for internal use only)
-------------	---

**Return values**

<i>out</i>	score
<i>dout</i>	Derivatives w.r.t. hyperparameters

Papers:

- "Predictive Approaches for Choosing Hyperparameters in Gaussian Process", S. Sundararajan, S.S. Keerthu, 1999

**7.4.3.27** `function rcValues ( )` `[inherited]`

Quantifies magnification of noise (lower is better)

Robustness-criterion (In theory useful only for ordinary kriging).

**Return values**

<i>rc</i>	robustness-criterion
-----------	----------------------

Returns a 2xn matrix:

- the first row contains the absolute robustness
- the second row contains the relative robustness Papers:
- "Kriging models that are robust w.r.t. simulation errors" A.Y.D. Siem, D. den Hertog (tech report)

**7.4.3.28** `function regressionFunction ( var options )` `[inherited]`

Returns the regression function.

Example: `[regressionFcn expression terms] = regressionFunction( this, struct('latex', true, 'precision', '%.5g') )`

The expression is based on the scaled data

## Parameters

<i>options</i>	Options struct
----------------	----------------

## Return values

<i>regressionFcn</i>	Degree matrix representing the regression function
<i>expression</i>	Symbolic expression
<i>terms</i>	Cell array of the individual terms

7.4.3.29 function regressionMatrix ( var *points* ) [protected],[inherited]

Constructs regression matrix.

Regression matrix (model matrix, Vandermonde matrix, ...) for a set of points Based on this.regressionFcn.

## Parameters

<i>points</i>	input point matrix (optional)
---------------	-------------------------------

## Return values

<i>F</i>	Model matrix
<i>dF</i>	Derivative of model matrix w.r.t. the hyperparameters OR the input points

7.4.3.30 function setData ( var *samples*, var *values* ) [protected]

Sets samples and values matrix.

Scales samples and values, passing the scaled dataset to the underlying base method.

## Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix

7.4.3.31 function setOption ( var *key*, var *value* ) [inline],[inherited]

Sets a value in the options structure.

## Parameters

<i>key</i>	option name
<i>value</i>	option value

7.4.3.32 function tuneParameters ( var *F* ) [protected],[inherited]

Hyperparameter optimization.

Setups and invokes the optimizer.

## Parameters

<i>F</i>	model matrix
----------	--------------

## Return values

<i>optimHp</i>	optimized hyperparameters
<i>perf</i>	Performance score (likelihood score)

**7.4.3.33** `function updateModel ( var F, var hp )` `[protected], [inherited]`

Constructs model.

Full update of the model (regression + correlation part)

**Parameters**

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

**7.4.3.34** `function updateRegression ( var F, var hp )` `[protected], [inherited]`

Constructs regression part.

Updates regression part of the model.

**Parameters**

<i>F</i>	model matrix
<i>hp</i>	new hyperparameters

**Return values**

<i>err</i>	error string (if any)
------------	-----------------------

**Todo** Rho is only used by co-kriging, can we abstract this somehow ?

**7.4.3.35** `function updateStochasticProcess ( var hp )` `[protected], [inherited]`

Constructs correlation part.

Updates correlation part of the model.

**Parameters**

<i>hp</i>	hyperparameters
-----------	-----------------

**Return values**

<i>err</i>	error string (if any)
<i>dpsi</i>	Derivative of correlation matrix w.r.t. the hyperparameters

**7.4.4 Member Data Documentation****7.4.4.1** `var alpha` `[protected], [inherited]`

Regression coefficients.

**7.4.4.2** `var C` `[protected], [inherited]`

Choleski Decomposition of extrinsic + intrinsic matrices.

**7.4.4.3** `var C_reinterp` `[protected], [inherited]`

Reinterpolation version of C.

**7.4.4.4** `var correlationFcn` [protected],[inherited]

string -> function handle

**7.4.4.5** `var dist` [protected],[inherited]

sample inter-distance

**7.4.4.6** `var distIdxPsi` [protected],[inherited]

indexing needed to calculate psi from D

**7.4.4.7** `var Ft` [protected],[inherited]

decorrelated model matrix

**7.4.4.8** `var Ft_reinterp` [protected],[inherited]

Reinterpolation version of Ft.

**7.4.4.9** `var gamma` [protected],[inherited]

'Correlation part' coefficients

**7.4.4.10** `const var HP` [inherited]

index of the correlation function parameters

**7.4.4.11** `var hyperparameters` [protected],[inherited]

correlation parameters

**7.4.4.12** `var hyperparameters0` [protected],[inherited]

initial hp values OR the real ones (if optimization is done outside the class)

**7.4.4.13** `const var LAMBDA` [inherited]

index of the lambda parameter

**7.4.4.14** `var optimIdx` [protected],[inherited]

logical indices to parameters that are optimized

**7.4.4.15** `var optimNrParameters` [protected],[inherited]

number of optimization parameter (vector; one entry per type of parameter)

**7.4.4.16** `var options` [protected],[inherited]

**7.4.4.17** `var R` [protected],[inherited]

from QR decomposition of regression part

**7.4.4.18** `var R_reinterp` [protected],[inherited]

Reinterpolation version of R.

**7.4.4.19** `var regressionFcn` [protected],[inherited]

degrees matrix (strings are converted)

**7.4.4.20 const var RHO** [inherited]

index of the rho parameter

**7.4.4.21 var rho** [protected],[inherited]**7.4.4.22 var Sigma** [protected],[inherited]

intrinsic covariance matrix (amount of regression of stochastic part)

**7.4.4.23 const var SIGMA2** [inherited]

index of the sigma2 parameter

**7.4.4.24 var sigma2** [protected],[inherited]

process variance of the GP (extrinsic variance )

**7.4.4.25 var sigma2\_reinterp** [protected],[inherited]

Reinterpolation version of `sigma2`.

**Note**

reinterpolation: might be nicer to just construct and keep a sub-GP... takes more space, some calculations are done twice but performance shouldn't take a very big hit...

**7.4.4.26 var tau2** [protected],[inherited]

intrinsic variance

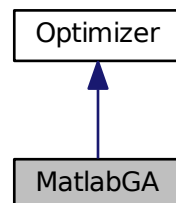
The documentation for this class was generated from the following files:

- [/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/Kriging.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/cvpe.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/getExpression.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/predict.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/predict\\_derivatives.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/@Kriging/setData.m](#)

**7.5 MatlabGA Class Reference**

Wrapper around the matlab optimizers.

Inheritance diagram for MatlabGA:



## Public Member Functions

- function [MatlabGA](#) (var varargin)  
*Class constructor.*
- function [optimize](#) (var arg)  
*This function optimizes the given function handle.*
- function [getPopulationSize](#) ()  
*Get the number of individuals.*
- function [setInputConstraints](#) (var con)  
*Sets input constraints.*
- function [getBounds](#) ()  
*Returns bounds for optimizers that need it.*
- function [setBounds](#) (var LB, var UB)  
*Sets bounds for optimizers that need it.*
- function [getInitialPopulation](#) ()  
*Gets the starting positions for the search.*
- function [setInitialPopulation](#) (var pop)  
*Sets the starting positions for the search.*
- function [getInputDimension](#) ()  
*Returns the number of input variables.*
- function [getOutputDimension](#) ()  
*Returns the number of cost functions.*
- function [setDimensions](#) (var inDim, var outDim)  
*Sets the number of input and output dimensions.*
- function [setHint](#) (var key, var value)  
*Gives a hint to the optimizer.*
- function [getHint](#) (var key)  
*Gets a hint to the optimizer.*
- function [setState](#) (var state)  
*Sets some extra information.*
- function [getState](#) ()  
*Gets some extra information.*

## 7.5.1 Detailed Description

Wrapper around the matlab optimizers.

## 7.5.2 Constructor &amp; Destructor Documentation

7.5.2.1 function **MatlabGA** ( var varargin ) `[inline]`

Class constructor.

## Returns

instance of the class

## 7.5.3 Member Function Documentation

## 7.5.3.1 function getBounds ( ) [inherited]

Returns bounds for optimizers that need it.

Only needed for optimization methods that support it.

## Return values

<i>LB</i>	lower bound
<i>UB</i>	upper bound

## 7.5.3.2 function getHint ( var key ) [inherited]

Gets a hint to the optimizer.

Returns a particular hint setting.

## Parameters

<i>key</i>	property name
------------	---------------

## Return values

<i>value</i>	property value
--------------	----------------

## 7.5.3.3 function getInitialPopulation ( ) [inherited]

Gets the starting positions for the search.

pop may be a matrix for population-based, multi-start, etc. methods

## Return values

<i>startx</i>	matrix of initial values
---------------	--------------------------

## 7.5.3.4 function getInputDimension ( ) [inherited]

Returns the number of input variables.

## Return values

<i>nvars</i>	Number of input variables
--------------	---------------------------

## 7.5.3.5 function getOutputDimension ( ) [inherited]

Returns the number of cost functions.

Returns the number of output variables.

## Return values

<i>nobjectives</i>	Number of cost functions
--------------------	--------------------------

## 7.5.3.6 function getPopulationSize ( )

Get the number of individuals.



**7.5.3.7 function getState ( ) [inline],[inherited]**

Gets some extra information.

Return values

<i>state</i>	structure
--------------	-----------

**7.5.3.8 function optimize ( var *arg* )**

This function optimizes the given function handle.

**7.5.3.9 function setBounds ( var *LB*, var *UB* ) [inherited]**

Sets bounds for optimizers that need it.

Only needed for optimization methods that support it.

Parameters

<i>LB</i>	lower bound
<i>UB</i>	upper bound

**7.5.3.10 function setDimensions ( var *inDim*, var *outDim* ) [inherited]**

Sets the number of input and output dimensions.

Includes some input checking to ensure that the bounds and the initial population are still correct.

Parameters

<i>inDim</i>	Number of input variables
<i>outDim</i>	Number of cost functions

If not, these variables are reset to their default values without warning!

**7.5.3.11 function setHint ( var *key*, var *value* ) [inherited]**

Gives a hint to the optimizer.

Sets a hint that may or not be honored by the optimizer (depends on the type...).

Parameters

<i>key</i>	property name
<i>value</i>	property value

Only supports 'maxTime', time atm.

**7.5.3.12 function setInitialPopulation ( var *pop* ) [inherited]**

Sets the starting positions for the search.

Sets the initial population.

Parameters

<i>pop</i>	matrix of initial values
------------	--------------------------

**7.5.3.13 function setInputConstraints ( var *con* )**

Sets input constraints.

#### 7.5.3.14 function setState ( var state ) [inline],[inherited]

Sets some extra information.

##### Parameters

<i>state</i>	structure
--------------	-----------

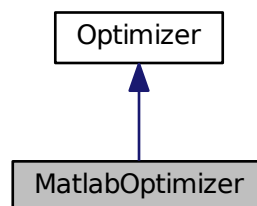
The documentation for this class was generated from the following files:

- /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/MatlabGA.m
- /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/getPopulationSize.m
- /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/optimize.m
- /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/setInputConstraints.m

## 7.6 MatlabOptimizer Class Reference

Wrapper around the matlab optimizers.

Inheritance diagram for MatlabOptimizer:



### Public Member Functions

- function [MatlabOptimizer](#) (var varargin)  
*Creates an [MatlabOptimizer](#) object.*
- function [optimize](#) (var arg)  
*The hint 'maxTime' is supported.*
- function [setInputConstraints](#) (var con)  
*Supports non-linear and linear inequality constraints.*
- function [getBounds](#) ()  
*Returns bounds for optimizers that need it.*
- function [setBounds](#) (var LB, var UB)  
*Sets bounds for optimizers that need it.*
- function [getInitialPopulation](#) ()  
*Gets the starting positions for the search.*
- function [setInitialPopulation](#) (var pop)  
*Sets the starting positions for the search.*
- function [getInputDimension](#) ()  
*Returns the number of input variables.*
- function [getOutputDimension](#) ()

- Returns the number of cost functions.*
- function [setDimensions](#) (var inDim, var outDim)  
*Sets the number of input and output dimensions.*
- function [setHint](#) (var key, var value)  
*Gives a hint to the optimizer.*
- function [getHint](#) (var key)  
*Gets a hint to the optimizer.*
- function [setState](#) (var state)  
*Sets some extra information.*
- function [getState](#) ()  
*Gets some extra information.*
- function [getPopulationSize](#) ()  
*Get the number of individuals in the population.*

### 7.6.1 Detailed Description

Wrapper around the matlab optimizers.

The matlab Optimization toolbox is required. If no bounds are set 'fmincon' will be used, else 'fminunc'.

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 function MatlabOptimizer ( var varargin ) [inline]

Creates an [MatlabOptimizer](#) object.

Takes the same option as the base class + an options structure (see optimset)

#### Parameters

<i>nvars</i>	Number of dimensions
<i>nobjectives</i>	Number of cost functions
<i>options</i>	Option structure

#### Returns

instance of the [Optimizer](#) class

### 7.6.3 Member Function Documentation

#### 7.6.3.1 function getBounds ( ) [inherited]

Returns bounds for optimizers that need it.

Only needed for optimization methods that support it.

#### Return values

<i>LB</i>	lower bound
<i>UB</i>	upper bound

#### 7.6.3.2 function getHint ( var key ) [inherited]

Gets a hint to the optimizer.

Returns a particular hint setting.

## Parameters

<i>key</i>	property name
------------	---------------

## Return values

<i>value</i>	property value
--------------	----------------

## 7.6.3.3 function getInitialPopulation ( ) [inherited]

Gets the starting positions for the search.

pop may be a matrix for population-based, multi-start, etc. methods

## Return values

<i>startx</i>	matrix of initial values
---------------	--------------------------

## 7.6.3.4 function getInputDimension ( ) [inherited]

Returns the number of input variables.

## Return values

<i>nvars</i>	Number of input variables
--------------	---------------------------

## 7.6.3.5 function getOutputDimension ( ) [inherited]

Returns the number of cost functions.

Returns the number of output variables.

## Return values

<i>nobjectives</i>	Number of cost functions
--------------------	--------------------------

## 7.6.3.6 function getPopulationSize ( ) [inherited]

Get the number of individuals in the population.

The base method assumes only 1 individual.

## Return values

<i>size</i>	Population size
-------------	-----------------

Population-based optimization methods should override this

## 7.6.3.7 function getState ( ) [inline],[inherited]

Gets some extra information.

## Return values

<i>state</i>	structure
--------------	-----------

## 7.6.3.8 function optimize ( var arg )

The hint 'maxTime' is supported.

**7.6.3.9 function setBounds ( var *LB*, var *UB* )** [inherited]

Sets bounds for optimizers that need it.

Only needed for optimization methods that support it.

**Parameters**

<i>LB</i>	lower bound
<i>UB</i>	upper bound

**7.6.3.10 function setDimensions ( var *inDim*, var *outDim* )** [inherited]

Sets the number of input and output dimensions.

Includes some input checking to ensure that the bounds and the initial population are still correct.

**Parameters**

<i>inDim</i>	Number of input variables
<i>outDim</i>	Number of cost functions

If not, these variables are reset to their default values without warning!

**7.6.3.11 function setHint ( var *key*, var *value* )** [inherited]

Gives a hint to the optimizer.

Sets a hint that may or not be honored by the optimizer (depends on the type...).

**Parameters**

<i>key</i>	property name
<i>value</i>	property value

Only supports 'maxTime', time atm.

**7.6.3.12 function setInitialPopulation ( var *pop* )** [inherited]

Sets the starting positions for the search.

Sets the initial population.

**Parameters**

<i>pop</i>	matrix of initial values
------------	--------------------------

**7.6.3.13 function setInputConstraints ( var *con* )**

Supports non-linear and linear inequality constraints.

**7.6.3.14 function setState ( var *state* )** [inline],[inherited]

Sets some extra information.

**Parameters**

<i>state</i>	structure
--------------	-----------

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

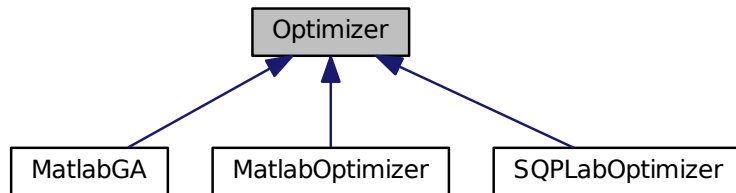
- /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/MatlabOptimizer.m

- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/optimize.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/setInputConstraints.m](#)

## 7.7 Optimizer Class Reference

Abstract base class for an optimizer.

Inheritance diagram for Optimizer:



### Public Member Functions

- function [Optimizer](#) (var varargin)  
*Creates an [Optimizer](#) object, not to be called directly.*
- function [getBounds](#) ()  
*Returns bounds for optimizers that need it.*
- function [setBounds](#) (var LB, var UB)  
*Sets bounds for optimizers that need it.*
- function [getInitialPopulation](#) ()  
*Gets the starting positions for the search.*
- function [setInitialPopulation](#) (var pop)  
*Sets the starting positions for the search.*
- function [getInputDimension](#) ()  
*Returns the number of input variables.*
- function [getOutputDimension](#) ()  
*Returns the number of cost functions.*
- function [setDimensions](#) (var inDim, var outDim)  
*Sets the number of input and output dimensions.*
- function [setHint](#) (var key, var value)  
*Gives a hint to the optimizer.*
- function [getHint](#) (var key)  
*Gets a hint to the optimizer.*
- function [setState](#) (var state)  
*Sets some extra information.*
- function [getState](#) ()  
*Gets some extra information.*
- function [getPopulationSize](#) ()  
*Get the number of individuals in the population.*
- function [setInputConstraints](#) (var con)  
*Sets input constraints.*
- function [optimize](#) (var arg)  
*This function optimizes the given function handle,.*

## 7.7.1 Detailed Description

Abstract base class for an optimizer.

Optimization methods are to be derived from this class. It provides a logger object for the derived classes and instantiates all constraints

## 7.7.2 Constructor &amp; Destructor Documentation

7.7.2.1 function `Optimizer ( var varargin )` `[inline]`

Creates an `Optimizer` object, not to be called directly.

The constructor of the derived class should be called.

## Parameters

<i>nvars</i>	Number of dimensions
<i>nobjectives</i>	Number of cost functions

## Returns

instance of the `Optimizer` class

## 7.7.3 Member Function Documentation

7.7.3.1 function `getBounds ( )`

Returns bounds for optimizers that need it.

Only needed for optimization methods that support it.

## Return values

<i>LB</i>	lower bound
<i>UB</i>	upper bound

7.7.3.2 function `getHint ( var key )`

Gets a hint to the optimizer.

Returns a particular hint setting.

## Parameters

<i>key</i>	property name
------------	---------------

## Return values

<i>value</i>	property value
--------------	----------------

7.7.3.3 function `getInitialPopulation ( )`

Gets the starting positions for the search.

pop may be a matrix for population-based, multi-start, etc. methods

## Return values

<i>startx</i>	matrix of initial values
---------------	--------------------------

## 7.7.3.4 function getInputDimension ( )

Returns the number of input variables.

## Return values

<i>nvars</i>	Number of input variables
--------------	---------------------------

## 7.7.3.5 function getOutputDimension ( )

Returns the number of cost functions.

Returns the number of output variables.

## Return values

<i>nobjectives</i>	Number of cost functions
--------------------	--------------------------

## 7.7.3.6 function getPopulationSize ( )

Get the number of individuals in the population.

The base method assumes only 1 individual.

## Return values

<i>size</i>	Population size
-------------	-----------------

Population-based optimization methods should override this

## 7.7.3.7 function getState ( ) [inline]

Gets some extra information.

## Return values

<i>state</i>	structure
--------------	-----------

## 7.7.3.8 function optimize ( var arg )

This function optimizes the given function handle,.

Dummy function. Subclasses should implement this function.

subject to constraints

## Parameters

<i>arg</i>	function handle
------------	-----------------

## Return values

<i>x</i>	optimal input point(s)
<i>fval</i>	optimal function value(s)



**7.7.3.9 function setBounds ( var *LB*, var *UB* )**

Sets bounds for optimizers that need it.

Only needed for optimization methods that support it.

**Parameters**

<i>LB</i>	lower bound
<i>UB</i>	upper bound

**7.7.3.10 function setDimensions ( var *inDim*, var *outDim* )**

Sets the number of input and output dimensions.

Includes some input checking to ensure that the bounds and the initial population are still correct.

**Parameters**

<i>inDim</i>	Number of input variables
<i>outDim</i>	Number of cost functions

If not, these variables are reset to their default values without warning!

**7.7.3.11 function setHint ( var *key*, var *value* )**

Gives a hint to the optimizer.

Sets a hint that may or not be honored by the optimizer (depends on the type...).

**Parameters**

<i>key</i>	property name
<i>value</i>	property value

Only supports 'maxTime', time atm.

**7.7.3.12 function setInitialPopulation ( var *pop* )**

Sets the starting positions for the search.

Sets the initial population.

**Parameters**

<i>pop</i>	matrix of initial values
------------	--------------------------

**7.7.3.13 function setInputConstraints ( var *con* )**

Sets input constraints.

By default the optimizer doesn't support constraints (gives error).

**Parameters**

<i>con</i>	constraint objects (cell array)
------------	---------------------------------

Derived classes should override this method if they do support it.

**7.7.3.14 function setState ( var *state* ) [inline]**

Sets some extra information.

## Parameters

<code>state</code>	structure
--------------------	-----------

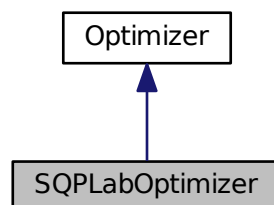
The documentation for this class was generated from the following files:

- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/Optimizer.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getBounds.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getHint.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInitialPopulation.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInputDimension.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getOutputDimension.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getPopulationSize.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/optimize.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setBounds.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setDimensions.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setHint.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInitialPopulation.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInputConstraints.m](#)

## 7.8 SQPLabOptimizer Class Reference

Wrapper around the SQPLab optimization package.

Inheritance diagram for SQPLabOptimizer:



### Public Member Functions

- function [SQPLabOptimizer](#) (var varargin)  
*Class constructor.*
- function [optimize](#) (var arg)  
*This function optimizes the given function handle.*
- function [getBounds](#) ()  
*Returns bounds for optimizers that need it.*
- function [setBounds](#) (var LB, var UB)  
*Sets bounds for optimizers that need it.*
- function [getInitialPopulation](#) ()  
*Gets the starting positions for the search.*
- function [setInitialPopulation](#) (var pop)  
*Sets the starting positions for the search.*

- function [getInputDimension](#) ()  
*Returns the number of input variables.*
- function [getOutputDimension](#) ()  
*Returns the number of cost functions.*
- function [setDimensions](#) (var inDim, var outDim)  
*Sets the number of input and output dimensions.*
- function [setHint](#) (var key, var value)  
*Gives a hint to the optimizer.*
- function [getHint](#) (var key)  
*Gets a hint to the optimizer.*
- function [setState](#) (var state)  
*Sets some extra information.*
- function [getState](#) ()  
*Gets some extra information.*
- function [getPopulationSize](#) ()  
*Get the number of individuals in the population.*
- function [setInputConstraints](#) (var con)  
*Sets input constraints.*

### 7.8.1 Detailed Description

Wrapper around the SQPLab optimization package.

### 7.8.2 Constructor & Destructor Documentation

#### 7.8.2.1 function **SQPLabOptimizer** ( var *varargin* ) [inline]

Class constructor.

#### Returns

instance of the class

### 7.8.3 Member Function Documentation

#### 7.8.3.1 function **getBounds** ( ) [inherited]

Returns bounds for optimizers that need it.

Only needed for optimization methods that support it.

#### Return values

<i>LB</i>	lower bound
<i>UB</i>	upper bound

#### 7.8.3.2 function **getHint** ( var *key* ) [inherited]

Gets a hint to the optimizer.

Returns a particular hint setting.

#### Parameters

<i>key</i>	property name
------------	---------------

## Return values

<i>value</i>	property value
--------------	----------------

7.8.3.3 function `getInitialPopulation ( )` [inherited]

Gets the starting positions for the search.

pop may be a matrix for population-based, multi-start, etc. methods

## Return values

<i>startx</i>	matrix of initial values
---------------	--------------------------

7.8.3.4 function `getInputDimension ( )` [inherited]

Returns the number of input variables.

## Return values

<i>nvars</i>	Number of input variables
--------------	---------------------------

7.8.3.5 function `getOutputDimension ( )` [inherited]

Returns the number of cost functions.

Returns the number of output variables.

## Return values

<i>nobjectives</i>	Number of cost functions
--------------------	--------------------------

7.8.3.6 function `getPopulationSize ( )` [inherited]

Get the number of individuals in the population.

The base method assumes only 1 individual.

## Return values

<i>size</i>	Population size
-------------	-----------------

Population-based optimization methods should override this

7.8.3.7 function `getState ( )` [inline],[inherited]

Gets some extra information.

## Return values

<i>state</i>	structure
--------------	-----------

7.8.3.8 function `optimize ( var arg )`

This function optimizes the given function handle.

7.8.3.9 function `setBounds ( var LB, var UB )` [inherited]

Sets bounds for optimizers that need it.

Only needed for optimization methods that support it.

## Parameters

<i>LB</i>	lower bound
<i>UB</i>	upper bound

**7.8.3.10** function setDimensions ( var *inDim*, var *outDim* ) [inherited]

Sets the number of input and output dimensions.

Includes some input checking to ensure that the bounds and the initial population are still correct.

## Parameters

<i>inDim</i>	Number of input variables
<i>outDim</i>	Number of cost functions

If not, these variables are reset to their default values without warning!

**7.8.3.11** function setHint ( var *key*, var *value* ) [inherited]

Gives a hint to the optimizer.

Sets a hint that may or not be honored by the optimizer (depends on the type...).

## Parameters

<i>key</i>	property name
<i>value</i>	property value

Only supports 'maxTime', time atm.

**7.8.3.12** function setInitialPopulation ( var *pop* ) [inherited]

Sets the starting positions for the search.

Sets the initial population.

## Parameters

<i>pop</i>	matrix of initial values
------------	--------------------------

**7.8.3.13** function setInputConstraints ( var *con* ) [inherited]

Sets input constraints.

By default the optimizer doesn't support constraints (gives error).

## Parameters

<i>con</i>	constraint objects (cell array)
------------	---------------------------------

Derived classes should override this method if they do support it.

**7.8.3.14** function setState ( var *state* ) [inline],[inherited]

Sets some extra information.

## Parameters

<i>state</i>	structure
--------------	-----------

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

- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/SQPLabOptimizer.m](#)
- [/home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/optimize.m](#)

## 8 File Documentation

### 8.1 [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/BasicGaussianProcess.m](#) File Reference

#### Classes

- class [BasicGaussianProcess](#)  
*A kriging surrogate model (also known as a Gaussian Process)*

#### 8.1.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [Basic-GaussianProcess](#)

### 8.2 [/home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/correlationFunction.m](#) File Reference

#### 8.2.1 Detailed Description

#### Authors

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[correlationFcn expression] = correlationFunction( this, options )

**8.3 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/correlationFunction.m File Reference****8.3.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[correlationFcn expression] = correlationFunction( this, varargin )

## 8.4 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/cvpe.m File Reference

### 8.4.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature out = cvpe(this)

## 8.5 /home/ilm/projecten/meta/scripts/ooDACE/@Kriging/cvpe.m File Reference

### 8.5.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:



- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature out = cvpe(this)

## **8.6 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/extrinsicCorrelationMatrix.m File Reference**

### **8.6.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [psi dpsj] = extrinsicCorrelationMatrix(this, points1, points2 )

## **8.7 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/extrinsicCorrelationMatrix.m File Reference**

### **8.7.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [psi  
dpsi] = extrinsicCorrelationMatrix(this, points1, points2)

**8.8 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/fit.m File Reference****8.8.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this =  
fit( this, samples, values )

## 8.9 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/fit.m File Reference

### 8.9.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this IK] = fit( this, samples, values )

## 8.10 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/fit.m File Reference

### 8.10.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = fit( this, samples, values )

## **8.11 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/generateDegrees.m File Reference**

### **8.11.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [degrees usedIdx] = generateDegrees( this, dj )

## **8.12 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/getExpression.m File Reference**

### **8.12.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature expression = getExpression( this, outputIndex )

**8.13 /home/ilm/projecten/meta/scripts/ooDACE/@Kriging/getExpression.m File Reference****8.13.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature expression = getExpression( this, outputIndex )

## 8.14 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/imse.m File Reference

### 8.14.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature out = imse( this )

## 8.15 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/intrinsicCovarianceMatrix.m File Reference

### 8.15.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[Sigma dSigma] = intrinsicCovarianceMatrix(this, points1, points2 )

## **8.16 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/intrinsicCovarianceMatrix.m File Reference**

### **8.16.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[Sigma dSigma] = intrinsicCovarianceMatrix(this, points1, points2 )

## **8.17 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/likelihood.m File Reference**

### **8.17.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [out  
dout] = likelihood( this, F, hp )

## 8.18 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/marginalLikelihood.m File Reference

### 8.18.1 Detailed Description

Authors

Ivo Couckuyt

Version

1.4 (\$Revision\$)

Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [out  
dout] = marginalLikelihood( this, dpsl, dsigma2 )



## 8.19 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/mseTestset.m File Reference

### 8.19.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature out = mseTestset(this, testx, testy)

## 8.20 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/plotLikelihood.m File Reference

### 8.20.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature h = plotLikelihood(this, func, param, perf)

## 8.21 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/plotVariogram.m File Reference

### 8.21.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature handle = plotVariogram(this)

## 8.22 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/predict.m File Reference

### 8.22.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [y sigma2] = predict(this, points)

## 8.23 /home/ilm/projecten/meta/scripts/ooDACE/@Kriging/predict.m File Reference

### 8.23.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [y sigma2] = predict(this, points)

## **8.24 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/predict\_derivatives.m File Reference**

### **8.24.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [dysigma2] = predict\_derivatives(this, point)

## **8.25 /home/ilm/projecten/meta/scripts/ooDACE/@Kriging/predict\_derivatives.m File Reference**

### **8.25.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [dy dsigma2] = predict\_derivatives(this, point)

## **8.26 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/pseudoLikelihood.m File Reference**

### **8.26.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [out dout] = pseudoLikelihood(this, dpsi, dsigma2)

## **8.27 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/rcValues.m File Reference**

### **8.27.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature rc = rcValues(this)

## 8.28 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/regressionFunction.m File Reference

### 8.28.1 Detailed Description

Authors

Ivo Couckuyt

Version

1.4 (\$Revision\$)

Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [regressionFcn expression terms] = regressionFunction( this, options )

## 8.29 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/regressionFunction.m File Reference

## 8.29.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[regressionFcn expression terms] = regressionFunction( this, options )

## 8.30 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/regressionFunction.m File Reference

## 8.30.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[regressionFcn expression terms] = regressionFunction( this, varargin )

## **8.31 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/regressionMatrix.m File Reference**

### **8.31.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [F dF]  
= regressionMatrix(this, points)

## **8.32 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/regressionMatrix.m File Reference**

### **8.32.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)



**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [F dF]  
= regressionMatrix(this, points)

**8.33 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/regressionMatrix.m File Reference****8.33.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [F dF]  
= regressionMatrix(this, points)

## **8.34 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/tuneParameters.m File Reference**

### **8.34.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this optimHp perf] = tuneParameters( this, F )

## **8.35 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateModel.m File Reference**

### **8.35.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = updateModel( this, F, hp )

## **8.36 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateRegression.m File Reference**

### **8.36.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this err dsigma2] = updateRegression( this, F, hp )

## **8.37 /home/ilm/projecten/meta/scripts/ooDACE/@BasicGaussianProcess/updateStochasticProcess.m File Reference**

### **8.37.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this err dpsi] = updateStochasticProcess( this, hp )

## 8.38 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/BlindKriging.m File Reference

## Classes

- class [BlindKriging](#)  
*A blind kriging surrogate model.*

## 8.38.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.

- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [Blind-Kriging](#)

## 8.39 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/polynomialCoding.m File Reference

### 8.39.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- [Blind Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [U dU]  
= polynomialCoding( this, samples, m, k, delta )

## 8.40 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/posteriorBeta.m File Reference

### 8.40.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature beta = posteriorBeta(this, R, U)

## 8.41 /home/ilm/projecten/meta/scripts/ooDACE/@BlindKriging/Rmatrix.m File Reference

## 8.41.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature out = Rmatrix(this, usedIdx)

## 8.42 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/CoKriging.m File Reference

### Classes

- class [CoKriging](#)  
*A cokriging surrogate model.*

### 8.42.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [Co-Kriging](#)

## 8.43 /home/ilm/projecten/meta/scripts/ooDACE/@CoKriging/setData.m File Reference

### 8.43.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setData(this, samples, values)

**8.44 /home/ilm/projecten/meta/scripts/ooDACE/@Kriging/setData.m File Reference****8.44.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setData(this, samples, values)



## 8.45 /home/ilm/projecten/meta/scripts/ooDACE/@Kriging/Kriging.m File Reference

### Classes

- class [Kriging](#)  
*A kriging surrogate model.*

### 8.45.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, *Advances in Engineering Software*, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, *International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE)*, Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [Kriging](#)

## 8.46 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrcubic.m File Reference

### Functions

- function [corrcubic](#) (var theta, var d)  
*Cubic correlation function.*

### 8.46.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dtheta, rho] = corrcubic(theta, d)

## 8.46.2 Function Documentation

8.46.2.1 function corrcubic ( var *theta*, var *d* )

Cubic correlation function.

## 8.47 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/correx.m File Reference

## Functions

- function [correx](#) (var *theta*, var *d*)  
*Exponential correlation function.*

## 8.47.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dtheta, rho] = correxp(theta, d)

## 8.47.2 Function Documentation

### 8.47.2.1 function correxp ( var theta, var d )

Exponential correlation function.

## 8.48 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrgauss.m File Reference

### Functions

- function [corrgauss](#) (var theta, var d)  
*Gaussian correlation function.*

### 8.48.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dtheta, rho] = corrgauss(theta, d)

### 8.48.2 Function Documentation

#### 8.48.2.1 function corrgauss ( var *theta*, var *d* )

Gaussian correlation function.

## 8.49 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrgaussp.m File Reference

### Functions

- function `corrgaussp` (var *hp*, var *d*)  
*Gaussian correlation function.*

### 8.49.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp rho] = corrgaussp(hp, d)

### 8.49.2 Function Documentation

#### 8.49.2.1 function corrgaussp ( var *hp*, var *d* )

Gaussian correlation function.

where *p* is also a hyperparameter

## 8.50 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrln.m File Reference

## Functions

- function `corrln` (var theta, var d)  
*Linear correlation function,.*

## 8.50.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dtheta, rho] = corrln(theta, d)

## 8.50.2 Function Documentation

## 8.50.2.1 function corrln ( var theta, var d )

Linear correlation function,.

## 8.51 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32.m File Reference

## Functions

- function `corrmatern32` (var hp, var d)  
*Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).*

## 8.51.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp, rho] = corrmatern32(hp, d)

## 8.51.2 Function Documentation

## 8.51.2.1 function corrmatern32 ( var hp, var d )

Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).

Implementation based on the book of Rasmussen et al.

- "Gaussian Processes for Machine Learning" (Chapter 4), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

The correlation function is:  $\text{corr} = f(\sqrt{\text{nud}} * \text{dist2}) * \exp(-\sqrt{\text{nud}} * \text{dist2})$  with  $\text{nud} = 2 * \nu$

## 8.52 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32\_iso.m File Reference

## Functions

- function [corrmatern32\\_iso](#) (var hp, var d)  
Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).

## 8.52.1 Detailed Description

**Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp, rho] = corrmatern32\_iso(hp, d)

**8.52.2 Function Documentation****8.52.2.1 function corrmatern32\_iso ( var hp, var d )**

Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).

Implementation based on the book of Rasmussen et al.

- "Gaussian Processes for Machine Learning" (Chapter 4), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

The correlation function is:  $\text{corr} = f(\sqrt{\text{nud}} \cdot \text{dist2}) \cdot \exp(-\sqrt{\text{nud}} \cdot \text{dist2})$  with  $\text{nud} = 2 \cdot \nu$

**8.53 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32\_variance.m File Reference****Functions**

- function [corrmatern32\\_variance](#) (var hp, var d)  
*Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).*

**8.53.1 Detailed Description****Authors**

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp, rho] = corrmatern32\_variance(hp, d)

## 8.53.2 Function Documentation

## 8.53.2.1 function corrmatern32\_variance ( var hp, var d )

Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).

Implementation based on the book of Rasmussen et al.

- "Gaussian Processes for Machine Learning" (Chapter 4), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

The correlation function is:  $\text{corr} = f(\sqrt{\text{nud}} \cdot \text{dist2}) \cdot \exp(-\sqrt{\text{nud}} \cdot \text{dist2})$  with  $\text{nud} = 2 \cdot \nu$

## 8.54 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m File Reference

## Functions

- function [corrmatern32iso](#) (var hp, var d)  
*Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).*

## 8.54.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)



## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp, rho] = corrmatern32iso(hp, d)

## 8.54.2 Function Documentation

## 8.54.2.1 function corrmatern32iso ( var hp, var d )

Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).

Implementation based on the book of Rasmussen et al.

- "Gaussian Processes for Machine Learning" (Chapter 4), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

The correlation function is:  $\text{corr} = f(\sqrt{\text{nud}} * \text{dist2}) * \exp(-\sqrt{\text{nud}} * \text{dist2})$  with  $\text{nud} = 2 * \nu$

## 8.55 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern52.m File Reference

## Functions

- function [corrmatern52](#) (var hp, var d)  
*Class of Matern covariance functions ( $\nu = \{1/2 \ 3/2 \ 5/2\}$ ).*

## 8.55.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp, rho] = corrmatern52(hp, d)

## 8.55.2 Function Documentation

## 8.55.2.1 function corrmatern52 ( var hp, var d )

Class of Matern covariance functions ( $\nu = \{1/2, 3/2, 5/2\}$ ).

Implementation based on the book of Rasmussen et al.

- "Gaussian Processes for Machine Learning" (Chapter 4), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

The correlation function is:  $\text{corr} = f(\sqrt{\text{nud}} \cdot \text{dist2}) \cdot \exp(-\sqrt{\text{nud}} \cdot \text{dist2})$  with  $\text{nud} = 2 \cdot \nu$

## 8.56 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrspherical.m File Reference

## Functions

- function [corrspherical](#) (var theta, var d)  
*Spherical correlation function.*

## 8.56.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dtheta, rho] = corrspherical(theta, d)

## 8.56.2 Function Documentation

## 8.56.2.1 function corrspherical ( var theta, var d )

Spherical correlation function.

## 8.57 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrspline.m File Reference

## Functions

- function [corrspline](#) (var theta, var d)  
*Cubic spline correlation function,.*

## 8.57.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dtheta, rho] = corrspline(theta, d)

## 8.57.2 Function Documentation

### 8.57.2.1 function corrspline ( var *theta*, var *d* )

Cubic spline correlation function,.

## 8.58 /home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/covmatern32.m File Reference

### Functions

- function [corrmatern32\\_variance](#) (var hp, var d)  
*Class of Matern covariance functions ( $\nu = \{1/2\ 3/2\ 5/2\}$ ).*

### 8.58.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [corr, dx, dhp, rho] = corrmatern32\_variance(hp, d)

### 8.58.2 Function Documentation

#### 8.58.2.1 function corrmatern32\_variance ( var *hp*, var *d* )

Class of Matern covariance functions ( $\nu = \{1/2, 3/2, 5/2\}$ ).

Implementation based on the book of Rasmussen et al.

- "Gaussian Processes for Machine Learning" (Chapter 4), C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006

The correlation function is:  $\text{corr} = f(\sqrt{\text{nud}} * \text{dist2}) * \exp(-\sqrt{\text{nud}} * \text{dist2})$  with  $\text{nud} = 2 * \nu$

## 8.59 /home/ilm/projecten/meta/scripts/ooDACE/dacefit.m File Reference

### Functions

- function `dacefit` (var *samples*, var *values*, var *regr*, var *corr*, var *theta0*, var *lb*, var *ub*)  
*Creates and fits a kriging model.*

### 8.59.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [krige perf] = dacefit(samples, values, regr, corr, theta0, lb, ub )

### 8.59.2 Function Documentation

#### 8.59.2.1 function dacefit ( var *samples*, var *values*, var *regr*, var *corr*, var *theta0*, var *lb*, var *ub* )

Creates and fits a kriging model.

DACE toolbox compatible interface to ooDACE (wrapper)

## Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix
<i>regr</i>	regression function (string)
<i>corr</i>	correlation function (string)
<i>theta0</i>	initial hyperparameter values
<i>lb</i>	lower bound of hyperparameters
<i>ub</i>	upper bound of hyperparameters

## Return values

<i>krige</i>	a ready-to-use kriging model
<i>perf</i>	a structure with some useful metrics

## 8.60 /home/ilm/projecten/meta/scripts/ooDACE/datasets/generateDatasets.m File Reference

## Functions

- function [generateDatasets](#) ()  
*Generates some example datasets used by the demo.*
- function [branin](#) (var x)
- function [birdfcn](#) (var x)

## 8.60.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [generateDatasets\(\)](#)

## 8.60.2 Function Documentation

## 8.60.2.1 function birdfcn ( var x )

## 8.60.2.2 function branin ( var x )

## 8.60.2.3 function generateDatasets ( )

Generates some example datasets used by the demo.

## 8.61 /home/ilm/projecten/meta/scripts/ooDACE/demo.m File Reference

## Functions

- function `demo` (var id)  
*Provides some examples on how to use the ooDACE toolbox.*

## 8.61.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, *Advances in Engineering Software*, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, *International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E)*, Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature k = demo(id)

## 8.61.2 Function Documentation

## 8.61.2.1 function demo ( var id )

Provides some examples on how to use the ooDACE toolbox.

- Test** Test case 1: Fits an ordinary kriging model on the branin function Covers: ordinary kriging interpolation, marginalLikelihood
- Test case 2: Fits an ordinary kriging model on the bird function Covers: ordinary kriging regression, re-interpolation of variance, pseudoLikelihood, likelihood debug plot
- Test case 3: Fits a blind kriging model on the branin function. Covers: blind kriging
- Test case 4: Fits a cokriging model on a mathematical 1D function. Covers: cokriging
- Test case 5: Fits a stochastic kriging model on the branin function plus some stochastic noise. Covers: stochastic kriging, sigma2 optimization

**Parameters**

<i>id</i>	id of the dataset to use (optional)
-----------	-------------------------------------

**Return values**

<i>k</i>	a ready-to-use kriging model
----------	------------------------------

**8.62 /home/ilm/projecten/meta/scripts/ooDACE/doc/mainpage.m File Reference****8.63 /home/ilm/projecten/meta/scripts/ooDACE/oodacefit.m File Reference****Functions**

- function [oodacefit](#) (var samples, var values, var userOpts)  
*Creates and fits a kriging model with sensible options.*

**8.63.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature k = oodacefit( samples, values, userOpts )



## 8.63.2 Function Documentation

8.63.2.1 function oodacefit ( var *samples*, var *values*, var *userOpts* )

Creates and fits a kriging model with sensible options.

```
For easy fitting when you don't care much about parameters.
The function tries to determine the best kriging model to use:
- samples is numeric array -> kriging
- samples is cell array -> cokriging
```

## Parameters

<i>samples</i>	input sample matrix
<i>values</i>	output value matrix
<i>userOpts</i>	struct of user options (optional) <ul style="list-style-type: none"> <li>• <i>userOpts.type</i> overrides the type of kriging model to use, e.g., "BasicGaussianProcess", "Kriging", "CoKriging", etc.</li> <li>• For regression kriging set the following fields: <ul style="list-style-type: none"> <li>– <i>userOpts.lambda0</i> = 0;</li> <li>– <i>userOpts.lambdaBounds</i> = [-5 ; 5]; % log scale</li> </ul> </li> <li>• For stochastic kriging set the following fields: <ul style="list-style-type: none"> <li>– <i>userOpts.Sigma</i> = <i>SigmaVector</i>; % variance of the output values</li> <li>– <i>userOpts.sigma20</i> = 1;</li> <li>– <i>userOpts.sigma2Bounds</i> = [0.001 ; 150];</li> </ul> </li> <li>• Please see the <a href="#">BasicGaussianProcess::getDefaultOptions()</a>, <a href="#">Kriging::getDefaultOptions()</a>, etc. methods for more options.</li> </ul>

## Return values

<i>k</i>	a ready-to-use kriging model
----------	------------------------------

## 8.64 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/getPopulationSize.m File Reference

## 8.64.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature size = getPopulationSize(this )

## 8.65 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getPopulationSize.m File Reference

### 8.65.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature size = getPopulationSize(this)

## 8.66 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/MatlabGA.m File Reference

#### Classes

- class [MatlabGA](#)  
*Wrapper around the matlab optimizers.*

### 8.66.1 Detailed Description

**Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[MatlabGA](#)

**8.67 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/optimize.m File Reference****8.67.1 Detailed Description****Authors**

Ivo Couckuyt

**Version**

1.4 (\$Revision\$)

**Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.

- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this, x, fval] = optimize(this, arg )

## 8.68 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/optimize.m File Reference

### 8.68.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this, xmin, fmin] = optimize(this, arg )

## 8.69 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/optimize.m File Reference

### 8.69.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this, x, fval] = optimize(this, arg )

## 8.70 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQLabOptimizer/optimize.m File Reference

## 8.70.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [this, x, fval] = optimize(this, arg )

## 8.71 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabGA/setInputConstraints.m File Reference

### 8.71.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setInputConstraints( this, con )

## 8.72 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/setInputConstraints.m File Reference

### 8.72.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setInputConstraints( this, con )

## 8.73 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInputConstraints.m File Reference

### 8.73.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setInputConstraints( this, con )

## 8.74 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@MatlabOptimizer/MatlabOptimizer.m File Reference

#### Classes

- class [MatlabOptimizer](#)  
*Wrapper around the matlab optimizers.*

## 8.74.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[MatlabOptimizer](#)

## 8.75 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getBounds.m File Reference

## 8.75.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.



- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [LB UB] = getBounds(this)

## 8.76 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getHint.m File Reference

### 8.76.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature value = getHint(this, key)

## 8.77 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInitialPopulation.m File Reference

### 8.77.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature pop = getInitialPopulation(this)

8.78 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getInputDimension.m File Reference

8.78.1 Detailed Description

Authors

Ivo Couckuyt

Version

1.4 (\$Revision\$)

Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature nvars = getInputDimension(this)

## 8.79 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/getOutputDimension.m File Reference

### 8.79.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature nobjectives = getOutputDimension(this)

## 8.80 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/Optimizer.m File Reference

### Classes

- class [Optimizer](#)  
*Abstract base class for an optimizer.*

### 8.80.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [Optimizer](#)

## 8.81 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setBounds.m File Reference

## 8.81.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setBounds(this, LB, UB)

## 8.82 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setDimensions.m File Reference

## 8.82.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setDimensions(this,inDim,outDim)

## 8.83 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setHint.m File Reference

## 8.83.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setHint(this, key, value)

## **8.84 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@Optimizer/setInitialPopulation.m File Reference**

### **8.84.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### **Date**

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature this = setInitialPopulation(this,pop)

## **8.85 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/simulator.m File Reference**

### **8.85.1 Detailed Description**

#### **Authors**

Ivo Couckuyt

#### **Version**

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature  
[outdic,f,ci,ce,cs,g,ai,ae] = simulator(indic, x, func, this, varargin )

#### 8.86 /home/ilm/projecten/meta/scripts/ooDACE/optimizers/@SQPLabOptimizer/SQPLabOptimizer.m File Reference

#### Classes

- class [SQPLabOptimizer](#)  
*Wrapper around the SQPLab optimization package.*

#### 8.86.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [SQPLabOptimizer](#)

## 8.87 /home/ilm/projecten/meta/scripts/ooDACE/plotKrigingModel.m File Reference

## Functions

- function `plotKrigingModel` (var *k*, var *LB*, var *UB*)  
Generate some plots of a *Kriging* model.

## 8.87.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind *Kriging*: Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature *h* = `plotKrigingModel(k, LB, UB)`

## 8.87.2 Function Documentation

8.87.2.1 function `plotKrigingModel` ( var *k*, var *LB*, var *UB* )

Generate some plots of a *Kriging* model.

Only for 1D/2D models.

## Parameters

<i>k</i>	<i>Kriging</i> model
<i>LB</i>	lowerbound of input parameters (optional)
<i>UB</i>	upperbound of input parameters (optional)

## Return values

<i>h</i>	handles to the figure windows
----------	-------------------------------



## 8.88 /home/ilm/projecten/meta/scripts/ooDACE/predictor.m File Reference

## Functions

- function [predictor](#) (var points, var krige)  
*Calculates prediction of a kriging model.*

## 8.88.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [y, or1, or2, dmse] = predictor(points, krige)

## 8.88.2 Function Documentation

## 8.88.2.1 function predictor ( var points, var krige )

Calculates prediction of a kriging model.

DACE toolbox compatible interface to ooDACE (wrapper)

## Parameters

<i>points</i>	input points matrix
<i>krige</i>	kriging model

## Return values

<i>y</i>	prediction values
<i>or1</i>	
<i>or2</i>	
<i>dmse</i>	

## 8.89 /home/ilm/projecten/meta/scripts/ooDACE/runBlindKrigingExamples.m File Reference

### Functions

- function [runBlindKrigingExamples](#) ()  
*Fits blind kriging models to some datasets.*

### 8.89.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [runBlindKrigingExamples\(\)](#)

### 8.89.2 Function Documentation

#### 8.89.2.1 function runBlindKrigingExamples ( )

Fits blind kriging models to some datasets.

## 8.90 /home/ilm/projecten/meta/scripts/ooDACE/runRegressionTests.m File Reference

### Functions

- function [runRegressionTests](#) (var idx, var regressionTestsDir, var saveResults)  
*Runs the regression test suite.*
- function [cmp\\_combinedRelative](#) (var a, var b)

## 8.90.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature run-RegressionTests(idx, regressionTestsDir, saveResults)

## 8.90.2 Function Documentation

## 8.90.2.1 function cmp\_combinedRelative ( var a, var b )

## 8.90.2.2 function runRegressionTests ( var idx, var regressionTestsDir, var saveResults )

Runs the regression test suite.

## Parameters

<i>idx</i>	vector indicating the tests to run. Default: 1:5 (optional)
<i>regressionTestsDir</i>	directory of results (optional)
<i>saveResults</i>	saves the results to regressionTestsDir. Default: false (optional)

## Note

Tests should be deterministic, no fixed random state needed (yet)  
The symbolic expression for the regression, correlation function and getExpression can not be compared directly

## 8.91 /home/ilm/projecten/meta/scripts/ooDACE/startup.m File Reference

## Functions

- function `startup` ()  
*Initializes the ooDACE toolbox.*

### 8.91.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [startup\(\)](#)

### 8.91.2 Function Documentation

#### 8.91.2.1 function `startup` ( )

Initializes the ooDACE toolbox.

Setup the toolbox path. Needs to be called once before using the ooDACE toolbox.

## 8.92 /home/ilm/projecten/meta/scripts/ooDACE/tools/averageEuclideanError.m File Reference

## Functions

- function `averageEuclideanError` (var a, var b)  
*Computes the average euclidean error (AEE) between a (true) and b (predicted).*

## 8.92.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature e = averageEuclideanError( a,b )

## 8.92.2 Function Documentation

## 8.92.2.1 function averageEuclideanError ( var a, var b )

Computes the average euclidean error (AEE) between a (true) and b (predicted).

From: Rong Li and Zhanlue Zhao, Evaluation of estimation algorithms part I: incomprehensive measures of performance, IEEE Transactions on Aerospace and Electronic Systems, vol. 42, no. 4, pp. 1340-1358, 2006

## 8.93 /home/ilm/projecten/meta/scripts/ooDACE/tools/buildVandermondeMatrix.m File Reference

## Functions

- function [buildVandermondeMatrix](#) (var samples, var degrees, var baseFunctions)  
*Build multidimensional Vandermonde like matrix for interpolation and/or evaluation of multidimensional polynomials.*

## 8.93.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature [m dm] = buildVandermondeMatrix( samples, degrees, baseFunctions )

## 8.93.2 Function Documentation

## 8.93.2.1 function buildVandermondeMatrix ( var samples, var degrees, var baseFunctions )

Build multidimensional Vandermonde like matrix for interpolation and/or evaluation of multidimensional polynomials.

## 8.94 /home/ilm/projecten/meta/scripts/ooDACE/tools/cfix.m File Reference

## Functions

- function [cfix](#) (var x, var d, var err)  
*This function "fixes" a cell array.*

## 8.94.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature y = cfix( x,d,err )

#### 8.94.2 Function Documentation

##### 8.94.2.1 function cfix ( var x, var d, var err )

This function “fixes” a cell array.

Either a constant, a one element cell array or a length ‘d’ cell array can be passed to this function. The function will return a length ‘d’ cell array, duplicating the input if necessary

#### 8.95 /home/ilm/projecten/meta/scripts/ooDACE/tools/makeEvalGrid.m File Reference

##### Functions

- function [makeEvalGrid](#) (var gridpoints, var gridsize)  
*Low-level procedure, makes a ‘prod(gridsize)’ by ‘length(gridsize)’ array, where each row is a different vector, where the i’th element is a number out of ‘gridpoints[i]’.*

##### 8.95.1 Detailed Description

###### Authors

Ivo Couckuyt

###### Version

1.4 (\$Revision\$)

###### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature eval-grid = makeEvalGrid( gridpoints, gridsize )

## 8.95.2 Function Documentation

## 8.95.2.1 function makeEvalGrid ( var gridpoints, var gridsize )

Low-level procedure, makes a 'prod(gridsize)' by 'length(gridsize)' array, where each row is a different vector, where the i'th element is a number out of 'gridpoints{i}'.

The 'gridsize' parameter may be omitted, in that case it's generated from 'gridpoints'. This function produces grids in the expected order, which is now also used for gridded dataset files. Use this instead of makeEvalGridInverted if possible.

Example: makeEvalGrid( { [-1, .5], [-1 0 1], [.2 .3] }, [2 3 2] ) ans = -1.0000 -1.0000 0.2000 -1.0000 -1.0000 0.3000 -1.0000 0.0000 0.2000 -1.0000 0.0000 0.3000 -1.0000 1.0000 0.2000 -1.0000 1.0000 0.3000 0.5000 -1.0000 0.2000 0.5000 -1.0000 0.3000 0.5000 0.0000 0.2000 0.5000 0.0000 0.3000 0.5000 1.0000 0.2000 0.5000 1.0000 0.3000

## 8.96 /home/ilm/projecten/meta/scripts/ooDACE/tools/makeGrid.m File Reference

## Functions

- function [makeGrid](#) (var sizes)

*Construct a matrix of size prod(sizes) by length(sizes)" where the rows represent all gridpoints on asizes" sized grid.*

## 8.96.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$

Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature g = makeGrid( sizes )



## 8.96.2 Function Documentation

### 8.96.2.1 function makeGrid ( var sizes )

Construct a matrix of size `prod(sizes)` by `length(sizes)` where the rows represent all gridpoints on a `sizes` sized grid.

Example:

```
makeGrid( [3 1 2 2] )
```

```
ans = 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2 2 2 1 1 1 2 1 1 2 2 1 2 1 2 1 2 2 3 1 1 1 3 1 1 2 3 1 2 1 3 1 2 2
```

## 8.97 /home/ilm/projecten/meta/scripts/ooDACE/tools/mergeStruct.m File Reference

### Functions

- function `mergeStruct` (var s1, var s2, var destFieldExist)  
*Copies field of s2 over to s1.*

### 8.97.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, *Advances in Engineering Software*, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, *International Journal of RF and Microwave Computer-Aided Engineering (RFMiCAE)*, Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature o = mergeStruct( s1, s2, destFieldExist )

## 8.97.2 Function Documentation

### 8.97.2.1 function mergeStruct ( var s1, var s2, var destFieldExist )

Copies field of s2 over to s1.

## Parameters

<i>s1</i>	destination structure
<i>s2</i>	source structure
<i>destFieldExist</i>	<ul style="list-style-type: none"> <li>• -1: always copy</li> <li>• false: only copy when destination field does NOT exist</li> <li>• true: only copy when destination field exist</li> </ul>

## 8.98 /home/ilm/projecten/meta/scripts/ooDACE/tools/plotScatteredData.m File Reference

## Functions

- function [plotScatteredData](#) (var varargin)  
*A nice, easy way to create a surf plot of scatterd data.*

## 8.98.1 Detailed Description

## Authors

Ivo Couckuyt

## Version

1.4 (\$Revision\$)

## Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Advances in Engineering Software, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E), Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature h = plotScatteredData(varargin)

## 8.98.2 Function Documentation

## 8.98.2.1 function plotScatteredData ( var varargin )

A nice, easy way to create a surf plot of scatterd data.

## 8.99 /home/ilm/projecten/meta/scripts/ooDACE/tools/powerBase.m File Reference

### Functions

- function `powerBase` (var x, var d)  
*Powers of all the variables as a base for the interpolant.*

### 8.99.1 Detailed Description

#### Authors

Ivo Couckuyt

#### Version

1.4 (\$Revision\$)

#### Date

\$LastChangedDate\$  
Copyright 2010-2013

This file is part of the ooDACE toolbox and you can redistribute it and/or modify it under the terms of the GNU Affero General Public License version 3 as published by the Free Software Foundation. With the additional provision that a commercial license must be purchased if the ooDACE toolbox is used, modified, or extended in a commercial setting. For details see the included LICENSE.txt file. When referring to the ooDACE toolbox please make reference to the corresponding publications:

- Blind [Kriging](#): Implementation and performance analysis I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, *Advances in Engineering Software*, Vol. 49, pp. 1-13, July 2012.
- Surrogate-based infill optimization applied to electromagnetic problems I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, *International Journal of RF and Microwave Computer-Aided Engineering (RFMiCA-E)*, Special Issue on Advances in Design Optimization of Microwave/RF Circuits and Systems, Vol. 20, No. 5, pp. 492-501, September 2010.

Contact : [ivo.couckuyt@ugent.be](mailto:ivo.couckuyt@ugent.be) - <http://sumo.intec.ugent.be/?q=ooDACE> Signature base  
= powerBase( x, d )

### 8.99.2 Function Documentation

#### 8.99.2.1 function powerBase ( var x, var d )

Powers of all the variables as a base for the interpolant.

## Index

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 105

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 106

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 107

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 108

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 109

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 109

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 110

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 111

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 112

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 113

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 114

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 115

/home/ilm/projecten/meta/scripts/ooDACE/basisfunctions/corrmatern32iso.m, 116

/home/ilm/projecten/meta/scripts/ooDACE/dacefit.m, 117

/home/ilm/projecten/meta/scripts/ooDACE/datasets/generate-Datasets.m, 118

/home/ilm/projecten/meta/scripts/ooDACE/demo.m, 119

/home/ilm/projecten/meta/scripts/ooDACE/doc/mainpage.m, 120

/home/ilm/projecten/meta/scripts/ooDACE/oodacefit.m, 120

/home/ilm/projecten/meta/scripts/ooDACE/plotKriging-Model.m, 136

/home/ilm/projecten/meta/scripts/ooDACE/predictor.m, 137

/home/ilm/projecten/meta/scripts/ooDACE/runBlind-KrigingExamples.m, 138

/home/ilm/projecten/meta/scripts/ooDACE/runRegression-Tests.m, 138

/home/ilm/projecten/meta/scripts/ooDACE/startup.m, 139

/home/ilm/projecten/meta/scripts/ooDACE/tools/average-EuclideanError.m, 140

/home/ilm/projecten/meta/scripts/ooDACE/tools/build-VandermondeMatrix.m, 141

/home/ilm/projecten/meta/scripts/ooDACE/tools/cfix.m, 142

/home/ilm/projecten/meta/scripts/ooDACE/tools/make-EvalGrid.m, 143

/home/ilm/projecten/meta/scripts/ooDACE/tools/make-Grid.m, 144

/home/ilm/projecten/meta/scripts/ooDACE/tools/merge-Struct.m, 145

/home/ilm/projecten/meta/scripts/ooDACE/tools/plot-ScatteredData.m, 146

/home/ilm/projecten/meta/scripts/ooDACE/tools/power-Base.m, 147

alpha

BasicGaussianProcess, 19

BlindKriging, 33

Cokriging, 47

Kriging, 60

averageEuclideanError

averageEuclideanError.m, 141

averageEuclideanError.m

averageEuclideanError, 141

BasicGaussianProcess, 7

alpha, 19

BasicGaussianProcess, 11

BasicGaussianProcess, 11

C, 19

C\_reinterp, 19

cleanup, 11

correlationFcn, 19

correlationFunction, 11

cvpe, 11

display, 11

dist, 19

distIdxPsi, 19

extrinsicCorrelationMatrix, 12

fit, 12

Ft, 19

Ft\_reinterp, 19

gamma, 19

generateDegrees, 12

getCorrelationMatrix, 13

getDefaultOptions, 13

getExpression, 13

getHyperparameters, 13

getProcessVariance, 13

getRegressionMatrix, 13

getRho, 13

getSamples, 13

getSigma, 14

getValues, 14

HP, 19

hyperparameters, 19

hyperparameters0, 19

imse, 14

intrinsicCovarianceMatrix, 14

LAMBDA, 20

marginalLikelihood, 14

mseTestset, 15

optimIdx, 20

optimNrParameters, 20

- options, 20
- plotVariogram, 15
- predict, 15
- predict\_derivatives, 16
- predict\_limit, 16
- pseudoLikelihood, 16
- R, 20
- R\_reinterp, 20
- RHO, 20
- rcValues, 17
- regressionFcn, 20
- regressionFunction, 17
- regressionMatrix, 17
- rho, 20
- SIGMA2, 20
- setData, 17
- setOption, 18
- Sigma, 20
- sigma2, 20
- sigma2\_reinterp, 20
- tau2, 20
- tuneParameters, 18
- updateModel, 18
- updateRegression, 18
- updateStochasticProcess, 18
- birdfcn
  - generateDatasets.m, 119
- BlindKriging, 21
  - alpha, 33
  - BlindKriging, 25
  - BlindKriging, 25
  - C, 33
  - C\_reinterp, 33
  - cleanup, 25
  - correlationFcn, 33
  - correlationFunction, 25
  - cvpe, 25
  - display, 25
  - dist, 33
  - distIdxPsi, 33
  - extrinsicCorrelationMatrix, 25
  - fit, 26
  - Ft, 33
  - Ft\_reinterp, 33
  - gamma, 33
  - generateDegrees, 26
  - getCorrelationMatrix, 26
  - getDefaultOptions, 26
  - getExpression, 27
  - getHyperparameters, 27
  - getProcessVariance, 27
  - getRegressionMatrix, 27
  - getRho, 27
  - getSamples, 27
  - getSigma, 27
  - getStatistics, 27
  - getValues, 28
  - HP, 33
  - hyperparameters, 34
  - hyperparameters0, 34
  - imse, 28
  - intrinsicCovarianceMatrix, 28
  - LAMBDA, 34
  - marginalLikelihood, 28
  - mseTestset, 29
  - optimIdx, 34
  - optimNrParameters, 34
  - options, 34
  - plotVariogram, 29
  - predict, 29
  - predict\_derivatives, 30
  - predict\_limit, 30
  - pseudoLikelihood, 30
  - R, 34
  - R\_reinterp, 34
  - RHO, 34
  - rcValues, 31
  - regressionFcn, 34
  - regressionFunction, 31
  - regressionMatrix, 31
  - rho, 34
  - SIGMA2, 34
  - setData, 31
  - setOption, 32
  - Sigma, 34
  - sigma2, 34
  - sigma2\_reinterp, 34
  - tau2, 35
  - tuneParameters, 32
  - updateModel, 32
  - updateRegression, 32
  - updateStochasticProcess, 33
- branin
  - generateDatasets.m, 119
- buildVandermondeMatrix
  - buildVandermondeMatrix.m, 142
- buildVandermondeMatrix.m
  - buildVandermondeMatrix, 142
- C
  - BasicGaussianProcess, 19
  - BlindKriging, 33
  - CoKriging, 47
  - Kriging, 60
- C\_reinterp
  - BasicGaussianProcess, 19
  - BlindKriging, 33
  - CoKriging, 47
  - Kriging, 60
- cfix
  - cfix.m, 143
- cfix.m
  - cfix, 143
- cleanup
  - BasicGaussianProcess, 11
  - BlindKriging, 25
  - CoKriging, 38

- Kriging, [52](#)
- cmp\_combinedRelative
  - runRegressionTests.m, [139](#)
- CoKriging, [35](#)
  - alpha, [47](#)
  - C, [47](#)
  - C\_reinterp, [47](#)
  - cleanup, [38](#)
  - CoKriging, [38](#)
  - CoKriging, [38](#)
  - correlationFcn, [47](#)
  - correlationFunction, [38](#)
  - cvpe, [39](#)
  - display, [39](#)
  - dist, [47](#)
  - distIdxPsi, [47](#)
  - extrinsicCorrelationMatrix, [39](#)
  - fit, [40](#)
  - Ft, [47](#)
  - Ft\_reinterp, [47](#)
  - gamma, [47](#)
  - generateDegrees, [40](#)
  - getCorrelationMatrix, [40](#)
  - getDefaultOptions, [40](#)
  - getExpression, [40](#)
  - getHyperparameters, [41](#)
  - getProcessVariance, [41](#)
  - getRegressionMatrix, [41](#)
  - getRho, [41](#)
  - getSamples, [41](#)
  - getSamplesIdx, [41](#)
  - getSigma, [41](#)
  - getValues, [41](#)
  - getValuesIdx, [41](#)
  - HP, [47](#)
  - hyperparameters, [47](#)
  - hyperparameters0, [47](#)
  - imse, [42](#)
  - intrinsicCovarianceMatrix, [42](#)
  - LAMBDA, [47](#)
  - marginalLikelihood, [42](#)
  - mseTestset, [43](#)
  - optimIdx, [48](#)
  - optimNrParameters, [48](#)
  - options, [48](#)
  - plotVariogram, [43](#)
  - predict, [43](#)
  - predict\_derivatives, [44](#)
  - predict\_limit, [44](#)
  - pseudoLikelihood, [44](#)
  - R, [48](#)
  - R\_reinterp, [48](#)
  - RHO, [48](#)
  - rcValues, [44](#)
  - regressionFcn, [48](#)
  - regressionFunction, [45](#)
  - regressionMatrix, [45](#)
  - rho, [48](#)
  - SIGMA2, [48](#)
  - setData, [45](#)
  - setOption, [45](#)
  - Sigma, [48](#)
  - sigma2, [48](#)
  - sigma2\_reinterp, [48](#)
  - tau2, [48](#)
  - tuneParameters, [46](#)
  - updateModel, [46](#)
  - updateRegression, [46](#)
  - updateStochasticProcess, [46](#)
- corrcubic
  - corrcubic.m, [106](#)
- corrcubic.m
  - corrcubic, [106](#)
- correlationFcn
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
  - CoKriging, [47](#)
  - Kriging, [60](#)
- correlationFunction
  - BasicGaussianProcess, [11](#)
  - BlindKriging, [25](#)
  - CoKriging, [38](#)
  - Kriging, [52](#)
- correx
  - correx.m, [107](#)
- correx.m
  - correx, [107](#)
- corrgauss
  - corrgauss.m, [108](#)
- corrgauss.m
  - corrgauss, [108](#)
- corrgaussp
  - corrgaussp.m, [108](#)
- corrgaussp.m
  - corrgaussp, [108](#)
- corrlin
  - corrlin.m, [109](#)
- corrlin.m
  - corrlin, [109](#)
- corrmatern32
  - corrmatern32.m, [110](#)
- corrmatern32.m
  - corrmatern32, [110](#)
- corrmatern32\_iso
  - corrmatern32\_iso.m, [111](#)
- corrmatern32\_iso.m
  - corrmatern32\_iso, [111](#)
- corrmatern32\_variance
  - corrmatern32\_variance.m, [112](#)
  - covmatern32.m, [117](#)
- corrmatern32\_variance.m
  - corrmatern32\_variance, [112](#)
- corrmatern32iso
  - corrmatern32iso.m, [113](#)
- corrmatern32iso.m
  - corrmatern32iso, [113](#)

- corrmatern52
  - corrmatern52.m, [114](#)
- corrmatern52.m
  - corrmatern52, [114](#)
- corrnspherical
  - corrnspherical.m, [115](#)
- corrnspherical.m
  - corrnspherical, [115](#)
- corrnspline
  - corrnspline.m, [116](#)
- corrnspline.m
  - corrnspline, [116](#)
- covmatern32.m
  - corrmatern32\_variance, [117](#)
- cvpe
  - BasicGaussianProcess, [11](#)
  - BlindKriging, [25](#)
  - CoKriging, [39](#)
  - Kriging, [53](#)
- dacefit
  - dacefit.m, [117](#)
- dacefit.m
  - dacefit, [117](#)
- demo
  - demo.m, [119](#)
- demo.m
  - demo, [119](#)
- display
  - BasicGaussianProcess, [11](#)
  - BlindKriging, [25](#)
  - CoKriging, [39](#)
  - Kriging, [53](#)
- dist
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
  - CoKriging, [47](#)
  - Kriging, [60](#)
- distIdxPsi
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
  - CoKriging, [47](#)
  - Kriging, [60](#)
- extrinsicCorrelationMatrix
  - BasicGaussianProcess, [12](#)
  - BlindKriging, [25](#)
  - CoKriging, [39](#)
  - Kriging, [53](#)
- fit
  - BasicGaussianProcess, [12](#)
  - BlindKriging, [26](#)
  - CoKriging, [40](#)
  - Kriging, [53](#)
- Ft
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
  - CoKriging, [47](#)
  - Kriging, [60](#)
- Ft\_reinterp
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
  - CoKriging, [47](#)
  - Kriging, [60](#)
- gamma
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
  - CoKriging, [47](#)
  - Kriging, [60](#)
- generateDatasets
  - generateDatasets.m, [119](#)
- generateDatasets.m
  - birdfcn, [119](#)
  - branin, [119](#)
  - generateDatasets, [119](#)
- generateDegrees
  - BasicGaussianProcess, [12](#)
  - BlindKriging, [26](#)
  - CoKriging, [40](#)
  - Kriging, [53](#)
- getBounds
  - MatlabGA, [63](#)
  - MatlabOptimizer, [67](#)
  - Optimizer, [71](#)
  - SQPLabOptimizer, [75](#)
- getCorrelationMatrix
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [26](#)
  - CoKriging, [40](#)
  - Kriging, [54](#)
- getDefaultOptions
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [26](#)
  - CoKriging, [40](#)
  - Kriging, [54](#)
- getExpression
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [27](#)
  - CoKriging, [40](#)
  - Kriging, [54](#)
- getHint
  - MatlabGA, [63](#)
  - MatlabOptimizer, [67](#)
  - Optimizer, [71](#)
  - SQPLabOptimizer, [75](#)
- getHyperparameters
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [27](#)
  - CoKriging, [41](#)
  - Kriging, [54](#)
- getInitialPopulation
  - MatlabGA, [64](#)
  - MatlabOptimizer, [67](#)
  - Optimizer, [71](#)
  - SQPLabOptimizer, [76](#)
- getInputDimension

- MatlabGA, [64](#)
- MatlabOptimizer, [68](#)
- Optimizer, [72](#)
- SQPLabOptimizer, [76](#)
- getOutputDimension
  - MatlabGA, [64](#)
  - MatlabOptimizer, [68](#)
  - Optimizer, [72](#)
  - SQPLabOptimizer, [76](#)
- getPopulationSize
  - MatlabGA, [64](#)
  - MatlabOptimizer, [68](#)
  - Optimizer, [72](#)
  - SQPLabOptimizer, [76](#)
- getProcessVariance
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [27](#)
  - CoKriging, [41](#)
  - Kriging, [54](#)
- getRegressionMatrix
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [27](#)
  - CoKriging, [41](#)
  - Kriging, [54](#)
- getRho
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [27](#)
  - CoKriging, [41](#)
  - Kriging, [54](#)
- getSamples
  - BasicGaussianProcess, [13](#)
  - BlindKriging, [27](#)
  - CoKriging, [41](#)
  - Kriging, [54](#)
- getSamplesIdx
  - CoKriging, [41](#)
- getSigma
  - BasicGaussianProcess, [14](#)
  - BlindKriging, [27](#)
  - CoKriging, [41](#)
  - Kriging, [55](#)
- getState
  - MatlabGA, [64](#)
  - MatlabOptimizer, [68](#)
  - Optimizer, [72](#)
  - SQPLabOptimizer, [76](#)
- getStatistics
  - BlindKriging, [27](#)
- getValues
  - BasicGaussianProcess, [14](#)
  - BlindKriging, [28](#)
  - CoKriging, [41](#)
  - Kriging, [55](#)
- getValuesIdx
  - CoKriging, [41](#)
- HP
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [33](#)
- CoKriging, [47](#)
- Kriging, [60](#)
- hyperparameters
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [34](#)
  - CoKriging, [47](#)
  - Kriging, [61](#)
- hyperparameters0
  - BasicGaussianProcess, [19](#)
  - BlindKriging, [34](#)
  - CoKriging, [47](#)
  - Kriging, [61](#)
- imse
  - BasicGaussianProcess, [14](#)
  - BlindKriging, [28](#)
  - CoKriging, [42](#)
  - Kriging, [55](#)
- intrinsicCovarianceMatrix
  - BasicGaussianProcess, [14](#)
  - BlindKriging, [28](#)
  - CoKriging, [42](#)
  - Kriging, [55](#)
- Kriging, [49](#)
  - alpha, [60](#)
  - C, [60](#)
  - C\_reinterp, [60](#)
  - cleanup, [52](#)
  - correlationFcn, [60](#)
  - correlationFunction, [52](#)
  - cvpe, [53](#)
  - display, [53](#)
  - dist, [60](#)
  - distIdxPsi, [60](#)
  - extrinsicCorrelationMatrix, [53](#)
  - fit, [53](#)
  - Ft, [60](#)
  - Ft\_reinterp, [60](#)
  - gamma, [60](#)
  - generateDegrees, [53](#)
  - getCorrelationMatrix, [54](#)
  - getDefaultOptions, [54](#)
  - getExpression, [54](#)
  - getHyperparameters, [54](#)
  - getProcessVariance, [54](#)
  - getRegressionMatrix, [54](#)
  - getRho, [54](#)
  - getSamples, [54](#)
  - getSigma, [55](#)
  - getValues, [55](#)
  - HP, [60](#)
  - hyperparameters, [61](#)
  - hyperparameters0, [61](#)
  - imse, [55](#)
  - intrinsicCovarianceMatrix, [55](#)
  - Kriging, [52](#)
  - LAMBDA, [61](#)
  - marginalLikelihood, [56](#)



- mseTestset, 56
  - optimIdx, 61
  - optimNrParameters, 61
  - options, 61
  - plotVariogram, 56
  - predict, 57
  - predict\_derivatives, 57
  - predict\_limit, 57
  - pseudoLikelihood, 57
  - R, 61
  - R\_reinterp, 61
  - RHO, 61
  - rcValues, 58
  - regressionFcn, 61
  - regressionFunction, 58
  - regressionMatrix, 58
  - rho, 61
  - SIGMA2, 61
  - setData, 59
  - setOption, 59
  - Sigma, 61
  - sigma2, 61
  - sigma2\_reinterp, 61
  - tau2, 62
  - tuneParameters, 59
  - updateModel, 59
  - updateRegression, 59
  - updateStochasticProcess, 60
- LAMBDA
- BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 47
  - Kriging, 61
- makeEvalGrid
- makeEvalGrid.m, 144
- makeEvalGrid.m
- makeEvalGrid, 144
- makeGrid
- makeGrid.m, 145
- makeGrid.m
- makeGrid, 145
- marginalLikelihood
- BasicGaussianProcess, 14
  - BlindKriging, 28
  - CoKriging, 42
  - Kriging, 56
- MatlabGA, 62
- getBounds, 63
  - getHint, 63
  - getInitialPopulation, 64
  - getInputDimension, 64
  - getOutputDimension, 64
  - getPopulationSize, 64
  - getState, 64
  - MatlabGA, 63
  - MatlabGA, 63
  - optimize, 64
  - setBounds, 64
  - setDimensions, 64
  - setHint, 65
  - setInitialPopulation, 65
  - setInputConstraints, 65
  - setState, 65
- MatlabOptimizer, 65
- getBounds, 67
  - getHint, 67
  - getInitialPopulation, 67
  - getInputDimension, 68
  - getOutputDimension, 68
  - getPopulationSize, 68
  - getState, 68
  - MatlabOptimizer, 67
  - MatlabOptimizer, 67
  - optimize, 68
  - setBounds, 68
  - setDimensions, 68
  - setHint, 69
  - setInitialPopulation, 69
  - setInputConstraints, 69
  - setState, 69
- mergeStruct
- mergeStruct.m, 145
- mergeStruct.m
- mergeStruct, 145
- mseTestset
- BasicGaussianProcess, 15
  - BlindKriging, 29
  - CoKriging, 43
  - Kriging, 56
- oodacefit
- oodacefit.m, 121
- oodacefit.m
- oodacefit, 121
- optimIdx
- BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- optimNrParameters
- BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- optimize
- MatlabGA, 64
  - MatlabOptimizer, 68
  - Optimizer, 72
  - SQPLabOptimizer, 76
- Optimizer, 69
- getBounds, 71
  - getHint, 71
  - getInitialPopulation, 71
  - getInputDimension, 72
  - getOutputDimension, 72
  - getPopulationSize, 72

- getState, 72
  - optimize, 72
  - Optimizer, 71
  - setBounds, 72
  - setDimensions, 73
  - setHint, 73
  - setInitialPopulation, 73
  - setInputConstraints, 73
  - setState, 73
- options
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- plotKrigingModel
  - plotKrigingModel.m, 136
- plotKrigingModel.m
  - plotKrigingModel, 136
- plotScatteredData
  - plotScatteredData.m, 146
- plotScatteredData.m
  - plotScatteredData, 146
- plotVariogram
  - BasicGaussianProcess, 15
  - BlindKriging, 29
  - CoKriging, 43
  - Kriging, 56
- powerBase
  - powerBase.m, 147
- powerBase.m
  - powerBase, 147
- predict
  - BasicGaussianProcess, 15
  - BlindKriging, 29
  - CoKriging, 43
  - Kriging, 57
- predict\_derivatives
  - BasicGaussianProcess, 16
  - BlindKriging, 30
  - CoKriging, 44
  - Kriging, 57
- predict\_limit
  - BasicGaussianProcess, 16
  - BlindKriging, 30
  - CoKriging, 44
  - Kriging, 57
- predictor
  - predictor.m, 137
- predictor.m
  - predictor, 137
- pseudoLikelihood
  - BasicGaussianProcess, 16
  - BlindKriging, 30
  - CoKriging, 44
  - Kriging, 57
- R
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- R\_reinterp
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- RHO
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- rcValues
  - BasicGaussianProcess, 17
  - BlindKriging, 31
  - CoKriging, 44
  - Kriging, 58
- regressionFcn
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- regressionFunction
  - BasicGaussianProcess, 17
  - BlindKriging, 31
  - CoKriging, 45
  - Kriging, 58
- regressionMatrix
  - BasicGaussianProcess, 17
  - BlindKriging, 31
  - CoKriging, 45
  - Kriging, 58
- rho
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- runBlindKrigingExamples
  - runBlindKrigingExamples.m, 138
- runBlindKrigingExamples.m
  - runBlindKrigingExamples, 138
- runRegressionTests
  - runRegressionTests.m, 139
- runRegressionTests.m
  - cmp\_combinedRelative, 139
  - runRegressionTests, 139
- SIGMA2
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- SQPLabOptimizer, 74
  - getBounds, 75
  - getHint, 75
  - getInitialPopulation, 76
  - getInputDimension, 76
  - getOutputDimension, 76

- getPopulationSize, 76
- getState, 76
- optimize, 76
- SQPLabOptimizer, 75
- setBounds, 76
- setDimensions, 77
- setHint, 77
- setInitialPopulation, 77
- setInputConstraints, 77
- setState, 77
- SQPLabOptimizer, 75
- setBounds
  - MatlabGA, 64
  - MatlabOptimizer, 68
  - Optimizer, 72
  - SQPLabOptimizer, 76
- setData
  - BasicGaussianProcess, 17
  - BlindKriging, 31
  - CoKriging, 45
  - Kriging, 59
- setDimensions
  - MatlabGA, 64
  - MatlabOptimizer, 68
  - Optimizer, 73
  - SQPLabOptimizer, 77
- setHint
  - MatlabGA, 65
  - MatlabOptimizer, 69
  - Optimizer, 73
  - SQPLabOptimizer, 77
- setInitialPopulation
  - MatlabGA, 65
  - MatlabOptimizer, 69
  - Optimizer, 73
  - SQPLabOptimizer, 77
- setInputConstraints
  - MatlabGA, 65
  - MatlabOptimizer, 69
  - Optimizer, 73
  - SQPLabOptimizer, 77
- setOption
  - BasicGaussianProcess, 18
  - BlindKriging, 32
  - CoKriging, 45
  - Kriging, 59
- setState
  - MatlabGA, 65
  - MatlabOptimizer, 69
  - Optimizer, 73
  - SQPLabOptimizer, 77
- Sigma
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- sigma2
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- sigma2\_reinterp
  - BasicGaussianProcess, 20
  - BlindKriging, 34
  - CoKriging, 48
  - Kriging, 61
- startup
  - startup.m, 140
- startup.m
  - startup, 140
- tau2
  - BasicGaussianProcess, 20
  - BlindKriging, 35
  - CoKriging, 48
  - Kriging, 62
- tuneParameters
  - BasicGaussianProcess, 18
  - BlindKriging, 32
  - CoKriging, 46
  - Kriging, 59
- updateModel
  - BasicGaussianProcess, 18
  - BlindKriging, 32
  - CoKriging, 46
  - Kriging, 59
- updateRegression
  - BasicGaussianProcess, 18
  - BlindKriging, 32
  - CoKriging, 46
  - Kriging, 59
- updateStochasticProcess
  - BasicGaussianProcess, 18
  - BlindKriging, 33
  - CoKriging, 46
  - Kriging, 60