

# Sgtelib user guide

Bastien Talgorn  
McGill University

January 11, 2017

## Contents

<b>1</b>	<b>General use of sgtelib</b>	<b>2</b>
1.1	GENERAL . . . . .	2
1.2	PREDICT . . . . .	3
1.3	BEST . . . . .	3
1.4	SERVER . . . . .	3
1.5	MODEL . . . . .	3
1.6	FIELD . . . . .	3
<b>2</b>	<b>Types of models</b>	<b>4</b>
2.1	PRS . . . . .	4
2.2	PRS_EDGE . . . . .	4
2.3	PRS_CAT . . . . .	4
2.4	RBF . . . . .	4
2.5	KS . . . . .	5
2.6	KRIGING . . . . .	5
2.7	LOWESS . . . . .	5
2.8	ENSEMBLE . . . . .	6
<b>3</b>	<b>Main model parameters</b>	<b>6</b>
3.1	TYPE . . . . .	6
3.2	DEGREE . . . . .	6
3.3	RIDGE . . . . .	7
3.4	KERNEL_TYPE . . . . .	7
3.5	KERNEL_COEF . . . . .	7
3.6	DISTANCE_TYPE . . . . .	8
3.7	WEIGHT . . . . .	8
3.8	OUTPUT . . . . .	8
<b>4</b>	<b>Parameter optimization and selection</b>	<b>8</b>
4.1	OPTIM . . . . .	8
4.2	METRIC . . . . .	9
4.3	BUDGET . . . . .	9

<b>5</b>	<b>Interface with Matlab</b>	<b>9</b>
5.1	sgtelib_server_start . . . . .	9
5.2	sgtelib_server_newdata . . . . .	10
5.3	sgtelib_server_predict . . . . .	10
5.4	sgtelib_server_info . . . . .	10
5.5	sgtelib_server_metric . . . . .	10
5.6	sgtelib_server_reset . . . . .	10
5.7	sgtelib_server_stop . . . . .	10

# 1 General use of sgtelib

## 1.1 GENERAL

sgtelib is a dynamic surrogate modeling library. Given a set of data points  $[X, z(X)]$ , it allows to estimate the value of  $z(x)$  for any  $x$ .

**sgtelib can be called in 5 modes:**

- **-predict:** build a model on a set of data points and perform a prediction on a set of prediction points. See **PREDICT** for more information. This requires the definition of a model with the option **-model**, see **MODEL**.  
`sgtelib.exe -model <model description> -predict <input/output files>`  
`sgtelib.exe -model TYPE PRS DEGREE 2 -predict x.txt z.txt xx.txt zz.txt`
- **-server:** starts a server that can be interrogated to perform predictions or compute the error metric of a model. The server should be used via the Matlab interface (see **SERVER**). This requires the definition of a model with the option **-model**, see **MODEL**.  
`sgtelib.exe -server -model <model description>`  
`sgtelib.exe -server -model TYPE LOWESS SHAPE_COEF OPTIM`
- **-best:** returns the best type of model for a set of data points  
`sgtelib.exe -best <x file name> <z file name>`  
`sgtelib.exe -best x.txt z.txt`
- **-help:** allows to ask for some information about some keyword.  
`sgtelib.exe -help keyword`  
`sgtelib.exe -help DEGREE`  
`sgtelib.exe -help LOWESS`  
`sgtelib.exe -help`
- **-test:** runs a test of the sgtelib library.  
`sgtelib.exe -test`

## 1.2 PREDICT

Performs a prediction in command line on a set of data provided through text files. If no ZZ file is provided, the predictions are displayed in the terminal. If no model is provided, the default model is used.

**Example:**

```
sgtelib.exe -predict <x file name> <z file name> <xx file name>
sgtelib.exe -predict x.txt z.txt xx.txt -model TYPE PRS DEGREE 2
sgtelib.exe -predict x.txt z.txt xx.txt zz.txt
```

## 1.3 BEST

Displays the description of the model that best fit the data provided in text files.

**Example:**

```
sgtelib.exe -best x.file.txt z.file.txt
```

## 1.4 SERVER

Starts a sgtelib server. See **MATLAB\_SERVER** for more details.

**Example:**

```
sgtelib.exe -server -model TYPE LOWESS DEGREE 1 KERNEL_SHAPE OPTIM
```

## 1.5 MODEL

Models in sgtelib are defined by using a succession of field names (see **FIELD** for the list of possible fields) and field values. Each field name is made of one single word. Each field value is made of one single word or numerical value. It is good practice to start by the field name **TYPE**, followed by the model type.

**Possible field names:**

- **TYPE**: mandatory field that specifies the type of model
- **DEGREE**: degree of the model for PRS and LOWESS models
- **RIDGE**: regularization parameter for PRS, RBF and LOWESS models
- **KERNEL\_TYPE**: Kernel function for RBF, KS, LOWESS and KRIGING models
- **KERNEL\_SHAPE**: Shape coefficient for RBF, KS and LOWESS models
- **METRIC\_TYPE**: Error metric used as criteria for model parameter optimization/selection
- **DISTANCE\_TYPE**: Metric used to compute the distance between points
- **PRESET**: Special information for some types of model
- **WEIGHT\_TYPE**: Defines how the weights of Ensemble of model are computed
- **BUDGET**: Defines the parameter optimization budget
- **OUTPUT**: Defines the output text file

## 1.6 FIELD

A model description is composed of field names and field values.

**Example:**

```
TYPE <model type> FIELD1 <field 1 value> FIELD2 <field 2 value>
```

## 2 Types of models

### 2.1 PRS

PRS (Polynomial Response Surface) is a type of model.

**Authorized fields for this type of model:**

- DEGREE (Can be optimized)
- RIDGE (Can be optimized)
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE PRS DEGREE 2
TYPE PRS DEGREE OPTIM RIDGE OPTIM
```

### 2.2 PRS\_EDGE

PRS\_EDGE (Polynomial Response Surface EDGE) is a type of model that allows to model discontinuities at 0 by using additional basis functions.

**Authorized fields for this type of model:**

- DEGREE (Can be optimized)
- RIDGE (Can be optimized)
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE PRS_EDGE DEGREE 2
TYPE PRS_EDGE DEGREE OPTIM RIDGE OPTIM
```

### 2.3 PRS\_CAT

PRS\_CAT (Categorical Polynomial Response Surface) is a type of model that allows to build one PRS model for each different value of the first component of x.

**Authorized fields for this type of model:**

- DEGREE (Can be optimized)
- RIDGE (Can be optimized)
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE PRS_CAT DEGREE 2
TYPE PRS_CAT DEGREE OPTIM RIDGE OPTIM
```

### 2.4 RBF

RBF (Radial Basis Function) is a type of model.

**Authorized fields for this type of model:**

- KERNEL\_TYPE (Can be optimized)
- KERNEL\_COEF (Can be optimized)
- DISTANCE\_TYPE (Can be optimized)

- RIDGE (Can be optimized)
- PRESET: "O" for RBF with linear terms and orthogonal constraints, "R" for RBF with linear terms and regularization term, "I" for RBF with incomplete set of basis functions. This parameter cannot be optimized.
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE RBF KERNEL_TYPE D1 KERNEL_SHAPE OPTIM DISTANCE_TYPE NORM2
```

## 2.5 KS

KS (Kernel Smoothing) is a type of model.

**Authorized fields for this type of model:**

- KERNEL\_TYPE (Can be optimized)
- KERNEL\_COEF (Can be optimized)
- DISTANCE\_TYPE (Can be optimized)
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE KS KERNEL_TYPE OPTIM KERNEL_SHAPE OPTIM
```

## 2.6 KRIGING

KRIGING is a type of model.

**Authorized fields for this type of model:**

- RIDGE (Can be optimized)
- DISTANCE\_TYPE (Can be optimized)
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE KRIGING
```

## 2.7 LOWESS

LOWESS (Locally Weighted Regression) is a type of model.

**Authorized fields for this type of model:**

- DEGREE: Must be 1 (default) or 2 (Can be optimized).
- RIDGE (Can be optimized)
- KERNEL\_TYPE (Can be optimized)
- KERNEL\_COEF (Can be optimized)
- DISTANCE\_TYPE (Can be optimized)
- BUDGET: Defines the budget allocated for parameter optimization.
- OUTPUT: Defines the output text file.

**Example:**

```
TYPE LOWESS DEGREE 1
TYPE LOWESS DEGREE OPTIM KERNEL_SHAPE OPTIM KERNEL_TYPE D1
```

TYPE LOWESS DEGREE OPTIM KERNEL\_SHAPE OPTIM KERNEL\_TYPE OPTIM DISTANCE\_TYPE OPTIM

## 2.8 ENSEMBLE

ENSEMBLE is a type of model.

**Authorized fields for this type of model:**

- **WEIGHT:** Defines how the ensemble weights are computed.
- **METRIC:** Defines which metric is used to compute the weights.
- **BUDGET:** Defines the budget allocated for parameter optimization.
- **DISTANCE\_TYPE:** This parameter is transferred to the models contained in the Ensemble.
- **OUTPUT:** Defines the output text file.

**Example:**

```
TYPE ENSEMBLE WEIGHT SELECT METRIC OECV          TYPE ENSEMBLE WEIGHT OPTIM METRIC
RMSECV DISTANCE_TYPE NORM2 BUDGET 100
```

## 3 Main model parameters

### 3.1 TYPE

The field name **TYPE** defines which type of model is used.

**Possible model type:**

- **PRS:** Polynomial Response Surface
- **KS:** Kernel Smoothing
- **PRS\_EDGE:** PRS EDGE model
- **PRS\_CAT:** PRS CAT model
- **RBF:** Radial Basis Function Model
- **LOWESS:** Locally Weighted Regression
- **ENSEMBLE:** Ensemble of surrogates
- **KRIGING:** Kriging model
- **CN:** Closest neighbor

**Example:**

```
TYPE PRS: defines a PRS model.
TYPE ENSEMBLE: defines an ensemble of models.
```

### 3.2 DEGREE

The field name **DEGREE** defines the degree of a polynomial response surface. The value must be an integer  $\geq 1$ .

**Allowed for models of type:** PRS, PRS\_EDGE, PRS\_CAT, LOWESS.

**Default values:**

- For PRS models, the default degree is 2.
- For LOWESS models, the degree must be 1 (default) or 2.

**Example:**

```
TYPE PRS DEGREE 3 defines a PRS model of degree 3.
TYPE PRS_EDGE DEGREE 2 defines a PRS_EDGE model of degree 2.
TYPE LOWESS DEGREE OPTIM defines a LOWESS model where the degree is optimized.
```

### 3.3 RIDGE

The field name **RIDGE** defines the regularization parameter of the model.

**Allowed for models of type:** PRS, PRS\_EDGE, PRS\_CAT, LOWESS, RBF.

**Possible values:** Real value  $\geq 0$ . Recommended values are 0 and 0.001.

**Default values:** Default value is 0.01.

**Example:**

TYPE PRS DEGREE 3 RIDGE 0 defines a PRS model of degree 3 with no ridge.

TYPE PRS DEGREE OPTIM RIDGE OPTIM defines a PRS model where the degree and ridge coefficient are optimized.

### 3.4 KERNEL\_TYPE

The field name **KERNEL\_TYPE** defines the type of kernel used in the model. The field name **KERNEL** is equivalent.

**Allowed for models of type:** RBF, RBF1, Kriging, LOWESS and KS.

**Possible values:**

- D1: Gaussian kernel (default)
- D2: Inverse Quadratic Kernel
- D3: Inverse Multiquadratic Kernel
- D4: Bi-quadratic Kernel
- D5: Tri-cubic Kernel
- D6: Exponential Sqrt Kernel
- D7: Epanechnikov Kernel
- I0: Multiquadratic Kernel
- I1: Polyharmonic splines, degree 1
- I2: Polyharmonic splines, degree 2
- I3: Polyharmonic splines, degree 3
- I4: Polyharmonic splines, degree 4
- OPTIM: The type of kernel is optimized

**Example:**

TYPE KS KERNEL\_TYPE D2 defines a KS model with Inverse Quadratic Kernel

TYPE KS KERNEL\_TYPE OPTIM KERNEL\_SHAPE OPTIM defines a KS model with optimized kernel shape and type

### 3.5 KERNEL\_COEF

The field name **KERNEL\_COEF** defines the shape coefficient of the kernel function. Note that this field name has no impact for KERNEL\_TYPES I1, I2, I3 and I4 because these kernels do not include a shape parameter.

**Allowed for models of type:** RBF, KS, KRIGING, LOWESS.

**Possible values:** Real value  $\geq 0$ . Recommended range is  $[0.1, 10]$ . For KS and LOWESS model, small values lead to smoother models.

**Default values:** By default, the kernel coefficient is optimized.

**Example:**

TYPE RBF KERNEL\_COEF 10 defines a RBF model with a shape coefficient of 10.

**TYPE KS KERNEL\_TYPE OPTIM KERNEL\_SHAPE OPTIM** defines a KS model with optimized kernel shape and type

### 3.6 DISTANCE\_TYPE

The field name **DISTANCE\_TYPE** defines the distance function used in the model.

**Allowed for models of type:** RBF, RBF, KS, LOWESS.

**Possible values:**

- **NORM1:** Euclidian distance
- **NORM2:** Distance based on norm 1
- **NORMINF:** Distance based on norm  $\infty$
- **NORM2\_IS0:** Tailored distance for discontinuity in 0.
- **NORM2\_CAT:** Tailored distance for categorical models.

**Default values:** Default value is **NORM2**.

**Example:**

**TYPE KS DISTANCE NORM2\_IS0** defines a KS model tailored for VAN optimization.

### 3.7 WEIGHT

The field name **WEIGHT** defines the method used to compute the weights **w** of the ensemble of models. The keyword **WEIGHT\_TYPE** is equivalent. **Allowed for models of type:** ENSEMBLE.

**Possible values:**

- **WTA1:**  $w_k \propto \mathcal{E}_{sum} - \mathcal{E}_k$  (default)
- **WTA3:**  $w_k \propto (\mathcal{E}_k + \alpha \mathcal{E}_{mean})^\beta$
- **SELECT:**  $w_k \propto 1$  if  $\mathcal{E}_k = \mathcal{E}_{min}$
- **OPTIM:** **w** minimizes  $\mathcal{E}(\mathbf{w})$

**Example:**

**TYPE ENSEMBLE WEIGHT SELECT METRIC RMSECV** defines an ensemble of models which selects the model that has the best RMSECV.

**TYPE ENSEMBLE WEIGHT OPTIM METRIC RMSECV** defines an ensemble of models where the weights **w** are computed to minimize the RMSECV of the model.

### 3.8 OUTPUT

Defines a text file in which model information are recorded. **Allowed for ALL types of model:**

## 4 Parameter optimization and selection

### 4.1 OPTIM

The field value **OPTIM** indicate that the model parameter must be optimized. The default optimization criteria is the AOECV error metric.

**Parameters that can be optimized:**

- **DEGREE**
- **RIDGE**
- **KERNEL\_TYPE**



- KERNEL\_COEF
- DISTANCE\_TYPE

**Example:**

```
TYPE PRS DEGREE OPTIM
TYPE LOWESS DEGREE OPTIM KERNEL_TYPE OPTIM KERNEL_SHAPE OPTIM METRIC ARMSECV
```

## 4.2 METRIC

The field name **METRIC** defines the metric used to select the parameters of the model (including the weights of Ensemble models).

**Allowed for ALL types of model:**

**Possible values:**

- EMAX: Error Max
- EMAXCV: Error Max with Cross-Validation
- RMSE: Root Mean Square Error
- RMSECV: RMSE with Cross-Validation
- OE: Order Error
- OECV: Order Error with Cross-Validation
- LINV: Invert of the Likelihood
- AOE: Aggregate Order Error
- AOECV: Aggregate Order Error with Cross-Validation

**Default values:** AOECV.

**Example:**

TYPE ENSEMBLE WEIGHT SELECT METRIC RMSECV defines an ensemble of models which selects the model that has the best RMSECV.

## 4.3 BUDGET

Budget for model parameter optimization. The number of sets of model parameters that are tested is equal to the optimization budget multiplied by the the number of parameters to optimize.

**Allowed for ALL types of model:**

**Default values:** 20

**Example:**

```
TYPE LOWESS KERNEL_SHAPE OPTIM METRIC AOECV BUDGET 100
TYPE ENSEMBLE WEIGHT OPTIM METRIC RMSECV BUDGET 50
```

# 5 Interface with Matlab

## 5.1 sgtelib\_server\_start

Command from Matlab. See **example** directory for more details.

Start a sgtelib model in a server from Matlab.

**Example:**

`sgtelib_server_start('TYPE PRS');` Start a sgtelib server with a PRS model

`sgtelib_server_start('TYPE LOWESS DEGREE 1');` Start a Lowess model

`sgtelib_server_start(model_name,true);` Start a model defined in model\_name and keep the window open

## 5.2 sgtelib\_server\_newdata

Command from Matlab. See **example** directory for more details.  
Add data points to the sgtelib model from Matlab.

**Example:**

```
sgtelib_server_newdata(X,Z); Add data points [X,Z]
```

## 5.3 sgtelib\_server\_predict

Command from Matlab. See **example** directory for more details.  
Perform a prediction from Matlab.

**Example:**

```
[ZZ,std,ei,cdf] = sgtelib_server_predict(XX); Prediction at points XX.
```

## 5.4 sgtelib\_server\_info

Command from Matlab. See **example** directory for more details.  
Use sgtelib\_server\_info to display information about the model.

## 5.5 sgtelib\_server\_metric

Command from Matlab. See **example** directory for more details.  
Use sgtelib\_server\_stop(metric\_name) to access the error metric of the model.

**Example:**

```
m = sgtelib_server_metric('OECV'); Return the OECV error metric  
m = sgtelib_server_metric('RMSE'); Return the RMSE error metric
```

## 5.6 sgtelib\_server\_reset

Command from Matlab. See **example** directory for more details.  
Reset the model of the sgtelib server from Matlab.

## 5.7 sgtelib\_server\_stop

Command from Matlab. See **example** directory for more details.  
Stop the sgtelib server from Matlab.