



Quick tutorial

Mohamed Amine Bouhlel

December 2, 2016

MDOlab, University of Michigan

Main functions

```
# Import a model.
from smt.model import MODEL
# Initialization of the model.
t = MODEL({List of the model-options},{List of
the print-options})
# Add the training points.
t.add_training_pts('exact',xt,yt)
# Add the gradient information
for i in xrange(dim):
    t.add_training_pts('exact',xt,yd[:, i],kx=i)
# Train the model.
t.train()
# Prediction of x.
y = t.predict(x)
```



LS

```
# Import the LS model.  
from smt.ls import LS  
# Initialization of the model.  
t = LS({'name':'LS'},{})  
# Add the training points.  
t.add_training_pts('exact',xt,yt)  
# Train the model.  
t.train()  
# Prediction of x.  
y = t.predict(x)
```

PA2

```
# Import the PA2 model.  
from smt.pa2 import PA2  
# Initialization of the model.  
t = PA2({'name':'PA2'},{})  
# Add the training points.  
t.add_training_pts('exact',xt,yt)  
# Train the model.  
t.train()  
# Prediction of x.  
y = t.predict(x)
```

The IDW model

IDW

```
# Import the IDW model.  
from smt.idw import IDW  
# Initialization of the model.  
t = IDW({'name':'IDW'},{})  
# Add the training points.  
t.add_training_pts('exact',xt,yt)  
# Train the model.  
t.train()  
# Prediction of x.  
y = t.predict(x)
```

The ordinary Kriging model

KRG

```
# Import the KRG model.  
from smt.kpls import KPLS  
# Initialization of the model. 'name', 'n_comp' and  
# 'theta0' must be equal to 'KRG', dim and a list of  
# length dim, respectively.  
t = KPLS({'name':'KRG','n_comp':dim,'theta0':  
[1e-2]*dim},{})  
# Add the training points.  
t.add_training_pts('exact',xt,yt)  
# Train the model.  
t.train()  
# Prediction of x.  
y = t.predict(x)
```

The KPLS model

KPLS

```
# Import the KPLS model.
from smt.kpls import KPLS
# Initialization of the model. 'name' must be equal to
# 'KPLS'. 'n_comp' and 'theta0' must be an integer in
# [1, dim] and a list of length n_comp, respectively.
# Here is an example using 1 principal component.
t = KPLS({'name':'KPLS','n_comp':1, 'theta0': [1e-2]},
        {})
# Add the training points.
t.add_training_pts('exact',xt,yt)
# Train the model.
t.train()
# Prediction of x.
y = t.predict(x)
```

The KPLSK model

KPLSK

```
# Import the KPLSK model.
from smt.kpls import KPLS

# Initialization of the model. 'name' must be equal to
# 'KPLSK'. 'n_comp' and 'theta0' must be an integer in
# [1, dim[ and a list of length n_comp, respectively.
t = KPLS({'name':'KPLSK','n_comp':1,'theta0': [1e-2]},
        {})

# Add the training points.
t.add_training_pts('exact',xt,yt)

# Train the model.
t.train()

# Prediction of x.
y = t.predict(x)
```


The GEKPLS model

GEKPLS

```
# Import the GEKPLS model.
from smt.kpls import KPLS
# Initialization of the model. 'name' must be equal to
# 'GEKPLS'. 'n_comp' and 'theta0' must be an integer in
# [1, dim[ and a list of length n_comp, respectively.
t = KPLS({'name':'GEKPLS','n_comp':1, /
'theta0': [1e-2], 'xlimits':xlimits},{})
# Add the training points.
t.add_training_pts('exact',xt,yt)
# Add the gradient information
for i in xrange(dim):
    t.add_training_pts('exact',xt,yd[:, i],kx=i)
# Train the model.
t.train()
# Prediction of x.
y = t.predict(x)
```