
matk Documentation

Release 0

Dylan R. Harp

October 23, 2013

CONTENTS

1	Indices and tables	5
	Python Module Index	7
	Index	9

Contents:

class `matk.matk` (***kwargs*)

Class for Model Analysis ToolKit (MATK) module

add_ins (*insfilenm, model_outfile*)

Add an instruction file to problem

add_obs (*name, **kwargs*)

Add observation to problem

Parameters

- **name** (*str*) – Name of observation
- **kwargs** – keyword arguments passed to observation class

add_par (*name, **kwargs*)

Add parameter to problem

Parameters

- **name** (*str*) – Name of parameter
- **kwargs** – keyword arguments passed to parameter class

add_tpl (*tplfilenm, model_infile*)

Add a template file to problem

calibrate ()

Calibrate MATK model

forward (*workdir=None, reuse_dirs=False*)

Run MATK model using current values

Parameters

- **workdir** (*str*) – Name of directory where model will be run. It will be created if it does not exist
- **reuse_dirs** – If True and workdir exists, the model will reuse the directory

Returns int – 0: Successful run, 1: workdir exists

get_obs_names ()

Get observation names

get_obs_values ()

Get observation values

get_par_dist_pars ()

Get parameters needed by parameter distributions

get_par_dists ()

Get parameter probabilistic distributions

get_par_maxs ()

Get parameter lower bounds

get_par_mins ()

Get parameter lower bounds

get_par_names ()

Get parameter names

get_par_nvals ()
Get parameter nvals (number of values for parameter studies)

get_par_values ()
Get parameter values

get_parstudy (*args, **kwargs)
Generate parameter study samples

Parameters

- **outfile** (*str*) – Name of file where samples will be written. If outfile=None, no file is written.
- ***args** – Number of values for each parameter. The order is expected to match order of matk.parlist (e.g. [p.name for p in matk.parlist])
- ****kwargs** – keyword arguments where keyword is the parameter name and argument is the number of desired values

Returns ndarray(fl64) – Array of samples

get_residuals ()
Get least squares values

get_samples (siz=None, noCorrRestr=False, corrmatrix=None, outfile=None, seed=None)
Draw lhs samples of parameter values from scipy.stats module distribution

Parameters

- **siz** (*int*) – Number of samples to generate, ignored if samples are provided
- **noCorrRestr** (*bool*) – If True, correlation structure is not enforced on sample
- **corrmatrix** (*matrix*) – Correlation matrix
- **outfile** (*string*) – Name of file where samples will be written. If outfile=None, no file is written.
- **seed** (*int*) – Random seed to allow replication of samples

Returns matrix – Parameter samples

get_sims ()
Get the current simulated values :returns: lst(fl64) – simulated values in order of matk.obslist

model None
Python function or system command to run model

nccpus None
Set number of ccpus to use for concurrent model evaluations

parameters_file None
Set the name of the parameters_file for parallel runs

read_model_files (workdir=None)
Write model files with current parameters

results_file None
Set the name of the results_file for parallel runs

run_parallel ()
Run models concurrently on multiprocessor machine

run_samples (*siz=None, noCorrRestr=False, corrmatrix=None, samples=None, outfile=None, parallel=False, ncpus=1, templatedir=None, workdir_base=None, seed=None, save=True, index_start=1, reuse_dirs=False*)

Run model using values in samples for parameter values. If samples are not specified, LHS samples are produced.

Parameters

- **siz** – Number of samples to generate, ignored if samples are provided
- **noCorrRestr** (*bool*) – If True, correlation structure is not enforced on sample
- **corrmatrix** (*matrix*) – Correlation matrix *npar* by *npar*
- **samples** (*matrix*) – Matrix of samples *npar* columns by *siz* rows
- **outfile** (*str*) – name of file where samples and responses will be written. If *outfile=None*, no file is written.
- **parallel** (*bool*) – If True, models run concurrently with ‘*ncpus*’ cpus
- **ncpus** (*int*) – number of cpus to use to run models concurrently
- **templatedir** (*str*) – Name of folder including files needed to run model (e.g. template files, instruction files, executables, etc.)
- **workdir_base** (*str*) – Base name for model run folders, run index is appended to *workdir_base*
- **seed** (*int*) – Random seed to allow replication of samples
- **save** (*bool*) – If True, model files and folders will not be deleted during parallel model execution
- **index_start** (*int*) – The initial index to be appended to working directories and output files
- **reuse_dirs** – Will use existing directories if True, will return an error if False and directory exists

Returns tuple(ndarray(float64), ndarray(float64)) - (Matrix of responses from sampled model runs *siz* rows by *npar* columns, Parameter samples, same as input samples if provided)

seed None

Set the seed for random sampling

set_obs_values (**args, **kwargs*)

Set simulated values using a dictionary or keyword arguments

set_par_values (**args, **kwargs*)

Set parameters using values in first argument

templatedir None

Set the name of the templatedir for parallel runs

workdir None

Set the base name for parallel working directories

workdir_base None

Set the base name for parallel working directories

workdir_index None

Set the working directory index for parallel runs

write_model_files (*workdir=None*)

Write model files with current parameters

class `matk.Parameter` (*name*, ***kwargs*)

MATK parameter class

dist `None`

Probabilistic distribution of parameter belonging to `scipy.stats` module

dist_pars `None`

Distribution parameters required by `self.dist` (e.g. if `dist == uniform`, `dist_pars = (min,max-min)`)

max `None`

Parameter upper bound

mean `None`

Parameter mean

min `None`

Parameter lower bound

name `None`

Parameter name

nvals `None`

Number of values the paramter will take for parameter studies

offset `None`

Offset to add to parameter

scale `None`

Scale factor to multiply parameter by

std `None`

Parameter st. dev.

value `None`

Parameter value

class `matk.Observation` (*name*, ***kwargs*)

MATK observation class

name `None`

Observation name

residual `None`

Observation value minus simulated value

sim `None`

Simulated value generated by MATK model

value `None`

Observation value

weight `None`

Weight to apply to simulated values

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

m

matk, 1

A

add_ins() (matk.matk method), 1
 add_obs() (matk.matk method), 1
 add_par() (matk.matk method), 1
 add_tpl() (matk.matk method), 1

C

calibrate() (matk.matk method), 1

D

dist (matk.Parameter attribute), 4
 dist_pars (matk.Parameter attribute), 4

F

forward() (matk.matk method), 1

G

get_obs_names() (matk.matk method), 1
 get_obs_values() (matk.matk method), 1
 get_par_dist_pars() (matk.matk method), 1
 get_par_dists() (matk.matk method), 1
 get_par_maxs() (matk.matk method), 1
 get_par_mins() (matk.matk method), 1
 get_par_names() (matk.matk method), 1
 get_par_nvals() (matk.matk method), 1
 get_par_values() (matk.matk method), 2
 get_parstudy() (matk.matk method), 2
 get_residuals() (matk.matk method), 2
 get_samples() (matk.matk method), 2
 get_sims() (matk.matk method), 2

M

matk (class in matk), 1
 matk (module), 1
 max (matk.Parameter attribute), 4
 mean (matk.Parameter attribute), 4
 min (matk.Parameter attribute), 4
 model (matk.matk attribute), 2

N

name (matk.Observation attribute), 4
 name (matk.Parameter attribute), 4

ncpus (matk.matk attribute), 2
 nvals (matk.Parameter attribute), 4

O

Observation (class in matk), 4
 offset (matk.Parameter attribute), 4

P

Parameter (class in matk), 3
 parameters_file (matk.matk attribute), 2

R

read_model_files() (matk.matk method), 2
 residual (matk.Observation attribute), 4
 results_file (matk.matk attribute), 2
 run_parallel() (matk.matk method), 2
 run_samples() (matk.matk method), 2

S

scale (matk.Parameter attribute), 4
 seed (matk.matk attribute), 3
 set_obs_values() (matk.matk method), 3
 set_par_values() (matk.matk method), 3
 sim (matk.Observation attribute), 4
 std (matk.Parameter attribute), 4

T

templatedir (matk.matk attribute), 3

V

value (matk.Observation attribute), 4
 value (matk.Parameter attribute), 4

W

weight (matk.Observation attribute), 4
 workdir (matk.matk attribute), 3
 workdir_base (matk.matk attribute), 3
 workdir_index (matk.matk attribute), 3
 write_model_files() (matk.matk method), 3