
matk Documentation

Release 0

Dylan R. Harp

October 30, 2013

CONTENTS

1	Indices and tables	7
	Python Module Index	9
	Index	11

Contents:

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

Class for Model Analysis ToolKit (MATK) module

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_sampleset (*name, samples, responses=None, indices=None, index_start=1*)

Add sample set to problem

Parameters

- **name** (*str*) – Name of sample set
- **samples** (*list(fl64), ndarray(fl64)*) – Matrix of parameter samples with npar columns in order of [p.name for p in matkobj.parlist]
- **responses** (*list(fl64), ndarray(fl64)*) – Matrix of associated responses with nobs columns in order of [o.name for o in matkobj.obslist] if observation exists (existence of observations is not required)
- **indices** (*list(int), ndarray(int)*) – Sample indices to use when creating working directories and output files

calibrate ()

Calibrate MATK model

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

Run MATK model using current values

Parameters

- **pardict** (*dict*) – Dictionary of parameter values keyed by parameter names
- **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_residuals()
Get least squares values

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

make_workdir (*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

model
Python function or system command to run model

ncpus
Set number of cpus to use for concurrent model evaluations

parameters_file
Set the name of the parameters_file for parallel runs

results_file
Set the name of the results_file for parallel runs

run_samples (*name=None, ncpus=1, templatedir=None, workdir_base=None, save=True, reuse_dirs=False*)
Run model using values in samples for parameter values If samples are not specified, LHS samples are produced

Parameters

- **name** – Name of MATK sample set object
- **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

- **save** (*bool*) – If True, model files and folders will not be deleted during parallel model execution
- **reuse_dirs** – Will use existing directories if True, will return an error if False and directory exists

Returns tuple(ndarray(fl64),ndarray(fl64)) - (Matrix of responses from sampled model runs *siz* rows by *npar* columns, Parameter samples, same as input samples if provided)

save_sampleset (*outfile*, *sampleset*)

Save sampleset to file

Parameters

- **outfile** (*str*) – Name of file where sampleset will be written
- **sampleset** (*str*) – Sampleset name

seed

Set the seed for random sampling

set_lhs_samples (*name*, *siz=None*, *noCorrRestr=False*, *corrmat=None*, *seed=None*, *index_start=1*)

Draw lhs samples of parameter values from scipy.stats module distribution

Parameters

- **name** (*str*) – Name of sample set to be created
- **siz** (*int*) – Number of samples to generate, ignored if samples are provided
- **noCorrRestr** (*bool*) – If True, correlation structure is not enforced on sample, use if *siz* is less than number of parameters
- **corrmat** (*matrix*) – Correlation matrix
- **seed** (*int*) – Random seed to allow replication of samples
- **index_start** – Starting value for sample indices

Type int

Returns matrix – Parameter samples

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

set_parstudy_samples (*name*, **args*, ***kwargs*)

Generate parameter study samples

Parameters

- **name** (*str*) – Name of sample set to be created
- **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

templatedir

Set the name of the templatedir for parallel runs

workdir

Set the base name for parallel working directories

workdir_base

Set the base name for parallel working directories

workdir_index

Set the working directory index for parallel runs

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

MATK parameter class

dist

Probabilistic distribution of parameter belonging to scipy.stats module

dist_pars

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

max

Parameter upper bound

mean

Parameter mean

min

Parameter lower bound

name

Parameter name

nvals

Number of values the paramter will take for parameter studies

offset

Offset to add to parameter

scale

Scale factor to multiply parameter by

std

Parameter st. dev.

value

Parameter value

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

MATK observation class

name

Observation name

residual

Observation value minus simulated value

sim

Simulated value generated by MATK model

value

Observation value

weight

Weight to apply to simulated values

class `matk.SampleSet` (*name, samples, index_start=1, **kwargs*)

MATK samples class - Stores information related to a sample including parameter samples, associated responses, and sample indices

corr (*type='pearson', plot=False*)

Calculate correlation coefficients of parameters and responses

Parameters **type** (*str*) – Type of correlation coefficient (pearson by default, spearman also available)

Returns `ndarray(fl64)` – Correlation coefficients

index_start

Starting integer value for sample indices

indices

Array of sample indices

name

Sample set name

obsnames

Array of observation names

parnames

Array of parameter names

responses

Ndarray of sample set responses, rows are samples, columns are responses associated with observations in order of `MATKobject.obslist`

samples

Ndarray of parameter samples, rows are samples, columns are parameters in order of `MATKobject.parlist`

INDICES AND TABLES

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

PYTHON MODULE INDEX

m

matk, [1](#)

INDEX

A

add_obs() (matk.matk method), 1
add_par() (matk.matk method), 1
add_sampleset() (matk.matk method), 1

C

calibrate() (matk.matk method), 1
corr() (matk.SampleSet method), 5

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), 2
get_par_mins() (matk.matk method), 2
get_par_names() (matk.matk method), 2
get_par_nvals() (matk.matk method), 2
get_par_values() (matk.matk method), 2
get_residuals() (matk.matk method), 2
get_sims() (matk.matk method), 2

I

index_start (matk.SampleSet attribute), 5
indices (matk.SampleSet attribute), 5

M

make_workdir() (matk.matk method), 2
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
name (matk.SampleSet attribute), 5
ncpus (matk.matk attribute), 2
nvals (matk.Parameter attribute), 4

O

Observation (class in matk), 4
obsnames (matk.SampleSet attribute), 5
offset (matk.Parameter attribute), 4

P

Parameter (class in matk), 4
parameters_file (matk.matk attribute), 2
parnames (matk.SampleSet attribute), 5

R

residual (matk.Observation attribute), 4
responses (matk.SampleSet attribute), 5
results_file (matk.matk attribute), 2
run_samples() (matk.matk method), 2

S

samples (matk.SampleSet attribute), 5
SampleSet (class in matk), 5
save_sampleset() (matk.matk method), 3
scale (matk.Parameter attribute), 4
seed (matk.matk attribute), 3
set_lhs_samples() (matk.matk method), 3
set_obs_values() (matk.matk method), 3
set_par_values() (matk.matk method), 3
set_parstudy_samples() (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), 4

workdir_base (matk.matk attribute), 4

workdir_index (matk.matk attribute), 4