pygpc

Generalized Polynomial Chaos for Python

Konstantin Weise, TU Ilmenau, MPI CBS - konstantin.weise@tu-ilmenau.de

Contents

1 Introduction

2 Classes and methods

3 Post-processing

3.1 Mean

`m = pygpc.mean(C)` – calculates the mean of the N_o expanded output variables

Parameter	`C`	*numpy array (2D), size:* $[N_c \times N_o]$
		N_c gpc coefficients for each N_o expanded output variable
Returns	`y`	*numpy array (1D) size:* $[1 \times N_o]$
		mean

3.2 Standard deviation

`s = pygpc.std(C)` – calculates the standard deviation of the N_o expanded output variables

Parameter	`C`	*numpy array (2D), size:* $[N_c \times N_o]$
		N_c gpc coefficients for each N_o expanded variable
Returns	`s`	*numpy array (1D)* $[1 \times N_o]$
		standard deviation

3.3 Sobol indices

`S, s_idx = pygpc.sobol(g, C)` – calculates the unnormalized Sobol indices of the N_o expanded output variables

Parameter	`g`	*object*
		gpc object
	`C`	*numpy array (2D), size:* $[N_c \times N_o]$
		N_c gpc coefficients for each N_o expanded output variable
Returns	`S`	*numpy array (2D), size:* $[N_s \times N_o]$
		Unnormalized sobol indices
		usually: normalization with respect to variance S/σ^2
		sorted in descending order (w.r.t. first column only)
	`s_idx`	*list of numpy arrays, length:* N_s
		List of corresponding variable combinations of sobol indices in rows of S

3.4 Global derivative based sensitivity coefficients

`G = pygpc.globalsens(g, C)` – calculates the global derivative based sensitivity coefficients of the N_o expanded output variables

Parameter	`g`	*object*
		gpc object
	`C`	*numpy array (2D), size:* $[N_c \times N_o]$
		N_c gpc coefficients for each N_o expanded output variable
Returns	`G`	*numpy array (2D), size:* $[d \times N_o]$
		Sensitivity indices for each of the d random input and N_o output variables

3.5 Local sensitivity coefficients

`L = pygpc.localsens(g, C)` – calculates the global derivative based sensitivity coefficients of the N_o expanded output variables

Parameter	`g`	*object*
		gpc object
	`C`	*numpy array (2D), size:* $[N_c \times N_o]$
		N_c gpc coefficients for each N_o expanded output variable
	ξ	*numpy array (1D), size:* $[1 \times d]$
		coordinates in normalized variable space to evaluate local sensitivity in coordinate space: $\beta(a, b, p, q)$: $[a, b] \rightarrow [-1, 1]$
Returns	`L`	*numpy array (2D), size:* $[d \times N_o]$
		Sensitivity indices for each of the d random input and N_o output variables