**PROCEDURE: SOBOL PARAMETER RANKING**

**NOTE TO SELF:** *Red texts indicate parts that I do not fully understand/can explain*

**[STEP 0]** *Select a set of key model parameters. Let = number of key model parameters which is indexed by . Thus the key parameters* *are indexed as follows:* *for*

* *If a model has many parameters the key parameter set can be selected on the basis of representativeness of R&D (parameters such as electricity price for example, while sensitive arent very representative of R&D and isnt something that is improved with research)*
* *If information regarding R&D representativeness isn’t available for parameter selection, as many parameters should be selected. Sobol Indices can be actually used to evaluate the parameter selection as it ranks parameters on the contribution to variance*
* *At TRL 2-3 the core phenomena is usually understood under First Order Principles, so it shouldn’t be difficult to distinguish the most important R&D parameters.*

**[STEP 1]** *Collect multiple sample points for each parameter in the process model*

* ***EX:*** *In the optimal operating conditions of a CCU process that has been designed via 1st principle model, what are the deterministic parameter values??*
* *The sample points need not necessarily be from optimality point and can be collected from pilot/lab-scale experiments or in the worst case, from literature study. The data collector should note discrepancies in the experimentation apparatus that might explain differences in parameter values.*

**[STEP 2]** *Establish probability distributions for each parameter using the Kernel Density Estimation (KDE) method. This is done by placing a Gaussian kernel at each sample point in* ***STEP 1*** *for each key parameter determined in* ***STEP 0****. The KDE generates a smooth continuous PDF as follows:*

*Where is the sample size (# of data samples), h is the bandwidth (curve spreadiness) and are the values of the data samples. With a Gaussian kernel, the equation becomes:*

* *One reason for using KDE for PDF generation is to avoid ambiguity issue in “assigning” a probability distribution for a parameter.*
* *Usually a PDF was assigned heuristically (assuming normal distribution, uniform etc). Generating a PDF that is defined by sample data points avoids the ambiguity issue as the PDF is tailored to available data. It can be proved that with infinite sample points, the KDE distribution converges to a Normal one*
* *Because the KDE distribution is sensitive to bandwidth h (which acts as “stdev”, it can be argued that choosing a value for still retains some level of ambiguity*

**[STEP 3]** *Determine the total number of Monte Carlo simulations to be performed, , where the # of simulations equals the # of sampled sets of Key Parameters and the # of Process Model evaluations. Each Monte Carlo CCUS evaluation in* *is indexed with* *(simulation in )*

* *Usually, enough simulations are required to guarantee smoothness in the distribution of the model outputs (when plotted).*
* *Alternatively, the minimum number of Monte Carlo runs can be determined by the formula proposed by Gu et al., Proceedia Engineering 99, 2015:*

*Here, z is the confidence level as number of standard deviations (z=3) and is the evaluation error which can be something like*

**[STEP 4]** *Define two IDENTICAL parameter spaces. One is called* ***“Parameter Space”*** *while the other is called the* ***“Complementary Space”****. These spaces are identical in the sense that the values in the spaces are derived from the same procedure, but the actual values themselves may not be identical on an index-to-index basis (see below). The parameter space is a matrix where for each Monte Carlo simulation, each* *key parameters are random sampled from their respective PDFs defined by the KDE method.*

* *The Parameter Space and Complementary Space are identical in dimension and definition, but because in each space the parameter values are RANDOM SAMPLED from their PDFs (identical procedure), the actual values of the parameters for a Monte Carlo simulation i could be different for the Parameter Space and Complementary Space*

**[STEP 5]** *Using the sampled parameter values in the* ***“Parameter Space”*** *perform evaluations of the CCUS Process Model. In other words, the values of the parameters are all from the “Parameter Space”. Label the resulting matrix as* ***M\_Evaluations.*** *M\_Evaluations is an matrix where corresponds to the number of model outputs we wish to observe the variance of.*

* *Are the outputs of the CCUS process model technoeconomic or LCA metrics (such as Unit Cost of Production or Specific GWI) or is it a “physical” metric such as Specific Yield or Boil-off rate??*
* *Any model output metric would work as long as it can be directly linked to the performance of the CCUS process. The significance being that the Sobol Indices would then be interpreted as the contribution of variance with respect to the total variance OF THAT METRIC (be it Unit Production Cost, yield, etc)*
* *For simplicity, choosing a single model output metrix ( vector) means that the key parameters can be ranked based on a single ranked list of Sobol indices and not multiple. For multiple outputs, we would have multiple parameter ranked lists, which would have to be compromised via averaging or some other methods (TBD)*

**[STEP 6]** *Evaluate the CCUS Process Model times for each key parameter (total evaluations). For each set of CCUS model evaluations for each key parameter, the values of key parameter at interest, should be extracted from the “Parameter Space” whereas the parameters complementary to* *(defined as ) derive their values from the “Complementary Space”. Label this matrix as* ***PS\_Evaluation****.*

* *For example, assume there are 3 key parameters: A, B, and C with 500 MC simulations.*
* *From* ***STEP 4,*** *500 Monte Carlo sampling sets (1 set = sampling for all selected parameters) are performed to form the “Parameter Space” by sampling values for A, B, and C from their respective PDFs. An additional 500 Monte Carlo sampling sets are performed to create the “Complementary Space” by again sampling the values for A, B, and C from their respective PDFs. However because the samplings are done randomly from PDFs, the precise values between the two spaces are different.*
* *In* ***STEP 6,*** *500 x 3 evaluations of the CCUS Process Model are performed to generate the* ***PS\_Evaluation*** *matrix. In the first set 500 evaluations, parameter is the parameter of interest. The values for parameter A are extracted from the “Parameter Space” while the values for parameters B and C () are extracted from the “Complementary Space”. The values for this first set of 500 evaluations fills the first column of* ***PS\_Evaluation***
* *In the next 500 CCUS model evaluations, parameter is the parameter of interest. The values for B are extracted from “Parameter Space” while the values for parameters A and C () are extracted from the “Complementary Space”. This fills the 2nd column of* ***PS\_Evaluation***
* *Repeat for parameter C to complete* ***PS\_Evaluation*** *matrix.*

**[STEP 7]** *Evaluate the CCUS Process Model times for each key parameter* *(total evaluations). For each set of* *CCUS model evaluations for each key parameter****,*** *the values of key parameter at interest,* ***,*** *should be extracted from the “Complementary Space” whereas the parameters complementary to () derive their values from the “Parameter Space”. Label the resulting matrix as* ***CS\_Evaluation.***

* *Quickly realize that* ***STEP 7*** *is virtually identical to* ***STEP 6*** *except that the Parameter and Complementary spaces are flipped.*
* *To generate Sobol Indices we 3 rounds of CCUS model evaluations are performed. In the first round* ***(STEP 5)*** *the model is evaluated only times with the values for the parameter set being derived from the Parameter Space. In the 2nd round* ***(STEP 6)*** *the model is evaluated times with Parameter Space for and Complementary Space for . In the 3rd round* ***(STEP 7)*** *the model is evaluated times with the Complementary Space for etc.*

**[STEP 8]** *Compute the integral of the CCUS model output, which can be estimated into the following summation if Monte Carlo results are available:*

* *This comes from the ANOVA function decomposition for conditionally independent parameters . It says that any function can be decomposed into summands of increasing dimensionality:*
* *is a constant. The integrals of every summand over its own variable set equals to zero, which implies that all the summands are orthogonal*
* *Thus only the constant term remains, i.e.: . In other words, the integral of the CCUS process model can be estimated by taking the average of process model evaluations from different Monte Carlo simulations.*

**[STEP 9]** *Compute the variance of the TOTAL model output, D. If there are more than 1 model outputs (see* ***STEP 5****) the total variances must be computed for each model output metric. The total variance can be computed as follows:*

*The integral estimation in* ***STEP 8*** *can be applied here also, as follows:*

**[STEP 10]** *Compute partial (1st order) variances for each parameter . The partial variance terms are denoted as . The 1st order variances are defined by the following expression:*

* *indicates its from Parameter Space whereas represents Complementary Space.*
* *Likewise, corresponds to* ***M\_Evaluations*** *in* ***STEP 5*** *with values for parameter sets extracted from the Parameter Space.*
* *corresponds to* ***PS\_Evaluations*** *matrix in* ***STEP 6*** *where for each parameter , values of are extracted from Parameter Space and the remaining parameters are extracted from complementary space.*
* *Since Monte Carlo simulations are performed, the derivation of can be discretized with the following expression:*

*In other words, for each parameter , in each -th Monte Carlo simulation, the differences between* ***M\_Evaluations*** *and* ***PS\_Evaluations*** *are squared, normalized by dividing the difference by , then subtracted from the Total Model Output variance . The subtractions are performed times for each parameter .*

* *Thus is a vector and the above iterative procedure is repeated for each key model parameters in*

**[STEP 11]** *Compute the total parameter variance, for each parameter . The total parameter variance is defined as the contribution to model output variance by parameter as well as interactions involving . The total parameter variance is defined as:*

* *corresponds to* ***CS\_Evaluations*** *in* ***STEP 7*** *where for each parameter , the values of are extracted from the Complementary Space and the remaining parameters get their values from the Parameter Space.*
* *For each parameter , in each -th Monte Carlo simulation, the differences between* ***M\_Evaluations*** *and* ***CS\_Evaluations*** *are squared, then normalized by dividing the difference by 2N.*
* *The differences are accumulated as the procedure is iterated for Monte Carlo simulations*

**[STEP 12]** *Compute the 1st Order () and Total Sobol () Indices, which are defined as follows:*

*and*

* *Parameters can be ranked according to their 1st order indices, or according to Total Sobol indices. However, it is more meaningful to rank the parameters according to Total Sobol indices since this takes into account parameter interactions with each other. This means Total Sobol Indices can be used as a way of performing Global Sensitivity Analysis by enumerating the sensitivity of Model Output on Parameter according to the parameter’s indices score. The higher the indices score, the greater the sensitivity.*
* *If the sum of 1st Order or Total Sobol Indices is GREATER than 1, it implies parameter correlations. The closer the sum is to 1, the less parameter correlations exist.*
* *Compared to NBC method, which ranks parameters based on risk, the Sobol method ranks parameters based on contribution to variance.*

**function** eval\_metrics = CCUS\_Model(kp) %kp = key parameters

% Key Parameters

x1 = kp(1) ;

x2 = kp(2) ; *.... etc*

% Process Outputs

Yield = 2\*x1 + 2\*x2

EnergyUse = 3\*x1 + x2

Size\_Reactor = x1.^2

Size\_Column = 4\*(x2/x1) *.... etc*

% Evaluation Metrics

CAPEX = 2\*Size\_Reactor + Size\_Column

OPEX = 5\*Yield + 2\*EnergyUse

Unit\_Prod\_Cost = CAPEX + OPEX

Specific\_GWI = 3\*EnergyUse *.... etc*

% Output Array

eval\_metrics = [CAPEX, OPEX, Unit\_Prod\_Cost, Specific\_GWI]

**end**