README File - Sensibility Analysis

Table of Contents

Introduction

This subfolder cointains all the code that was developped to perform a sensitivity analysis on the ERT Matlab simulator (present in the parent folder). This was part of a semester project done by Frédéric Berdoz and supervised by Dr Philippe Müllhaupt, during the fall semester of 2020. The report containing all the supporting theory can be found in the ERT drive.

Content

- `Main_NS.m`: Main executable for the Numerical Stability analysis.
- `Main_EE.m`: Main executable for the Elementary Effect analysis.
- `Main_SO.m`: Main executable for the Sobol analysis.
- `Main_Outputs`: Main to produce output plot (for illustration in the report).
- `NS_Results.m`: Auxiliary executable to produce plots for the NS analysis.
- `EE_Results.m`: Auxiliary executable to produce plots for the EE analysis.
- `SO_Results.m`: Auxiliary executable to produce plots for the Sobol analysis.
- Subfolder - `Figures`: Contains the figures (sorted into subfolders).
- Subfolder - `Helpers`: Contains the helper functions
- Subfolder - `Outputs`: Contains the results (variables) of several analysis (serves as backup folder for long simulations).
- Subfolder - `Simulators`: Contains the multilayerwindSimulator3D with different ODE solver (for the NS analysis).

How to Use

- **Numerical Stability Analysis**: Run the `Main_NS` executable and specify N (number of simulation) and sigma (size of input domain, in which paramter are sampled to run different simualtions). Also specify the rocket declaration, serving as the base values around which the input domain is created. The variables are stored in the subfolder `Outputs` and stored under the name prefixed by NS_ and corresponding to the parameter N. Then run `NS_Results` by specifying which output to analyse.

- **Elementary Effect Analysis**: Run the `Main_EE` executable and specify p (EE parameter, see report), r (number of repetition) and sigma (size of input domain). Also specify the rocket and environment declaration, serving as base values around which the input domain is created. Finally, modify Xid and Yid to specify which input and output should be monitored. The variables are stored in the subfolder `Outputs` and stored under the name prefixed by EE_ and corresponding to the parameters k, o and r (size of Xid, size of Yid and r, respectively). Then run `EE_Results` by specifying which output to analyse.

- **Sobol Analysis**: Run the `Main_SO` executable and specify N (total number of simulation for the double loop reordering technique, or number of simulation per parameter for the other techniques) and sigma (size of input domain). Also specify the rocket and environment declaration, serving as base values around which the input domain is created. Finally, modify Xid and Yid to specify which input and output should be monitored. If ONLYDLR is set to true, only the double loop reordering will be performed. The variables are stored in the subfolder `Outputs` and stored under the name prefixed by SO_ and corresponding to the parameters k, o and N (size of Xid, size of Yid and N, respectively). Then run `SO_Results` by specifying which output to analyse.

How to Modify

The code is designed to run several simulation in a row by feeding `SimAPI.m` with a simulator object, a list of inputs Xid, a list of output Yid and an input matrix X whose columns represent the different set of parameters (1 column = 1 distinct simulation). To create X, the code first reads the current set of parameters in the simulator object using the function `baseValues.m`. This function returns the input domain XX that is fed to `SOsampling.m` to create a matrix X. `SimAPI.m` then loops over the columns of X, sets the parameters in the simulator object to the values of the current column (using `setParam.m` and then run the corresponding simulation (monitoring the outputs Y specified in Yid). Y is then reused in the EE and Sobol analysis. `SimAPI_NS.m` is a modification of `SimAPI.m` so that multiple simulator objects can be used in parallel (for the NS analysis).

As of now, only the `multilayerwindSimulator3D` type is implemeneted. To implement another tye of simulator for the sensitivity analysis, one must modify `baseValues.m` (to read the parameter values from the simulator object), `setParam.m` (to set the parameter values in the simulator object) and `SimAPI.m` (to monitor the outputs of the simulation).

Published with MATLAB® R2019b