

**Portland State University**

---

**From the Selected Works of Wayne W. Wakeland**

---

Summer 2019

# Addressing parameter uncertainty in SD models with fit-to-history and Monte-Carlo sensitivity methods

Wayne W. Wakeland, *Portland State University*  
Jack Homer, *Homer Consulting*



Available at: [https://works.bepress.com/wayne\\_wakeland/117/](https://works.bepress.com/wayne_wakeland/117/)

## **Addressing parameter uncertainty in SD models with fit-to-history and Monte-Carlo sensitivity methods**

### **Abstract**

We present a practical guide, including a step-by-step flowchart, for establishing confidence intervals for key model outcomes in the face of uncertain parameters. The process starts with Powell optimization (e.g., using Vensim™) to find a set of uncertain parameters (the “optimum” parameter set or OPS) that minimize the model fitness error relative to available reference behavior data. The optimization process also helps in refinement of assumed parameter uncertainty ranges. Next, Markov Chain Monte Carlo (MCMC) or conventional Monte Carlo (MC) randomization is used to create a sample of parameter sets that fit the reference behavior data nearly as well as the OPS. Under the MC method, the entire parameter space is explored broadly (with a very large number of runs), and the results are sorted for selection of qualifying parameter sets (QPS) based on goodness-of-fit criteria. The statistical properties of the QPS parameter distributions are analyzed to ensure their centrality relative to the uncertainty ranges. Also, the full set of QPS outputs are graphed (as sensitivity graphs or box-and-whisker plots) for comparison with the reference behavior data. Finally, alternative policies and scenarios are run against the OPS and all QPS, and confidence intervals are found for key model outcomes. The method is demonstrated with a non-trivial model, and a narrative template is provided to illustrate how such analyses could be described to interested parties such as policy or decision makers.

## **1. Introduction**

System dynamics (SD) models frequently employ parameters for which solid empirical data is not available. Modelers often use expert judgment, either informally or formally, to provide estimates of parameter values for which empirical data is not available. When several experts are available, and a formal process such as a Delphi survey is used to obtain their judgments, model developers and users may have high confidence in the values of these parameters despite the lack of data. Nevertheless, these parameter values remain uncertain to a degree.

To address parameter uncertainty, SD modelers experiment extensively with alternative parameter values in order to understand their degree of influence in the model. Formal sensitivity analyses can (and should) be run using features provided in popular SD software packages. Results can be displayed in a table or portrayed graphically as a “Tornado” diagram (c.f. Wakeland and Hoarfrost 2005).

Once the modeler has identified influential parameters, additional effort is applied as time and budget allow to increase confidence that the values of these influential parameters are well supported. When reliable empirical data are available, ideally from multiple sources, the parameter value may be fixed and used with confidence. Usually, however, some, perhaps even many, parameter values remain uncertain. This paper describes how the model analysis features available in the Vensim™ software, and likely other software, can be used to represent and incorporate that uncertainty into the analysis of SD model results, including, potentially, providing confidence intervals informed by parameter uncertainty.

A brief background section will review the use of these methods in recent literature. A methods section provides a detailed diagram describing the process for incorporating parameter uncertainty in SD models, followed by a results section that provides an illustration of the use of the method. The paper closes with a discussion section that illustrates how the model results under uncertainty can be described for interested parties such as policy makers or other end users of the analyses.

## **2. Background**

Helton et al. (2006) provide a general overview of sampling-based methods for sensitivity analysis, including Monte Carlo (MC). Cheun and Beck (2009) explain Bayesian updating in MC processes, and background material on the mathematics of Markov Chain Monte Carlo (MCMC) is readily available (cf. ter Braak 2006 and Vrugt J.A. et al. 2009). Background on sensitivity analysis methods for dynamic modeling, and SD in particular, including the MC and MCMC methods being employed in this paper, may be found in Houghten et al. (2014), Houghten and Siegel (2015), Osgood (2013), and Osgood and Liu (2015). Using conventional MC methods to search a parameter space is sometimes considered to a brute force or totally random search, whereas the MCMC method, uses a Bayesian update process to guide the search of parameter space. In at least some contexts, it has been shown that MCMC optimally creates statistically valid samples.

Although publications describing the application of these methods are plentiful in some scientific and engineering disciplines, publications featuring their use with SD models are scant. A search for “system dynamics” AND (“MCMC” or “monte carlo”) return only a handful of publications. Two examples are Jeon and Chin (2014), who describe their use of MC with an SD model applied to a renewable energy study, and Sterman et al. (2018), who describe their use of MCMC in the evaluation of an SD forestry industry model. Conference papers are no more numerous. A search of the SD conference proceeds for MCMC or Monte return just one result for the period from 2015-2018, a methods paper (Fiddaman and Yeager, 2015).

### 3. Method

Figure 1 summarizes a detailed process for incorporating uncertainty analysis into SD models. Two alternative approaches are included. One approach employs a conventional MC-based search of parameter space, whereas the other uses the MCMC approach.

In the following text, **bold italics** are used to highlight the name of the block in Figure 1 being discussed. The process starts at **Create model & modify as needed**. This would be a model that employs uncertain parameters and includes dynamic outcome variables that strive to match the dynamics seen in real world reference behavior data. To use the methods described in Figure 1, we need to **Define error metric variables** and **Add error metrics to model**. A useful example could be to compute the mean absolute error (MAE) between the model calculated time series and the reference behavior time series for each outcome variable. Care must be taken to consider how to compute MAE when reference data is incomplete so as not to distort results. When different outcomes have very different scales, it is useful to use MAEM which is MAE over M, where M is the mean value of the metric. In addition, composite error statistics are added to the model, such as the average of the MAEMs over all the outcomes, and the maximum value of the individual MAEMs. These are used later in the process for identifying well-fitting parameter sets. There are statistical macros available for Vensim to help with this.

Another key step at the outset is to **Estimate parameter uncertainty ranges**, a lower and upper bound for each uncertain parameter, and to **Specify weights for the outcome variables**. These are needed for the algorithm used in a key part of the process: **Estimate Uncertain Parameter values**. This process employs a Powell optimization process that uses an objective function consisting of model outcomes vs. reference data. Each term of the objective function represents one of the outcome time series of interest. The algorithm strives to minimize the differences between model and reference data. Weights are needed because outcomes may have very different scales and variances. This tends to be an iterative process, so Figure 1 contains a feedback loop, and some of the connections are bidirectional. The end result of this step is a set of optimized parameter values for the uncertain parameters (OPS). Its average MAEM might be .1 and its max MAEM (worst MAEM for any one of the outcomes) might be .2.

### Process for Addressing SD Model Uncertainty\*

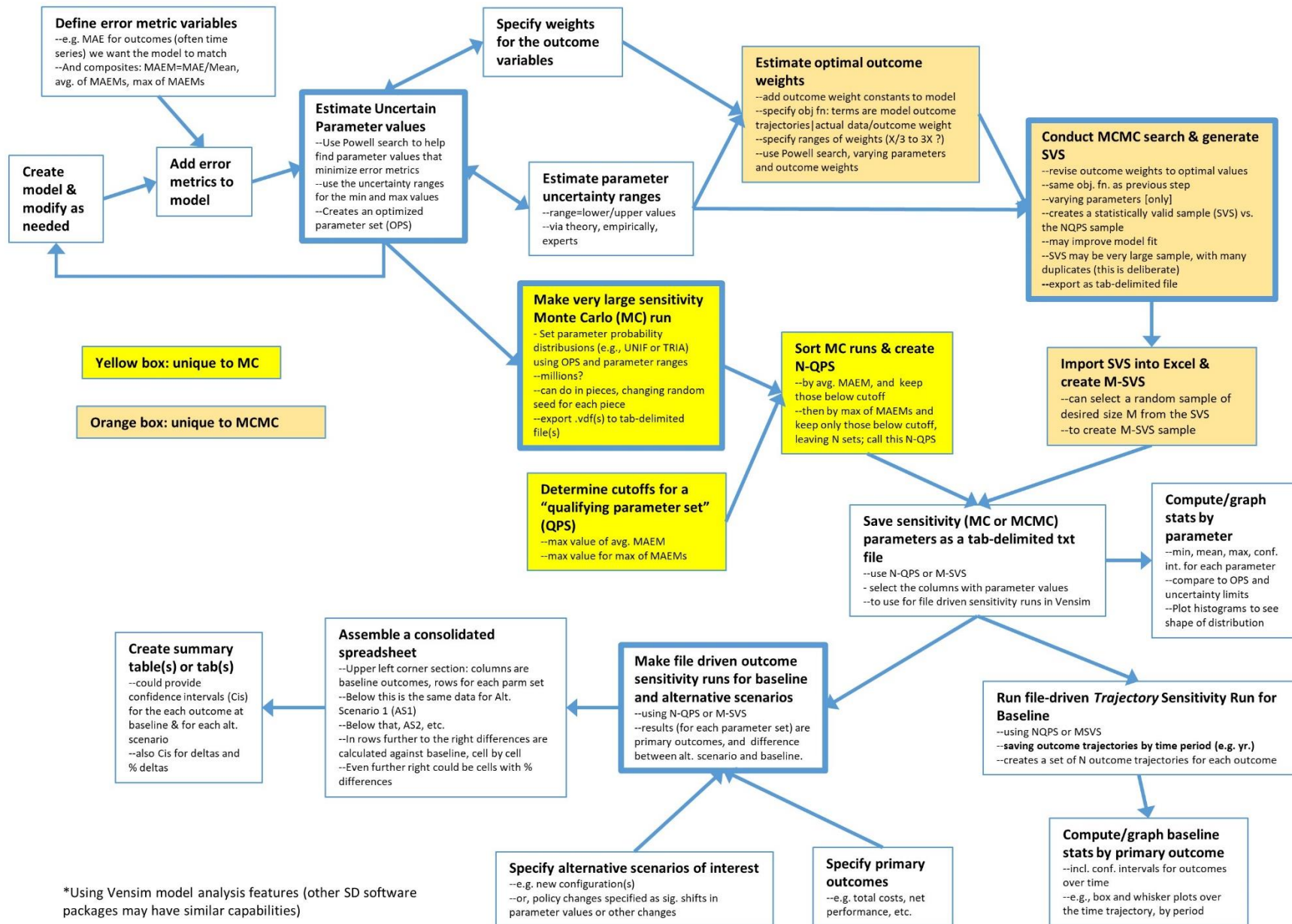


Figure 1. Process for addressing SD model uncertainty

At this point, the user may elect to use conventional Monte Carlo (MC) or Markov Chain Monte Carlo (MCMC). MC is discussed first.

**Make very large sensitivity Monte Carlo (MC) run** is another key step of the process. It employs Vensim's sensitivity feature to make a very large number of model runs, millions if there are many uncertain parameters. Using this feature requires the user to specify how many runs to make, the seed to start with, the type of sampling (e.g., multivariate, latin hypercube, etc.), and which parameters to vary and how. We used multi-variate, which is a totally random search process. Latin hypercube strives to cover a large parameter space more efficiently. Both may be valid choices. We experimented with uniform, but settled on Triangular with the mode specified as the value from the OPS. Since we are focused on overall error, we changed the SAVEPER to be the length of the run. This keeps the output file size manageable. We also specified some additional variables to be saved (all the parameters being varied are automatically saved). These were the avgMAEM and maxMAEM for the run.

Note that when >1M runs are needed, we found it practical to make one million at a time and to change the seed for each run. At the end of a sensitivity run, the .vdf file is exported to a tab-delimited file for further analysis.

Next, we **Determine cutoffs for a "qualified parameter set (QPS)** which involves determining how well-fitting a parameter set needs to be in order for it to be kept in the sample. The vast majority of the MC runs will not be well-fitting like the OPS. One rational approach would be to accept parameter sets that performed nearly as well as the OPS, say, for example average MAEM < .12 and max MAEM < .25.

**Sort MC runs and create N-QPS** employs Excel's data/import external data from file to read in the MC results. The result will be M columns x N rows, where M is the number parameters being varied + 4 and N is the number of runs. The 4 is the two saved variables saved at two time periods (start and end). The results are then sorted by avgMAEM and all rows > avgMAEM cutoff are discarded. The remainder is sorted by maxMAEM and all rows > maxMAEM are discarded. This will likely leave a very tiny fraction of runs, perhaps a few hundred out of a million. These N runs are the qualified parameter sets (N-QPS). **Save sensitivity (MC or MCMC) parameters as a tab-delimited txt file** creates the file needed for file-driven sensitivity runs.

Within Excel, the user can proceed to **Compute/graph stats by parameter** to see what the values in N-QPS file look like for each parameter. For example, to see if the entire range of possible values for a given parameter represented in N-QPS, and if the mean of this sample is near the value of the parameter in the OPS. Also to determine the confidence interval of the estimate for the mean of the parameter based on this sample and examine the distribution (shape) of the sample? Does this information raise any red flags with respect to the OPS or N-QPS? Does this information indicate that the N-QPS may be representative of the entire parameter space?

Moving on, the N-QPS file that can be used in two primary ways. One can **Run file-driven Trajectory Sensitivity Run for Baseline**, for which we change the SAVPER to a useful time-period such as by year in order to create a series of N trajectories for each outcome. The result is exported to a tab-delimited file that can be imported into Excel to **Compute/graph baseline stats by primary outcome**, including confidence intervals for the model calculated outcome at each time period, since there is now a sample of N model-calculated values at each time point. Excel can also be used to create box and whisker plots at each time point, with or without outliers. However, we found it more expedient to make these plots using Python.

For the other primary use of the N-QPS file (or the M-SVS file, which is explained later), we first need to: **Specify primary high level outcomes of interest** such as total cost and net performance. The previous sensitivity run was focused on behavior over time for all outcome time series, but to compare alternative scenarios in an overall sense, a few key end-of-run metrics are needed. And, we need to **Specify alternative scenarios of interest**; these might be different configurations or policy changes that could be implemented in the model as switches that are used in conjunction with "magnitude of impact" parameters and timing parameters linked to specific model constants.

Using the N-QPS or M-SVS, we change the SAVPER back to the End of Run and then **Make file driven sensitivity runs for baseline and each alt. scenario**. The results, for the baseline and for each scenario are matched pairs of data points (where all of the uncertain parameter values are the same for baseline and the alternative). This means that the distributions of the differences can be used to determine in a statistically valid fashion the confidence interval estimates for the mean of the difference by outcome between baseline and the alternative that can be attributed to parameter uncertainty.

We suggest to **Assemble a consolidated spreadsheet** to do these analyses. In the upper left corner, is  $O \times N$ , where O is the number of overall outcomes and N is the number of runs. Note that the raw sensitivity results file has columns for all the parameters as well as two columns for each outcome (start and end). You will be selecting and copying into the consolidated sheet just the end columns for each outcome. The data for Alternative 1 will be placed below the Baseline results; Alternative 2 below Alternative 1, etc. Then, starting at the first row of Alternative 1, columns will be added that compute the differences between the Alternative 1 numbers and baseline number, one column for each outcome. Similarly for Alternative 2 vs. baseline, and so forth. Columns for percentage differences can also be added.

Finally, **Create summary table(s) or tab(s)** is used to provide the results of relevant calculations, such as means and their confidence intervals for each outcome at baseline and for each alternative. Also for the differences by outcome between baseline and Alternative 1, baseline and Alternative 2, etc. Similarly for the percentage differences.

A very different approach for creating a set of runs to be used to evaluate the impacts of parameter uncertainty is to use the Markov Chain Monte Carlo (MCMC) approach. This begins

with **Estimate Optimal Outcome Weights**. First, one adds to the model a weight variable for each outcome instead of specifying the weights numerically as in the MC process, and then one specifies the objective function, which is similar to the prior approach except that the weights are variables within the model rather than numeric values. One also needs to specify the search range for outcome weight constants. The search ranges for the uncertain parameters can be the same as for the MC approach. Or, these ranges could be broader than those used with the prior approach. No sampling distribution such as Uniform or Triangular is specified because the search of parameter space is guided by a heuristic not random sampling. Guidance regarding how broad to set the ranges for MCMC varies by expert. Informally we've heard the comment that very broad ranges will give the algorithm "room to work." Others have suggested, again, informally, that ranges should not include implausible values. Both comments are sensible, suggesting that a careful study of this question is needed. To proceed, first a Powell search is done that optimizes the uncertain parameters and the weights simultaneously. In our experience, allowing larger ranges may allow the algorithm to find an optimum that achieves slightly lower avgMAEM, but higher maxMAEM. The reason is that some of the individual outcome are "sacrificed" to achieve lower overall error. However, these results may be less realistic.

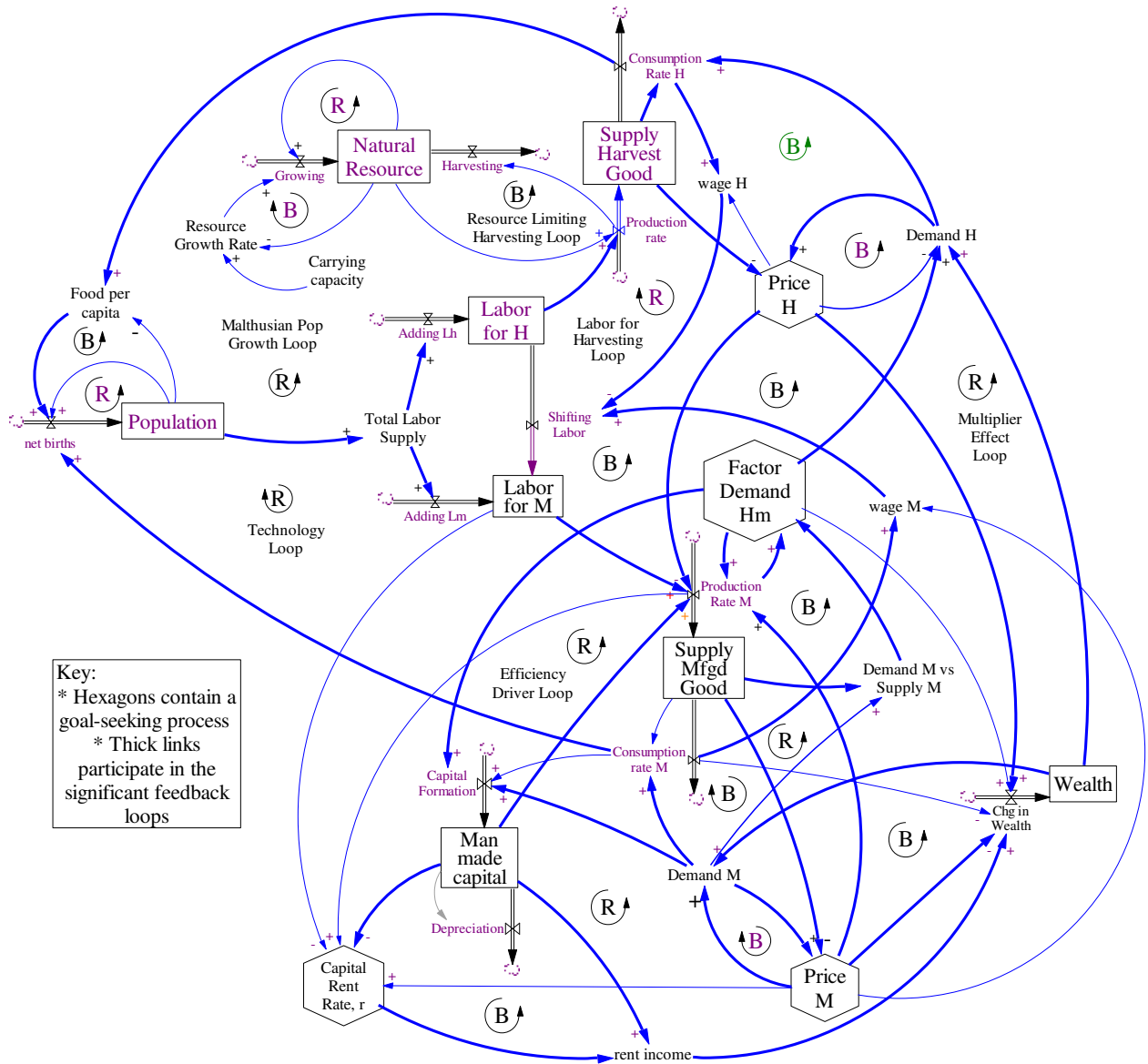
Next, is to **Conduct MCMC search & generate SVS**, which, like MC, varies the uncertain parameters over the specified ranges, but uses in the objective function the outcome weights that were found in previous step. The Markov Chain-driven search algorithm is designed to create a statistically valid sample. It can be quite large and contain many duplicates (which apparently helps to assure a sample with the correct properties). Model fitness may be further improved, but be less plausible, as mentioned earlier. Results are then exported as a tab-delimited file.

One then must **Import results into Excel**, and, if the population of runs produced by the MCMC search is excessively large, one can select a random sample of desired size M from the SVS. This subsample of size M, which is still a statistically valid sample (M-SVS), is comparable to the N-QPS set of runs created by the conventional MC process described earlier. One can proceed to computing stats by parameter using the M-SVS and/or use it to run file-driven sensitivity runs, analyze alternative scenarios, etc.

#### 4. Results

As an illustration, the MC method was applied to an SD model of medium complexity that represents a simple ecological economic system first published by Brander and Taylor (1999) and enhanced significantly by Uehara and colleagues (2013, 2015). The model contains 14 stocks, 25 constants, and 72 auxiliary variables. There are 1000s of feedback loops. Figure 2 provides a simplified diagram. Although inspired by real world events, the population overshoot and collapse on Easter Island, the model is conceptual and not strongly driven by empirical data. Key outcome variables are population, natural resources (vegetation). There is also a manufactured item in the economy, called Mgood.





**Figure 2. Model diagram showing stocks, flows, and primary feedback structures. Hexagons are goal seeking subsystems in which stocks seek to achieve a variable representing a dynamic goal. Thick lines are links that participate in feedback loops containing several model components labeled with “B” & “R.”**

To apply the parameter uncertainty analysis method, first error metrics are created, in this case MAEM (mean absolute error over mean) for the population and resource stock. We used the Vensim SSTATS macro (author unknown) attached herewith as Appendix 1. This code is inserted via text editor (we used Notepad) at the top of the model file starting at the second line. Appendix 2 shows the model code for calculating the statistics, which is also inserted into the model file via text editor, typically after the “Control” block, which is located after the user-defined model variables/constants section. We also needed to convert our reference behavior data (which we had originally entered as a time function with the “as graph” feature) to an actual data series. We created an Excel spreadsheet with a sheet named RBP that has

Time in Row 1 and the two RBP series in Rows 2 and 3. Note that one of these series had data for every 20 periods, and the other for every 10 time periods. The SSTATS macro calculations are designed to handle this correctly. The Vensim equation for reading this data is "GET XLS DATA('Demonstration model RBP data.xlsx','RBP','1','B2')"

Another step to get the model ready for model analysis is to add a custom graphs file via the Control Panel and add custom tables that display the variables calculated by the SSTATS macro (R2, MAPE, MAEM, RMSE, Um, Us, Uc, and Count), one table for each Outcome variable (Population L and Natural Resource R in the demonstration model). Finally, I/O Objects are added to one of the views to display the custom tables. Figure 3 shows the starting point outcome trajectories vs. RBP as well as the statistical parameter displays from SSTATS. Visually, it is clear that the Population L outcome is reproduced less accurately than Natural Resource S. The STATS bear this out: the MAEM Population L is .28 vs .11 for MAEM Natural Resource S.

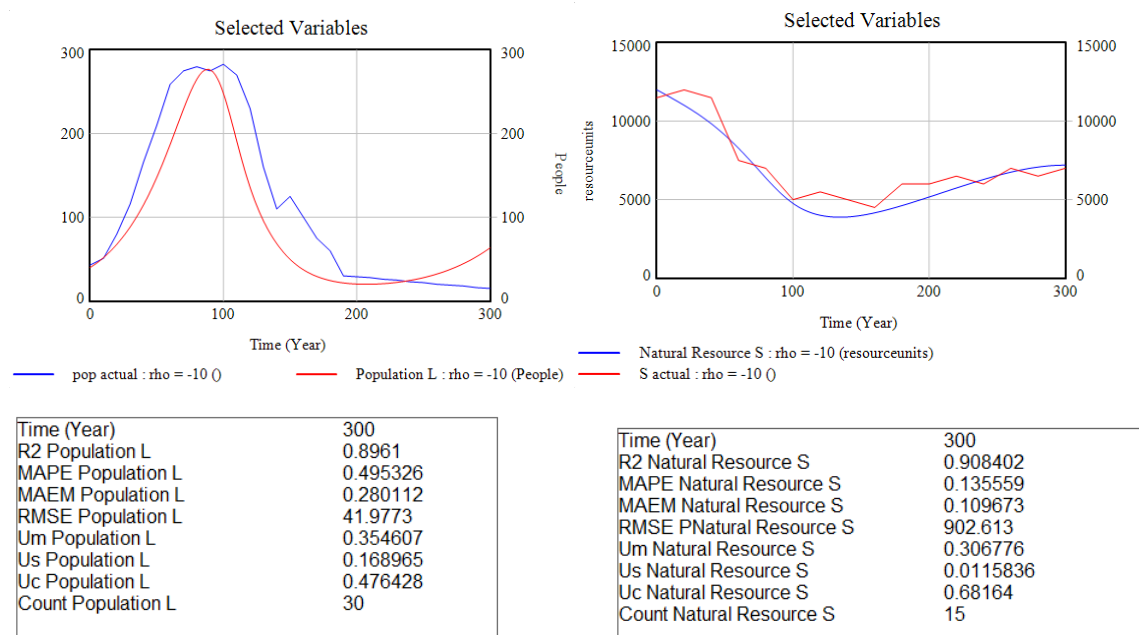


Figure 3: Initial outcome graphs, model vs. RBP for Population L and Natural Resource S, and SSTATS display

Before we can use optimization to estimate uncertain parameter values, we need to determine appropriate ranges for each parameter to be varied by the Powell search algorithm in order to bound its search. Appendix 3 provides an example vensim optimization control file (.voc) in which specifies nine uncertain parameters to be varied (see Table 1 below).

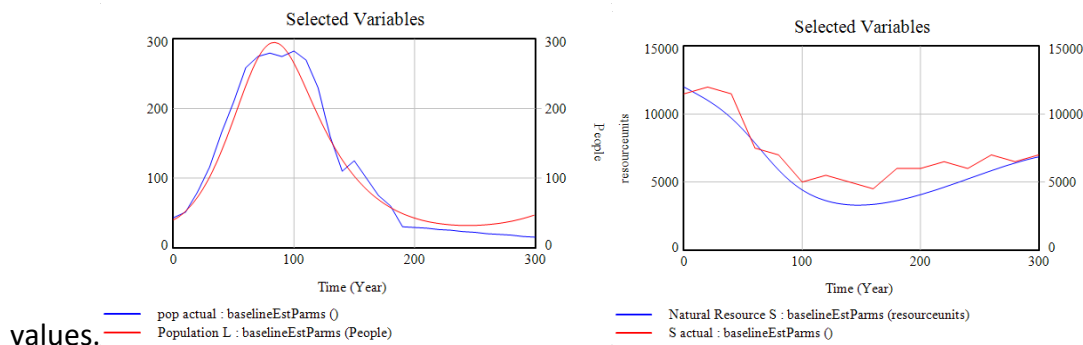
In addition, weights must be specified for the objective function, informed by the relative magnitudes and variances of the outcome variables. In this case, Population L weight is set to 1 and Natural Resource S weight set to 50. The optimization run using the CG (calibration

Gaussian) option ran for 20 minutes on a laptop computer. Table 1 shows the resulting optima reported in the Vensim “.out” file placed in the directory from which the model was run.

	Min	Start	Optima	Max
Regeneration Rate r	0.003	0.04	0.0156	0.1
Maximum birth rate b0	0.03	0.1	0.1002	0.5
Maximum death rate d0	0.03	0.2	0.0821	0.5
Sensitivity of birth rate to manufactured good intake b2	0.3	1	8.287	10
Sensitivity of birth rate to resource good intake b1	0.3	1	2.705	10
Sensitivity of death rate to manufactured good intake d2	0.3	1	10	10
Sensitivity of death rate to manufactured good intake d1	0.1	1	0.958	3
tau	1	20	83.8	100
sensitivity to price gap	0.001	.01	0.00196	0.05

**Table 1: Uncertain parameters, Min, Start, Optimal, Max**

All but two of the parameters were modified by a factor of at least 2 and as much as a factor of 10. One parameter was selected as the maximum value allowed, which raises a yellow flag. Was the upper limit truly an upper limit? The revised MEAMs are .13 for population and .17 for the resource. Figure 4 shows the model fit using the optimal



**Figure 4: Model fit using the optimal values**

The resource plot looks much worse, and population much better. There could be two reasons for this. One, the weight applied to the Natural Resource may have allowed the search to sacrifice its fitness; and two, the resource has half as many data points (every 20 time periods), so its standardized errors carry half as much weight overall. One could change the weights to find a more balanced solution. One could also increase the limits, if plausible, to allow the search more latitude. And, one must ask, are the large changes in the parameter values consistent with theory, etc.? For this demo, we will move on to the next step.

Making a large Monte Carlo (MC) run is easy using the sensitivity feature in Vensim, which creates a .vsc file. One option for exploring parameter space could be to use Uniform distributions between thoughtfully chosen minima and maxima. These values could be the

same as those used during calibration or perhaps narrowed somewhat. Another option could be to use a Triangular distribution with the optimum value as the mode, which would increase samples near the optima. In the demo, since several of the optima are near one of the limits set earlier, the min/max will be set to approximately the optimum/3 and 3\*optimum. We also need to specify a .lst file that lists what we want to be saved in each run. Sensitivity runs automatically save the varied parameters used for each run, and in addition, we want to know the maxMAEM and average MAEM (to be used to select which runs are qualified). Since we don't need to know the time trajectories, we set the model SAVEPER to the max time (300 in this case). This keeps the output file modest in size. Appendix 4 shows the .vsc and .lst files.

The simulation was set to create 20,000 runs, which took 100 minutes on a laptop. The resulting .vdf (vensim data file) was exported (via the model menu) to a tab-delimited file and read into Excel using data/get external data/from text (switch to "all files" to see .tab files). This was sorted by simple average of MAEMS. About 800 of the 20K runs were below .25 and kept. The other rows were deleted. The file was then sorted by max MAEM, and 213 were less than .25 on this metric. The rest were deleted. The two columns for "T1" were also deleted, leaving a matrix with 11 columns and 213 rows of data (the parameters for the best-fitting 213 of the 20K runs). Table 2 shows the first few rows of the spreadsheet.

**Table 2: Initial rows of the N-QPS file**

Sim#	Maximum birth rate b0	Maximum death rate d0	Regeneration Rate r	Sensitivity of birth rate to manufactured good intake b2	Sensitivity of birth rate to resource good intake b1	Sensitivity of death rate to manufactured good intake d2	Sensitivity of death rate to resource good intake d1	sensitivity to price gap	Tau	T2 Max of all MAEM	T2 Simple avg of all MAEM
15680	0.087207	0.143948	0.0142	8.28863	1.93021	22.321	2.22121	0.002694	139.841	0.161356	0.015478
7842	0.116089	0.108901	0.014539	6.24487	1.6217	19.6976	1.03318	0.001929	57.1568	0.166286	0.015137
2975	0.137109	0.095993	0.016006	9.77164	3.30537	8.86094	0.441421	0.001468	51.0105	0.166973	0.016589
6032	0.083993	0.110925	0.014528	12.2336	5.62019	11.3299	1.61066	0.004399	98.231	0.16705	0.016654
9965	0.122615	0.096897	0.015558	7.0892	1.07825	14.0869	1.99178	0.00213	75.8887	0.167135	0.014925
4540	0.107372	0.082053	0.015048	9.76103	2.14145	9.51846	1.40379	0.001687	124.302	0.168646	0.015623

After saving the spreadsheet, one can delete Col A and the metrics columns at the right, and then save the file as a .txt (tab-delimited file), the N-QPS. The N-QPS is used to run file-driven sensitivity runs in two ways. The first way is make a sensitivity run to obtain outcome trajectories for the different well-fitting runs and examine whether they are tightly grouped or spread out. The second usage is to capture final outcomes (one or more metrics regarding overall "goodness" or "badness" of the outcome) for base case and alternative configurations.

But first, we examine the properties of the parameter samples contained in the N-QPS. One can easily compute their range compared to the allowed (min to max) range, their mean value, and a confidence interval. This is easy, so there is no need to demonstrate this. One can also quickly create histogram plots. Figure 5 shows an example. This parameter was limited to be in the .03 to .3 range, so none of the well-fitting runs "hit the rails" so to speak, but much of the range was included. The mode is about .1 (the optimal value). No yellow flags here. One could examine all the parameter in this fashion to build confidence in the sample.

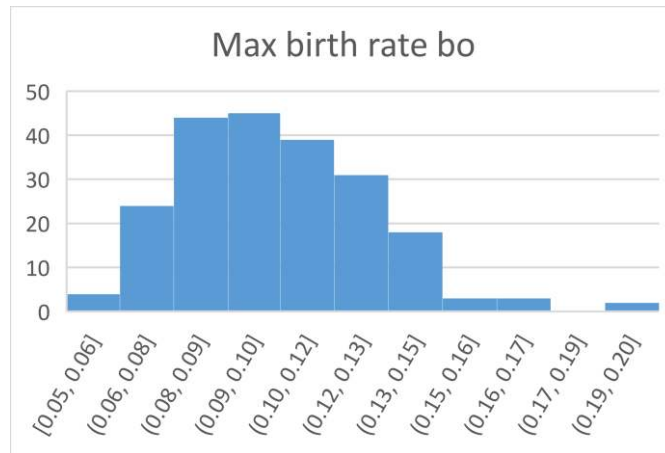


Figure 5: Example histogram for a parameter from the N-QPS

To put the N-QPS sample to use, a file-driven sensitivity run is done. The SAVEPER is changed to 10, and the Sensitivity tool is selected. Previous settings are cleared, and the Type is changed to “File.” The N-QPS text file is selected as the file. Next a save list is created. The previous one (with the stat variables) is cleared and the two primary outcome variables are entered, Population L, and Natural Resource S, for which there is RBP data (and, therefore, were used during the model calibration phase). This run took just a minute to complete, and the .vdf file it produced is exported via Vensim and imported into Excel.

The first 10 columns of this trajectory spreadsheet provide the run number (1-213) and the values of the nine uncertain parameter values used for each specific run. The remaining 62 columns are values at time 0 to 300 (by 10) for the two outcome variables. Amazingly, if one selects the 213 rows and the 31 columns for one of the outcome variables, and then requests a Box and Whisker plot, it works, although it does not know the time periods. Figure 6: shows the trajectory under uncertain for Natural Resource S using box and whiskers. Also, using the data in the spreadsheet, the confidence interval of the estimate at each time point can be calculated, as well as other statistics.

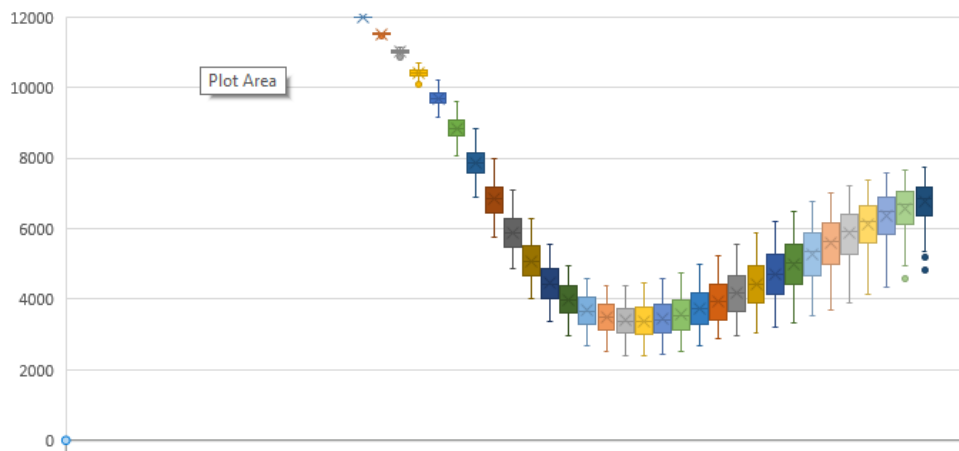


Figure 6: Box and whisker time series plot using Excel.

There is no easy way using Excel to superimpose the RBP on this plot, which would be very helpful. Python can be used create much better formatted and more complex charts as needed. Figure 9 shows the plots produced using Python showing confidence interval trajectories overlaid with the actual data. To prepare the file needed by Python, a version of the trajectory spreadsheet was created, named OutcomeTrajectory DataforPythonPlotting (OTDPP), in which the columns for run number and the parameter value were deleted. Also, two rows were added below the first row. Row 2 contains the actual data to be overlaid. For each outcome variable, the appropriate row of the RBP spreadsheet is copied above its corresponding section in OTDPP. Since RBP contains 31 values for each outcome trajectory, and since the data sections for each outcome in OTDPP is 31 columns wide, it is a simple copying process, even for several outcomes. Row 3 in OTDPP provides the time values, which can also be copied from RBP (31 cells copied once and pasted N times in OTDPP).

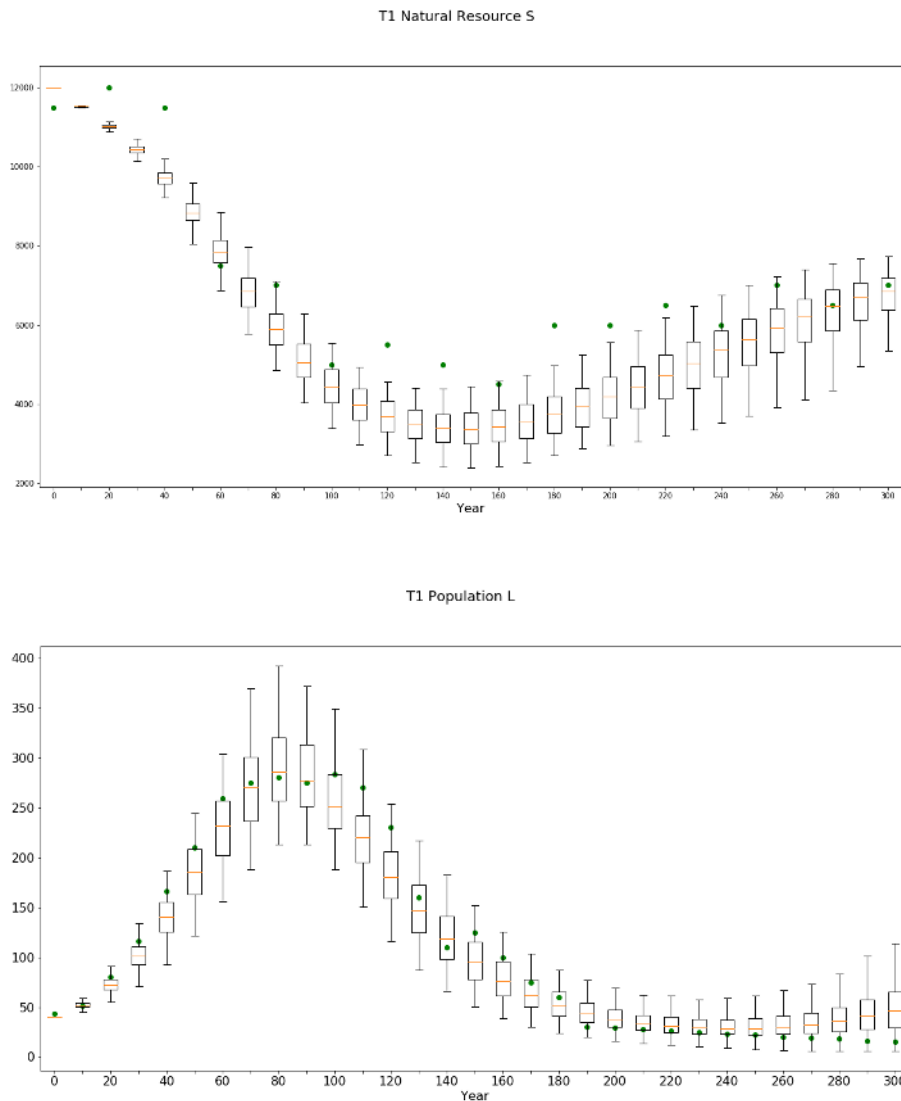


Figure 7: Outcome trajectories with confidence Intervals based on N-QPS, with actual data (dots)

Another important use of the N-QPS is to compare a baseline case with one or more alternative cases. For example, a situation where, as the Natural Resource declines, a new technology is developed at a particular point in time that effectively increases the carrying capacity by 30%. How much will this impact the most important outcome indicator(s)? That indicator could be the size of the net reduction in population from its peak to its trough. Such a metric could be added to the model and captured at the end of the run, both for baseline and the alternative. Both cases would use the N-QPS file. And the net change in the metric for the alternative case, run by run would be calculated, in absolute and percentage terms

To demonstrate, the variable Pop loss was added to the model, and the SAVEPER changed to 300. A basecase N-QPS sensitivity run was done, with the .lst file changed so that Pop loss was saved. The run took just a minute or two. Figure 8 shows the distribution of Pop loss in the baseline case. The Pop loss for the optimal parameter set was 260, which lies in the middle bar, but it appears that a more modest loss is more likely, urging caution when reporting “optimal” results. The mean of the sample is 246, and, although the sample clearly would not meet a stringent test for normality, the 95% confidence interval for this estimate (treating the sample as being “close enough” to normal) is 240 to 252. The result for the optimal parameter set (OPS) is not within this interval. However, such a difference may be inconsequential in terms of policy-making.

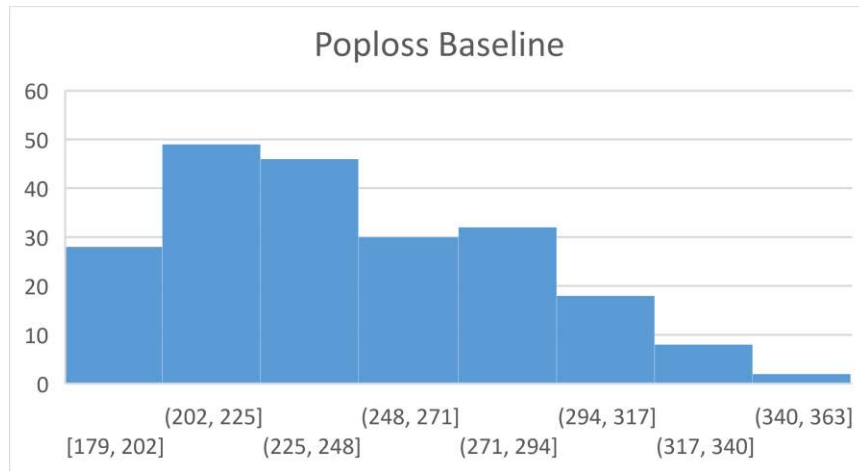


Figure 8: Histogram for outcome metric in baseline case

Next, the alternative scenario was created, and another file-drive sensitivity run was done. Figure 9 shows the Pop loss histogram for the alternative scenario, and the distribution of the matched pairs of differences. The mean and confidence interval for Pop loss under alternative 1 is 219 and [213, 225], and the mean and CI for the distribution of differences between baseline and Alternative 1 are -29 and [-31, -27]. The technology helps to a modest degree. This completes the demonstration.

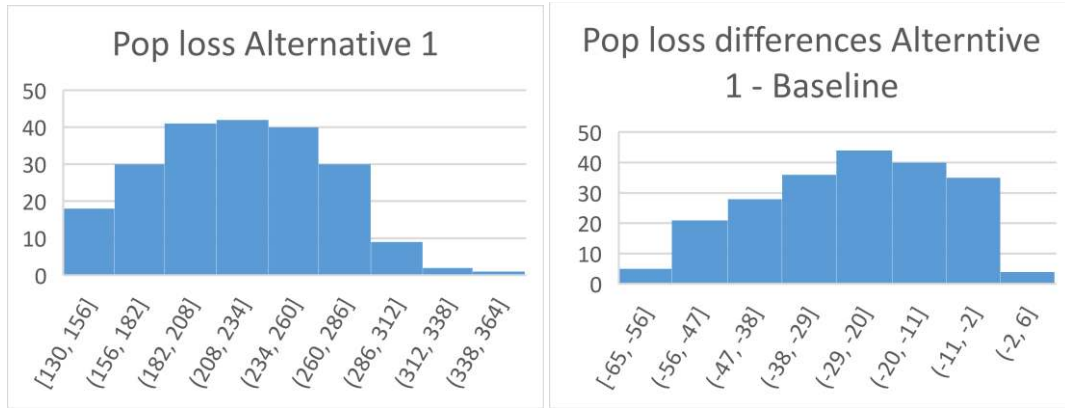


Figure 9: Alternative 1 Pop loss and differences vs. baseline

## Discussion

This section is a narrative adapted from a specific, but proprietary, model development and application that employed the uncertainty methods discussed in this paper. The narrative began by explaining that it accompanies a series of tables and graphs that lay out the assumptions and results of a complex system dynamics model, and that all of the graphs in the package are time plots covering a specific multi-year time period. It then went on to say that the purpose of the model is to determine the extent to which changes over time in key aspects of the system contributed to negative outcomes. The way we have analyzed this question is by comparing the results from historically accurate baseline simulations to those representing alternative system configurations. In these alternative simulations, we go back in time to ask “what if” things had been different, or what if things are different in the future?

### Part one: baseline modeling and parameter estimation

The model was developed and tested using the Vensim™ simulation software (version 7.2 Professional dated 2017). Model development was carried out according to established practices of system dynamics, which include information gathering, model formulation, model testing, and revising of model assumptions as necessary to improve model realism and usefulness. Information came from published studies as well as online data sources. During the process, the model progressed through many iterations.

The model contains hundreds of interacting equations, over a hundred input constants and time series, and over two hundred primary and secondary outputs. “Primary outputs” refers to nearly two dozen variables for which we had historical time-series data, covering many aspects of the system. A few time series included all of the historical data, but many provided historical data at only a few time points, averaging perhaps half of the historical period on average.

The model’s input constants and time series include many that were estimated with high confidence from empirical data, but nearly half of them must be considered to have a wider



range of uncertainty. The model's fixed constants are listed in a table, and the input time series are presented as figures. These materials also include data sources and derivations (associated logic).

We used Vensim's optimization feature (utilizing a standard Powell hill-climbing routine) to test the uncertain constants over their plausible ranges and found the best fit to the primary output data. The best-fit metric was the Mean Absolute Error as a percent of the Mean (MAEM), for each of the primary output variables and for all of them combined. From this testing came the "optimized" version of the model.

We also used Vensim's sensitivity testing feature (using standard Monte Carlo randomization) to perform millions of runs. In these runs, we allowed the uncertain constants to vary over their plausible (or allowed) ranges according to triangular probability distributions. We set each triangular peak at the constant's optimized value, the lower extreme at the minimum allowed value, and the upper extreme at the maximum allowed value.

From these runs we found N "qualifying parameter sets" (QPS) with a good fit to the primary output data, a fit comparable to that of the optimized run. In particular, we selected as QPS only those runs for which the maximum of the individual MAEMs for each of the primary outputs was no greater than XX% (the value achieved by the optimized run), and where the average across the MAEMs was no greater than YY%. The average MAEM was ZZ% for the optimized run, and ZZ was appreciably less than YY.

The model's uncertain constants are listed in a table, with relevant sources and notes, as well as their optimized values, the min and max allowed values for each, and summary statistics for the values of these constants in the QPS. These summary statistics include, for each constant, the QPS mean, the 95% sample interval based on the QPS (CI: the 2.5 percentile and 97.5 percentile), and the sample range as a percentage of the min-to-max allowed range. These summary statistics indicate whether the assumed triangular distributions are adequately wide enough to capture the universe of well-fitting parameter sets. If the QPS mean values strayed away from the optimized value, or if the sample range were close to 100% of the allowed range, it would indicate that the assumed distributions were skewed or insufficiently wide.

In fact, only a handful of the uncertain constants has a QPS mean value that differed by more than X% from the optimized value (and only one [or none] by more than YY%), and all of the CI ranges are well within the allowed range. We conclude that the assumed distributions for the uncertain constants are appropriate for finding the universe of well-fitting parameter sets.

Another table lists the MAEM values for the optimized run, including MAEMs for each of the primary outputs; for all data points combined; for the average of the outputs; and the maximum MAEM across all of the outputs. This table also lists MAEM summary statistics for

the N QPS runs; those statistics include, for each of the primary output variables, the mean of the N QPS runs and its 95% sample interval.

The optimized run provides a better fit to the data (a lower MAEM) than the mean of the N QPS runs for all but two of the outputs. Moreover, the average of the MAEMs (A%) is lower for the optimized run than the mean of average MAEMs for the N QPS runs (B%). On the other hand, the maximum of the MAEMs is a little greater for the optimized run (C%) than it is for the mean of the N QPS runs (D%).

From these results, we conclude that our Monte Carlo-based approach provided N qualifying parameter sets that yield a good fit to the data, nearly as good as that of the optimized run.

The good fits to history of the optimized run may be seen in several figures, where we show line graphs for the simulated trajectories for each of the outcomes superimposed on the historical data (vertical bars), along with data sources and derivations.

The good fits to history of the N QPS runs may be seen in additional figures which present box-and-whisker plots (see <http://mathworld.wolfram.com/Box-and-WhiskerPlot.html>) for the N QPS, one plot for each year over the modeled time period, superimposed on the historical data (green dots).

#### Part two: testing alternative situations

We defined several conditions representing alternatives to be tested using the optimized parameter set, as well as against the N QPS. Each alternative represents a change to the system, often at a particular point in time. These changes are often parameterized (made controllable) by adding a switch or flag that controls alternative logic (invokes alternative logic when the switch is “on.” Often, alternatives are simply parameter changes at particular points in time, with constants being added to the model to represent the magnitude of the change and the timing of the change.

We selected several outcome metrics for comparing alternatives against the baseline run. Five of these six metrics are key cumulative (end of run) statistics for key results of interest. The sixth is the number the state of the systems at the end of the time-period of interest. The results are presented in a series of tables, one for each outcome metric. These tables also describe what the alternative configuration represents.

Each table presents the outcome metric for the baseline and for each of the alternatives. They also show the difference between each alternative outcome and the baseline outcome (“change vs baseline”), as well as the percentage difference (“% change vs baseline”). These results and calculations are shown for the optimized parameter set; and also for the QPS (summarized in terms of mean and the 95% sample interval).

When changes against baseline have a negative sign, it means that the net impact of the alternative being tested was harmful. When these changes have a positive sign, it means the net impact of the alternative was beneficial.

The final section of the report from which this documentation sample was extracted discussed what the results (described here in completely generic terms) actually mean in real context. Of course, this is exactly how any report of simulation results should end: explaining what all the modeling and analysis reveals about the system of interest that was previously unknown and very important to know!

In conclusion, it is not difficult to evaluate the degree of uncertainty in model outcomes due to uncertainty in the parameters used in the model. This paper has provided a flowchart that summarizes two ways to do this, and demonstrates one of these approaches with a concrete example. A template for documenting results was also provided.

#### Acknowledgements

The authors gratefully acknowledge Kevin Stoltz who wrote the Python code for graphing confidence interval trajectories.

## References

- Brander JA, M.S. Taylor MS. (1998). The Simple Economics of Easter Island: A Ricardo-Malthus Model of Renewable Resource Use. *The American Economic Review* 88(1):119-138.
- Cheung SH, Beck JL. (2009). Bayesian Model Updating Using Hybrid Monte Carlo Simulation with Application to Structural Dynamic Models with Many Uncertain Parameters. *Journal of Engineering Mechanics* 135(4), 243–255.
- Fiddaman T, Yeager L. (2015). Vensim Calibration and Markov Chain Monte Carlo. *Proc 33rd Int Conf Syst Dyn Soc*.
- Helton JC, Johnson JD, Sallaberry CJ, Storlie CB. (2006). Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab Eng Syst Saf* 91(10-11):1175-1209.
- Houghton J, Siegel M. (2015). Advanced data analytics for system dynamics models using PySD. *revolution* 3(4).
- Houghton JA, Siegel M, Wirsch A, Moulton A, Madnick S, Goldsmith D. (2014). A survey of methods for data inclusion in System Dynamics models: methods, tools and applications. *Composite Information Systems Laboratory (CISL) Working Paper CISL 3:2014-03*.
- Jeon C, Shin J (2014). Long-term renewable energy technology valuation using system dynamics and Monte Carlo simulation: Photovoltaic technology case. *Energy* 66:447-457.
- Osgood N. (2013). Bayesian Parameter Estimation of System Dynamics Models Using Markov Chain Monte Carlo Methods: An Informal Introduction. *Proc 31st Int Conf Syst Dyn Soc*. #1391.
- Osgood ND, Liu J. (2015). Combining Markov Chain Monte Carlo approaches and dynamic modeling. In: *Analytical Methods for Dynamic Modelers*, eds. Rahmandad H, Oliva R, Osgood ND. Cambridge MA: MIT Press; Chapter 5, pp. 125-169.
- Sterman JD et al. (2018). Does replacing coal with wood lower CO2 emissions? Dynamic lifecycle analysis of wood bioenergy? *Environ. Res. Lett.* Vol 13 (in press).
- Vrugt JA, ter Braak CJF, Diks CGH, Robinson BA, Hyman JM, Higdon D. (2009). Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling. *International Journal of Nonlinear Sciences & Numerical Simulation* 10(3):271-288.
- ter Braak CJF. (2006). A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. *Stat Comput* 16:239–249.
- Uehara T. (2013). Ecological threshold and ecological economic threshold: Implications from an ecological economic model with adaptation. *Ecological Economics* 93:374-384.
- Uehara T, Nagase Y, Wakeland W. (2016). Integrating economics and system dynamics approaches for modelling an ecological–economic system. *Systems Research and Behavioral Science* 33(4), 515-531.
- Wakeland W, Hoarfrost M. (2005). The Case for Thoroughly Testing Complex System Dynamic Models. *Proc. 23rd Int Conf of the Syst Dyn Soc*.

## Appendix 1: SSTATS Vensim Macro (need to give credit)

```
:MACRO: SSTATS(historical,simulated:R2,MAE,MAE over Mean,MAPE,RMSE,MSE,Um,Us,Uc,Count\
)
SSTATS = residuals
~      simulated
~      Note that first argument (historical) must be data;
~      second argument (simulation) can be data or a simulation,
~      but if it is data, it will not be checked for existence.
~      This could easily be changed by modifying the code below.
~      Arguments following the : are outputs. They generate model
~      variables that are visible in the listing on the Variable
~      tab of the control panel, and can be used in equations
~      and custom graphs/tables/reports. However, they cannot be
~      made visible on diagrams.
|
R2 = r*r
~      Dimensionless
~      Correlation coefficient squared
|
MAE=
  ZIDZ(sum ae,Count)
~      simulated
~      mean absolute error
|
MAE over Mean=
  ZIDZ(MAE,meanx)
~      Dimensionless
~      Mean Absolute Error as a fraction of the mean of the data
|
MAPE = ZIDZ(Sum APE,Count)
~      Dimensionless
~      Mean Absolute Percent Error is reported as fraction rather than
~      percentage (x100) for consistency with other the other metrics
|
RMSE = SQRT(MSE)
~      simulated
~      Root Mean Square Error
|
MSE = dif mea + dif var + dif cov
~      simulated*simulated
~      Mean Square Error. The addition of the three components
|
Um = ZIDZ(dif mea,MSE)
~      Dimensionless
~      Bias inequality proportion
|
Us = ZIDZ(dif var,MSE)
~      Dimensionless
~      Variance inequality proportion
|
Uc = ZIDZ(dif cov,MSE)
~      Dimensionless
~      Covariance inequality proportion
|
Count = INTEG(pick/dt,0)
~      Dimensionless
~      Counter for # of points
|
residuals=
```

```

        IF THEN ELSE(pick, Yi - Xi, :NA:)
        ~      simulated
        ~      Errors
        |

r = MIN(1, ZIDZ (Mxy-(meanx*MeanY), Sx*Sy))
~      Dimensionless
~      Correlation coefficient. Calculated through the 'hand computation' formula.
~      Sterman (1984) pg. 63
|

sum ae= INTEG (
    ABS(Xi - Yi)/dt,
    0)
~      simulated
~      Sum of Absolute Errors
|

meanx = ZIDZ (Sum Xi, Count)
~      simulated
~      Mean of x (sum x)/n
|

Sum APE = INTEG (ABS(ZIDZ (Xi-Yi, Yi))/dt, 0)
~      Dimensionless
~      Sum of Absolute Percent Errors is reported as fraction rather than
~      percentage (x100) for consistency with other the other metrics
|

dif mea = (meanx-MeanY)*(meanx-MeanY)
~      simulated*simulated
~      Difference of Means (bias)
|

dif var = (Sx-Sy)*(Sx-Sy)
~      simulated*simulated
~      Difference of variances
|

dif cov = 2*Sx*Sy*(1-r)
~      simulated*simulated
~      Difference of covariances
|

pick= IF THEN ELSE(Y = :NA: :OR: X = :NA:, 0, 1)
~      Dimensionless
~      Takes a value of one for every data point available, assuming the data are \
~      available at intervals of Interval between the Start Time and End Time.
|

dt = TIME STEPS$
~      Time$
~      |

Yi = pick*Y
~      simulated
~      Sampled simulated variable
|

Xi = pick*X
~      simulated
~      The historic data series
|

Mxy = ZIDZ (SumXY, Count)
~      simulated*simulated
~      Mean of x*y (sum x*y)/n
|

MeanY = ZIDZ (Sum Yi, Count)
~      simulated

```

*Addressing parameter uncertainty in SD Models 9/20/2019, page 22*

```
~      Mean of y (sum y)/n
|
Sx = SQRT(MAX(0,MX2-(meanx*meanx)))
~      simulated
~      Standard Deviation of x. Calculated using the 'hand computation'
      formula to calculate the standard deviation without prior knowledge of
      the mean. Sterman (1984), pg. 64 MAX prevents spurious numerical
      problems from roundoff
|
Sy = SQRT(MAX(0,MY2-(MeanY*MeanY)))
~      simulated
~      Standard Deviation of y. Calculated using the 'hand computation'
      formula to calculate the standard deviation without prior knowledge of
      the mean. Sterman (1984), pg. 64
|
Sum Xi = INTEG(Xi/dt,0)
~      simulated
~      Sum of x's (simulated)
|
Y = simulated
~      simulated
~      The simulated data series
|
X :RAW: := historical
~      simulated
~      The historical data input
|
SumXY = INTEG(Xi*Yi/dt,0)
~      simulated*simulated
~      Sum of x*y
|
Sum Yi = INTEG(Yi/dt,0)
~      simulated
~      Sum of y
|
MX2=
      ZIDZ(SumX2,Count)
~      simulated*simulated
~      Mean of x^2
|
MY2 = ZIDZ(SumY2,Count)
~      simulated*simulated
~      Mean of y^2
|
SumX2 = INTEG(Xi*Xi/dt,0)
~      simulated*simulated
~      Sum of x^2
|
SumY2 = INTEG(Yi*Yi/dt,0)
~      simulated*simulated
~      Sum of y^2
|
:END OF MACRO:
```

## Appendix 2: Example Vensim Code to Specify Error Variables

```
*****
.Stats
*****~
    Summary Statistics
    |
Max of all MAEM = MAX(MAEM Natural Resource S, MAEM Population L)
~
  Dmnl
~
  |

Residuals Natural Resource S = SSTATS(S actual,Natural Resource S:R2 Natural Resource S\
,MAE Natural Resource S,MAEM Natural Resource S,MAPE Natural Resource S,RMSE
PNatural Resource S\
,MSE Natural Resource S,Um Natural Resource S,Us Natural Resource S,Uc Natural
Resource S\
,Count Natural Resource S)
~
  Dmnl
~
  |

Residuals Population L = SSTATS(pop actual,Population L:R2 Population L,MAE Population L\
,MAEM Population L,MAPE Population L,RMSE Population L,MSE Population L,Um
Population L\
,Us Population L,Uc Population L,Count Population L)
~
  Dmnl
~
  |

Simple avg of all MAEM = (MAEM Natural Resource S + MAEM Population L) / 2
~
  Dmnl
~
  |
```



### Appendix 3: Sample .voc and .vpd files for the “Estimate Uncertain Parameters” Step

#### .voc

```
:OPTIMIZER=Powell
:SENSITIVITY=Off
:MULTIPLE_START=RRandom
:RANDOM_NUMBER=Default
:OUTPUT_LEVEL=On
:TRACE=Off
:MAX_ITERATIONS=1000
:SIMS_MAX=10000
:RESTART_MAX=5
:PASS_LIMIT=2
:FRACTIONAL_TOLERANCE=0.0003
:TOLERANCE_MULTIPLIER=21
:ABSOLUTE_TOLERANCE=1
:SCALE_ABSOLUTE=1
:VECTOR_POINTS=25
:MCINITMETHOD=2
:MCPAYOFFTYPE=0
:MCRECORD=0
:MCSCHEDULE=0
:MCLIMIT=10000
:MCBURNIN=0
:MCNCHAINS=4
:MCOUTLIER=0.05
:MCGAMMA=0.5
:MCEPSILON=0.01
:MCDELTA=0.001
:MCJUMP=0.05
:MCUPDATEPAIRS=2
:MCXOVER=0.2
:MCTEMP=1
:MCFTEMP=1
:MCCOOLING=1000
0.003<=Regeneration Rate r=0.04<=0.1
0.03<=Maximum birth rate b0=0.1<=0.5
0.03<=Maximum death rate d0=0.2<=0.5
0.3<=Sensitivity of birth rate to manufactured good intake b2=1<=10
0.3<=Sensitivity of birth rate to resource good intake b1=1<=10
0.3<=Sensitivity of death rate to manufactured good intake d2=1<=10
0.1<=Sensitivity of death rate to resource good intake d1=1<=3
1<=tau=20<=100
.001<=sensitivity to price gap=.01<=.05
```

#### .vpd

```
*CG
Population L |pop actual/1
Natural Resource S |S actual/50
```

#### Appendix 4: Demo .vsc and .lst files for the Sensitivity MC run step

##### .VSC

```
100000,M,1234,,0
Maximum birth rate b0=RANDOM_TRIANGULAR(.03,.3,.03,.1,.3)
Maximum death rate d0=RANDOM_TRIANGULAR(.03,.3,.03,.08,.3)
Sensitivity of birth rate to manufactured good intake
    b2=RANDOM_TRIANGULAR(2,20,2,8.3,20)
Sensitivity of birth rate to resource good intake b1=RANDOM_TRIANGULAR(1,10,1,2.7,10)
Sensitivity of death rate to manufactured good intake
    d2=RANDOM_TRIANGULAR(3,30,3,10,30)
Sensitivity of death rate to resource good intake d1=RANDOM_TRIANGULAR(.3,3,.3,.96,3)
tau=RANDOM_TRIANGULAR(20,200,20,84,200)
sensitivity to price gap=RANDOM_TRIANGULAR(.0005,.005,.0005,.002,.005)
Regeneration Rate r=RANDOM_TRIANGULAR(.005,.05,.005,.016,.05)
```

##### .lst

```
Max of all MAEM
Simple avg of all MAEM
```