Building Confidence in Simulation Models using Automated Analyses

International System Dynamics Conference 2018
Reykjavík, Iceland ◇ August 6-10, 2018

Departement Wirtschaft
Introduction

- Model validation and documentation is a much discussed topic in SD
- Validation tests often lack details in their formulation and rely on the experience of the modeler (e.g. Sterman 2000, Forrester & Senge 1980)
- Modeling guidelines are often diverse (different variable types, different naming principles)
- Documentation is inconsistent (Rahmandad & Sterman 2012, Groesser & Tschupp 2012)
Introduction

- Model documentation has an excellent tool in SDM-Doc
- This has led to a standardization and improvement of model documentation
- PySD testing battery $\rightarrow$ Structure oriented behavior tests, not formulation
Objectives of the PySD testing battery

- Provide modelers with effective input for building model confidence
- Provide effective points of inquiry
- Test assumptions
Time Step test

“Cut the time step in half and test for changes in behavior.”

- $x_{ast} =$ value of stock s for timestep run with timestep a at timestep t
- $x_{bst} =$ value of stock s for timestep run with timestep b = 2a at timestep t
- t = 1, ..., m are integer values for each time step of the model runtime, with m being the final time step calculated as $m = \frac{\text{final time} - \text{initial time}}{\text{timestep}_b}$
- s = 1, ..., n are integer values assigned to all stocks in the model, with n being the total number of stocks in the model

$$\frac{1}{m} \sum_{s=1}^{m} \frac{1}{n} \sum_{t=1}^{n} \left| \frac{x_{ast} - x_{bst}}{x_{bst}} \right| \leq 0.025$$

- Iteratively test for timesteps from $\frac{1}{2^1}$ to $\frac{1}{2^9}$ until condition is met → b is optimal timestep
Extreme Condition test

“Subject model to large shocks and extreme conditions.”

<table>
<thead>
<tr>
<th>Exogenous variable name</th>
<th>Lower Bound Extreme value</th>
<th>Base Value</th>
<th>Upper Bound Extreme value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable a</td>
<td>0*x</td>
<td>x</td>
<td>10*x</td>
</tr>
<tr>
<td>Variable b</td>
<td>0*y</td>
<td>y</td>
<td>10*y</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## Extreme Condition test (2)

<table>
<thead>
<tr>
<th>Run</th>
<th>Variable</th>
<th>Flag Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exogenousvariable_setting</td>
<td>Variable where the flag occurred</td>
<td>Unexpected negative values</td>
</tr>
</tbody>
</table>

### Table

<table>
<thead>
<tr>
<th></th>
<th>Extreme run</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Base run</td>
<td>Only positive values</td>
<td>Not only positive values</td>
</tr>
<tr>
<td>Only positive values</td>
<td>Not flagged</td>
<td></td>
<td>Flagged</td>
</tr>
<tr>
<td>Not only positive values</td>
<td>Not flagged</td>
<td></td>
<td>Not flagged</td>
</tr>
</tbody>
</table>
Extreme Condition test (3)
Extreme Condition test (4)

For each table function, three runs are executed:

- Table value fixed at y-min
- Table value fixed at origin (0 or 1)
- Table value fixed at y-max
Extreme Condition test (5)
Error tracking

- Errors are tracked for every run
- Errors tracked are:
  - Division by 0
  - Floating point
  - Negative flows
  - Negative stocks
Discussion
Benefits of using automated analyses

- Time savings
- Comparability of results
- Rigor in formulation
- Rigor in test application
Future research

- Empirical validation of current tests
- Develop “error profiles” and automated interpretation
  - Preselection of output
- Develop model formulation standards driven by evaluation
  - Naming
  - Variable blocks