Model Debugging and Verification

Debugging

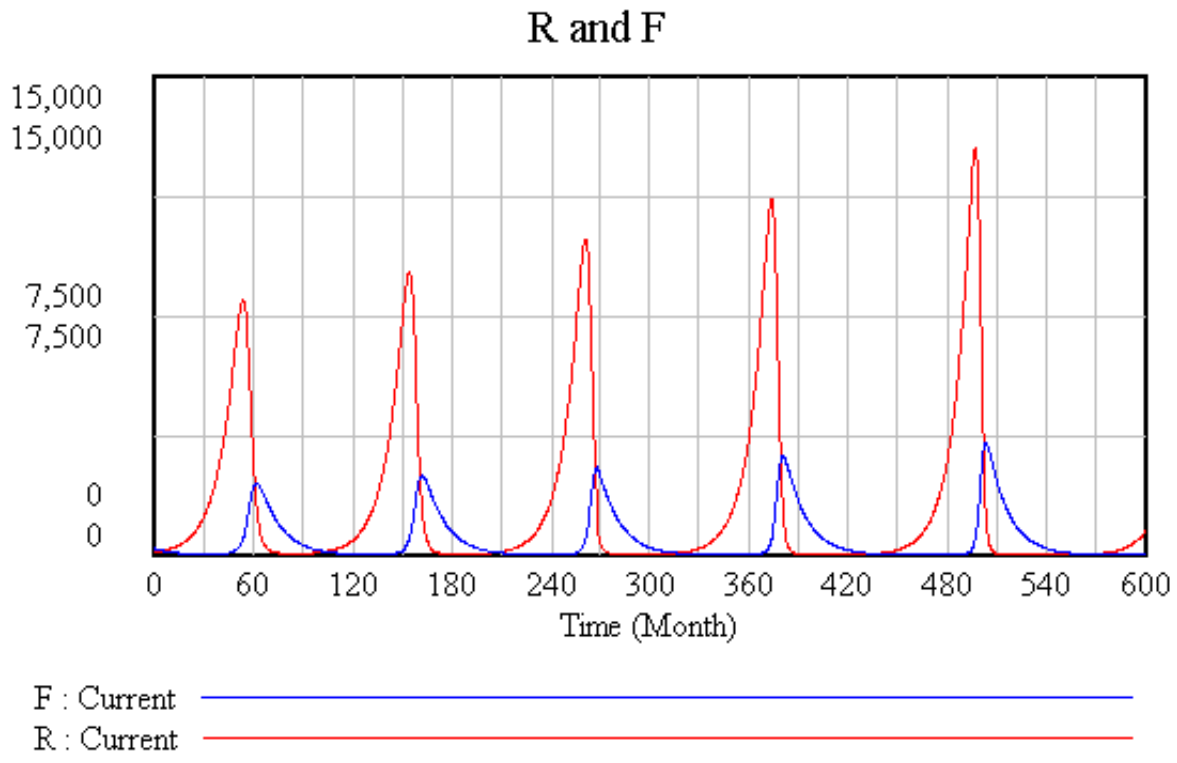
- Start very simple and don't add until simple part is working
- Use tables in addition to graphs (sometimes easier to spot problems)

  -- print out every dt
  -- shorten the run time way down
- When debugging or testing:
  - to neutralize an equation w/o deleting it, simply multiply by zero (may need enclose existing equation inside parentheses)
  - to replace an equation with a constant, neutralize it and add the constant, as shown below:

```
(resting_breaths_per_minute+.3*lung_delta)*0 + 60
```

- ALWAYS test your model with smaller DT and/or one of the sophisticated integration algorithms available in VENSIM

  -- e.g. slowly widening oscillations are a tipoff that there could be numerical problems
- The following plot was made using the predator prey model, but instead of using RK4 as with the original graph, Euler was used here:



-- this is an example of artifactual behavior or dynamics

Note: finding and studying loops may also help with debugging.

Verification vs. Validation

- Verification compares implemented (computer) model to conceptual model
- Validation compares conceptual model to reality
- Once verified, the implemented model can be compared to reality in order to validate it

Verification & Validation both involve Building Confidence:

- of the modeler(s)
  - doubting frame of mind
  - use external doubters
- of the user(s)
  - involvement throughout process
  - walkthroughs (helps modeler as well...)

Verification Intro.

Verification is the process of determining that a model operates as intended. Through the verification process, we try to find and remove unintentional errors in the logic of the model (debugging the model). Through this process we determine if we have built the model right (i.e., that it matches our mental model). The activities are restricted to the model itself; during verification, we make sure that the model contains as few errors as possible. This involves checking the design consistency (accuracy and correctness of modeling and solution methodologies, algorithms, numerical solutions) of the model.

Common Errors

- units of measure
- data (entry) errors--parameters, initial values, etc.
- arithmetic/equation errors (parentheses, formulas, ...)

  time scale errors

DT and artifactual delays & behavior

Artifactual delays

- storages are computed at a point in time
- flows take place between those points in time
- time jumps forward one dt at a time
- thus, any changes that occur *during* a dt are delayed until the next dt

Artifactual dynamics

- if dt is too big, sawtooth behavior is often a symptom
- if dt is large and the flow rate is changing rapidly, then changes in rate could be missed or "jumped over"

- the 1/2 dt test is used to identify artifactual behavior
  - reduce dt by 1/2 and rerun
  - is the behavior the same?

- Read **Sterman 21.1 - 21.2** (7 pgs), and
- Read **VENSIM Help (Technical docs) under topic Simulating Models (Chap 8) as TIME STEP,** DT situations requiring special care (~5 pgs).

More on Verification

The most difficult part of the verification process is not the correction of errors, but the isolation of errors.

The process of isolating errors involves:

- eliminating error messages (obviously)
- creating flow diagrams at the outset, to force the logic to be explicit during design
- creating documentation as one proceeds, to make the intended logic clear
- establishing a doubting frame of mind--the goal is to find scenarios that cause the model to fail so that we can isolate and correct errors.
- incorporating outside doubters--they should know and understand the modeled system and its intended operation, but they should not have been directly involved in its construction.
- conducting model walkthroughs--the modeler explains the model's logic to a small group of individuals who are familiar with the system being modeled
  - fresh eyes; improve group coordination
- performing test runs to reveal errors in the logic of the model
  - at first, test one portion of the model at a time; then test in combination later
- some ways of performing test runs in CSS are to:
  - set variables, including flows, temporarily equal to a constant
    - in VENSIM, to "preserve" an eqn., multiply it by zero, then add the desired constant
  - set initial conditions to zero
- hypothesis testing (fully exercising the model):
  - create a set of conditions
  - estimate performance that should result
  - run the model
  - compare
- use animation to "watch" the flows, storages, and auxiliary variables
- Other resources (available from the Instructor):
  - Modeling for Management, Vol. II, Ch.20, "Diagnosing Surprise Model Behavior" by N. Mass
  - Modeling for Management, Vol. II, Ch. 21, "Tests for Building Confidence in S-D Models" by Forrester & Senge
  - Richardson, Intro. to S-D Modeling w/DYNAMO, Ch.5.4

There is a Debugging and verification quiz associated with this page

At this point, you might want to learn about Model Dynamics (including how to diagnose surprising behavior).
