Using Insightmaker to Enhance Understanding in a First ODE Course

Erich McAlister

Fort Lewis College

SIMIODE Spring 2024 Webinars
April 17, 2024

FORT LEWIS COLLEGE
DURANGO, COLORADO

Outline

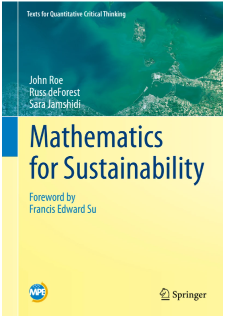
- What is Stock-Flow modeling via standard examples?
- An ODE modeling workflow example.
- Some Stock-Flow modeling idioms.
- Concluding discussion.

Potential appendix discussions (time permitting)

- Parsing Insightmaker files with Python (for advanced/more accurate analysis).
- The categorical framework for Stock-Flow models.

Background Story

- My first encounter with Insightmaker was through the text *Mathematics for Sustainability* by Roe et al.
- This is text is meant for a “gen ed” math course (without Calculus) focusing on sustainability issues.
- Using Stock-Flow models with Insightmaker makes ODE modeling accessible to this audience.
- Why shouldn’t I try this in a class that knows Calculus?



Class Parameters

- Calculus II (single variable) prerequisite.
- Mostly a service course for Engineering.
- Modeling focused, but perhaps not "modeling first". Maybe "modified modeling first".
- Using versions of SIMIODE scenarios and/or scenarios in SIMIODE textbook, which students are encouraged but not required to buy.
- Class is held in a computer lab at least one day per week (every day if possible).

Possible Student Outcomes

Possible student outcomes: A student comes away thinking one of two things

1. A differential equation in a bunch of stuff with an equals sign. Solving one means applying some procedure according to how it looks. At the beginning you do integrals, then quit for some reason.
2. A differential equation is a mathematical model relating changing quantities. Every symbol in a differential equation has a specific contextual meaning. Solving a differential equation means finding values of the changing quantities one way or another. I can do things that I am interested in using differential equations.

System Dynamics/Stock-Flow Modeling

Stock-flow (aka compartmental) models have the following components:

- **stocks $\leftrightarrow$ dependent variables**
- **variables $\leftrightarrow$ parameters and other quantities derived from the model**
- **links:** A link is an arrow between a stock, flow, and/or variable that allows the linked object to be used in the formula for the object to which the link points.

Briefly, the key formula to manage a stock-flow model is the following:

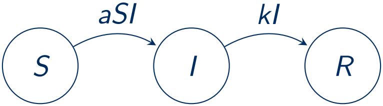
$$\text{Time Derivative of a Stock} = \sum \text{Inflows} - \sum \text{Outflows}$$

Example: The *SIR* model

The most common encounter with stock-flow diagrams for systems of ODEs occurs in infectious disease modeling. Here we have the *SIR* (susceptible-infected-recovered) model given by the ODEs

$$\begin{aligned} S' &= -aSI \\ I' &= aSI - kI \\ R' &= kI. \end{aligned}$$

These equations are commonly presented in graphical form as



Example: The *SIR* model

Let's explore what this looks like as a completed stock-flow model in Insightmaker, and how a student might approach the prompts in Sheila Miller's SIMIODE scenario (links in chat too).

- Modeling Scenario is [Mil22]
- Insight

Example: The *SIR* model

This modeling scenario is one among many that could be introduced in the first week of an ODE class. In addition to the graphical presentation of the model (which is ubiquitous in infectious disease modeling), we see how Insightmaker makes the introduction of several important habits/ideas/methods of ODE modeling seem natural. These include

1. the importance of units/dimensional analysis in ODE modeling,
2. different graphical representations of solutions (time series, phase portrait, and table),
3. and adjusting values of model parameters to fit real world data.

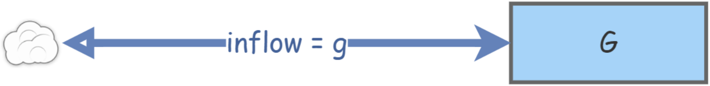
Remarks: For item (1), you *can* leave everything unitless. This is a battle you have to choose to fight. Items (2) and (3) will be discussed further.

(Fundamental) Example: The integral

Q: Why don't we see stock-flow models more outside of infectious disease modeling?

Possible A: We (at least initially) *really* like to see flows going from one stock to another. However, for that to happen the stocks must have the same dimension (such as individuals). This doesn't happen in many of the models we encounter in an introductory ODE course.

The solution is to allow for *partial flows*, i.e. ones without a stock upstream (or downstream, depending on signs). The basic example is the integrator.



Example: ODE modeling workflow with ripcord toys

Now let's work through another example from SIMIODE. It is the Ripcord Toys scenario by Brody Johnson.

- Modeling Scenario is [Joh23]
- Insight
- In case you don't know what a Ripcord Toy is.

The ODE model in question is

$$\frac{dv}{dt} = \begin{cases} F - kv, & 0 < t < t_0 \\ -kv, & t > t_0, \end{cases}$$

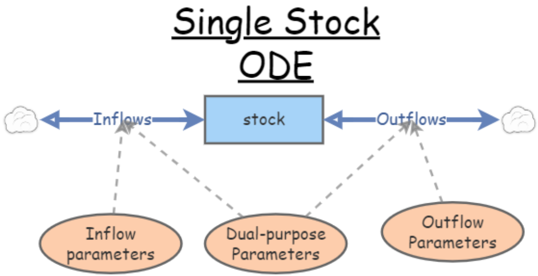
where F is the (normalized, i.e. divided by mass) frictional force from the ground (while the wheel is skidding), t_0 is the time skidding stops, and k is a resistance coefficient.

Example: ODE modeling workflow with ripcord toys

What are some takeaways from exploring this scenario with Insightmaker?

- We see a complete (and semantically meaningful) visual representation of an ODE modeling problem workflow:
  - The time derivative of each state variable is its inflow minus its outflow, and inflows and outflows can be broken down into meaningful pieces.
  - The idea that second order equations are just first order systems is quite intuitive.
  - Parameters are created and estimated from data in an accessible way for beginning students.
- Representations of systems in Insightmaker, as well as the syntax is very modern, and perhaps more familiar to our students.

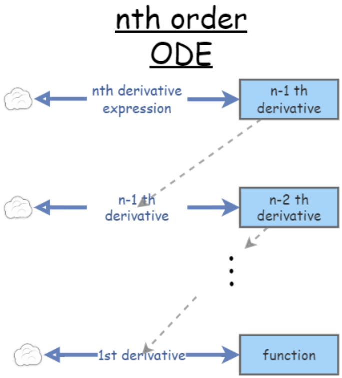
The First Order Equation



This is the generic diagram for a first-order equation. Depending on the scenario you may

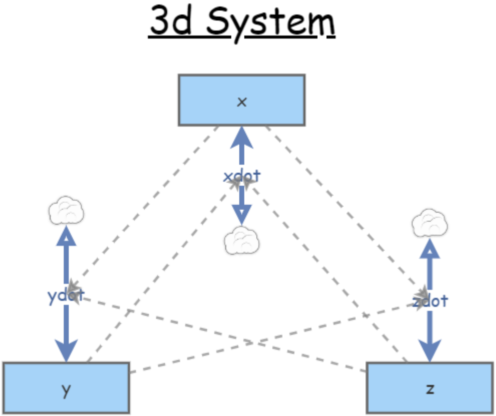
- restrict flows to positive rates only and
- break the inflows and outflows into more components.

A Higher Order Equation



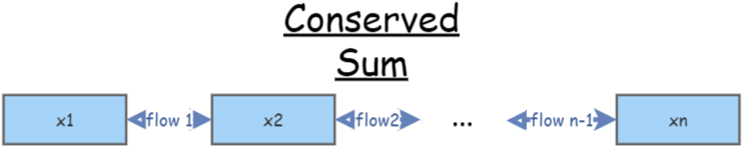
This is the generic diagram for a higher-order equation. Numerical parameters and external (time-dependent) forces may be added as variables.

A Coupled System



This is the diagram for a three-dimensional system where all the variables are interacting.

A Conserved Sum



This diagram translates to

$$\frac{d}{dt}\sum_{i=1}^{n} x_i = 0.$$

This idiom is present in the *SIR* model. It can also be useful in models involving energy transfer, such as this modified take on the Ripcord Toy model.

Take Home Tasks:

Here are a couple little tasks that you can do to further familiarize yourself with what we've seen in these examples. Each should begin with a clone of the linked *SIR* model.

- Take the data from the *SIR*-modeling scenario and create a converter then find the parameter values that minimize the SSE.
- Decide how you would modify the given *SIR* model in the following scenarios:
  - On one of the days (say day three), all of the susceptible students are isolated from all the infected students. (Follow-ups with a class may include deciding why this may be unreasonable as well as solving the post-quarantine ODEs analytically.)
  - Assuming individuals in each category are readily identifiable, how would you modify the model if some fraction of the susceptible population could be inoculated each day?

How does it fit into our classes?

My answer(s):

1. Modeling with Insightmaker does not (and cannot) replace all the analytic methods and qualitative analysis in the class I teach.
2. Analytic methods are deemed necessary by client disciplines (I don't ask them why) and they also serve as a "met before" for students. Modeling projects with Insightmaker (and other software) creates a nice positive reinforcement loop when combined with more traditional methods.
3. Insightmaker helps students' ODE literacy as well as their ODE communication skills.

What are Insightmaker files good for?

Insightmaker exports files that you can then import back to Insightmaker to run. Why might we want to work with these?

- They enable us to share Insights with our classes and have them turn in Insights for grading while keeping them private. *When you create and save an Insight, it defaults to being public. That's great, but if Insightmaker is used in teaching, we need to keep these things private.*
- Insightmaker files use xml to store the information in an Insight. This means we can parse the model in other languages (such as Python) if we desire. We'll now explore why (and how) we might do that.

Another Example

Here we are going to explore an Insight based on the "perturbed model system I" in [UI05].
The system of ODEs is given by

$$\begin{aligned} \frac{ds}{dt} &= s(1-s) - \frac{ms}{a+s} + s\sin(\omega t) \\ \frac{dm}{dt} &= -bm + \frac{\epsilon ms}{a+s} - m\sin(\omega t) \end{aligned}$$

(In the class scenario s is for snakes and m is for mongooses.)
Insight here. Python code available here (very beta).

Why export to other programs?

Umm... because it's fun to do stuff like that. Well, really...

- Improved graphics.
  - Graphs of all types just have better resolution in Python, Matlab, etc.
  - If you want things like slope/vector fields, you need something other than Insightmaker.
- Long runs with small step sizes go really slow in Insightmaker.
- The numerical algorithms available in Insightmaker are limited.
- If you want things like slope/vector fields, you need something other than Insightmaker.

On the other hand, some things (for example, discontinuous functions and positive flow rate only restrictions) are easy with Insightmaker.

How the Python parsing works.

What's interesting is the way you can parse the Insightmaker file into a system of ODEs for Python to deal with. The process is *roughly* as follows:

1. Find all the `Stock` tags and create a table with a row for each stock and with one column having the stock name and one for the initial value.
2. Find all `Flow` tags and create another table with a row for each flow and with columns for name, source, target, and formula. The source and target indicate where the flow starts and ends (indicating a positive direction).
3. Build another table for variables and assign their values.
4. Loop through the Stock data frame and build the system of ODEs from the Flow data frame adding a term for each flow with the given stock as target and subtracting a term for each flow with the given stock as source.

Now you have the system of ODEs built and you can do ODE things with them.

Category Theory?

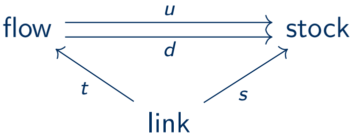
(At this point we may be going off the deep end, sorry.)
The strategy for parsing the Insightmaker files into systems of ODEs is actually based on realizing them as instances of a category. This is because a collection of connected tables (i.e. a relational database) is a functor $F : \mathscr{C} \to \mathtt{FiniteSets}$ for some finite category $\mathscr{C}$ as follows:

- There is one table (primary key) for each object in $\mathtt{H}$.
- If $A \in \mathtt{Ob}(\mathscr{C})$, the rows (records) of its corresponding table are labeled by elements of $F(A)$.
- Each column in a table corresponds to a morphism α with A as the domain, with entries given by $F(\alpha)$.

For this category theoretic discussion, the primary reference is [Bae+23].

Instances of a Category

Consider the (free) category, $\mathtt{H}$ given by the diagram:



For a particular stock-flow model we can define a functor $F : \mathtt{H} \to \mathtt{FiniteSets}$ where

- $F(stock)$ = the set of stocks in our stock-flow model,
- $F(flow)$ = the set of flows,
- $F(link)$ = the set of links between stocks and flows,
- $F(u)(a\ flow)$ and $F(d)(a\ flow)$ define the positive direction of a flow in the stock-flow model, and
- $F(s)(a\ link)$ and $F(t)(a\ link)$ define which flow is connected to a given stock by that link.

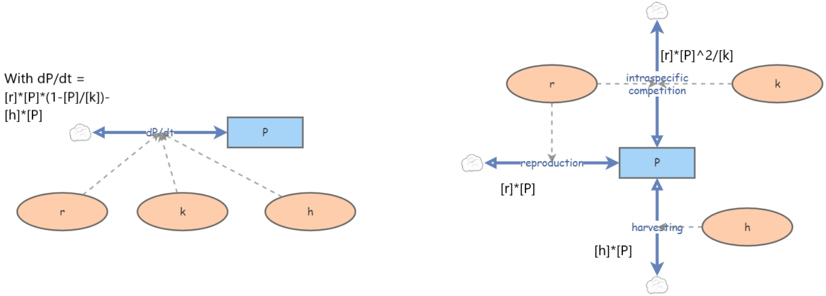
Why?!

Ok, why on Earth do I bring this up?
The categorical framework gives a (possible) answer to the question of why interacting with ODEs and stock-flow diagrams is different than interacting with them as equations.

- Stock-flow diagrams are realized as functors from the category $\mathtt{H}$ to the category of finite sets. We then obtain a category of stock-flow diagrams as the functor category $\mathtt{H}^{\mathtt{Set}}$ and morphisms are natural transformations.
- This imposes an idea of isomprphism that is stronger than the differential equations. That is, **a single system of ODEs has many non-isomorphic (in this functor category) stock-flow diagrams**.

Illustrative Example

The following two stock-flow diagrams represent the same ODE (logistic growth with harvesting).



They are not isomorphic in the category of stock-flow diagrams. (Which one is more communicative?)

Composition and Idioms

In addition to regarding stock-flow diagrams as instances of a category, pasting together stock-flow idioms may also be cast in category theoretic terms. (I'm guessing we're out of time here.)

References I

[UI05] Ranjit Kumar Upadhyay and S.R.K. Iyengar. "Effect of seasonality on the dynamics of 2 and 3 species prey–predator systems". In: *Nonlinear Analysis: Real World Applications* 6.3 (2005), pp. 509–530. ISSN: 1468-1218. DOI: https://doi.org/10.1016/j.nonrwa.2004.11.001. URL: https://www.sciencedirect.com/science/article/pii/S1468121804000975.

[Mil22] Sheila Miller. *6-001-Epidemic-ModelingScenario*. June 2022. DOI: doi:/10.25334/N6DF-2J82. URL: https://qubeshub.org/community/groups/simiode/publications?id=2956&v=1.

References II

[Bae+23] John Baez et al. "Compositional Modeling with Stock and Flow Diagrams". In: *Electronic Proceedings in Theoretical Computer Science* 380 (Aug. 2023), pp. 77–96. ISSN: 2075-2180. DOI: 10.4204/eptcs.380.5. URL: http://dx.doi.org/10.4204/EPTCS.380.5.

[Joh23] Brody Dylan Johnson. *3-100-Ripcord-Toys-ModelingScenario*. June 2023. DOI: doi:/10.25334/8CZ1-Y987. URL: https://qubeshub.org/community/groups/simiode/publications?id=3605&v=2.