Instructions for Lab 3

Modelling of Systems for Sustainability

12th October 2023

1 Introduction

1.1 Aim

The aims of this Lab session are to become familiar with the system dynamics framework in NetLogo, and gain intuition about system dynamics models by replicating some of the models in chapter 1 and chapter 2 of Meadows (2008).

1.2 Overview

You should work through all the instructions listed below. While the labs are not assessed, they are an integral part of the learning experience and will help you in getting the most out of this course. The labs are intended to be executed on DICE computers within the School of Informatics.

1.3 Prerequisites

You need to have NetLogo installed, and it would help to have finished Labs 1 and 2, so that you've gained some confidence with NetLogo.

2 Instructions for Lab 3

Note you may not get as far as, or complete, section 2.4. We may allocate more time for this in next week's lab.

2.1 NetLogo System dynamics tutorial: sheep and wolves

Do the tutorial on system dynamics modelling in NetLogo here: `https://ccl.northwestern.edu/netlogo/docs/systemdynamics.html`.

- Translation note: the tutorial and Meadows (2008) use the US English word "faucet" instead of the UK English word "tap" or "valve".
- In **Step 3 (Wolf predation)**, the instruction to "Add one more *Flow* from the wolves Stock to the Flow that goes out of the Sheep stock." should read "Add one more *Link*..." (our italics).
- If you don't get the behaviour you expect in any of the simulations, you will need to check the equations you used, especially in **Step 3 (Wolf predation)**.

2.2 Exploring and extending the NetLogo tutorial

1. Save your work so far, so you have a record of your work
2. Save your work to a new file name too, where you will make the following changes
3. Try changing some or the parameters (values in the Variables boxes, e.g., `sheep-birth-rate`, `predation-rate`). You may find that setting some values too large causes NetLogo to report numerical difficulties.
4. To help with the exploration, you may want to set the parameters from the main interface. To do this for the `sheep-birth-rate`:

(a) Add an **Input** from the main NetLogo window, Interface tab

(b) Give it the **Global Variable** `sheep-birth-rate-val` and give it the value of 0.04 (the original value of `sheep-birth-rate`).

(c) In the NetLogo System Dynamics Modeler window, edit the `sheep-birth-rate` to be `sheep-birth-rate-val` instead of a fixed number.

5. Can you find any sets of parameter values where the behaviour isn't oscillatory?

6. Can you find a set of parameter values that prevent the wolf population crashing below 1?

2.3 Implementing simple system dynamics models

Meadows (2008) has a number of examples of system dynamics models, complete with equations in the Appendix (from page 195 onwards). Try to replicate and investigate the following Figures in Meadows:

- Figure 10/11 (coffee cup). Note this is actually two models - one for coffee warming up and one for coffee cooling down.

- Figure 12/13 (interest-bearing account)

- Figure 27-28 (capital stock).

- Figure 15-18 (thermostat). Try to reproduce Figure 16 with the heat loss to the outside turned off, Figure 17 with the heating turned off, and Figure 18 with both heating and heat loss operative.

  In Netlogo for this model, each tick is one hour. Try changing `dt` in the system dynamics-modeler, which is 1.0 by default, for example to 0.3 and 0.1, and re-running, and notice the effect this has on the curves. Does it change the time taken (in ticks) to get close to the equilibrium? Is there overshoot? [1]

  Note that the `dt` parameter you see in the interface in the system-dynamics-modeler is in ticks too, so a `dt` of 0.1 means `dt` is 1/10th of a tick, so the system-dynamics-modeler procedure `system-dynamics-go increments <ticks>` by `dt` each time it is called. Ticks are the basic time unit in NetLogo. Similarly, flows are multiplied automatically `dt` (you can see it in the system-dynamics-code tab) so that the actual flow rate per tick is not dependent on `dt`. In a pure system-dynamics model (which this lab is about), this doesn't really matter, but it would in a hybrid Agent Based / System Dynamics model where the time stepping the AB part and the SD part would need to be treated consistently.

2.4 Implementing system dynamics models with delays

Meadows also has examples of system dynamics models with delays, which affect how feedback loops work. NetLogo does not have built-in support for delays, but we have created code that implements delays. We'll look at how to use this in some detail in a later lab, for now just use the model provided in the zipfile attached in the schedule on opencourse.

- Download the zipfile and unzip it into a folder. You will see two files

  ```
  delay.nls
  Figs30-36-inventory.nlogo
  ```

  Load the file `Figs30-36-inventory.nlogo` into NetLogo, it loads `delay.nls` itself.

- The model implements the diagram in Figure 31, and the interface has sliders to control the 3 delays built into the system. Read through the Meadows section "A System with Delays — Business Inventory and verify you can reproduce Figures 30, 32, 34, 35 and 36 using the delay settings given in the Appendix in the Business Inventory section

- Click on the "deliveries" flow in the system-dynamics-modeler window and click Edit in the toolbar.

- The "Expression" reads:

[1] Note if you want each click of your "go" button to execute one tick when `dt` is a fraction of a tick, you need to use something like the NetLogo "repeat" primitive to do (1/`dt`) system-dynamics-go commands, because each system-dynamics-go only advances ticks by `dt`.

```
deliveries-calc
```

This is a procedure call, the procedure must be in the main Code.

- Select the Code tab in the main interface (not the System Dynamics Modeler). Find the reporting procedure `deliveries-calc`
- You'll see that if `ticks > 5` then the value reported is

  ```
  get-delay delivery-delay "orders" 20
  ```

  This means 'Get the value of the "orders" variable `delivery-delay` time steps ago, and if the simulation time hasn't yet reached `delivery-delay` ticks, assume that the value of "orders" was 20.

This `get-delay` procedure is what you use to get delayed values of stocks, flows and variables. You have to be careful with flows, as they are scaled down by `dt`, so if you want to use the flow per tick, you need to scale the value back up (by dividing by `dt`. You can see an example of this in the code for `perceived-sales-calc`, which is working on the `sales` flow.

Note also that the time step parameter in the procdure get-delay (`delivery-delay` in this example) is in "ticks", i.e. the same ticks as when running an Agent Based Model. As discussed above, the `dt` parameter you see in the interface in the system-dynamics-modeler is in ticks too, so a `dt` of 0.1 means `dt` is 1/10th of a tick. Always think of the tick value when you are using delays, ignore `dt`.

When using delays, this means that if say `dt` is 0.1 then a delay of 1 tick in the `get-delay` call means that the value that is being delayed is delayed 10 `dts`, or 10 simulation steps in the system dynamics model, not 1. That's because we're producing ten values of the source per tick. The reason for this is that one often wants to experiment with `dt` to see what models the differential equations best - smaller `dt` is a better fit, but is also more computationally expensive.

Useful Links

Book	`https://read.kortext.com/reader/pdf/653121/Cover`
Book Resources	`https://www.railsback-grimm-abm-book.com/downloads-errata-2nd-edition/`
NetLogo	`https://ccl.northwestern.edu/NetLogo/`

References

Meadows, D. H. (2008). *Thinking in systems: a primer*. Chelsea Green.