Integrating real-time precision livestock data into dynamic models using R

Hector Menendez III, Jameson Brennan, Krista Ehlert, and Luis O. Tedeschi

South Dakota State University

OUTLINE

- Definitions
- Data/Code Access Links (See alberto folder)
- Dynamic Rumen Model (Tedeschi and Fox, 2018, 2020)
- How to transfer to Program R
- Simulation of dynamic model **(Program R)**
- Simulation of dynamic with points **(Program R)**
  - **Shiny App**

Definitions

- **Real-time:** The data are recorded and are simultaneously available.

  (Jacobs et al., 2022)
- **Precision Livestock Data:** Any data collection that is collect with hardware and software that can be automatically transferred digitally.
- (Menendez et al., 2022).
- **Dynamic Models:** Models account for “feedback” over time.
- (Sterman, 2000).
- **Vensim:** A visually based programing software.
- (https://vensim.com/free-download/)
- **Program R:** An open-source programming software. (https://www.rstudio.com/products/rstudio/download/)

Data Processing Pipelines in Program R for Multi-Integrated Technologies

Examples of precision data management and measurement tools that require data processing for research:

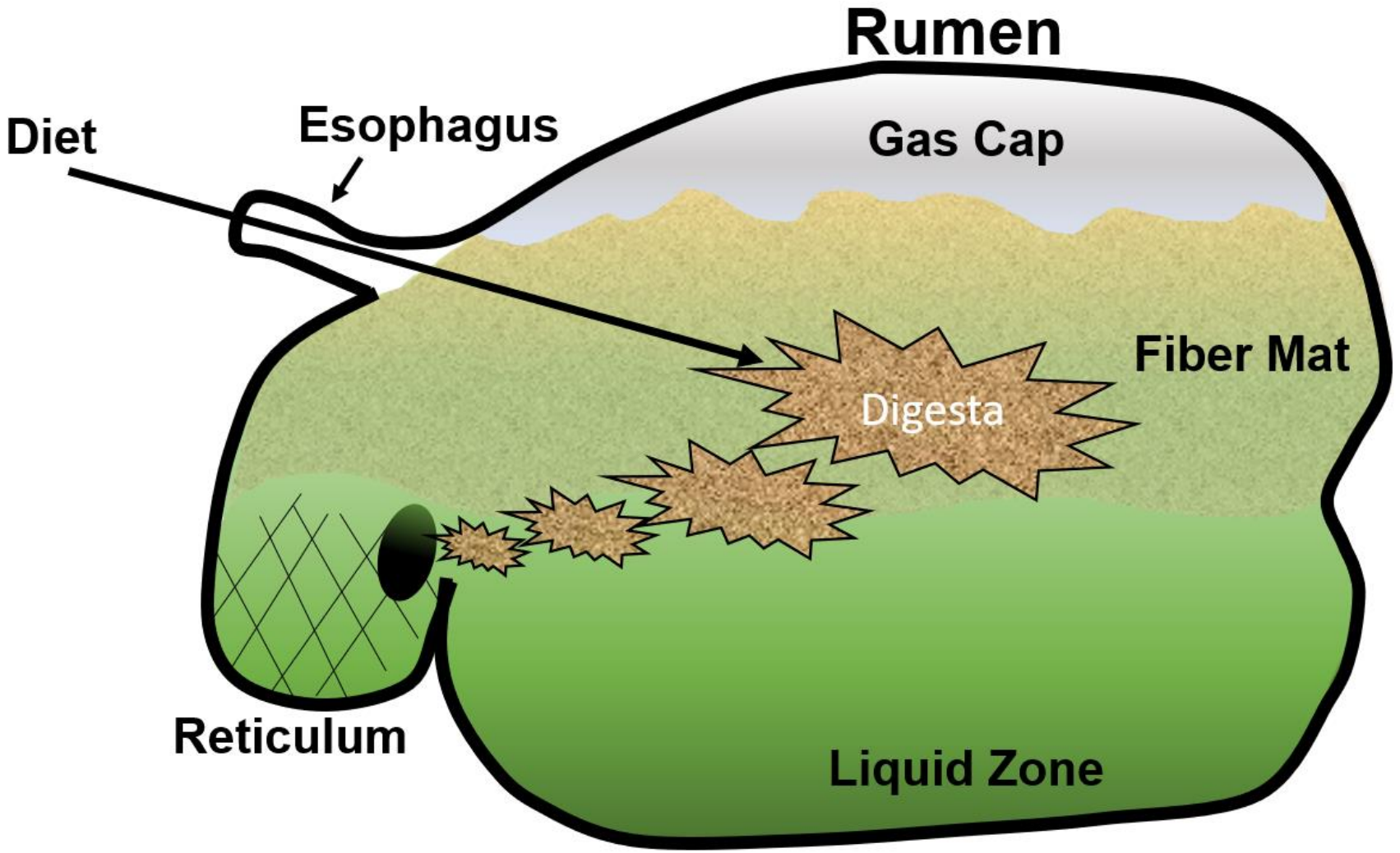
- GreenFeed™
- Smart Scales™
- Super Smart Feed Pro™
- Smart Feed Producer™
- Vence™
- Drones
- Soil Sensors


032


SMARTFEED


Super
SMARTFEED

Rumen Passage Rate: Aging Chain

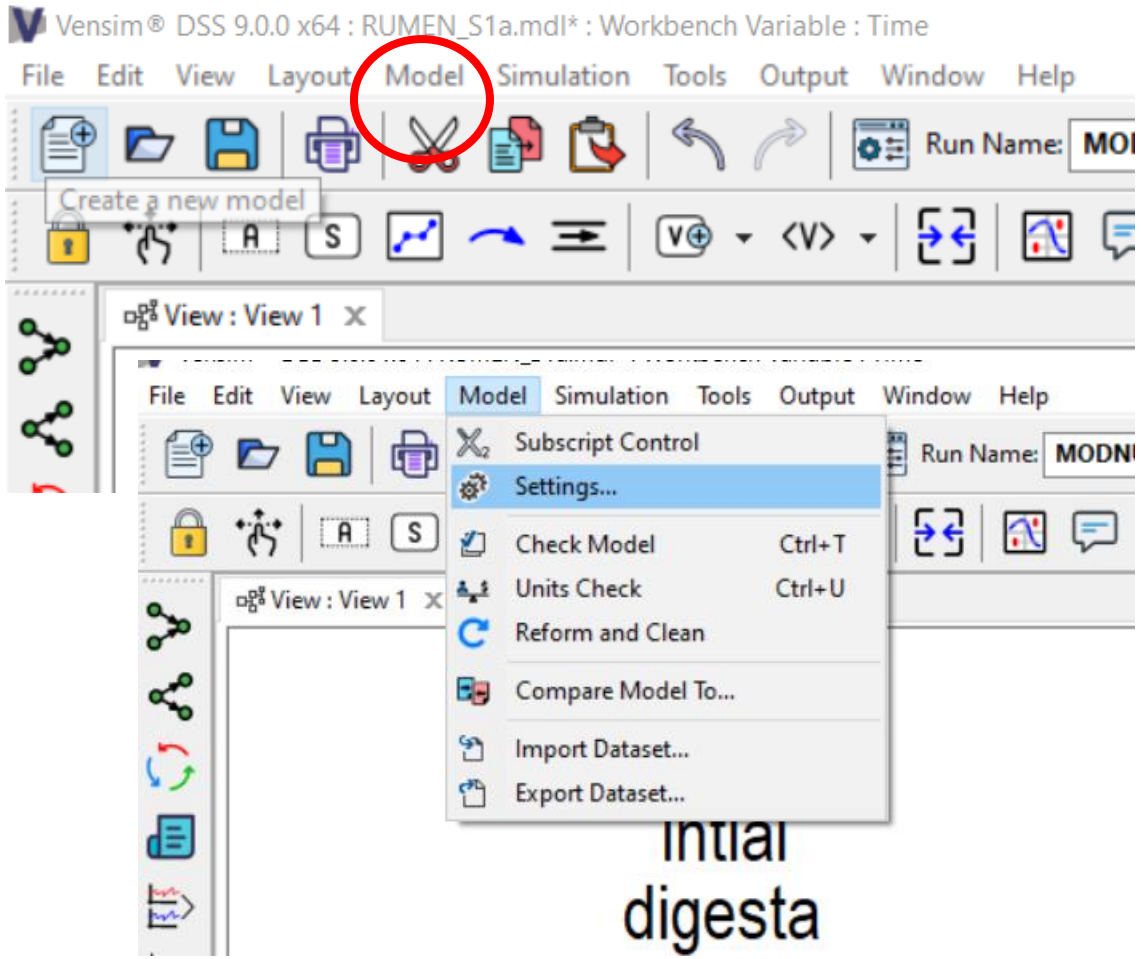


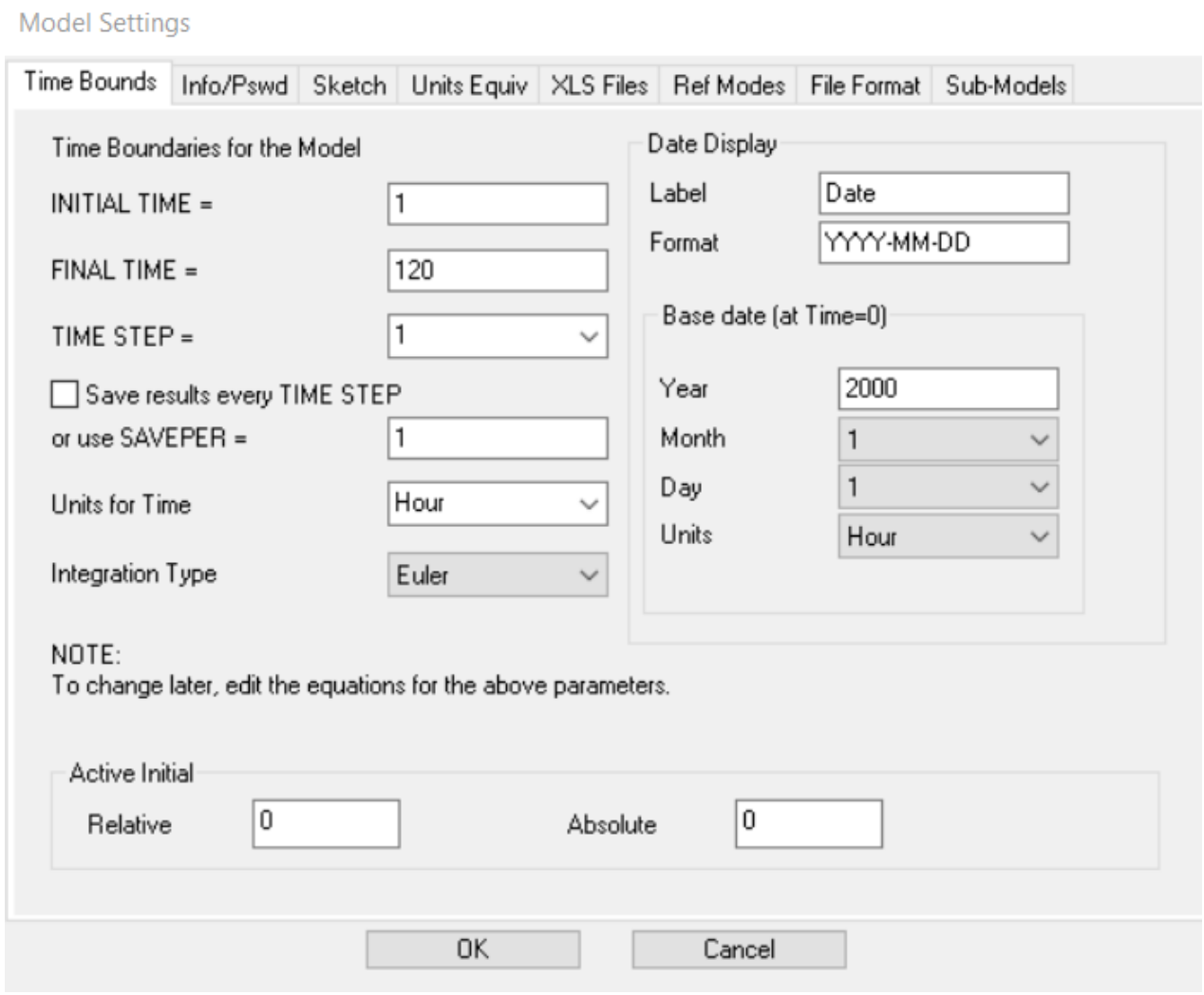
Model Components

- Continuous (Tedeschi and Fox, 2020)
- Perfect Mixing (Sterman, 2000)
- 4^{th} Order = 4 Stocks

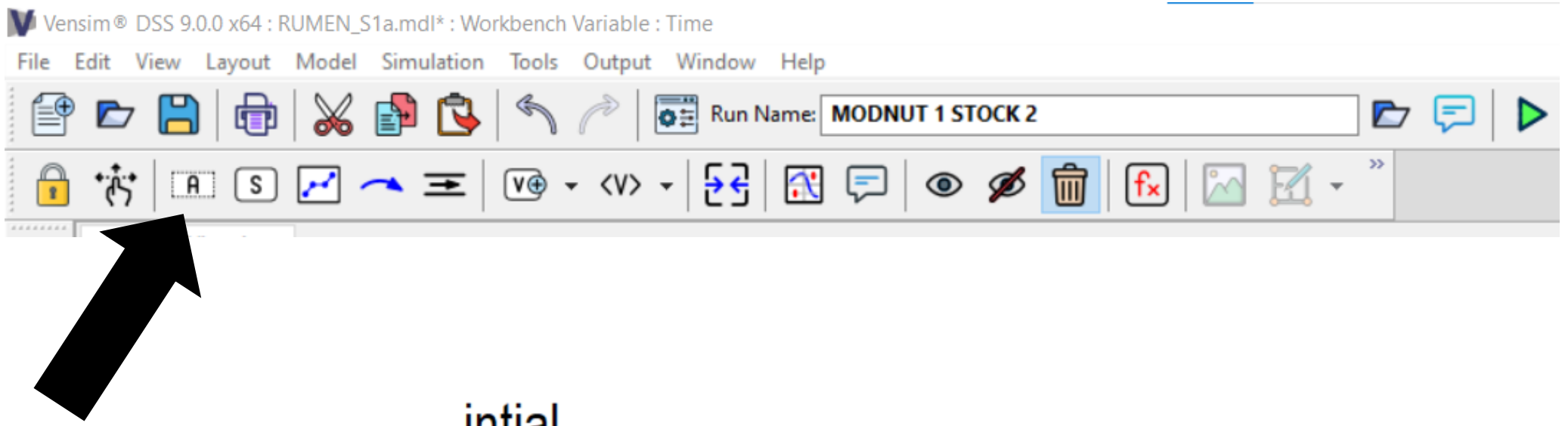
also known as “pools” or “levels”.

Step 1: Select Model and then Model Settings, set model parameters



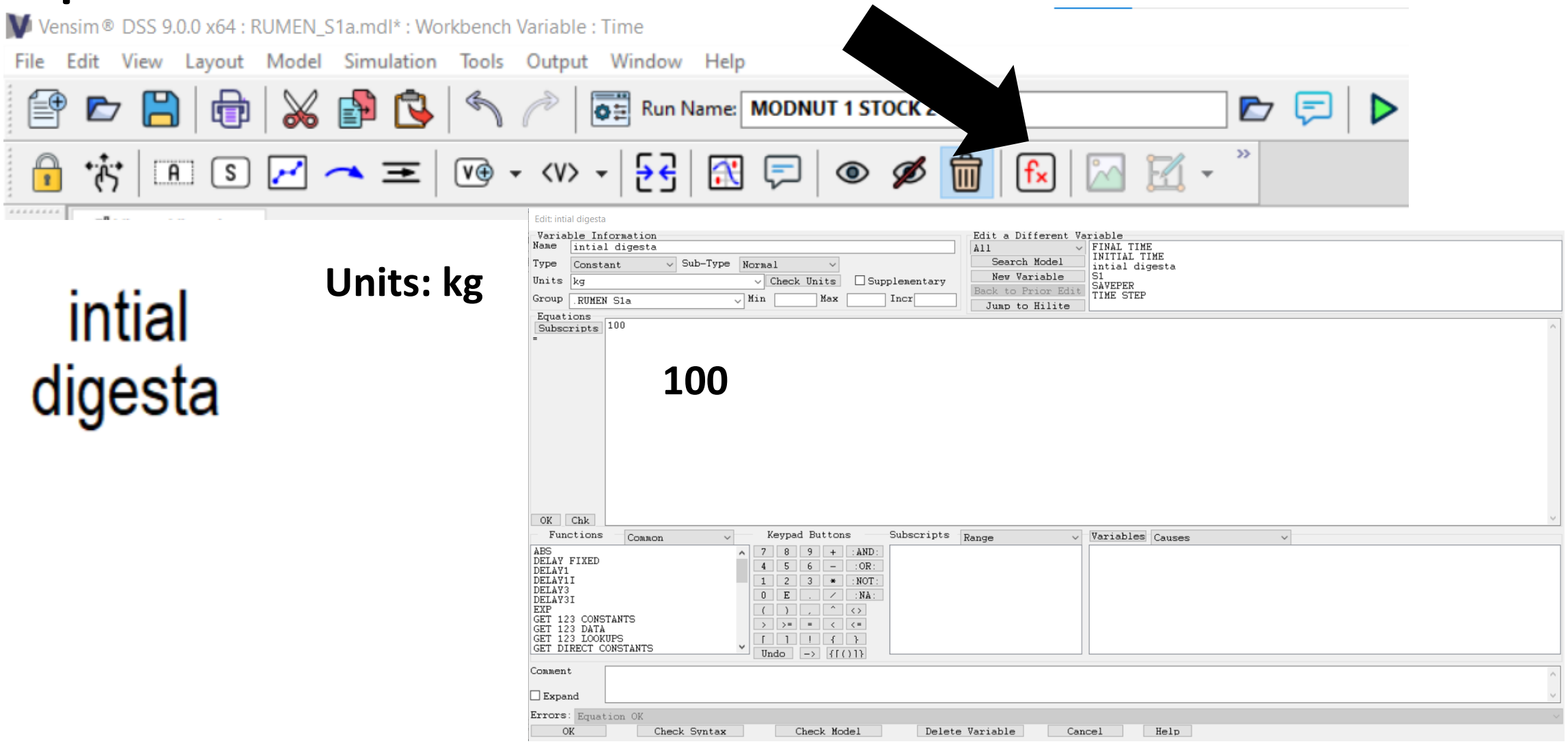


Step 2a: Create an initial value “initial digesta” add

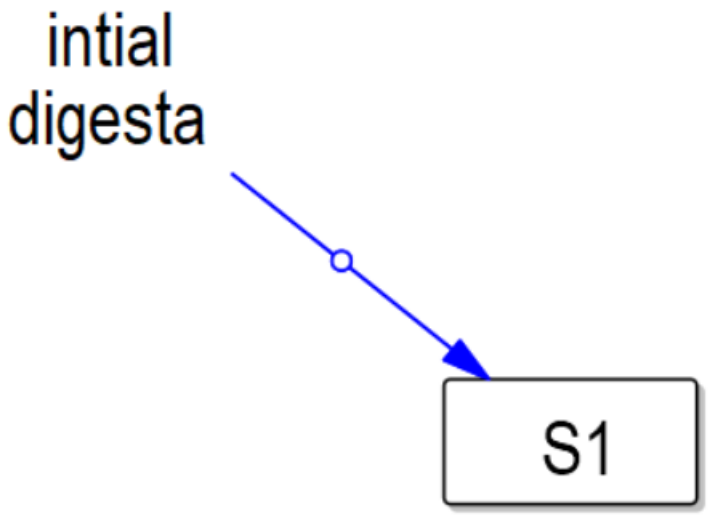


intial digesta

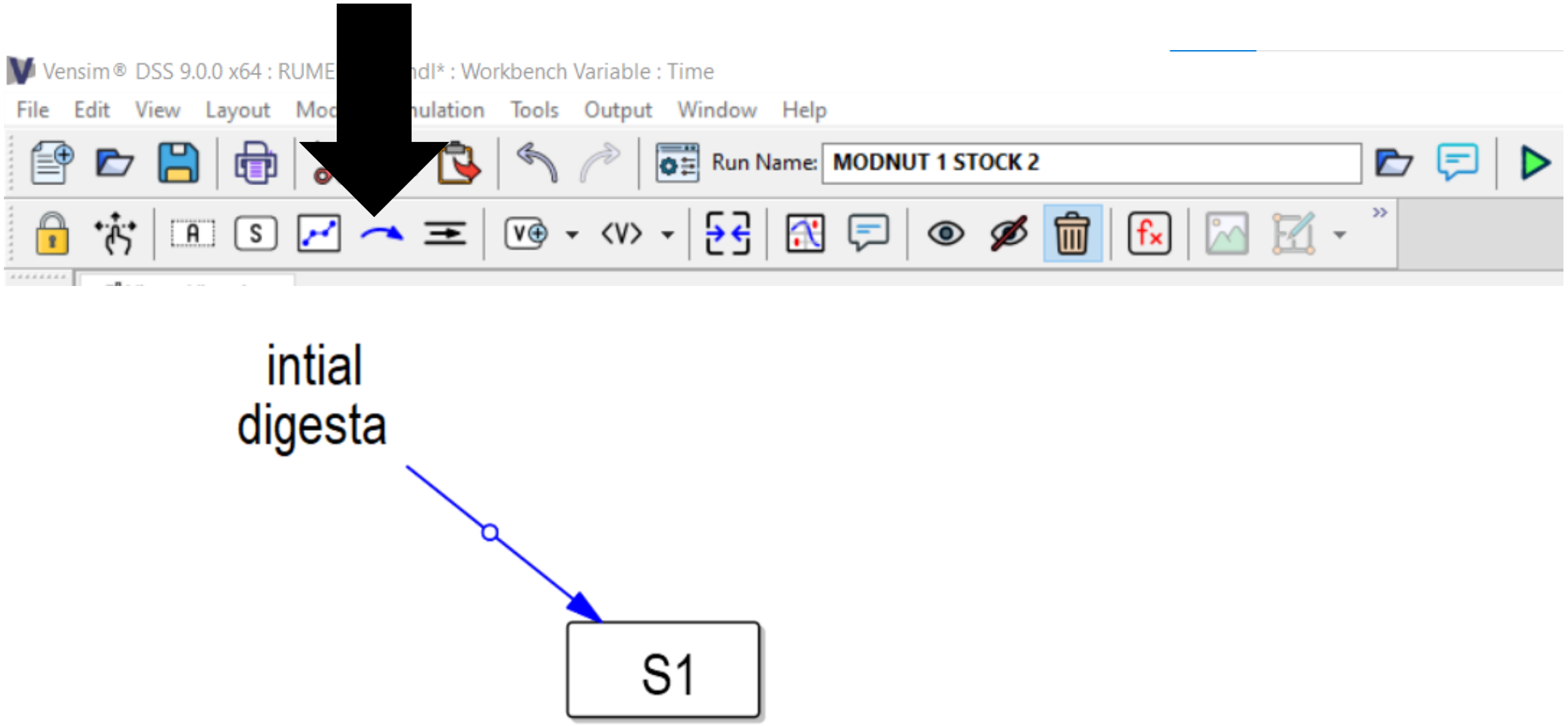
Step 2b: Use the equation editor to parameterize the variable.



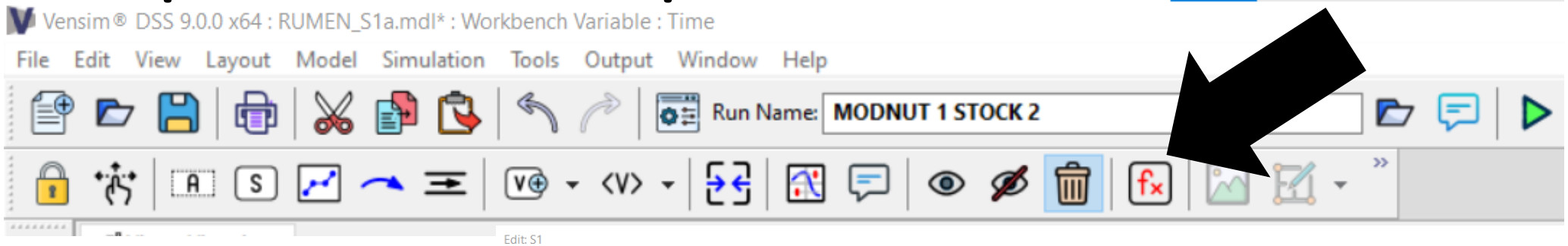
Step 2c: Create a stock called “S1”



Step 2d: Link “initial digesta” to the Stock (i.e., S1)

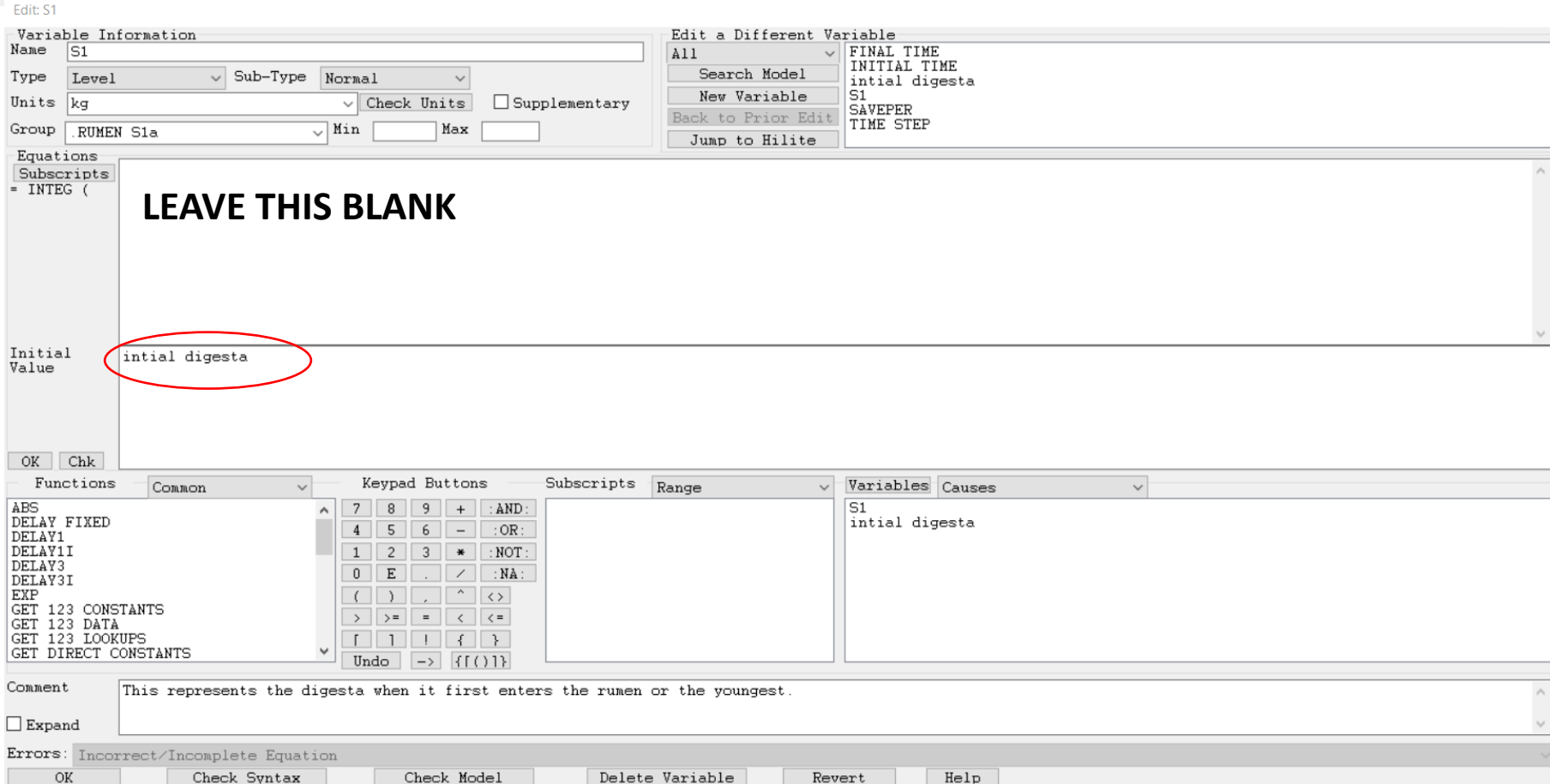


Step 2e: Set “S1” parameters and add units “kg”

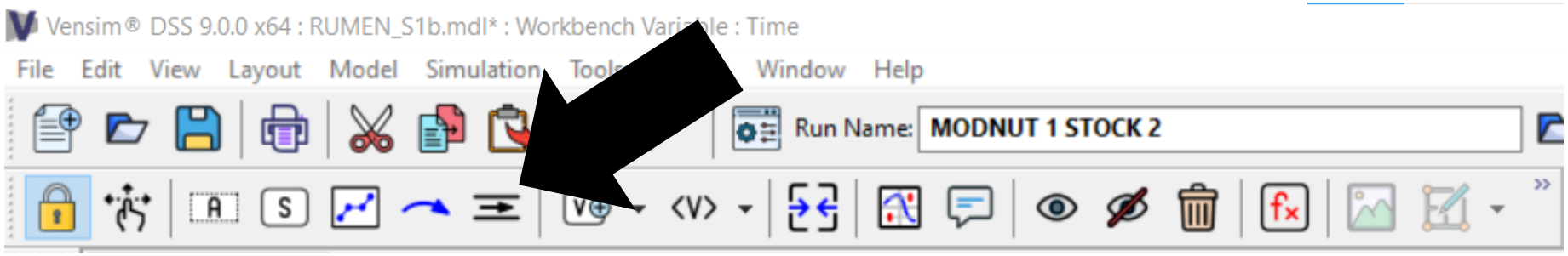


intial digesta

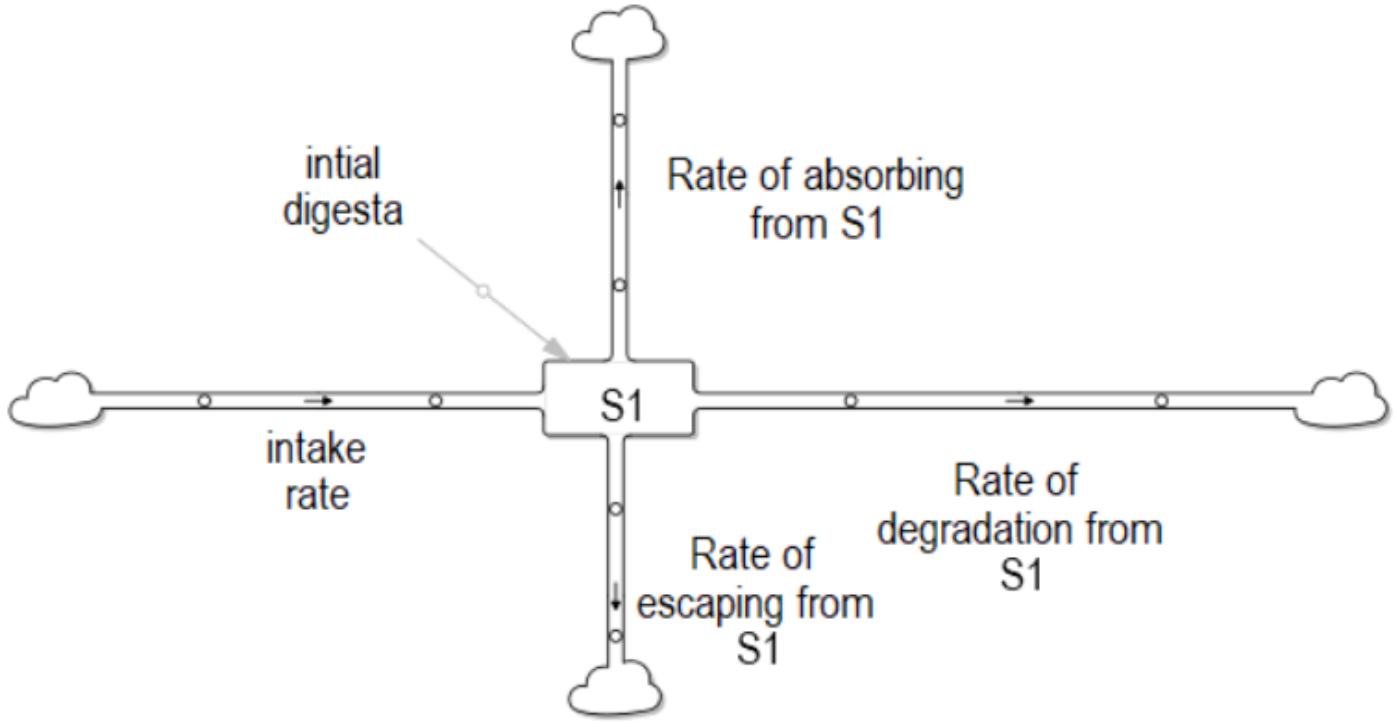
S1



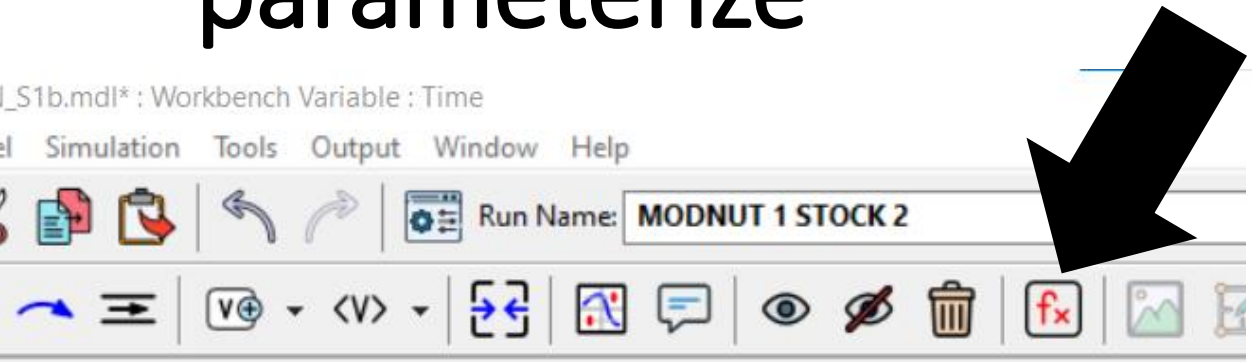
STEP 3a: Add Rates (i.e., inflow and outflows)

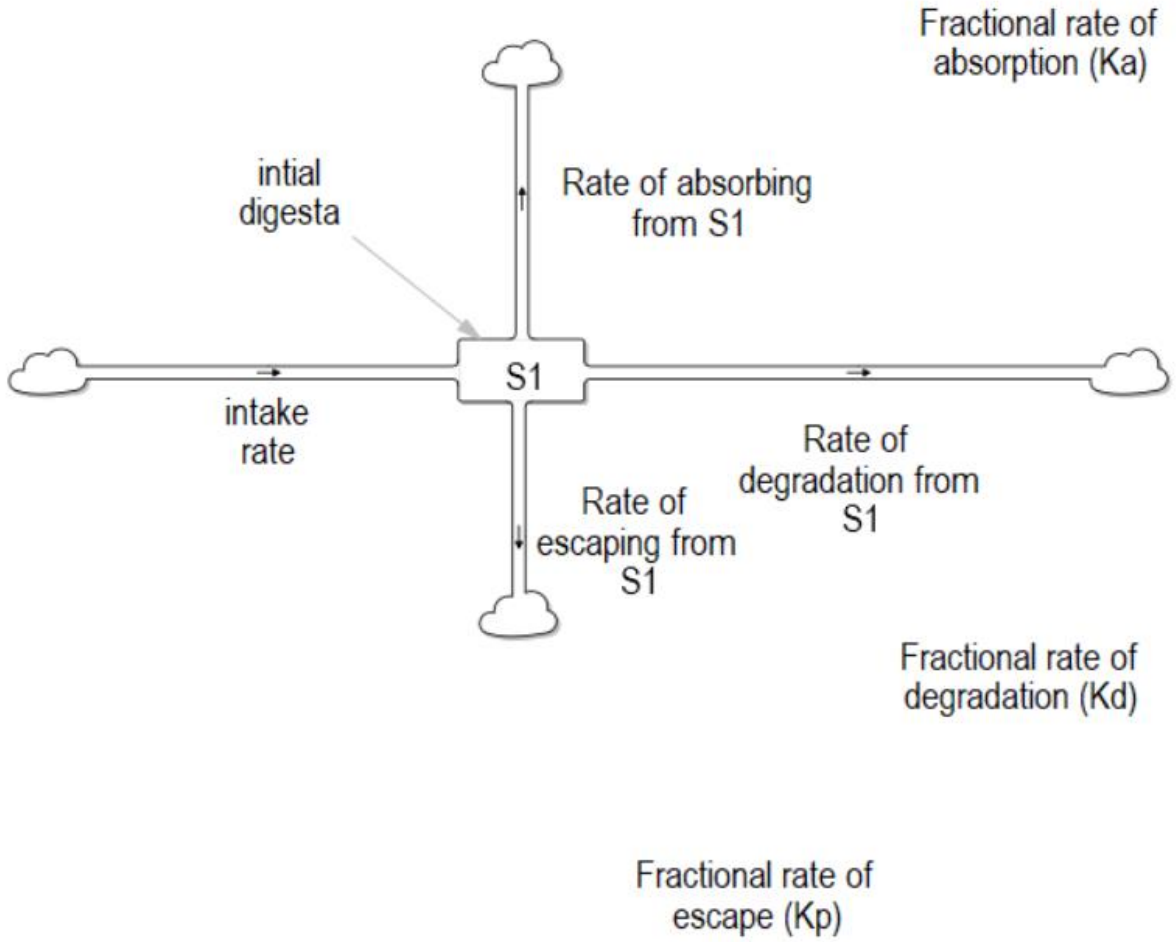


1. Intake rate
2. Rate of absorbing from S1
3. Rate of degradation from S1
4. Rate of escaping from S1



STEP 3c: Add auxiliary variables and parameterize





1. Fractional rate of absorption (Ka)

= 0.02, units = 1/hr

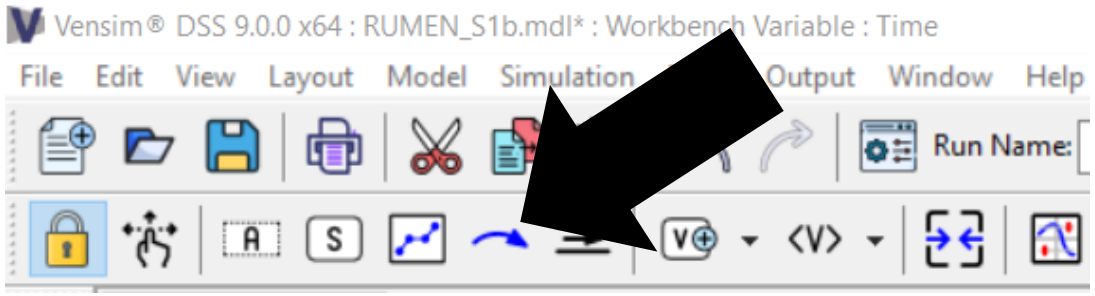
2. **Fractional rate of degradation (Kd)**

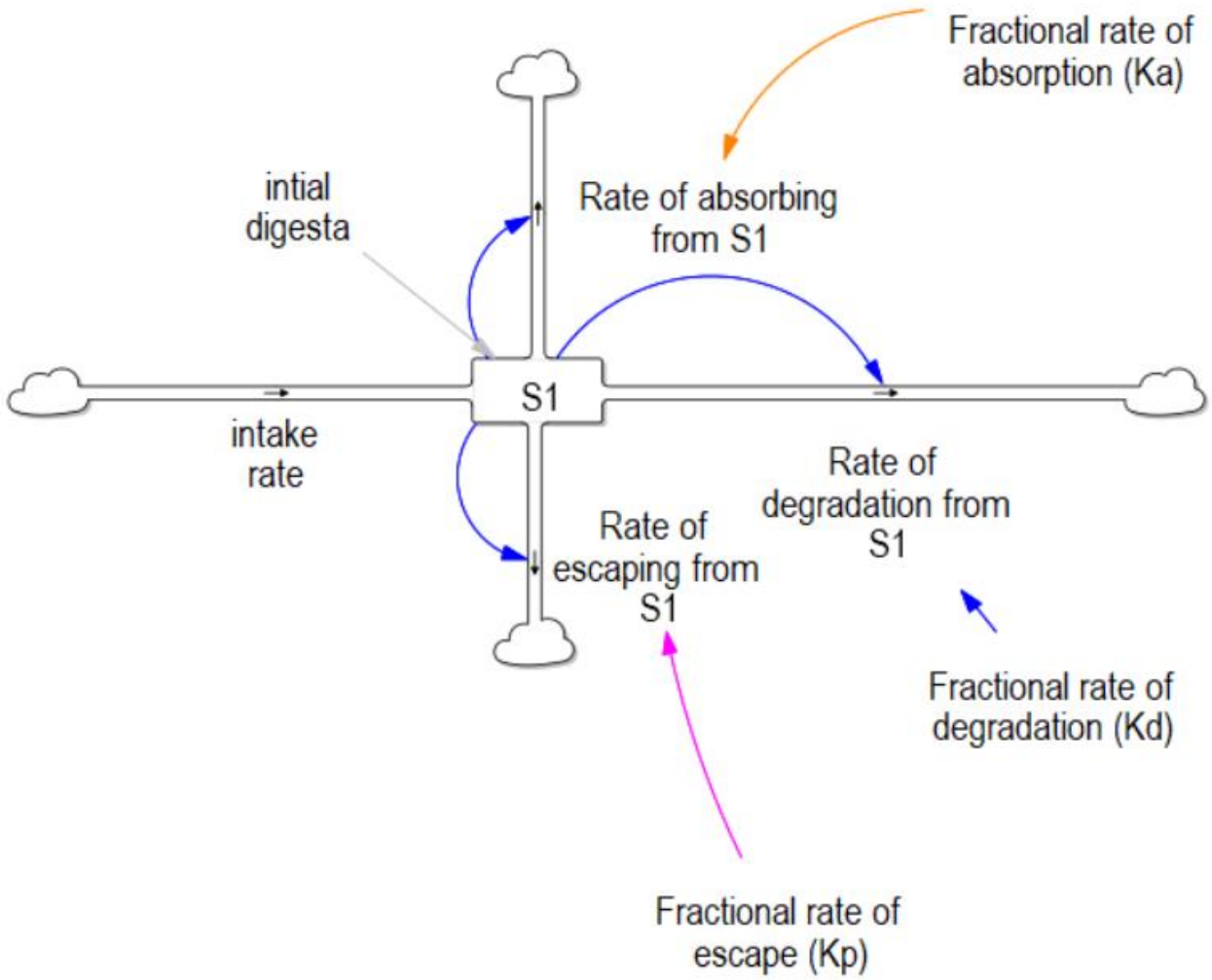
= 0.25, units = 1/hr

3. Fractional rate of escape

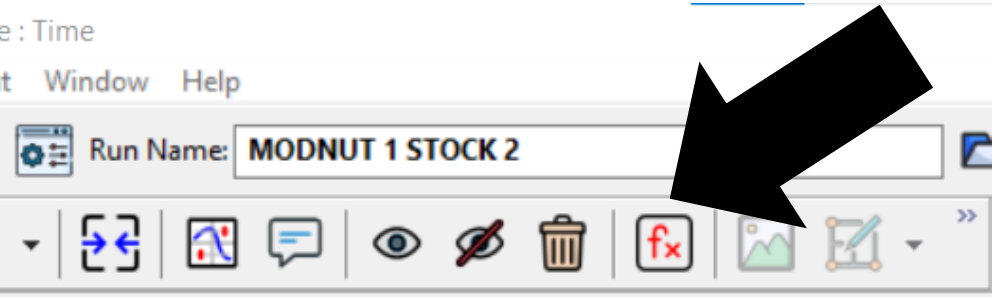
=0.05, units = 1/hr

STEP 3b: Add linkages (blue arrows) from “S1” to each rate, except “Intake rate”

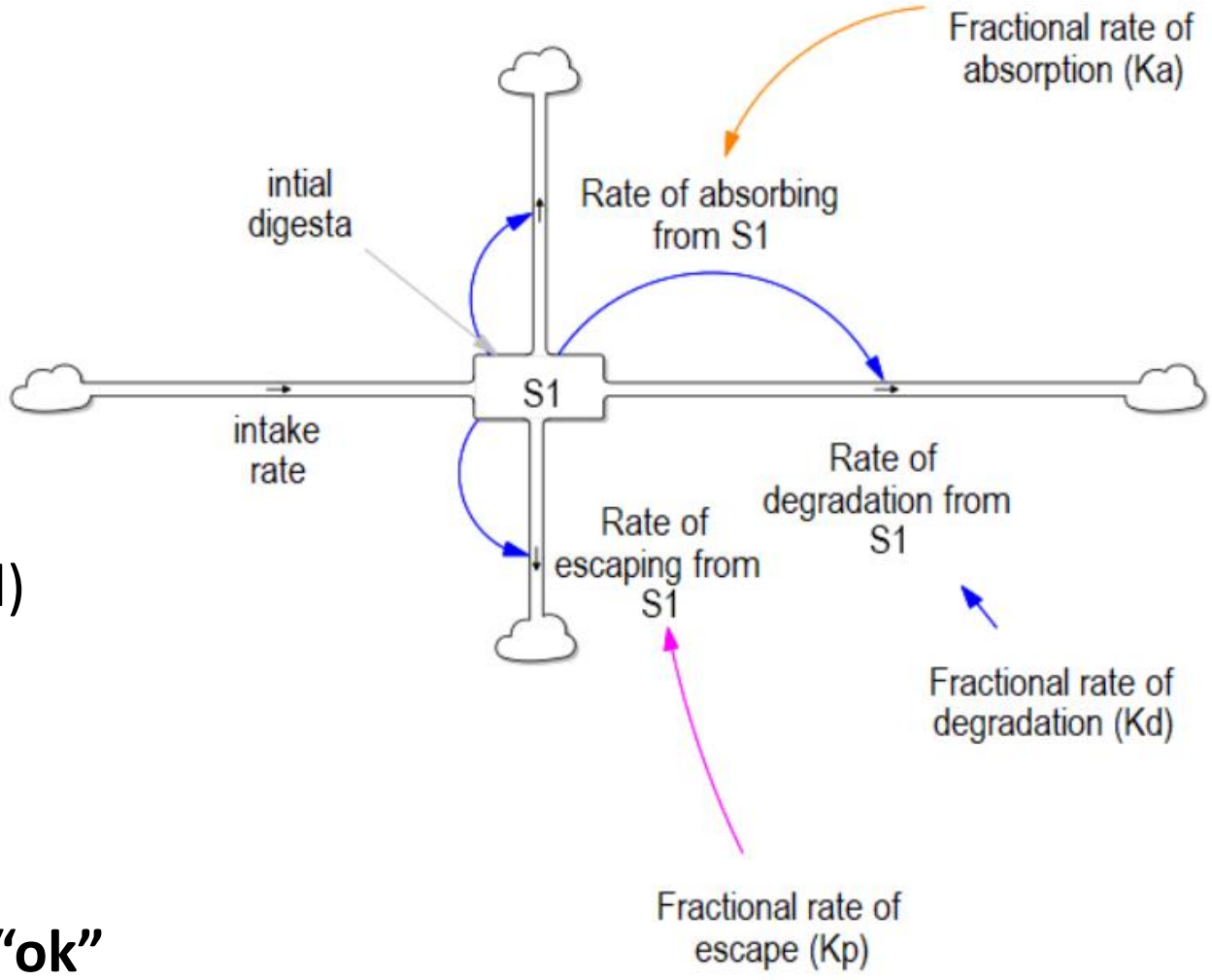




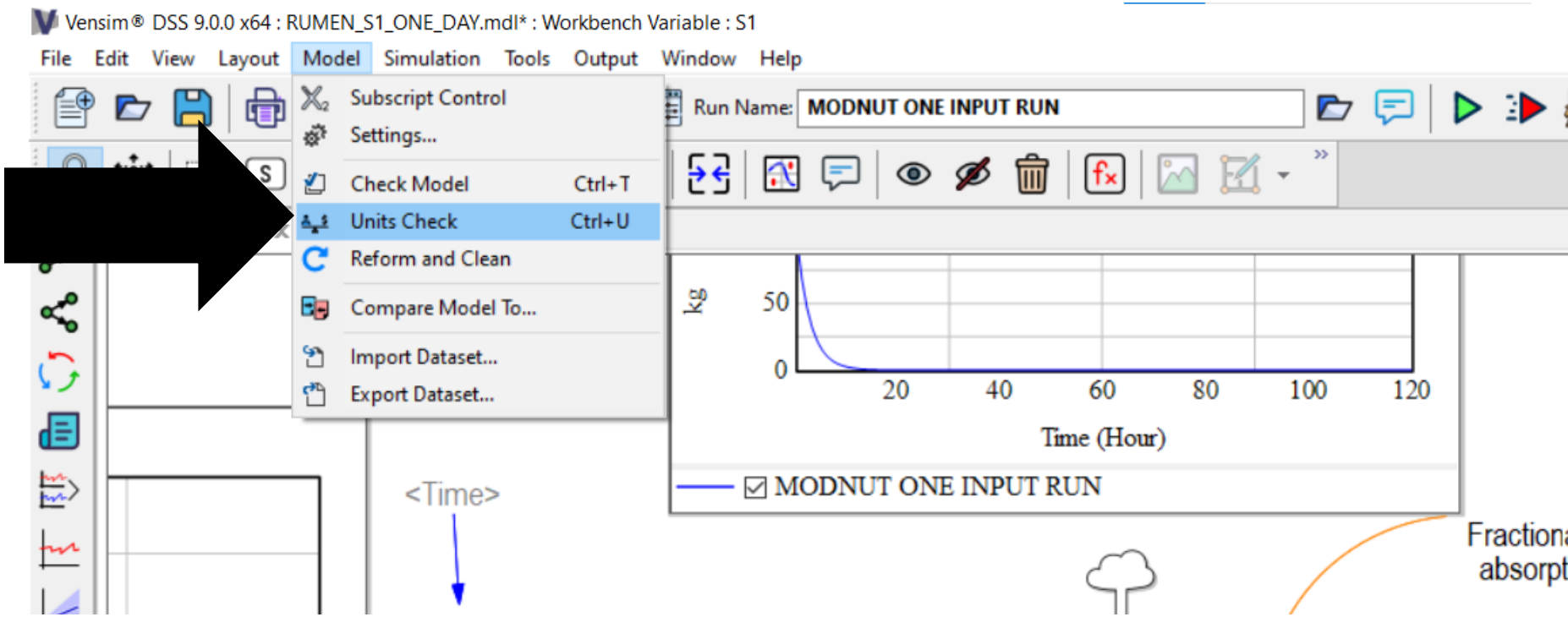
Step 3c: Parameterize rate variables



1. **Rate of absorbing from S1**
   = S1 * Fractional rate of absorption (Ka)
2. **Rate of degradation from S1**
   = S1 * Fractional rate of degradation (Kd)
3. **Rate of escaping from S1**
   = S1 * Fractional rate of escape (Kp)
4. **Intake rate**
   = 0
5. **Click on "S1" and press check units and "ok"**

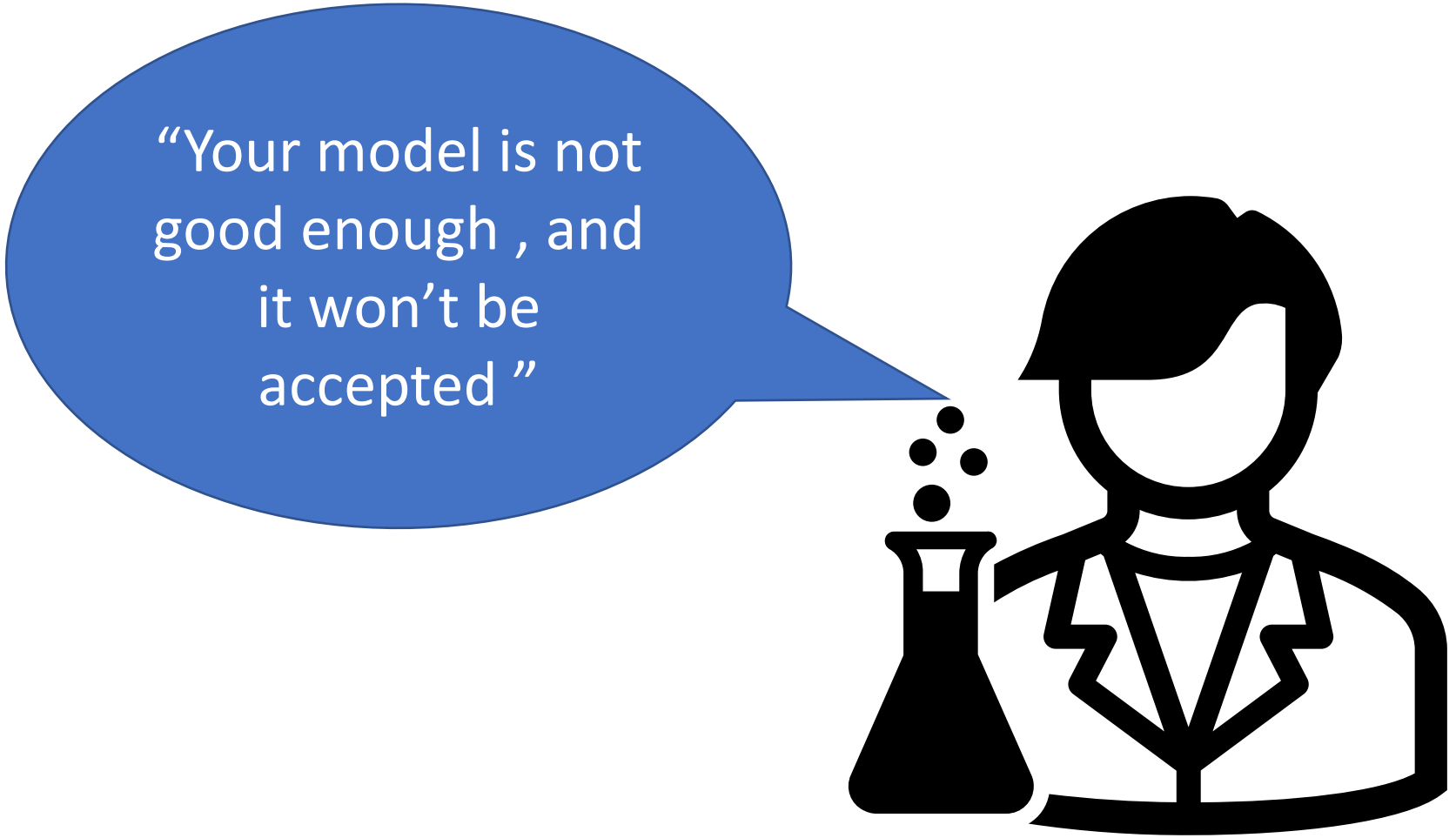


Step 4. Check Units



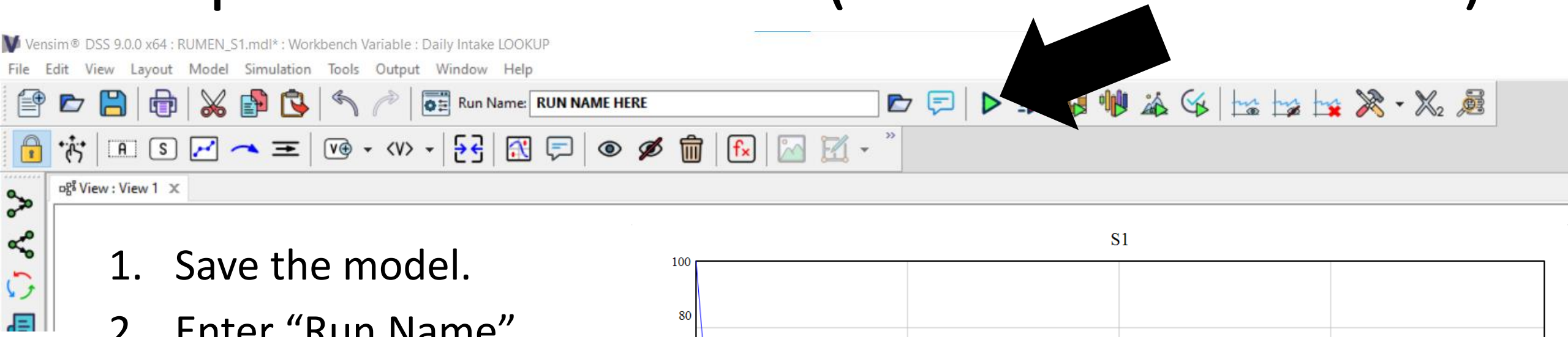
- THIS IS NOT AN OPTION!

If you do not check units, you will hear this from reviewers:

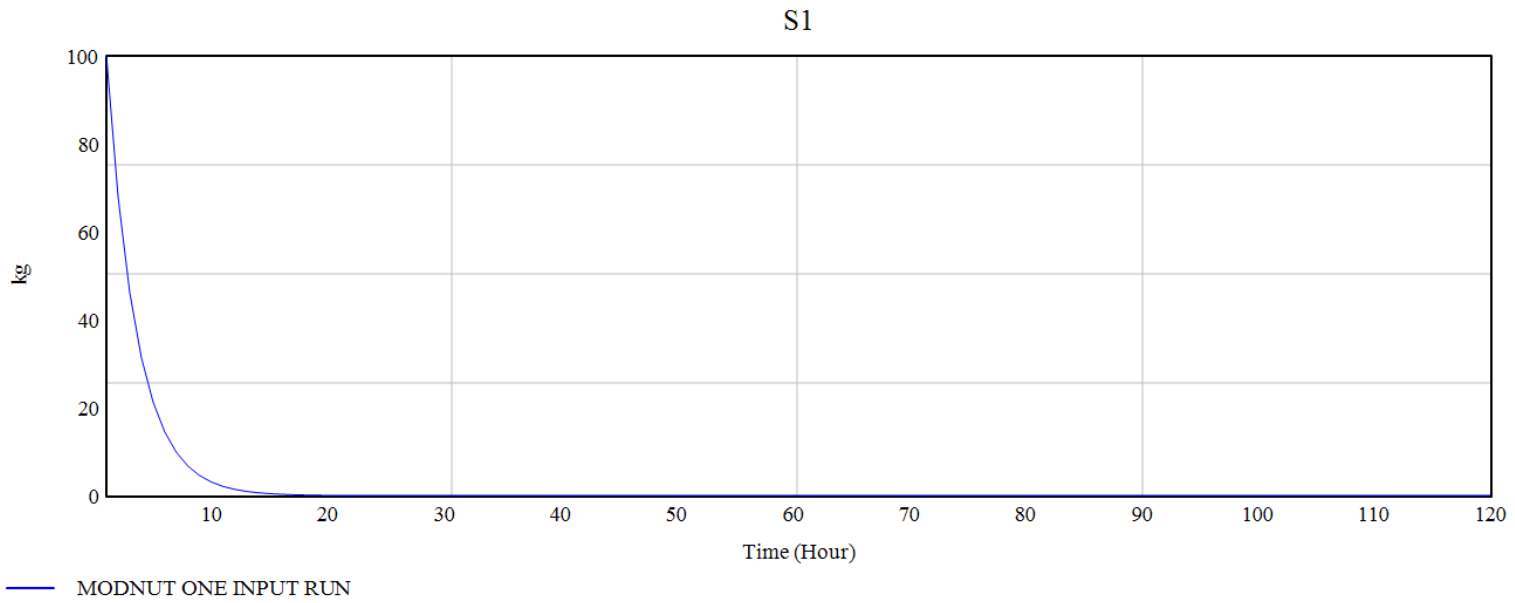


RUN THE MODEL

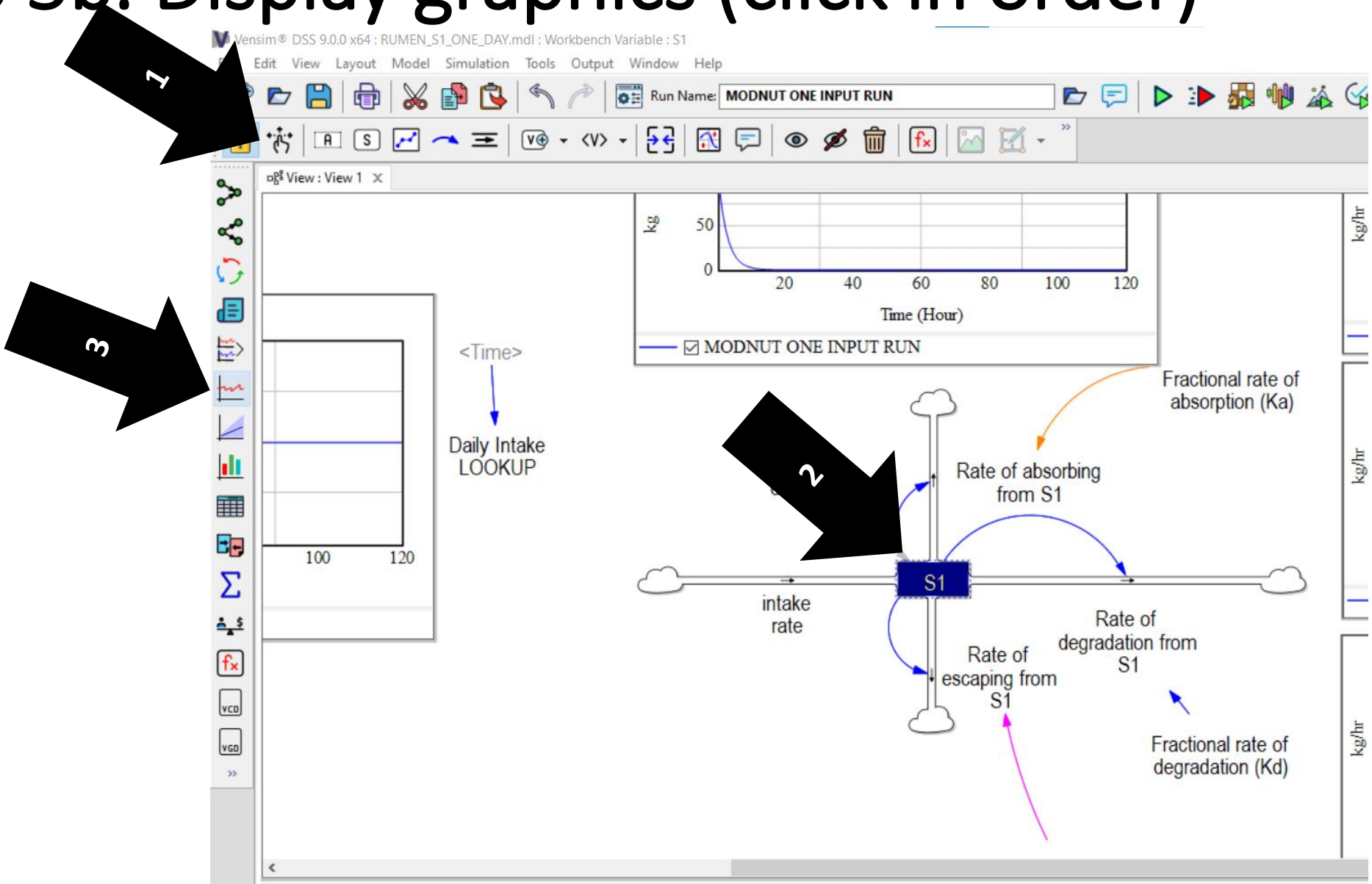
Step 5a: Run the Model! (See Model 1 in Folder)



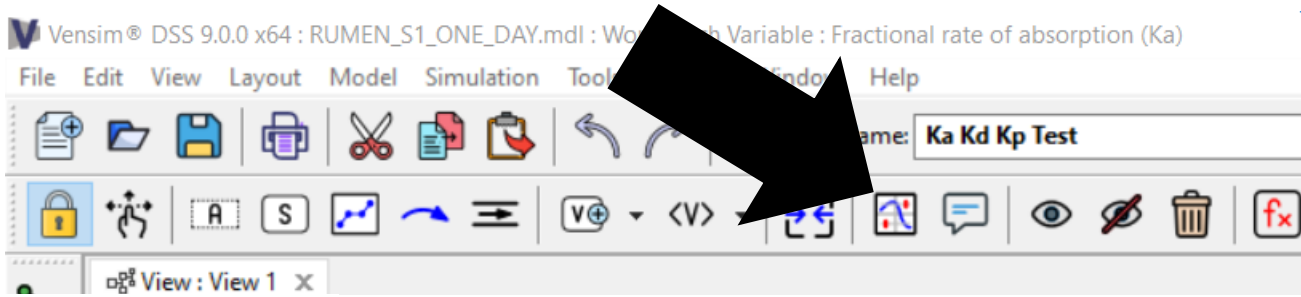
1. Save the model.
2. Enter “Run Name”
3. Press the green arrow
4. Let’s evaluate

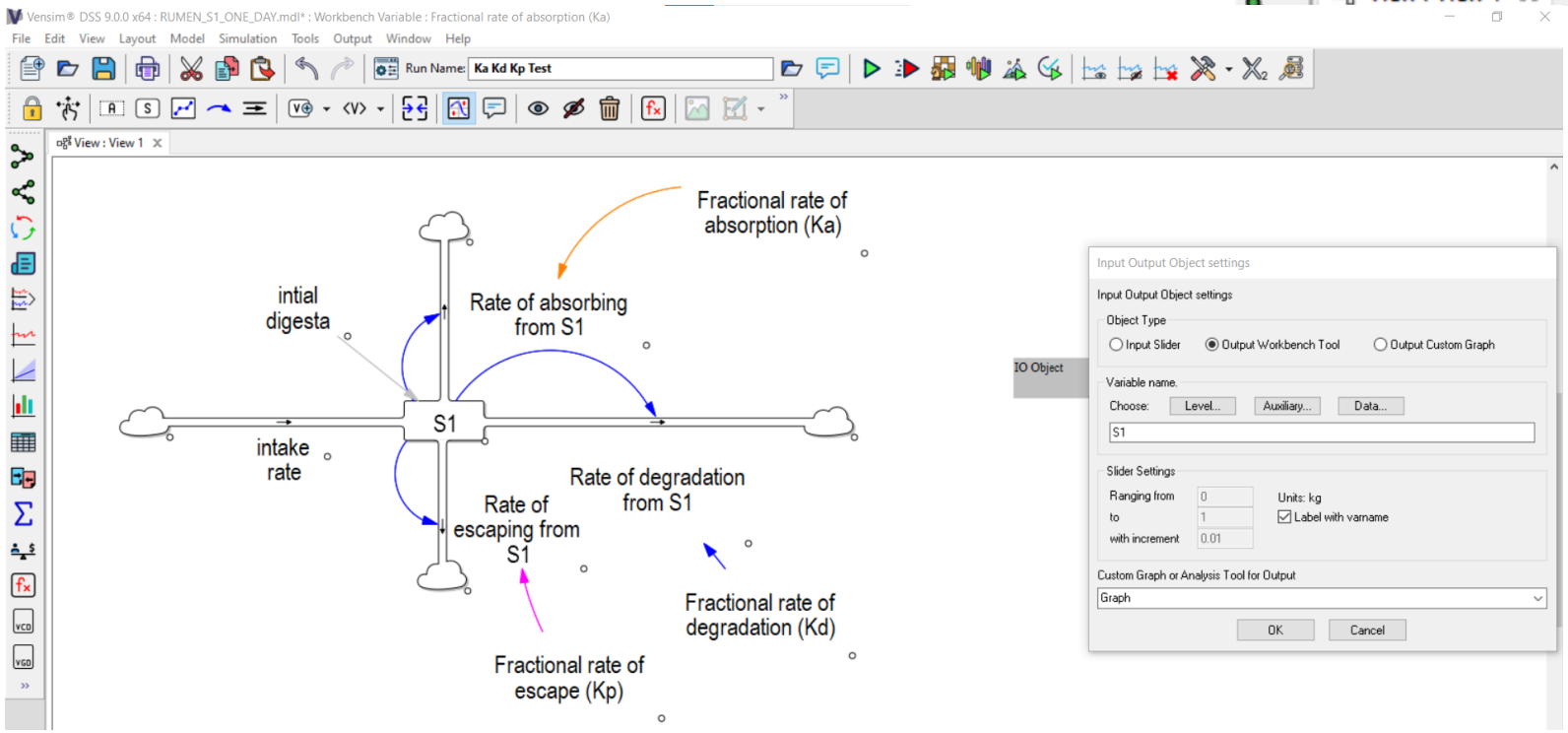


Step 5b. Display graphics (click in order)



Step5c. Make a graph



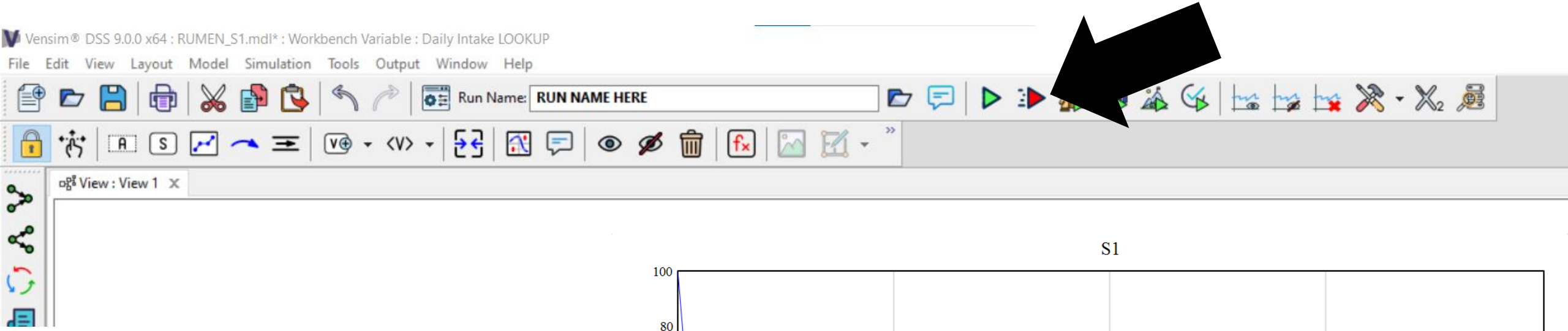


1. Click graph tool

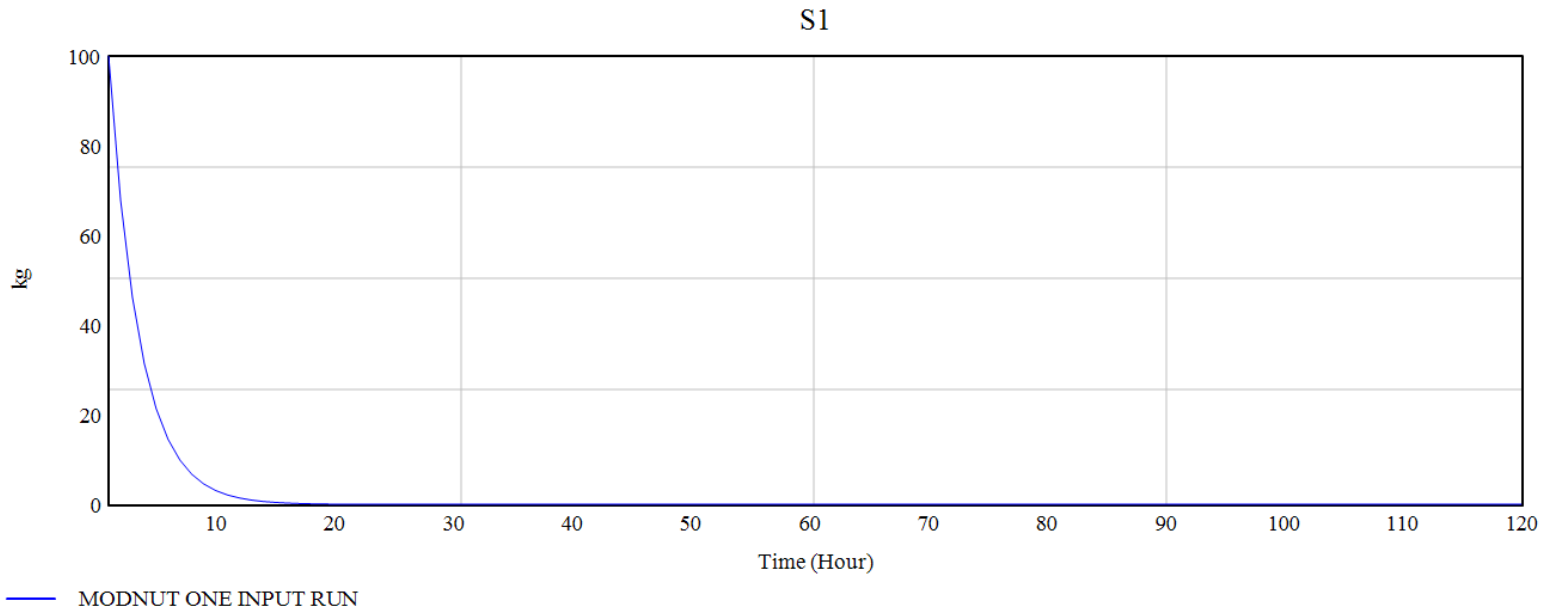
2. Click location on screen

3. Select:
   a) “**Output Workbench Tool**”
   b) Level, “**S1**”
   c) Pull down arrow “**Graph**”

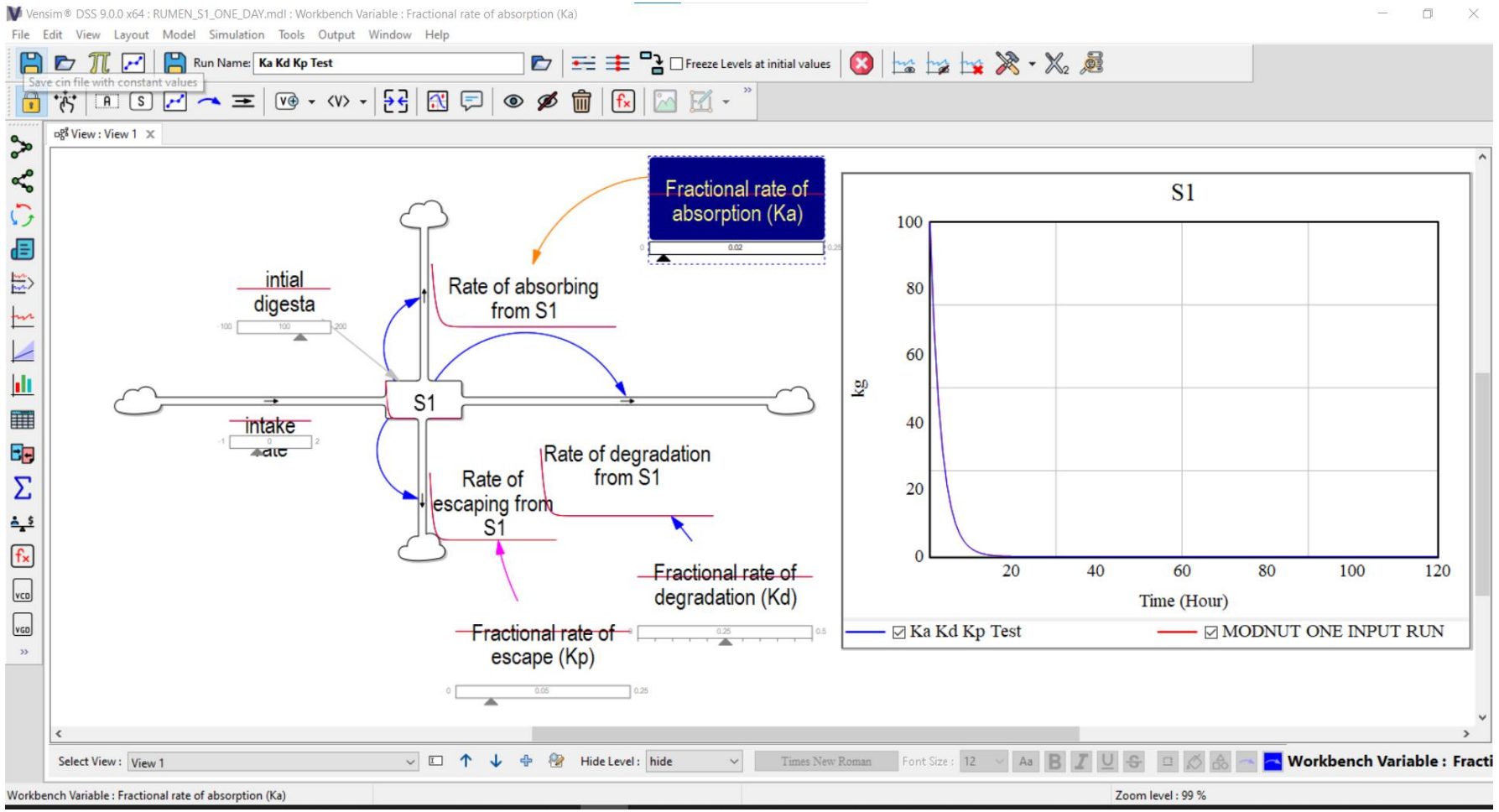
Step 5d: Evaluate Ka, Kd, Kp



1. Put a new “Run Name”
2. Click the red button.



Step 5e. Move the sliders

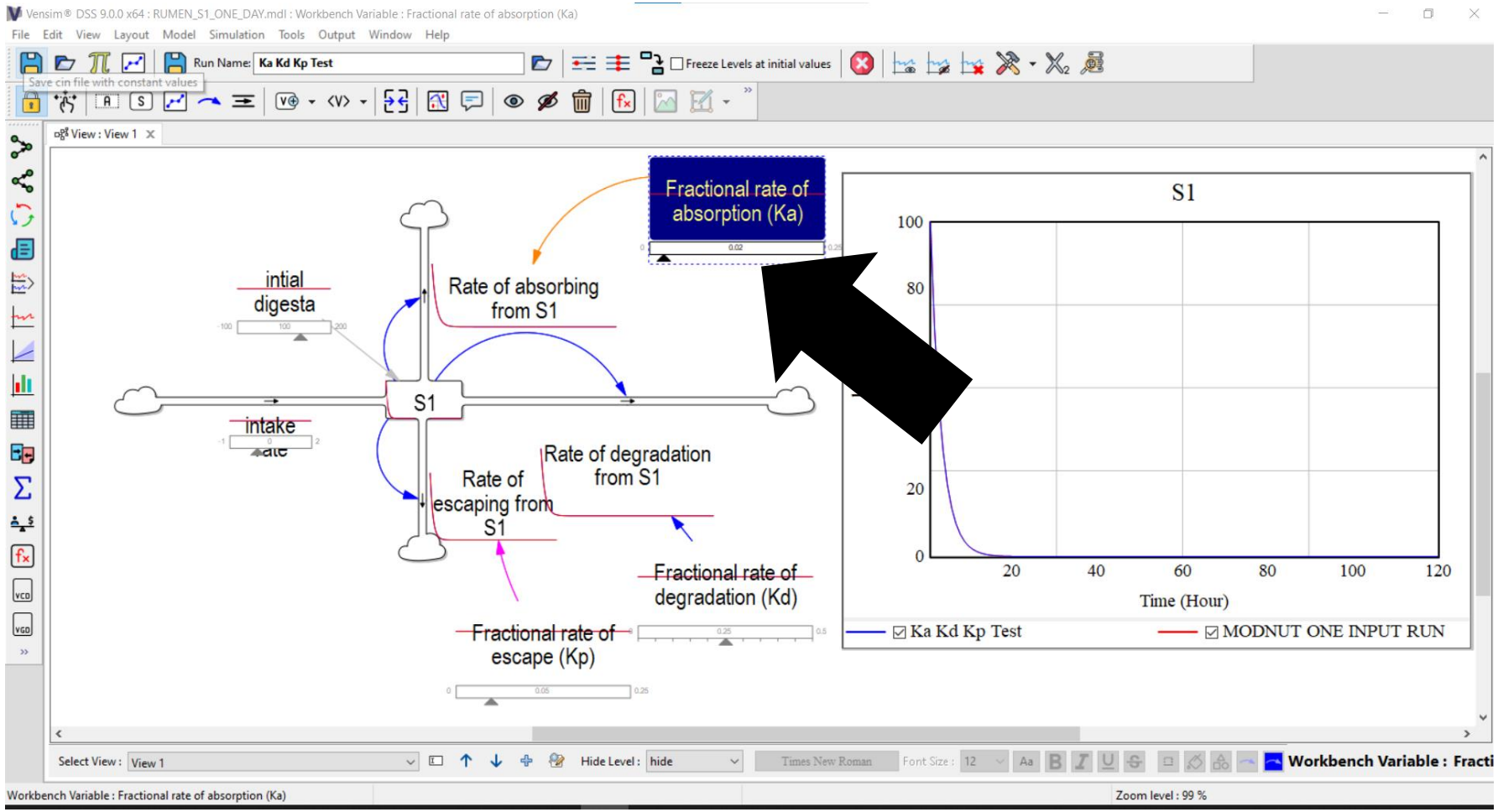


For **Ka**, **Kd**, and **Kp** move the “black arrow” under the variable name.

What happens?

You broke your model! Good!

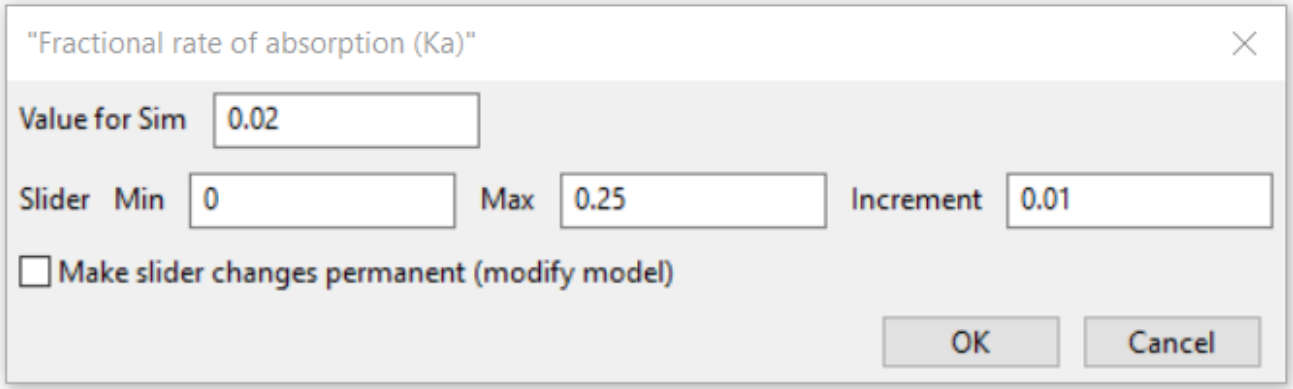
Step 5e. Move the sliders



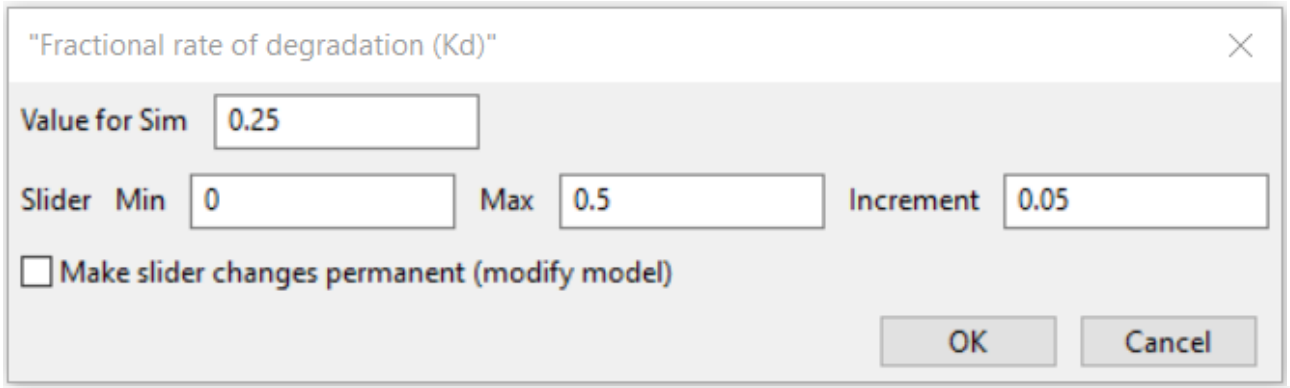
Double-click the "**white bar**" under **Ka, Kd, and Kp.**

Set the parameter bounds and select "**make slider change permanent**"

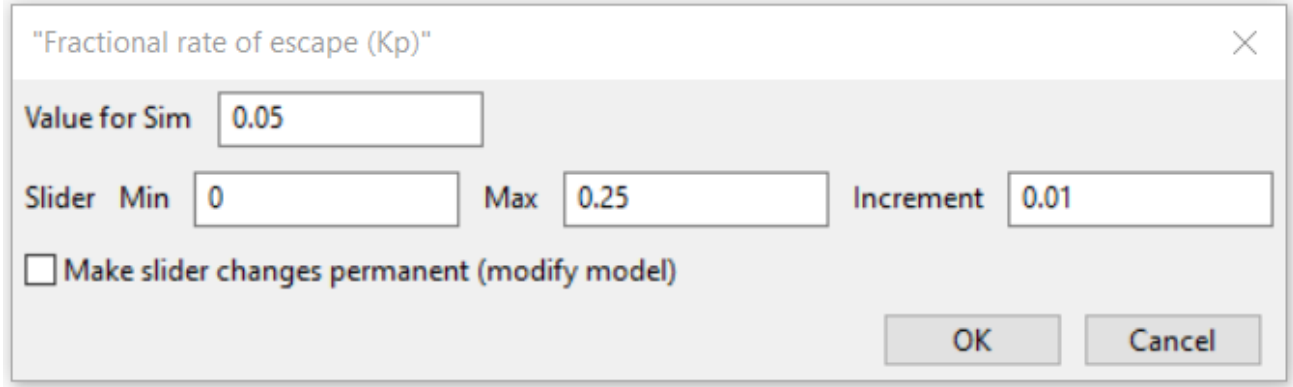
Step 5f. Set the input sliders



Ka



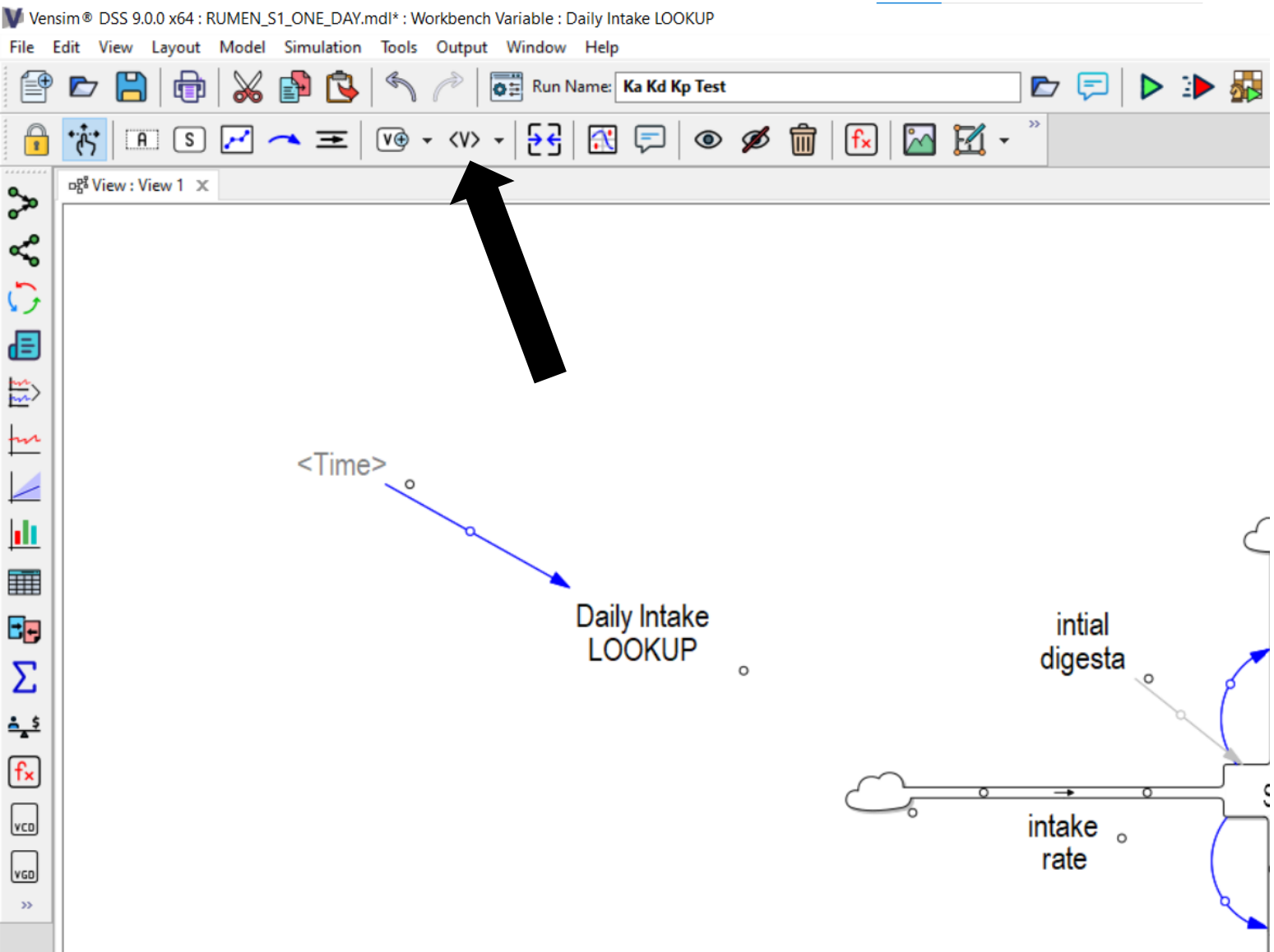
Kd



Kp

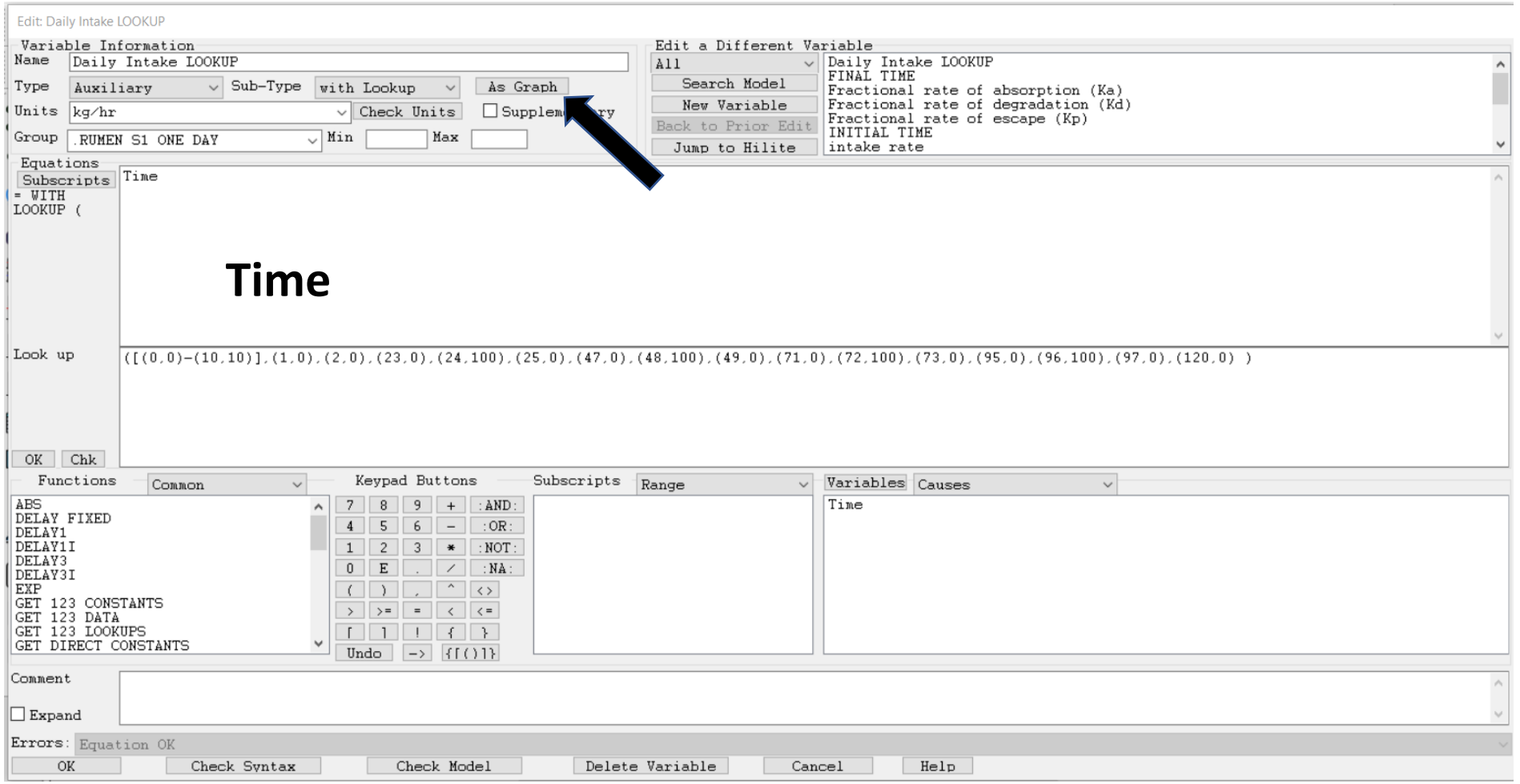
The sum of Ka, Kd, Kp fractions cannot be greater than 1!

Step 6a. Add “LOOKUP” or Table Function



1. Click the “**shadow variable**” button
2. Select “**Time**”
3. Make another auxiliary variable called “**Daily Intake LOOKUP**”
4. Link “**Time**” to “**Daily Intake LOOKUP**”

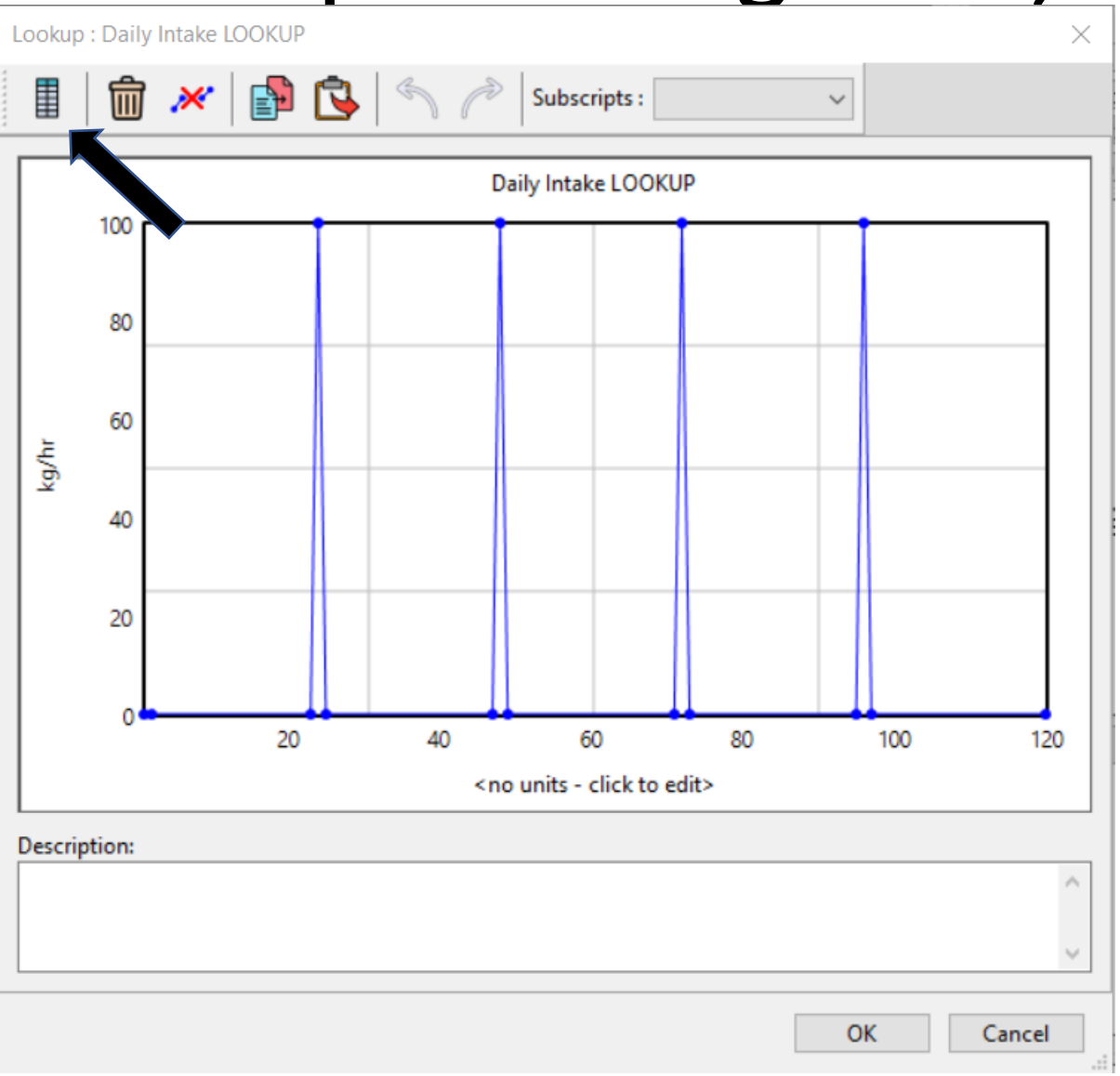
Step 6b: Parameterize the LOOKUP

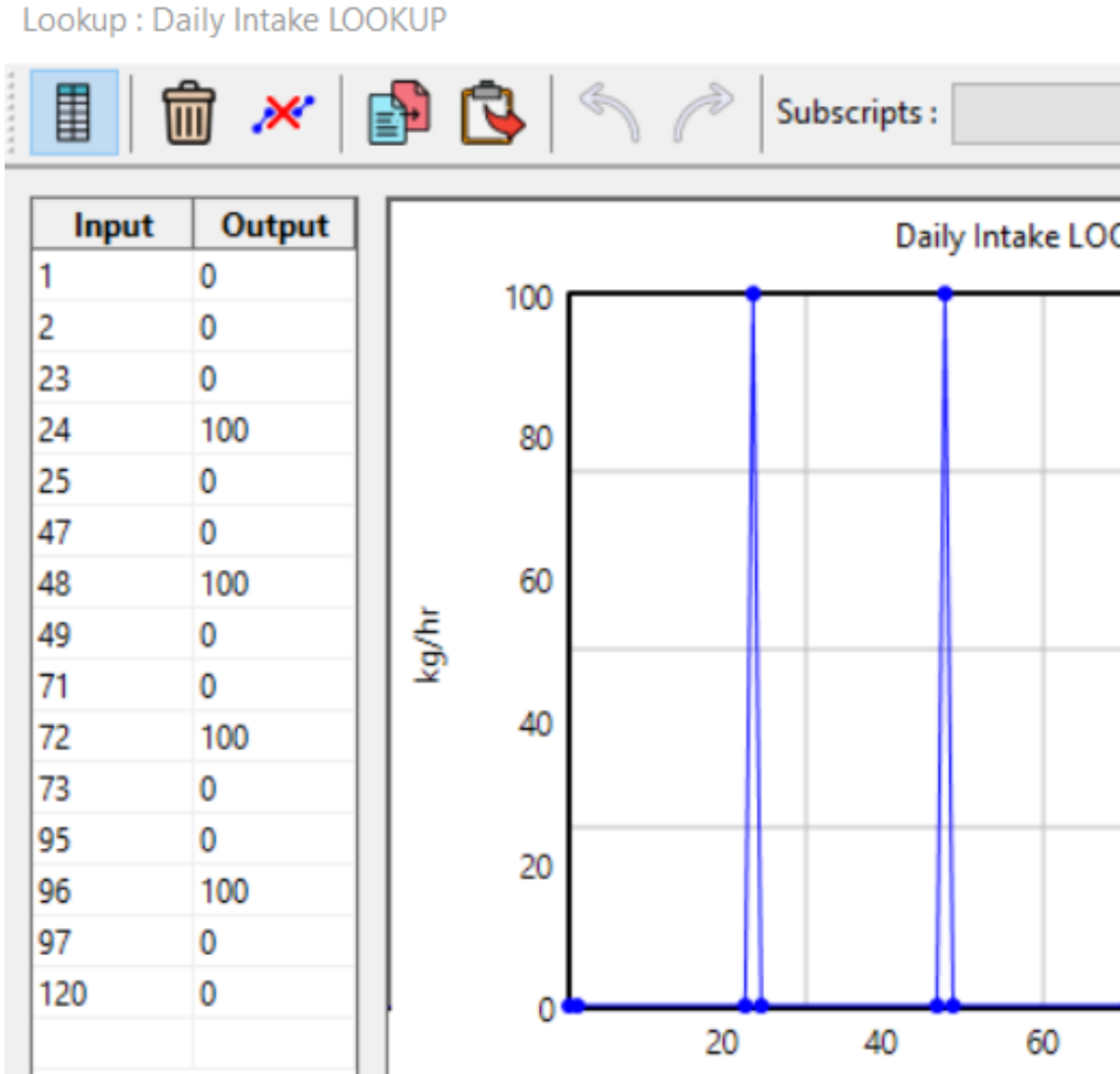


Units = kg/hr

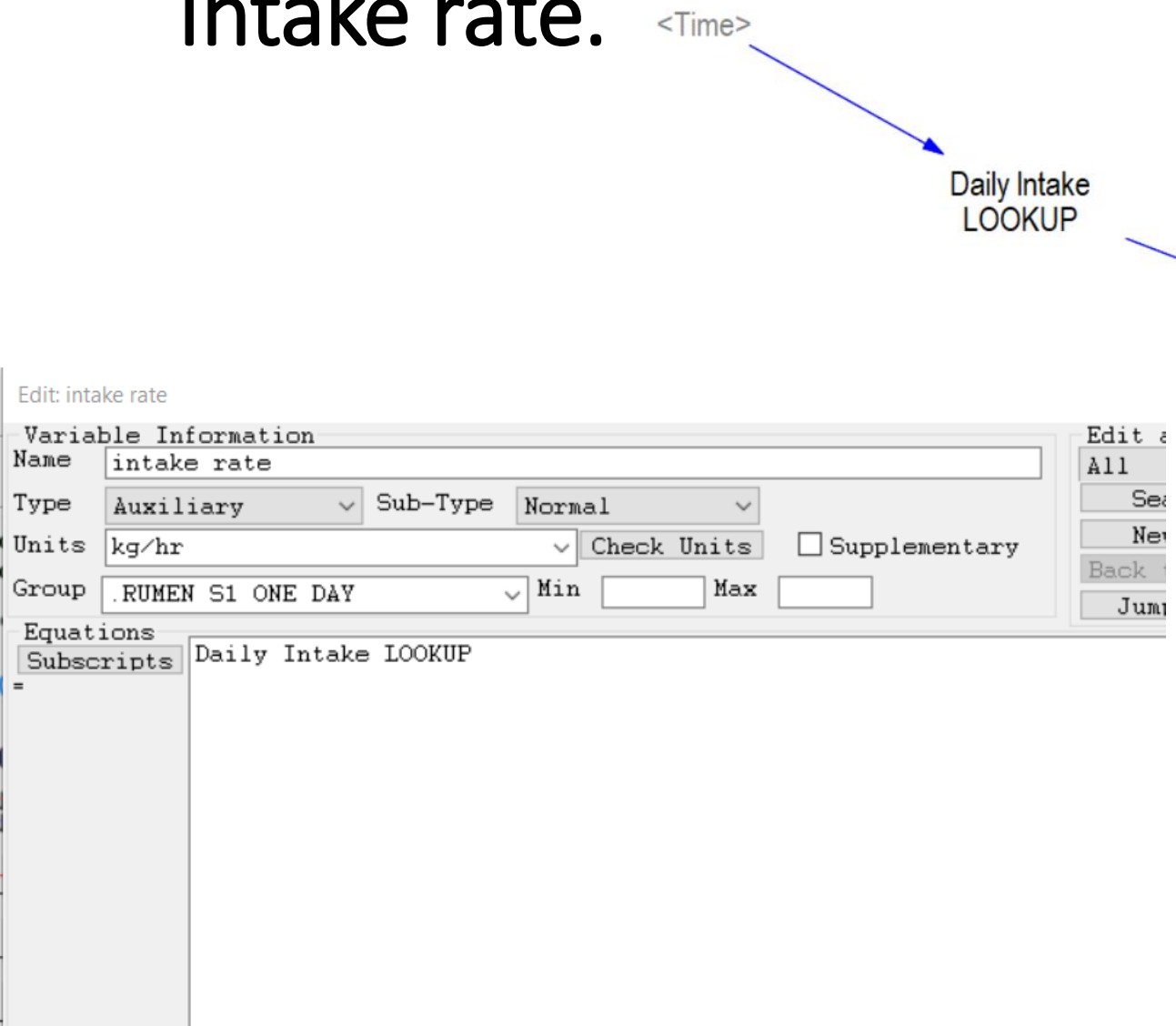
1. Select “**As Graph**”

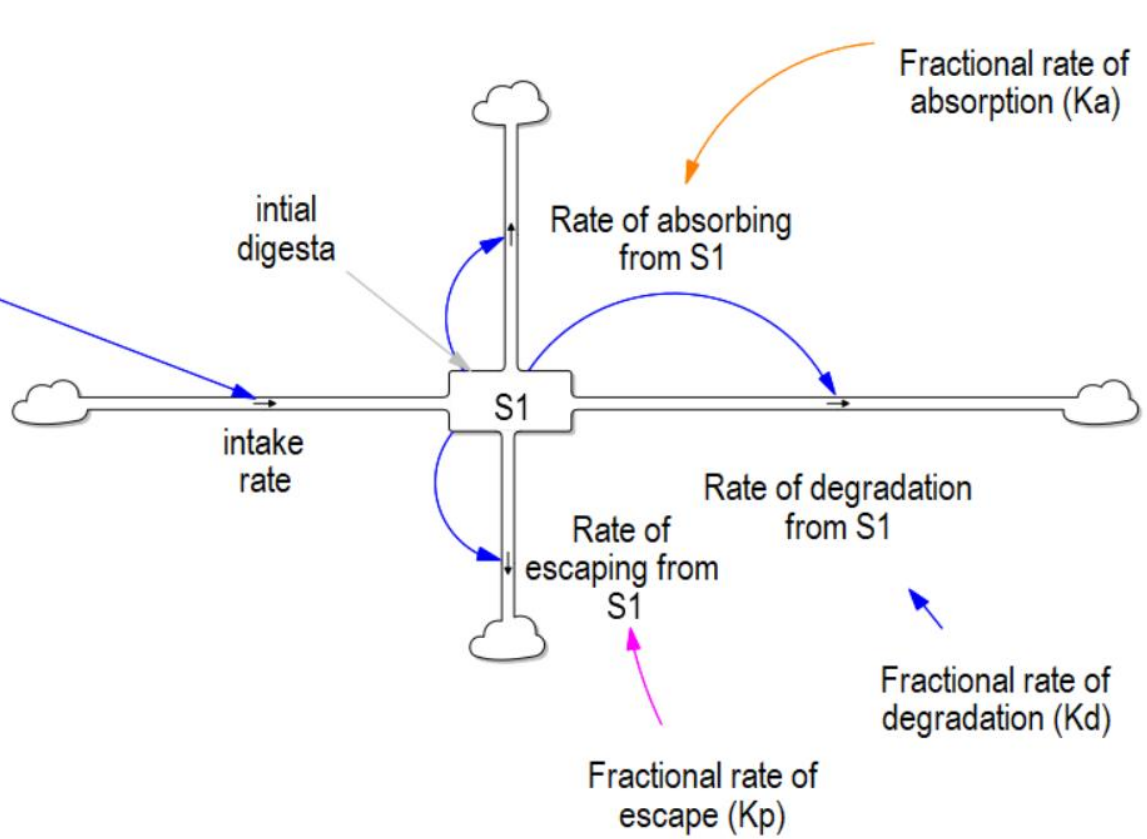
Step 6c. Using the X,Y input table





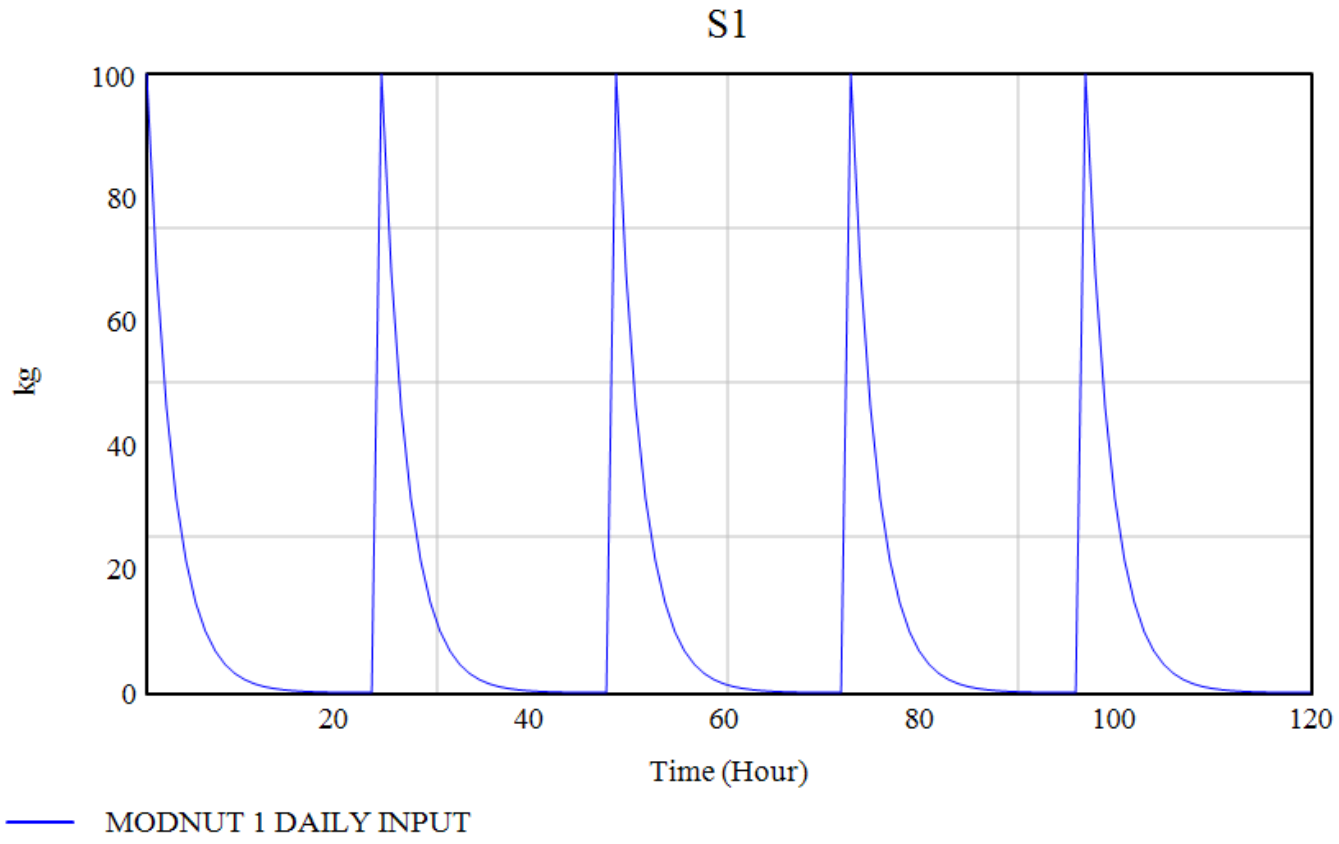
Step 6d. Link to “Intake rate” and use the equation editor to set “Daily Intake LOOKUP” as the input for Intake rate.

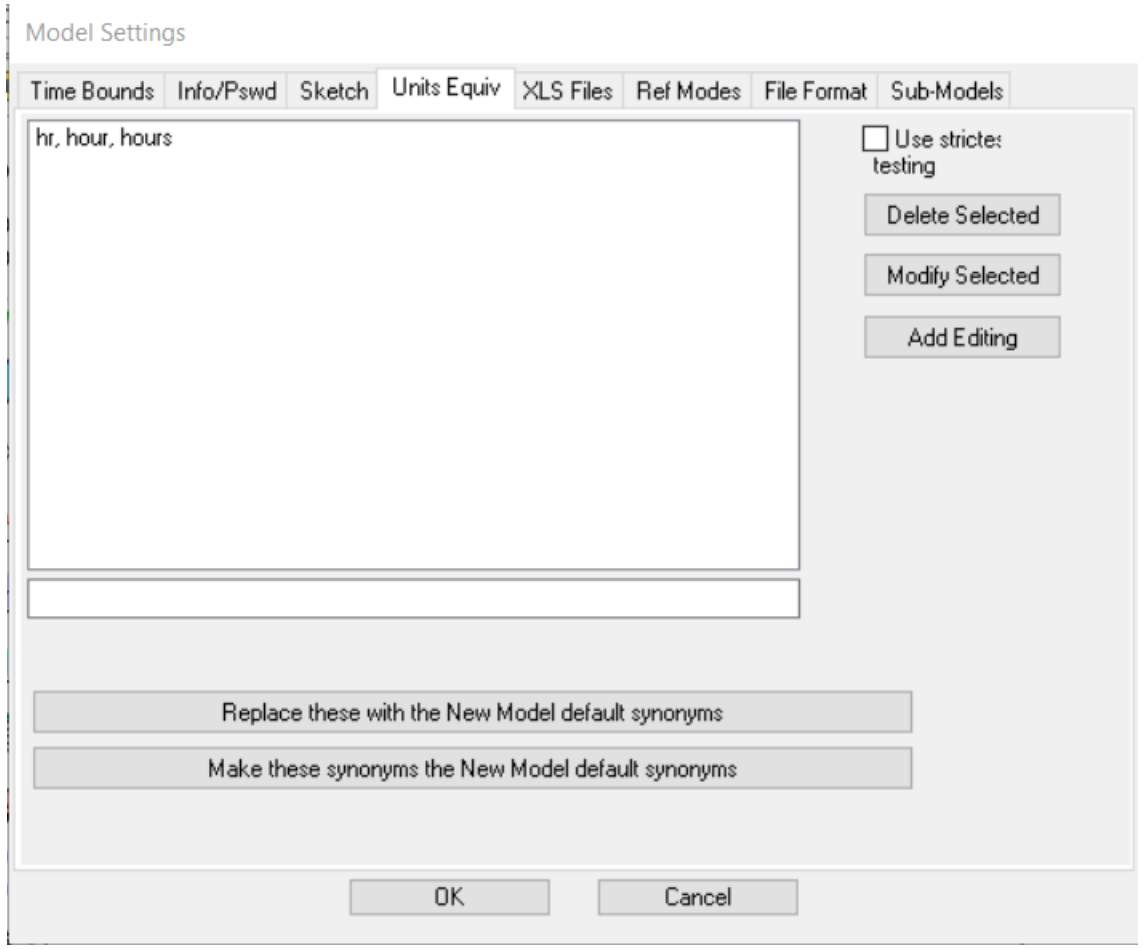




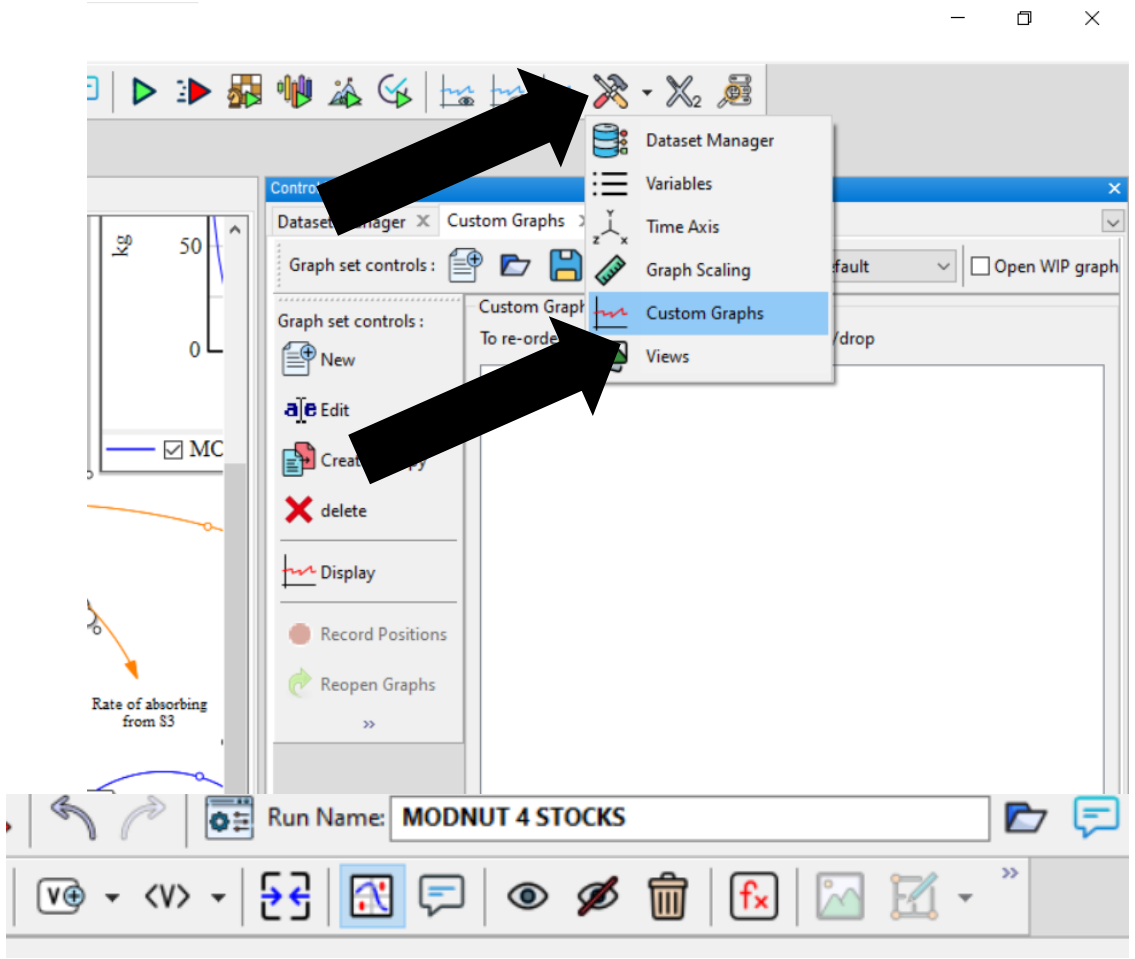
Step7. Open “Model 2” from files

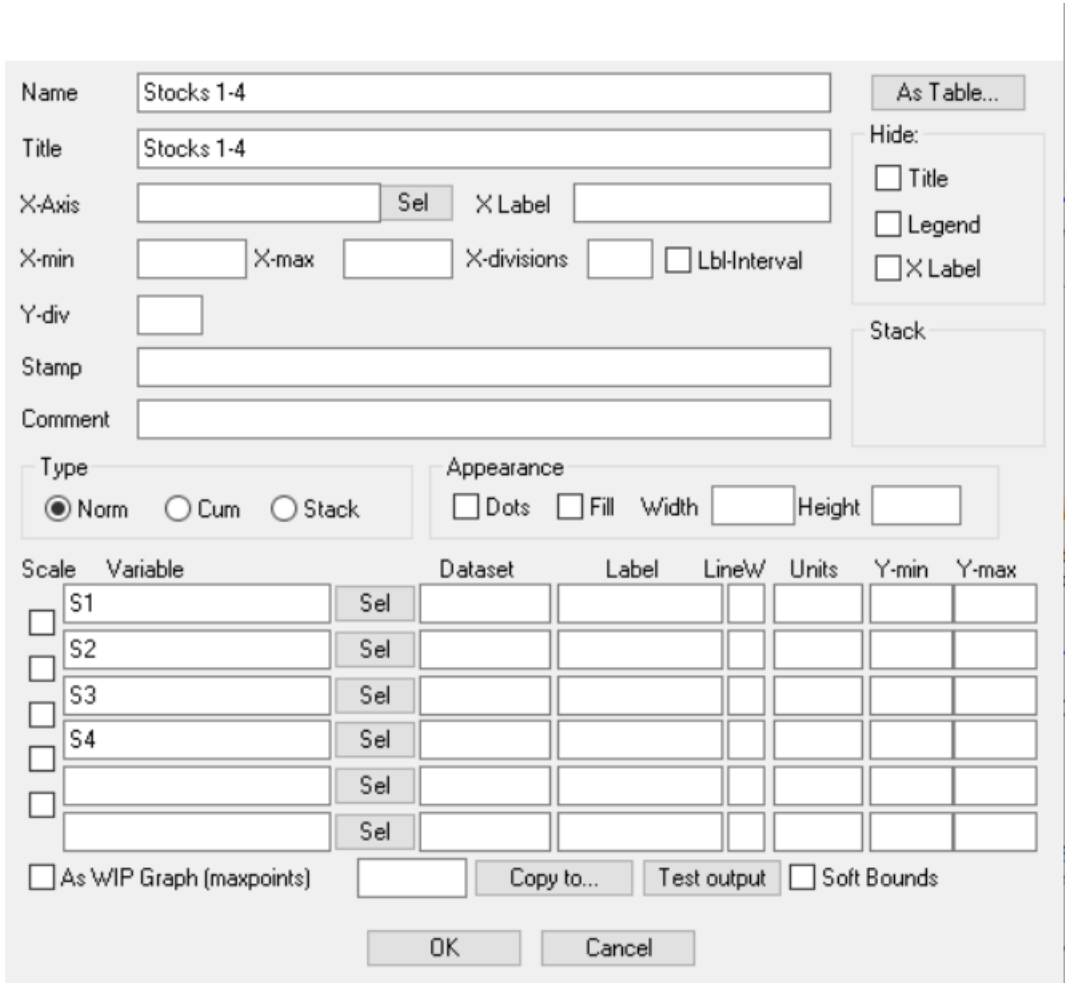
- Click the Green Run button.



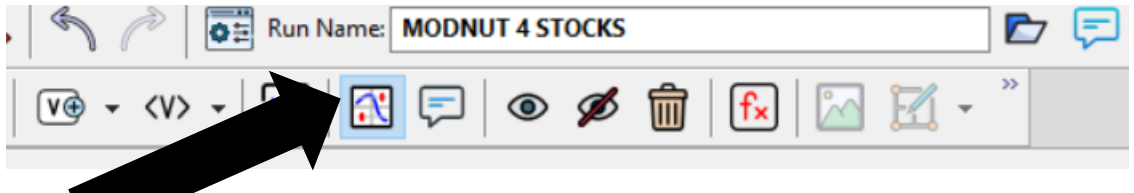


Custom Graphs

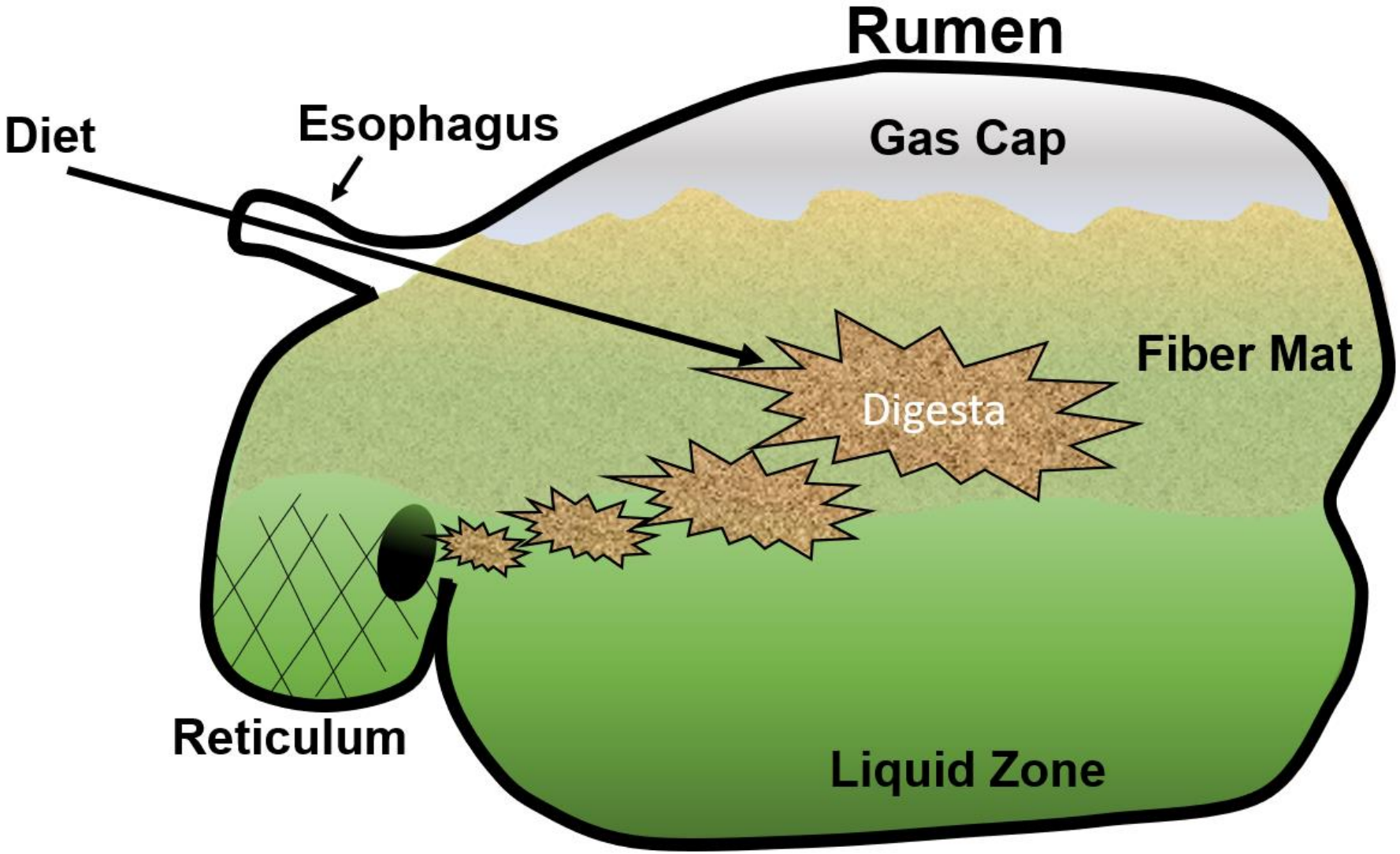




Then use the same process as creating a graph but select **“output custom graph”**

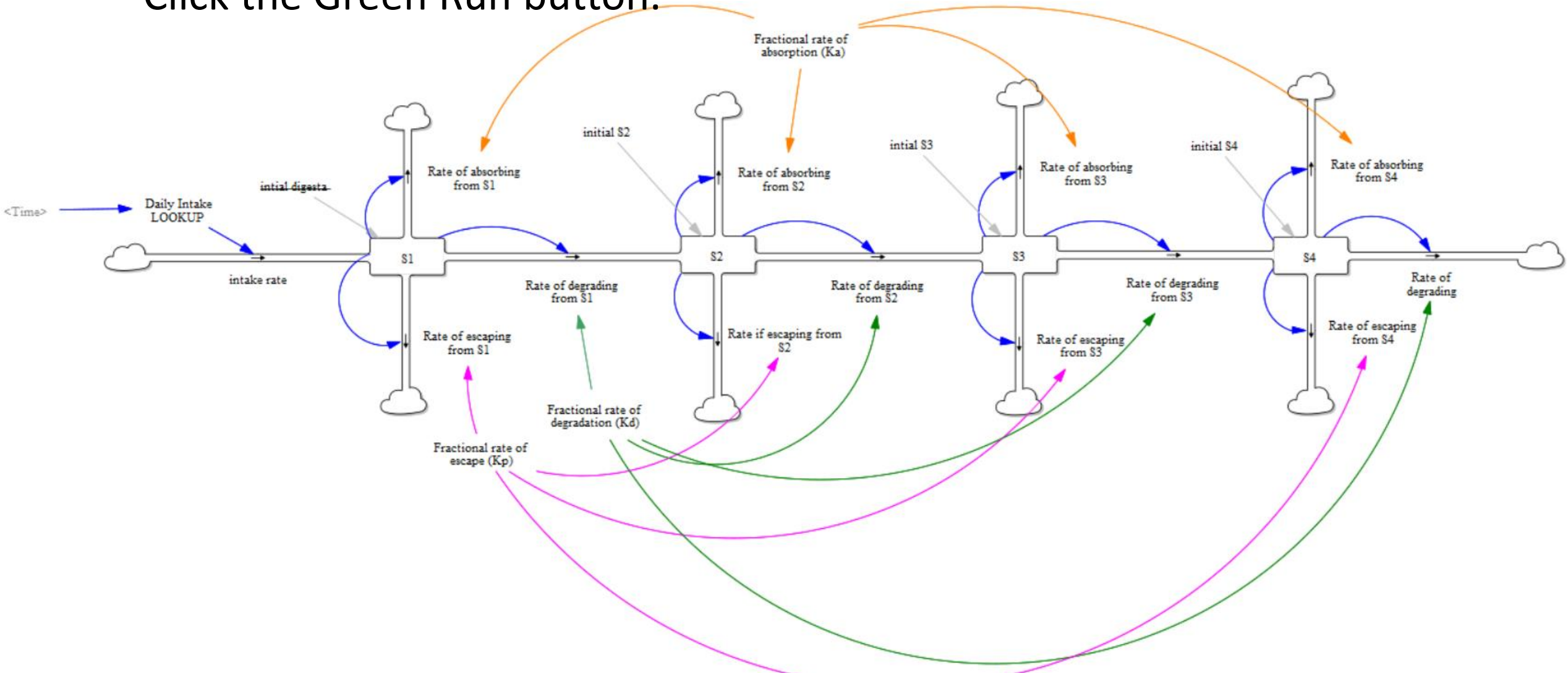


Making a four stock model to achieve an "**Aging Chain**"

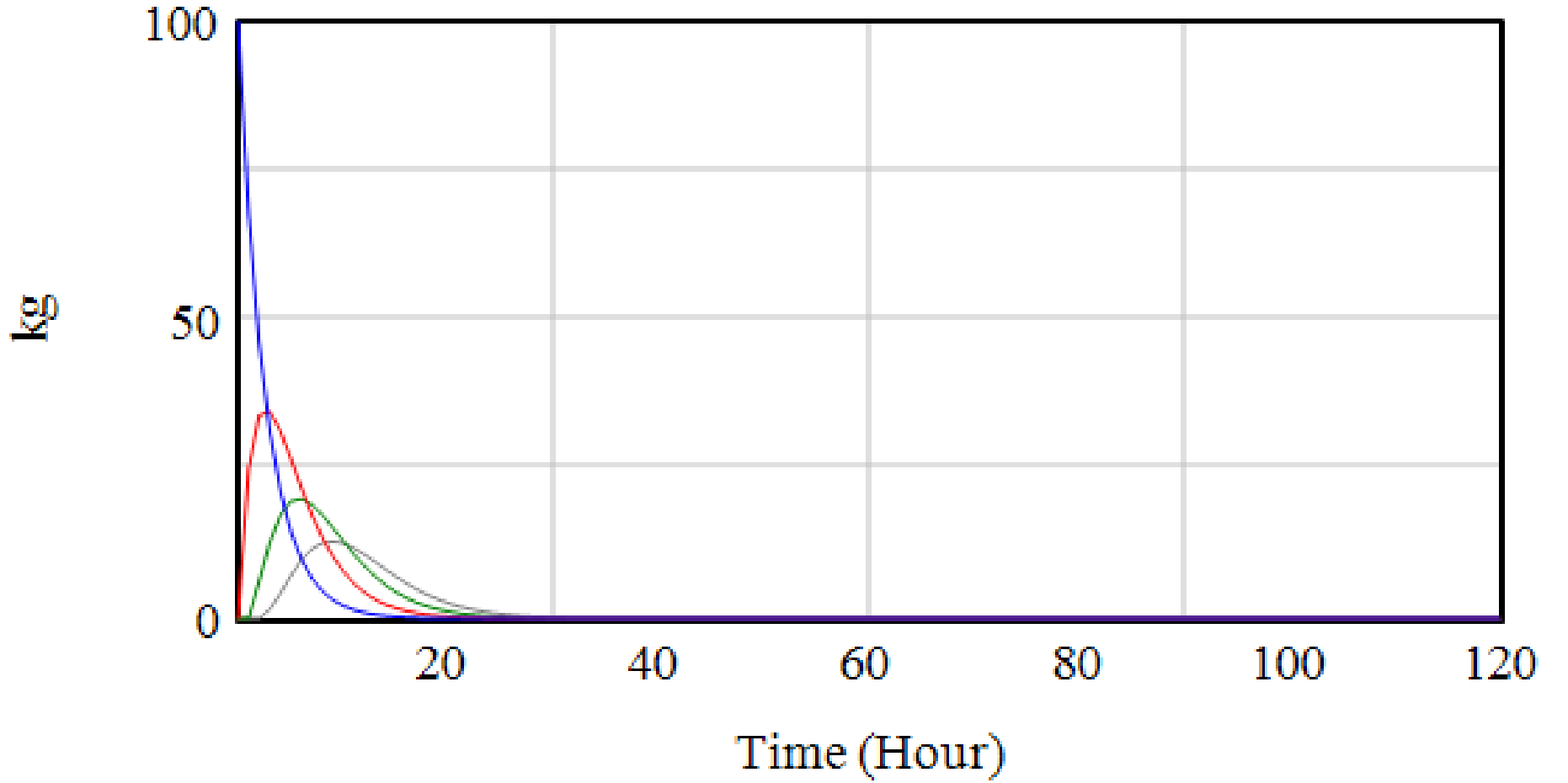


Step 8. Open “Model 3” from files

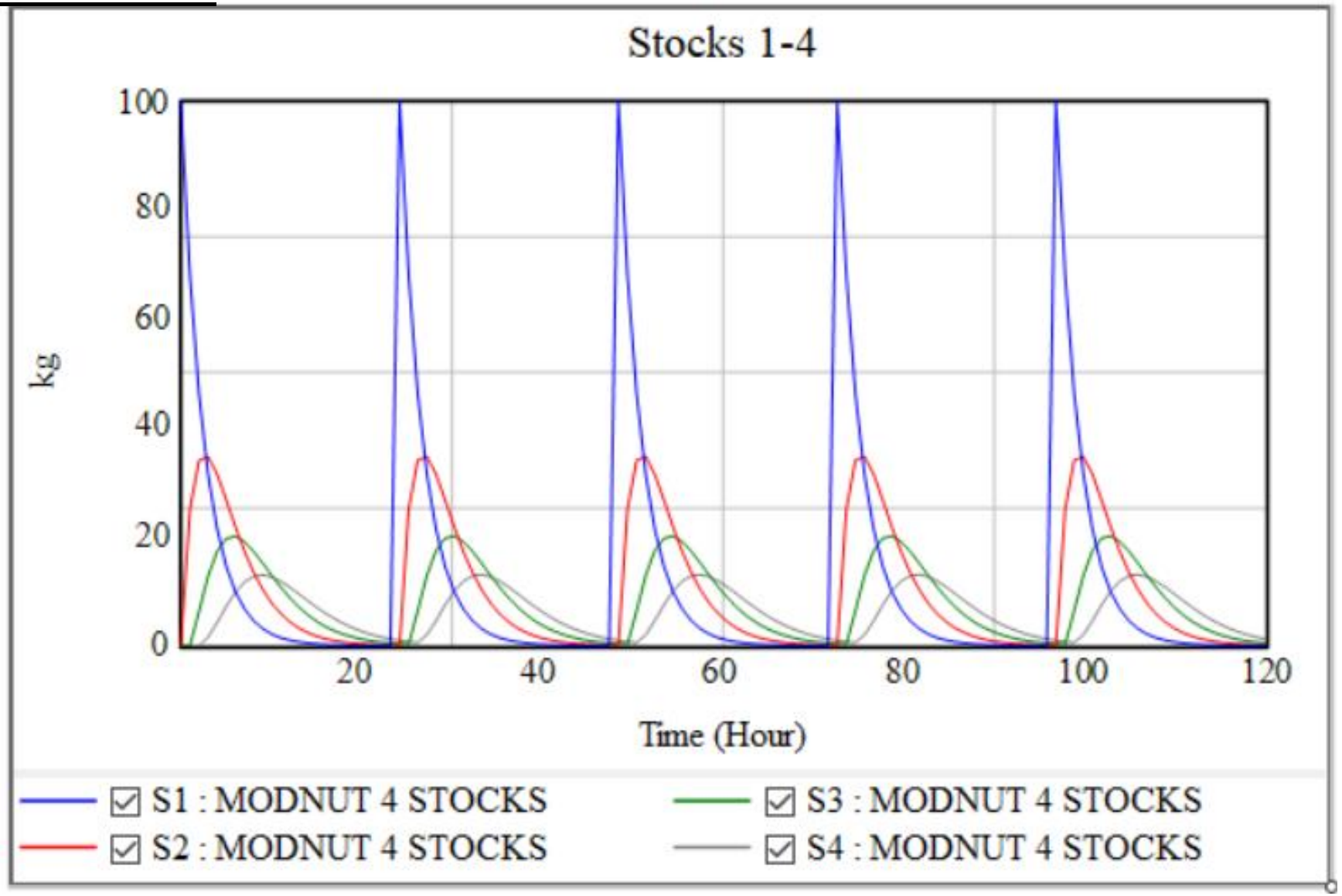
- Click the Green Run button.



With only the initial value of **100 kg** in S1



With only the "**initial digesta**" value of **zero** and the **Intake Rate** informed by the **daily intake LOOKUP** variable



How to transfer to Program R

Step 1. Open Program R Studio (See MODNUT.R file)

- Install packages (lines 1-2)

```
installed.packages("deSolve", "ggplot2", "gridExtra", "scales",
                   "readr", "data.table")
```

- Load packages

```
library(deSolve)
library(ggplot2)
require(gridExtra)
library(scales)
library(readr)
library(data.table)
```

Freely available code :
https://github.com/JimDuggan

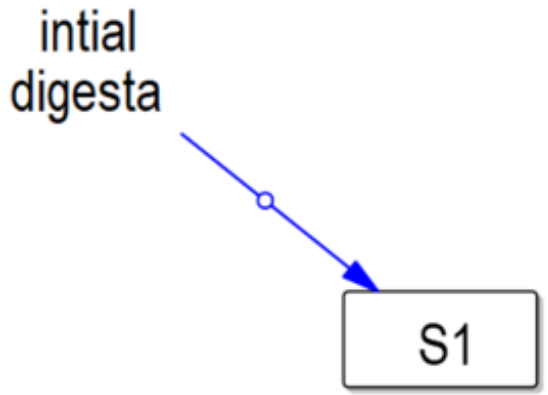
Goal of this Exercise

- **IS NOT** to become an R Wizard.

- **IT IS** to understand how to transfer a simple dynamic model to Program R. THE BASICS!

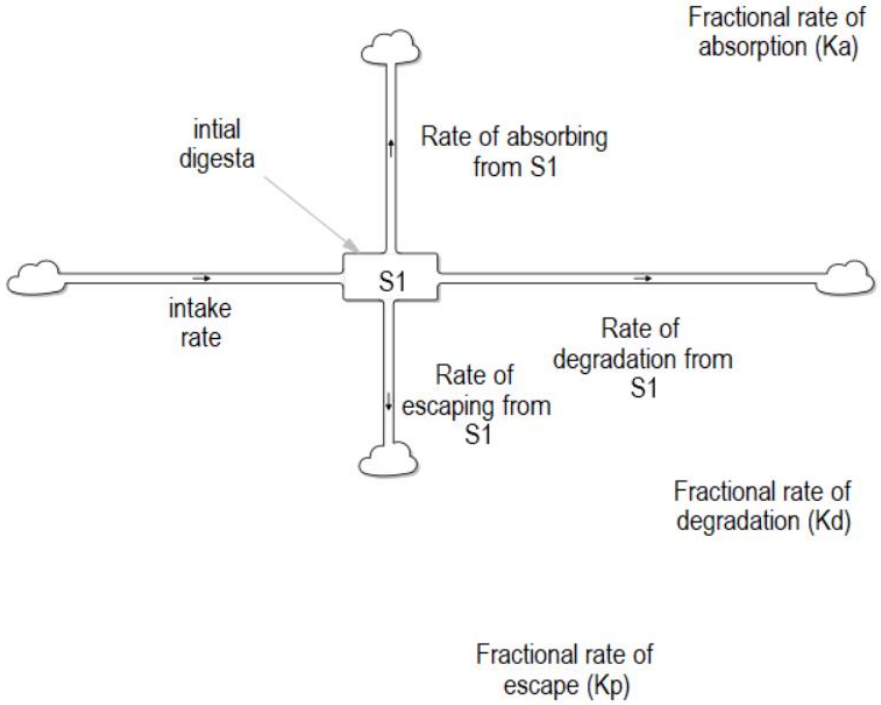
```
11  ###MODEL 1######
12
13  # Create the start time, finish time, and time step
14  START<-0; FINISH<-120; STEP<-1
15
16  # Create time vector
17  simtime <- seq(START, FINISH, by=STEP)
18
19  # Create stocks vector, with initial values
20  stocks  <- c(sS1=100)
21
22  # Create auxiliaries vector, with values
23  auxs    <- c(aKa=0.02, aKd=0.25, aKp=0.05)
24
```

- This is setting the initial conditions of the stock “S1” and the auxiliary variables “Ka, Kd, Kp”

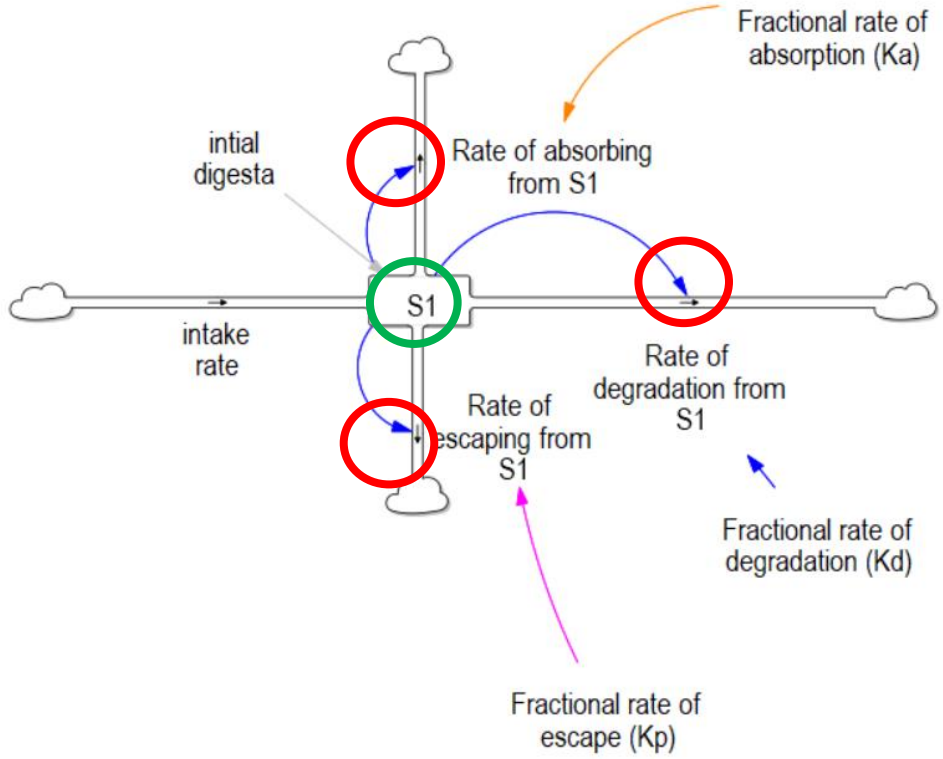


Step 2b and step 2c

- This is “model settings” see ppt: **Step 1**



```r
# write callback function (model equations)
model <- function(time, stocks, auxs){
  with(as.list(c(stocks, auxs)),{


    KaR <- sS1 * aKa
    KpR <- sS1 * aKp
    KdR <- sS1 * aKd

    d_sS1_dt  <- -(KaR+KpR+KdR)



    return (list(c(d_sS1_dt),
                 fKaR=KaR, fKpR=KpR))
  })
}

```



This makes a functions that calculates rates and the "delta" of the S1 stock.

```
43  # Call Solver, and store results in a data frame
44  o<-data.frame(ode(y=stocks, times=simtime, func = model,
45                    parms=auxs, method="euler"))
46
47  # Plot output
48  p1<-ggplot()+
49    geom_line(data=o,aes(time,o$ss1,color="1. S"))+
50    scale_y_continuous(labels = comma)+
51    ylab("Rumen Content")+
52    xlab("Day") +
53    labs(color="")+
54    theme(legend.position="bottom")
55
56  p1
57
```

Integration method

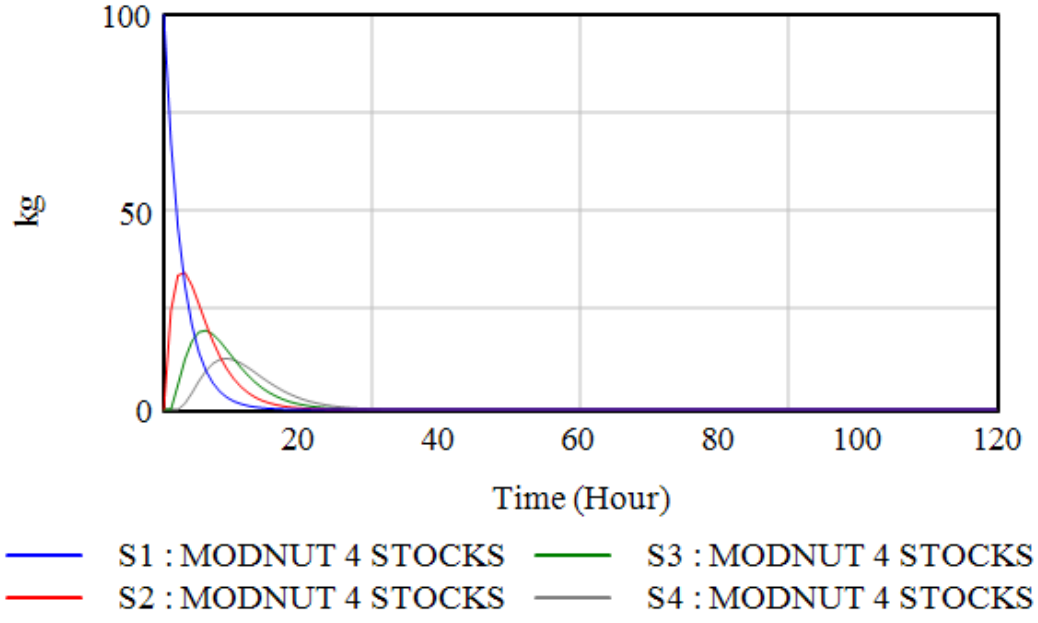
We then employ the Ordinary Differential Equation (ode) function in R.

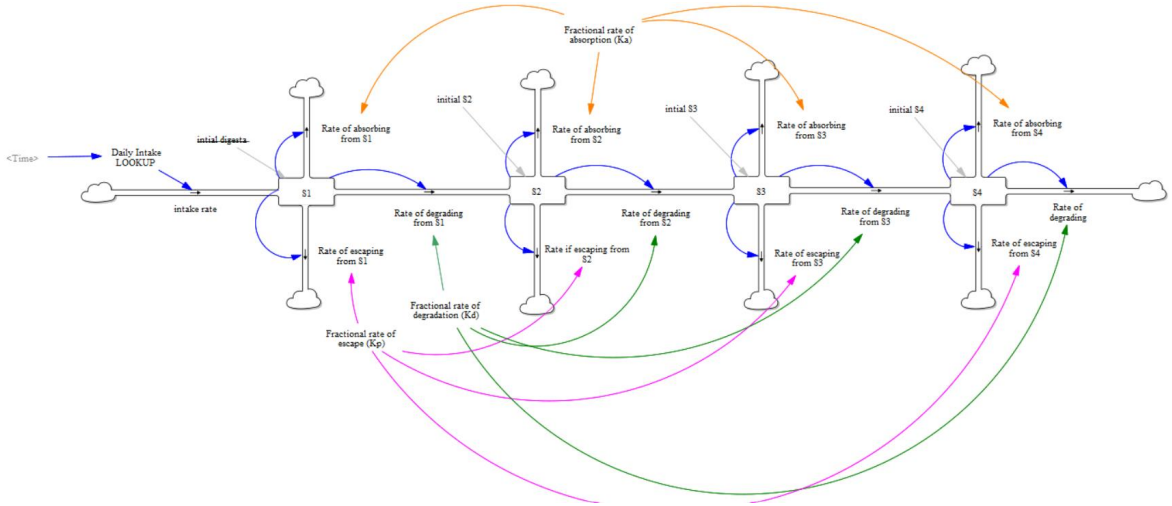
Next, we plot the stock value.

Models 2 (see “p2”)

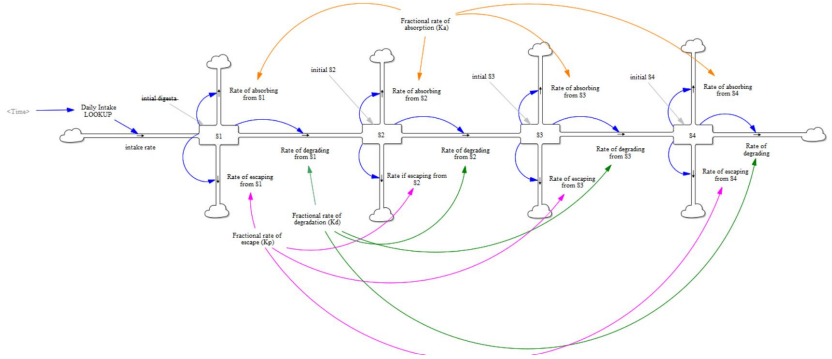
- This code only add additional **stocks** and **inputs**

With only the initial value of **<u>100 kg</u>** in S1





Models 3 (see “p3”)

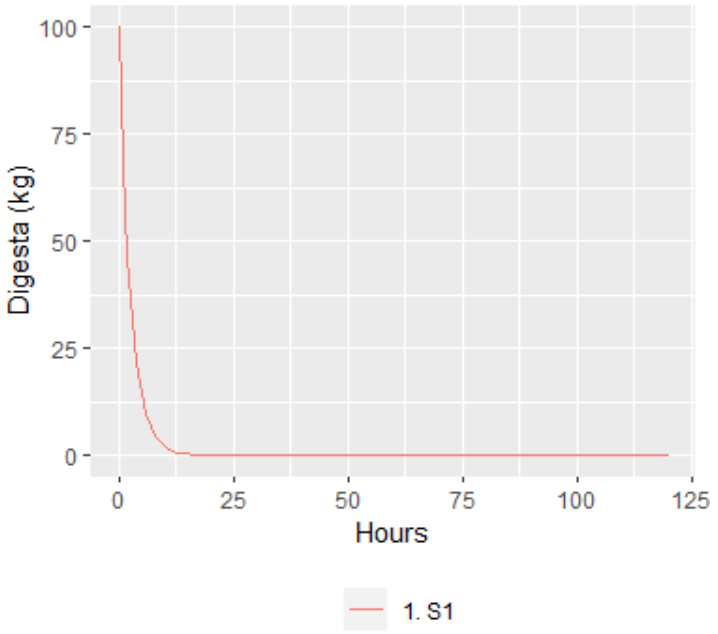


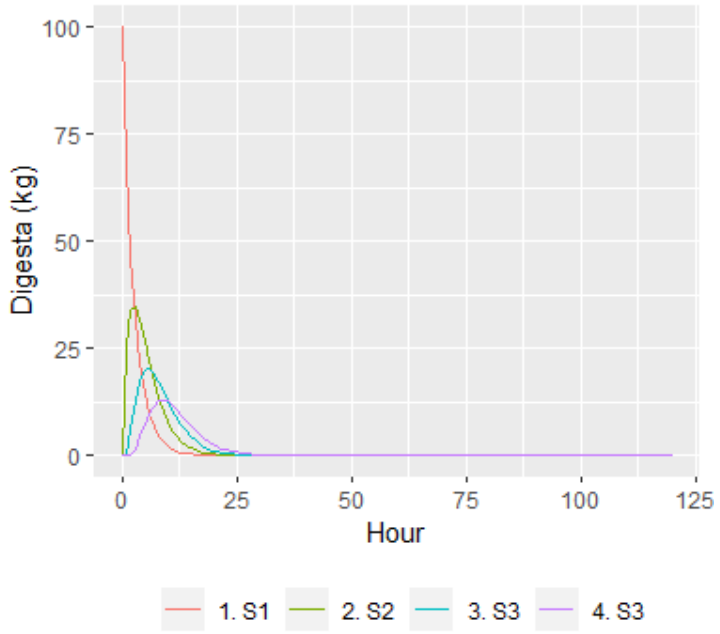
- This code imports precision DMI data from dry beef Angus cows (mean BW ~635 kg) consuming chopped hay (7 to 15 % CP) using **GitHub**

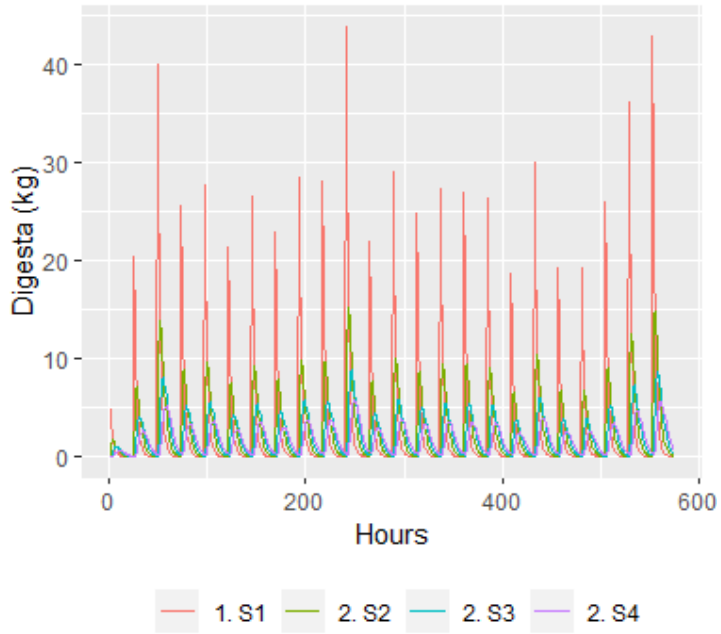
```
131  ### Model 3 Precision Data Integration ###
132
133  #load the data from gitub repo
134  dmi = read_csv('https://raw.githubusercontent.com/rhensen/modnut/main/dmi_example.csv')
135
```

- This code allows you to check the data and then subset data to assess a specific cow. In R [row#,column#]

```
146  #View data
147  head (dmi)
148
149  #call the first cow
150  dmi_2=dmi[1:24,2] #can subset by [row1:20,column]
151
```







Models 1-3 Result Output

Open Shiny App

- https://agland.sdstate.edu/rumen_kinetics/

Contact

Hector Menendez III
(Hector.Menendez@sdstate.edu)
Jameson R. Brennan
(Jameson.Brennan@sdstate.edu)
Krista Ehlert
(Krista.Ehlert@sdstate.edu)
Luis O. Tedeschi
(Luis.Tedeschi@tamu.edu)

References

- Duggan, J. (2016). System dynamics modeling with R (Vol. 501). Cham: Springer International Publishing.
- Jacobs, M., Remus, A., Gaillard, C., Menendez III, H. M., Tedeschi, L. O., Neethirajan, S., & Ellis, J. L. (2022). ASAS-NANP symposium: mathematical modeling in animal nutrition: limitations and potential next steps for modeling and modelers in the animal sciences. *Journal of Animal Science*, *100*(6), skac132.
- Menendez III, H. M., Brennan, J. R., Gaillard, C., Ehlert, K., Quintana, J., Neethirajan, S., ... & Tedeschi, L. O. (2022). ASAS–NANP Symposium: Mathematical Modeling in Animal Nutrition: Opportunities and challenges of confined and extensive precision livestock production. *Journal of Animal Science*, *100*(6), skac160.
- Sterman, J. (2010). *Business dynamics*. Irwin/McGraw-Hill 2000.
- TEDESCHI, L. O., & Fox, D. G. (2018; 2020). *RUMINANT NUTRITION SYSTEM,*. XANEDU PUBLISHING Incorporated.