CSI 709/CSS 739
Verification and Validation of Models

Computational Techniques to Support Simulation Model Validation

Dr. Hamdi Kavak
Computational and Data Sciences Department

http://www.hamdikavak.com
hkavak@gmu.edu

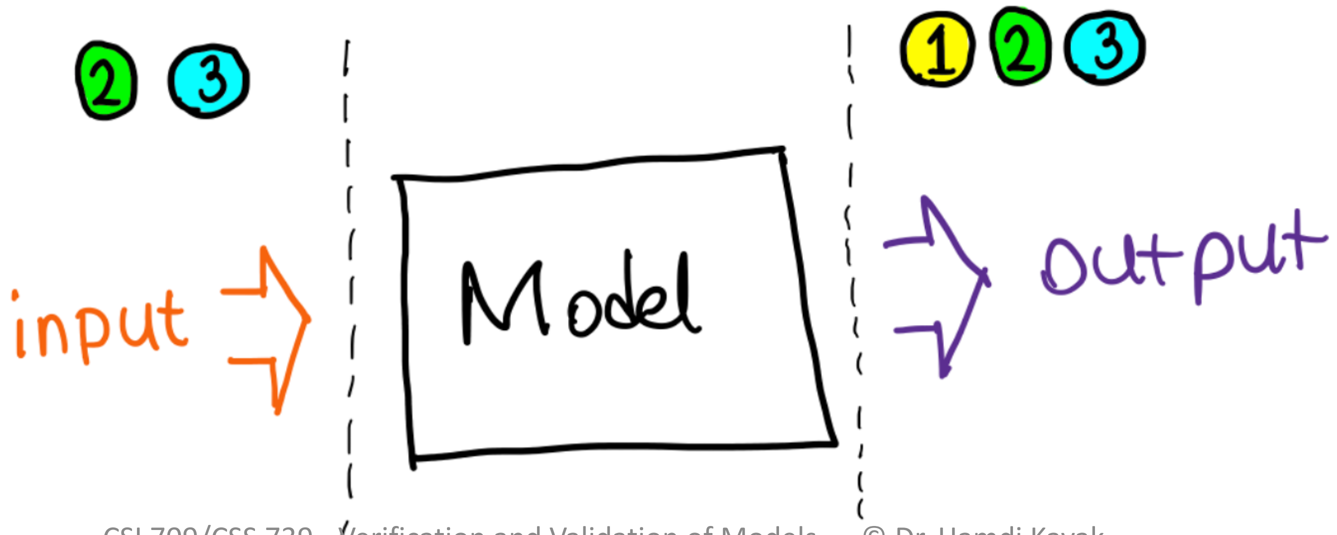
Objective of this lecture

- Exploring some computational techniques to conduct more thorough V&V for simulation models.

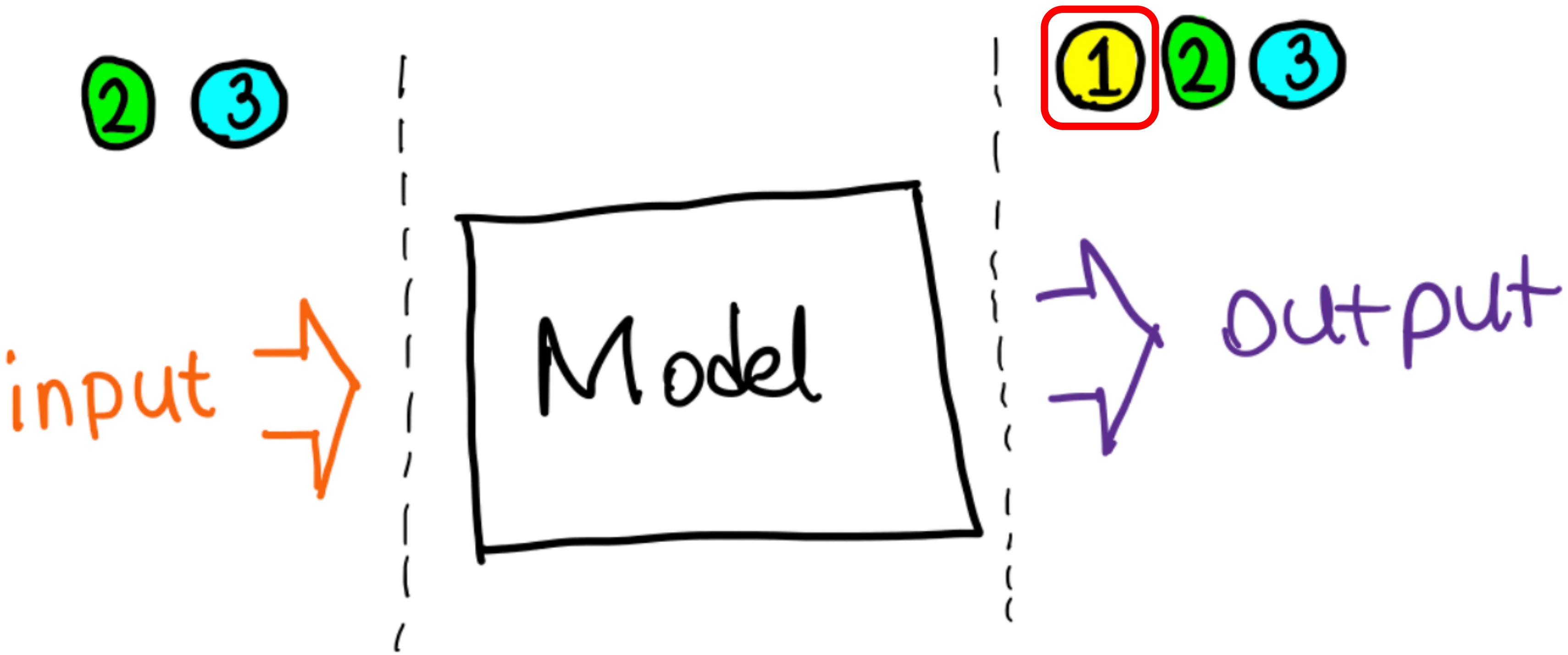
1 Non-terminating simulations

2 Model parameter value estimation

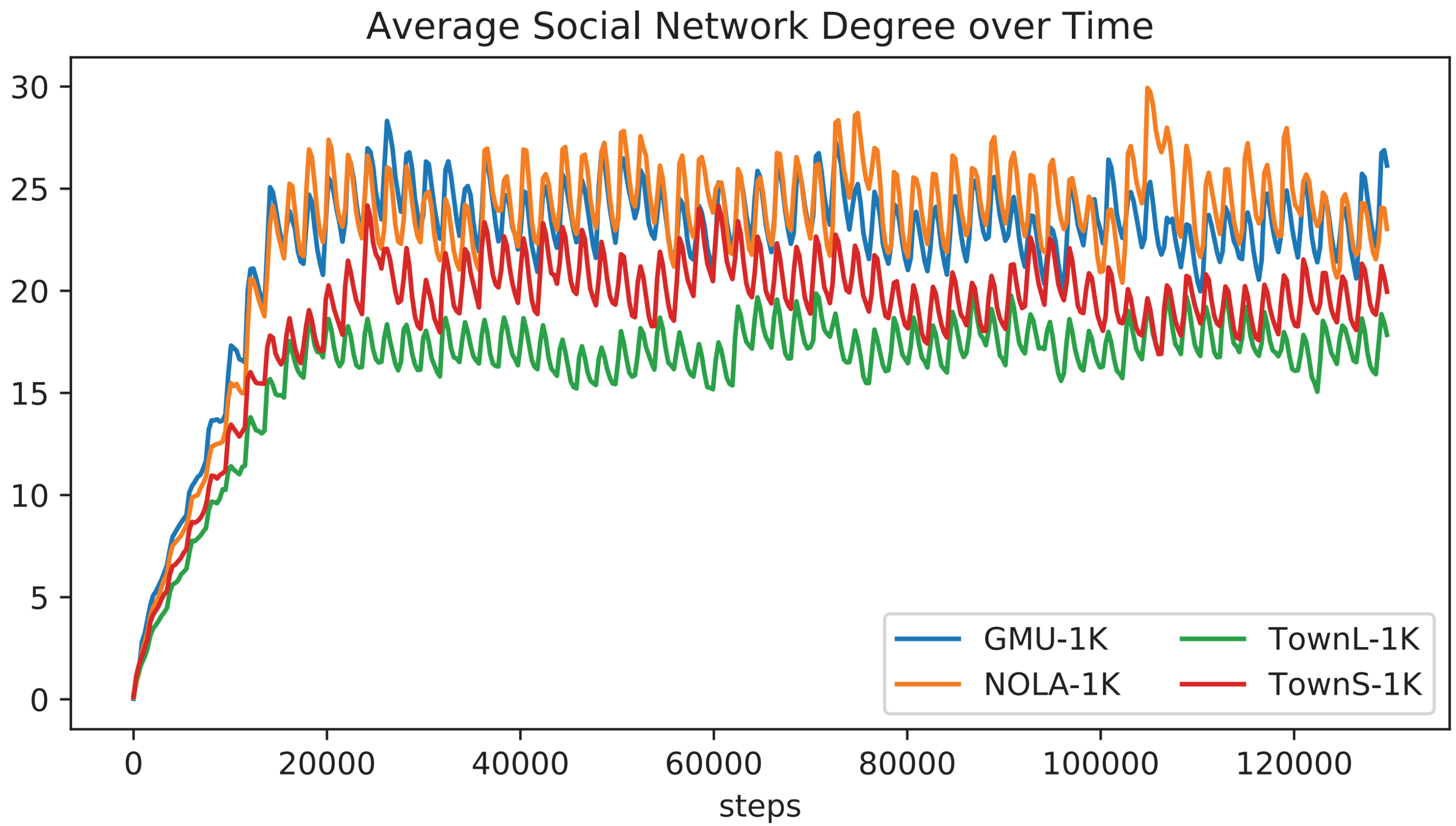
3 Model sensitivity



1. Non-terminating simulations
2. Model parameter value estimation
3. Model sensitivity



What are your observations about this plot?



Kim, J. S., Jin, H., Kavak, H., Rouly, O. C., Crooks, A., Pfoser, D., ... & Züfle, A. (2020). Location-based Social Network Data Generation Based on Patterns of Life. In *IEEE International Conference on Mobile Data Management (MDM'20). IEEE*.

Simulation types

Terminating simulation

- The behavior of interest in the model has a specific initial state and a natural ending event (E).
- Runs for a specific duration until E occurs.
- Examples:
  - A grocery store model with an opening time of 6:00 am and closing time of 10:00 pm when no more customers are inside.
  - A WMD model aims to simulate people's behavior to a nuclear bomb until the initial lethal effects of the bomb are no longer present.

Non-terminating simulation

- The behavior of interest in the model does not have a natural ending event.
- Runs for a duration identified by the modeler.
- The interest is usually to study long-term behaviors.
- Examples:
  - An assembly line simulation that runs almost non-stop with infrequent disruptions.
  - A patterns of life model that simulates the fabric of everyday life and people's movement.

Simulation types

Terminating simulation

- The behavior of interest in the model has a specific initial state and a natural ending event (E).
- Runs for a specific duration until E occurs.
- Examples:
  - A grocery store model with an opening time of 6:00 am and closing time of 10:00 pm when no more customers are inside.
  - A WMD model aims to simulate people's behavior to a nuclear bomb until the initial lethal effects of the bomb are no longer present.

Non-terminating simulation

- The behavior of interest in the model does not have a natural ending event.
- Runs for a duration identified by the modeler.
- The interest is usually to study long-term behaviors.
- Examples:
  - An assembly line simulation that runs almost non-stop with infrequent disruptions.
  - A patterns of life model that simulates the fabric of everyday life and people's movement.

To identify whether a simulation is terminating or not terminating, one needs to look at **"the objectives of the simulation"** and **"the nature of the system"** (Banks et al., 2013)

Terminating vs. not terminating categorization is usually considered for Discrete-Event Simulation (DES) models. However, it can be applied to other simulation model types (e.g., ABMs, hybrid models).

Non-terminating simulation types

STEADY-STATE

STEADY-STATE CYCLIC

OTHER

Non-terminating simulation types*

- **Steady-state**
  - Example manufacturing operation:
    - "Operation starts empty and idle at t = 0.
    - System operates **2 eight-hour shifts per day, 5 days per week**.
    - System state at day's end is the initial state for beginning the next day.
    - Interested in long-term average daily production."

*Source: Mielke

Non-terminating simulation types*

- **Steady-state**
  - Example manufacturing operation:
    - "Operation starts empty and idle at t = 0.
    - System operates **2 eight-hour shifts per day, 5 days per week**.
    - System state at day's end is the initial state for beginning the next day.
    - Interested in long-term average daily production."

- **Steady-state cyclic**
  - Example manufacturing operation:
    - "Operation starts empty and idle at t = 0.
    - System operates **2 eight-hour shifts per day, Monday through Friday**.
    - System operates **1 eight-hour shift on Saturday and Sunday**.
    - System state at day's end is the initial state for beginning the next day.
    - Interested in long-term average daily production."

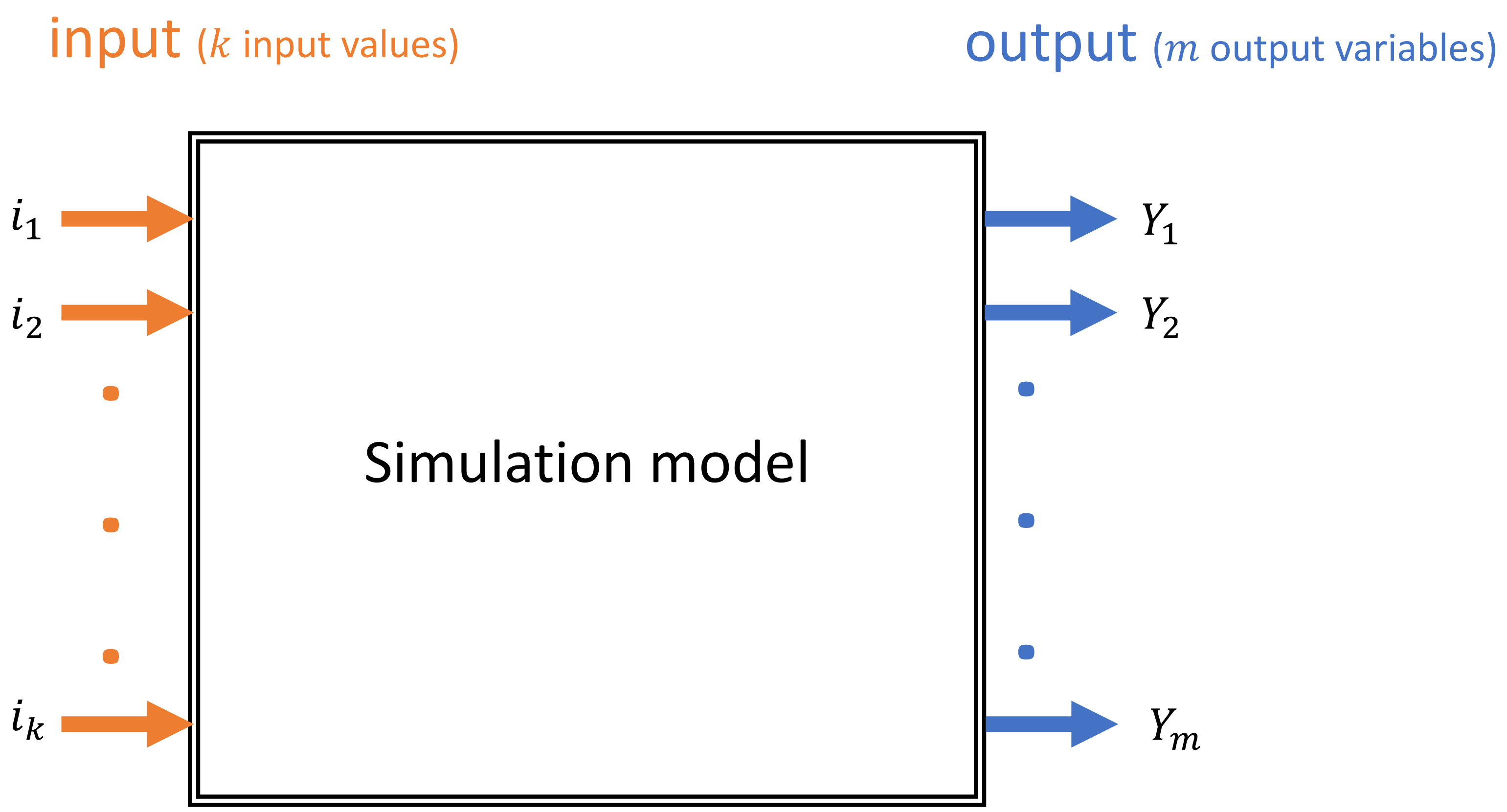
*Source: Mielke

Non-terminating simulation types*

- **Steady-state**
  - Example manufacturing operation:
    - "Operation starts empty and idle at t = 0.
    - System operates **2 eight-hour shifts per day, 5 days per week**.
    - System state at day's end is the initial state for beginning the next day.
    - Interested in long-term average daily production."
- **Other**
  - Example call center operation:
    - "Operation starts empty and idle at t = 0.
    - System operates **3 eight-hour shifts per day, 7 days per week**.
    - Call volume varies by the day of the week.
    - Weekly call volume depends on season of the year.
    - Yearly call volume varies with economy.
    - Interested in average delay experienced by a caller."
- **Steady-state cyclic**
  - Example manufacturing operation:
    - "Operation starts empty and idle at t = 0.
    - System operates **2 eight-hour shifts per day, Monday through Friday**.
    - System operates **1 eight-hour shift on Saturday and Sunday**.
    - System state at day's end is the initial state for beginning the next day.
    - Interested in long-term average daily production."

*Source: Mielke

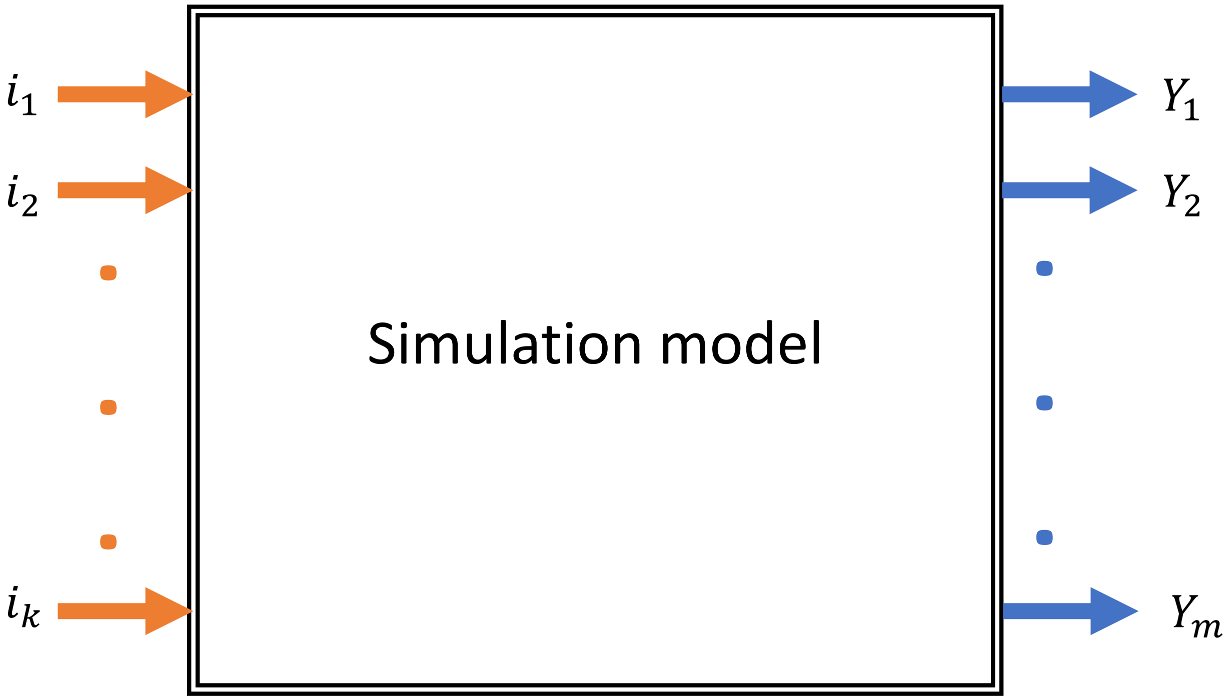
Simulation input/output formalization



Simulation input/output formalization

input (k input values) output (m output variables)

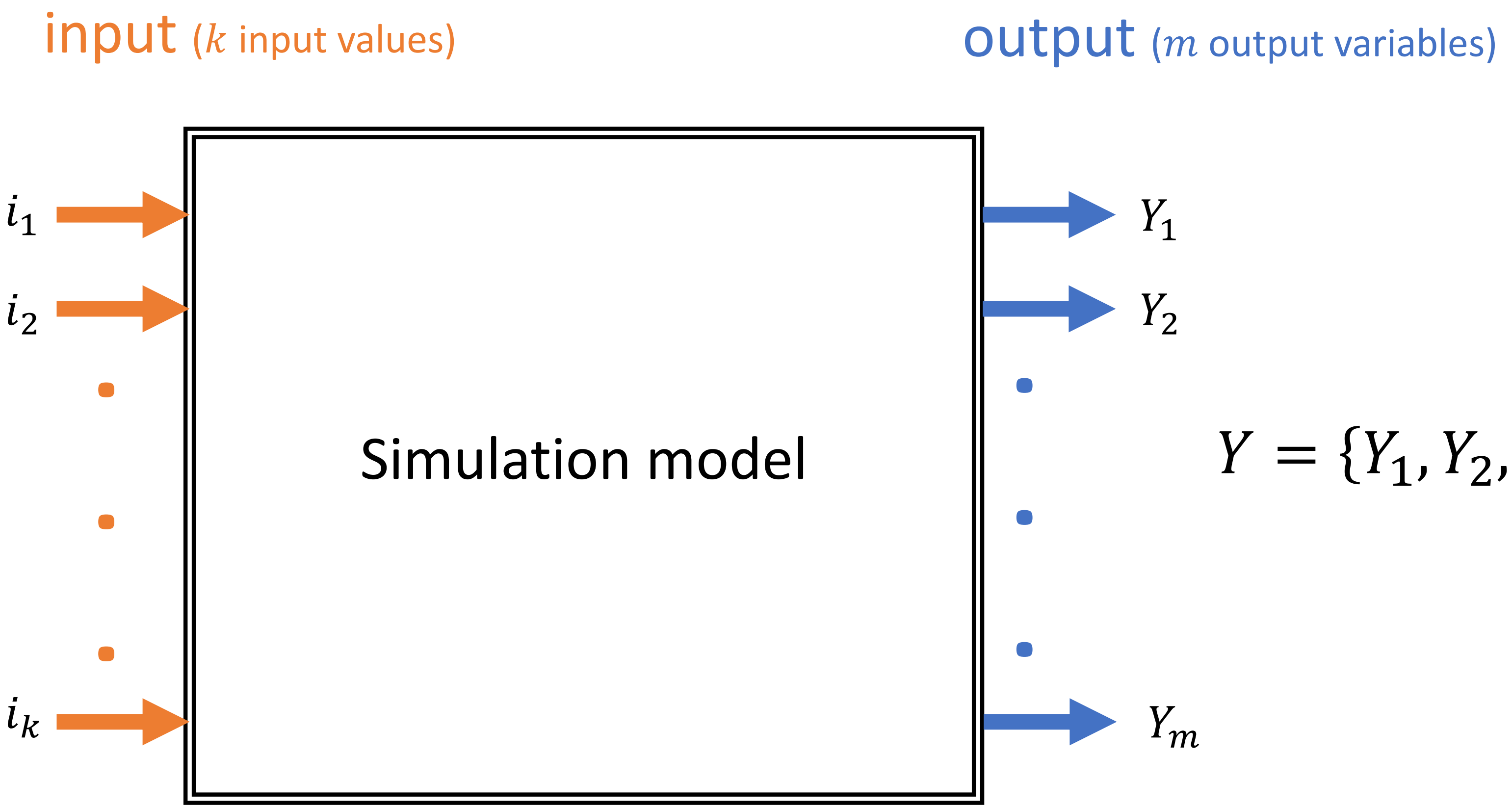
$$I = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_\mathrm{k}\}$$



$$Y = \{Y_1, Y_2, \ldots, Y_m\}$$

Simulation input/output formalization

$I = \{\mathrm{i}_1, \mathrm{i}_2, \dots, \mathrm{i}_\mathrm{k}\}$



$Y = \{Y_1, Y_2, \dots, Y_m\}$

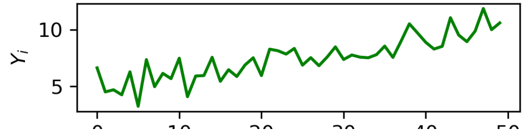
$$Y_i = \{y_{i1}, y_{i2}, \dots, y_{ij}\}$$

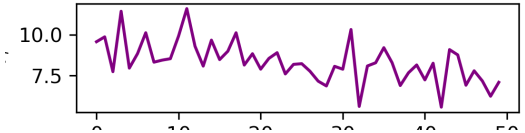
we observe j values for output variable i

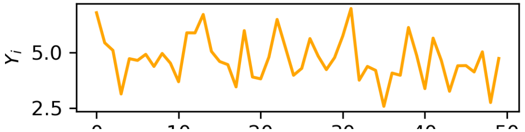
Simulation output data

- Most simulation models are stochastic and non-terminating.
- Assume that we are interested in the output variable $Y_i = \{y_{i1}, y_{i2}, \dots, y_{ij}\}$.
- Run $\boldsymbol{n}$ times w/ the same input set (I) and observe j values in each run.

$$\begin{array}{cccccccccl}
\boldsymbol{y_{i1}^{(1)}} & \boldsymbol{y_{i2}^{(1)}} & \boldsymbol{y_{i3}^{(1)}} & \cdots & \boldsymbol{y_{ik}^{(1)}} & \boldsymbol{y_{i(k+1)}^{(1)}} & \cdots & \boldsymbol{y_{ij}^{(1)}} & \longleftarrow & \text{Run } \mathbf{1} \\
\boldsymbol{y_{i1}^{(2)}} & \boldsymbol{y_{i2}^{(2)}} & \boldsymbol{y_{i3}^{(2)}} & \cdots & \boldsymbol{y_{ik}^{(2)}} & \boldsymbol{y_{i(k+1)}^{(2)}} & \cdots & \boldsymbol{y_{ij}^{(2)}} & \longleftarrow & \text{Run } \mathbf{2} \\
\cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot & & \\
\boldsymbol{y_{i1}^{(n)}} & \boldsymbol{y_{i2}^{(n)}} & \boldsymbol{y_{i3}^{(n)}} & \cdots & \boldsymbol{y_{ik}^{(n)}} & \boldsymbol{y_{i(k+1)}^{(n)}} & \cdots & \boldsymbol{y_{ij}^{(n)}} & \longleftarrow & \text{Run } \boldsymbol{n}
\end{array}$$



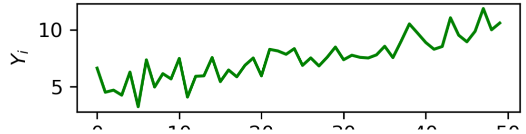


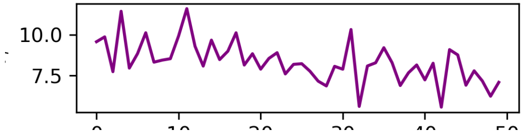


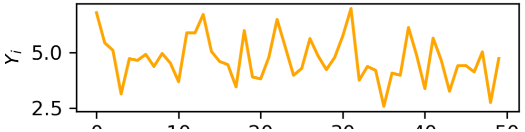
Simulation output data

- Most simulation models are stochastic and non-terminating.
- Assume that we are interested in the output variable $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{ij}\}$.
- Run n times w/ the same input set (I) and observe j values in each run.

$$
\begin{array}{cccccccccl}
\boxed{\begin{array}{c} \boldsymbol{y_{i1}^{(1)}} \\ \boldsymbol{y_{i1}^{(2)}} \\ \vdots \\ \boldsymbol{y_{i1}^{(n)}} \end{array}} &
\begin{array}{c} \boldsymbol{y_{i2}^{(1)}} \\ \boldsymbol{y_{i2}^{(2)}} \\ \vdots \\ \boldsymbol{y_{i2}^{(n)}} \end{array} &
\begin{array}{c} \boldsymbol{y_{i3}^{(1)}} \\ \boldsymbol{y_{i3}^{(2)}} \\ \vdots \\ \boldsymbol{y_{i3}^{(n)}} \end{array} &
\begin{array}{c} \cdots \\ \cdots \\ \\ \cdots \end{array} &
\begin{array}{c} \boldsymbol{y_{ik}^{(1)}} \\ \boldsymbol{y_{ik}^{(2)}} \\ \vdots \\ \boldsymbol{y_{ik}^{(n)}} \end{array} &
\begin{array}{c} \boldsymbol{y_{i(k+1)}^{(1)}} \\ \boldsymbol{y_{i(k+1)}^{(2)}} \\ \vdots \\ \boldsymbol{y_{i(k+1)}^{(n)}} \end{array} &
\begin{array}{c} \cdots \\ \cdots \\ \\ \cdots \end{array} &
\begin{array}{c} \boldsymbol{y_{ij}^{(1)}} \\ \boldsymbol{y_{ij}^{(2)}} \\ \vdots \\ \boldsymbol{y_{ij}^{(n)}} \end{array} &
\begin{array}{c} \longleftarrow \\ \longleftarrow \\ \\ \longleftarrow \end{array} &
\begin{array}{l} \text{Run } \mathbf{1} \\ \text{Run } \mathbf{2} \\ \\ \text{Run } \boldsymbol{n} \end{array}
\end{array}
$$





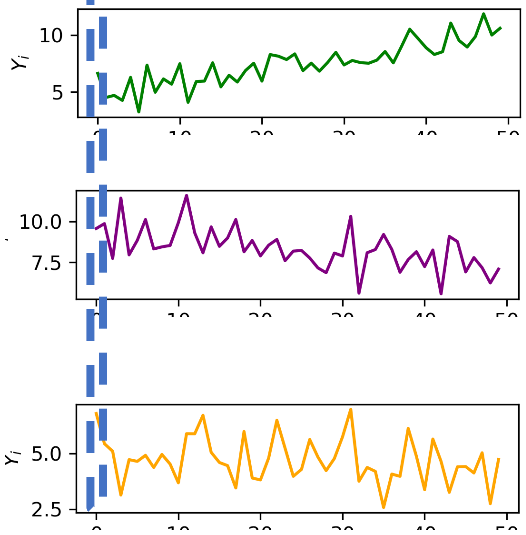


n samples from output variable Y_i at the first observation

Simulation output data

- Most simulation models are stochastic and non-terminating.
- Assume that we are interested in the output variable $Y_i = \{y_{i1}, y_{i2}, \dots, y_{ij}\}$.
- Run n times w/ the same input set (I) and observe j values in each run.

$$
\begin{array}{ccccccccl}
y_{i1}^{(1)} & y_{i2}^{(1)} & y_{i3}^{(1)} & \cdots & y_{ik}^{(1)} & y_{i(k+1)}^{(1)} & \cdots & y_{ij}^{(1)} & \longleftarrow \text{Run } 1 \\
y_{i1}^{(2)} & y_{i2}^{(2)} & y_{i3}^{(2)} & \cdots & y_{ik}^{(2)} & y_{i(k+1)}^{(2)} & \cdots & y_{ij}^{(2)} & \longleftarrow \text{Run } 2 \\
. & . & . & & . & . & & . & \\
y_{i1}^{(n)} & y_{i2}^{(n)} & y_{i3}^{(n)} & \cdots & y_{ik}^{(n)} & y_{i(k+1)}^{(n)} & \cdots & y_{ij}^{(n)} & \longleftarrow \text{Run } n
\end{array}
$$

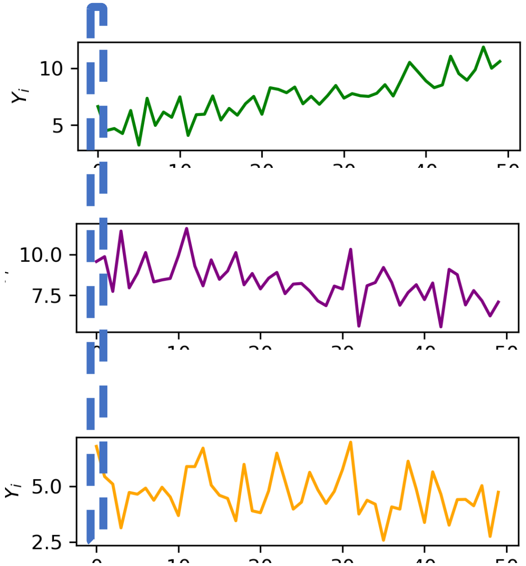


n samples from output variable Y_i at the first observation

Simulation output data

- Most simulation models are stochastic and non-terminating.
- Assume that we are interested in the output variable $Y_i = \{y_{i1}, y_{i2}, \dots, y_{ij}\}$.
- Run n times w/ the same input set (I) and observe j values in each run.

$$
\begin{array}{cccccccccl}
y_{i1}^{(1)} & y_{i2}^{(1)} & y_{i3}^{(1)} & \cdots & y_{ik}^{(1)} & y_{i(k+1)}^{(1)} & \cdots & y_{ij}^{(1)} & \longleftarrow & \text{Run } 1 \\
y_{i1}^{(2)} & y_{i2}^{(2)} & y_{i3}^{(2)} & \cdots & y_{ik}^{(2)} & y_{i(k+1)}^{(2)} & \cdots & y_{ij}^{(2)} & \longleftarrow & \text{Run } 2 \\
. & . & . & & . & . & & . & & \\
y_{i1}^{(n)} & y_{i2}^{(n)} & y_{i3}^{(n)} & \cdots & y_{ik}^{(n)} & y_{i(k+1)}^{(n)} & \cdots & y_{ij}^{(n)} & \longleftarrow & \text{Run } n
\end{array}
$$



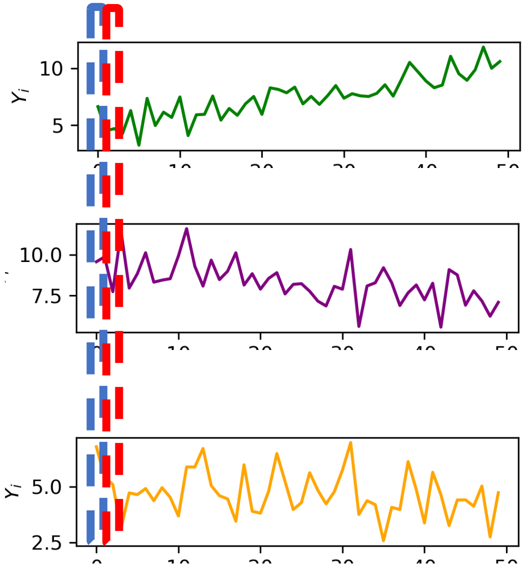
n samples from output variable Y_i at the second observation

n samples from output variable Y_i at the first observation

Simulation output data

- Most simulation models are stochastic and non-terminating.
- Assume that we are interested in the output variable $Y_i = \{y_{i1}, y_{i2}, \dots, y_{ij}\}$.
- Run $\boldsymbol{n}$ times w/ the same input set (I) and observe j values in each run.

$$
\begin{array}{cccccccccl}
\boldsymbol{y_{i1}^{(1)}} & \boldsymbol{y_{i2}^{(1)}} & \boldsymbol{y_{i3}^{(1)}} & \cdots & \boldsymbol{y_{ik}^{(1)}} & \boldsymbol{y_{i(k+1)}^{(1)}} & \cdots & \boldsymbol{y_{ij}^{(1)}} & \longleftarrow & \text{Run } \mathbf{1} \\
\boldsymbol{y_{i1}^{(2)}} & \boldsymbol{y_{i2}^{(2)}} & \boldsymbol{y_{i3}^{(2)}} & \cdots & \boldsymbol{y_{ik}^{(2)}} & \boldsymbol{y_{i(k+1)}^{(2)}} & \cdots & \boldsymbol{y_{ij}^{(2)}} & \longleftarrow & \text{Run } \mathbf{2} \\
. & . & . & & . & . & & . & & \\
\boldsymbol{y_{i1}^{(n)}} & \boldsymbol{y_{i2}^{(n)}} & \boldsymbol{y_{i3}^{(n)}} & \cdots & \boldsymbol{y_{ik}^{(n)}} & \boldsymbol{y_{i(k+1)}^{(n)}} & \cdots & \boldsymbol{y_{ij}^{(n)}} & \longleftarrow & \text{Run } \boldsymbol{n}
\end{array}
$$



n samples from output variable Y_i at the second observation

n samples from output variable Y_i at the first observation

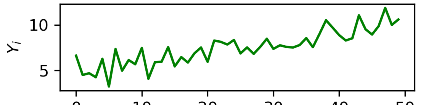
How to know if simulation is in steady-state?

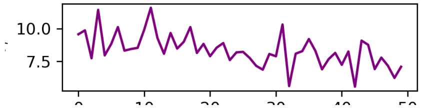
- There is often the first k observations such that distribution from $k+1$ and beyond are roughly the same, a.k.a., **steady-state**.
- Thus, the first k observations are usually called **transient**.
- We need to ignore transient state observations

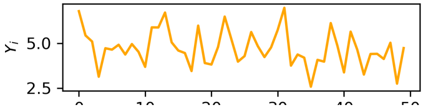
How to know if simulation is in steady-state?

- There is often the first k observations such that distribution from $k+1$ and beyond are roughly the same, a.k.a., **steady-state**.
- Thus, the first k observations are usually called **transient-state**.
- We need to ignore transient state observations

$$
\begin{array}{cccccccccl}
\boldsymbol{y_{i1}^{(1)}} & \boldsymbol{y_{i2}^{(1)}} & \boldsymbol{y_{i3}^{(1)}} & \cdots & \boldsymbol{y_{ik}^{(1)}} & \boldsymbol{y_{i(k+1)}^{(1)}} & \cdots & \boldsymbol{y_{ij}^{(1)}} & \longleftarrow & \text{Run } \mathbf{1} \\
\boldsymbol{y_{i1}^{(2)}} & \boldsymbol{y_{i2}^{(2)}} & \boldsymbol{y_{i3}^{(2)}} & \cdots & \boldsymbol{y_{ik}^{(2)}} & \boldsymbol{y_{i(k+1)}^{(2)}} & \cdots & \boldsymbol{y_{ij}^{(2)}} & \longleftarrow & \text{Run } \mathbf{2} \\
. & . & . & & . & . & & . & & \\
\boldsymbol{y_{i1}^{(n)}} & \boldsymbol{y_{i2}^{(n)}} & \boldsymbol{y_{i3}^{(n)}} & \cdots & \boldsymbol{y_{ik}^{(n)}} & \boldsymbol{y_{i(k+1)}^{(n)}} & \cdots & \boldsymbol{y_{ij}^{(n)}} & \longleftarrow & \text{Run } \boldsymbol{n}
\end{array}
$$



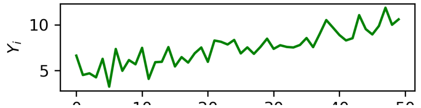


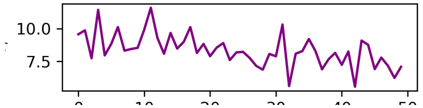


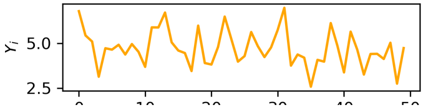
How to know if simulation is in steady-state?

- There is often the first k observations such that distribution from $k+1$ and beyond are roughly the same, a.k.a., **steady-state**.
- Thus, the first k observations are usually called **transient-state**.
- We need to ignore transient state observations.

$$\underbrace{\begin{matrix} y_{i1}^{(1)} & y_{i2}^{(1)} & y_{i3}^{(1)} & \cdots & y_{ik}^{(1)} \\ y_{i1}^{(2)} & y_{i2}^{(2)} & y_{i3}^{(2)} & \cdots & y_{ik}^{(2)} \\ . & . & . & & . \\ y_{i1}^{(n)} & y_{i2}^{(n)} & y_{i3}^{(n)} & \cdots & y_{ik}^{(n)} \end{matrix}}_{\text{transient-state}} \; \underbrace{\begin{matrix} y_{i(k+1)}^{(1)} & \cdots & y_{ij}^{(1)} \\ y_{i(k+1)}^{(2)} & \cdots & y_{ij}^{(2)} \\ . & & . \\ y_{i(k+1)}^{(n)} & \cdots & y_{ij}^{(n)} \end{matrix}}_{\text{steady-state}} \; \begin{matrix} \longleftarrow \text{Run } \mathbf{1} \\ \longleftarrow \text{Run } \mathbf{2} \\ \\ \longleftarrow \text{Run } \boldsymbol{n} \end{matrix}$$







How to know if simulation is in steady-state?

- There is often the first k observations such that distribution from $k + 1$ and beyond are roughly the same, a.k.a., **steady-state**.
- Thus, the first k observations are usually called **transient-state**.
- We need to ignore transient state observations.

$$
\begin{array}{ccccc|ccc}
y_{i1}^{(1)} & y_{i2}^{(1)} & y_{i3}^{(1)} & \cdots & y_{ik}^{(1)} & y_{i(k+1)}^{(1)} & \cdots & y_{ij}^{(1)} \\
y_{i1}^{(2)} & y_{i2}^{(2)} & y_{i3}^{(2)} & \cdots & y_{ik}^{(2)} & y_{i(k+1)}^{(2)} & \cdots & y_{ij}^{(2)} \\
. & . & . & & . & . & & . \\
y_{i1}^{(n)} & y_{i2}^{(n)} & y_{i3}^{(n)} & \cdots & y_{ik}^{(n)} & y_{i(k+1)}^{(n)} & \cdots & y_{ij}^{(n)}
\end{array}
$$

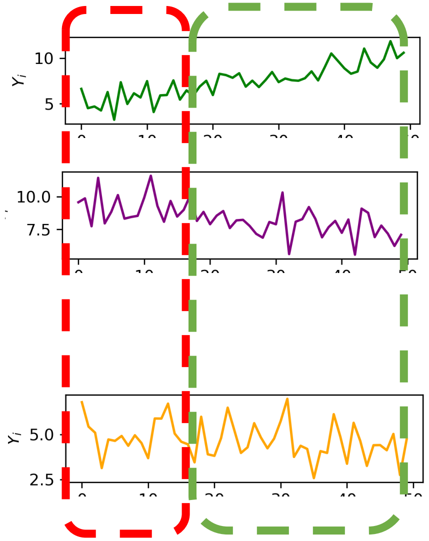
transient-state

steady-state

Run **1**

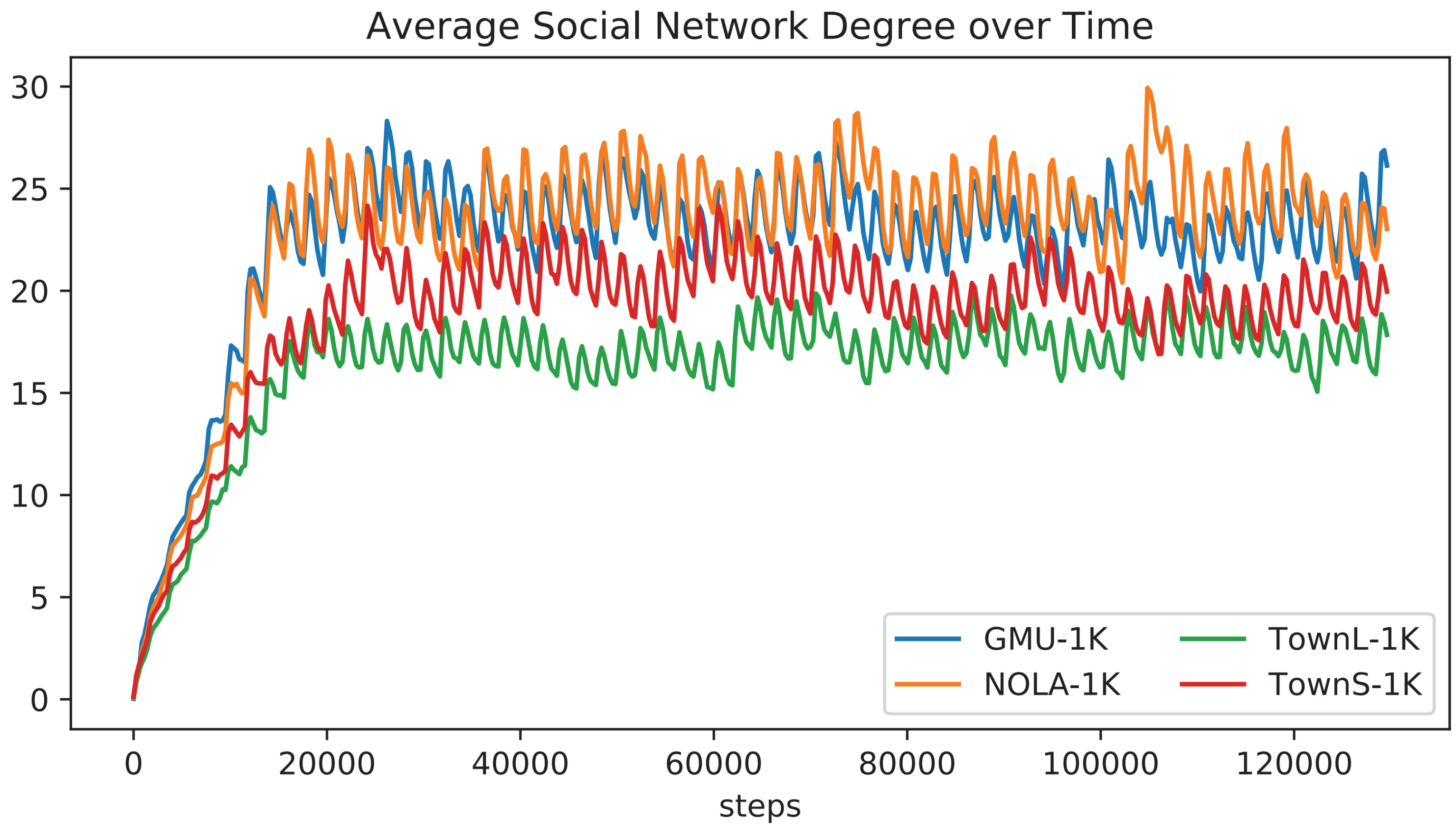
Run **2**

Run n



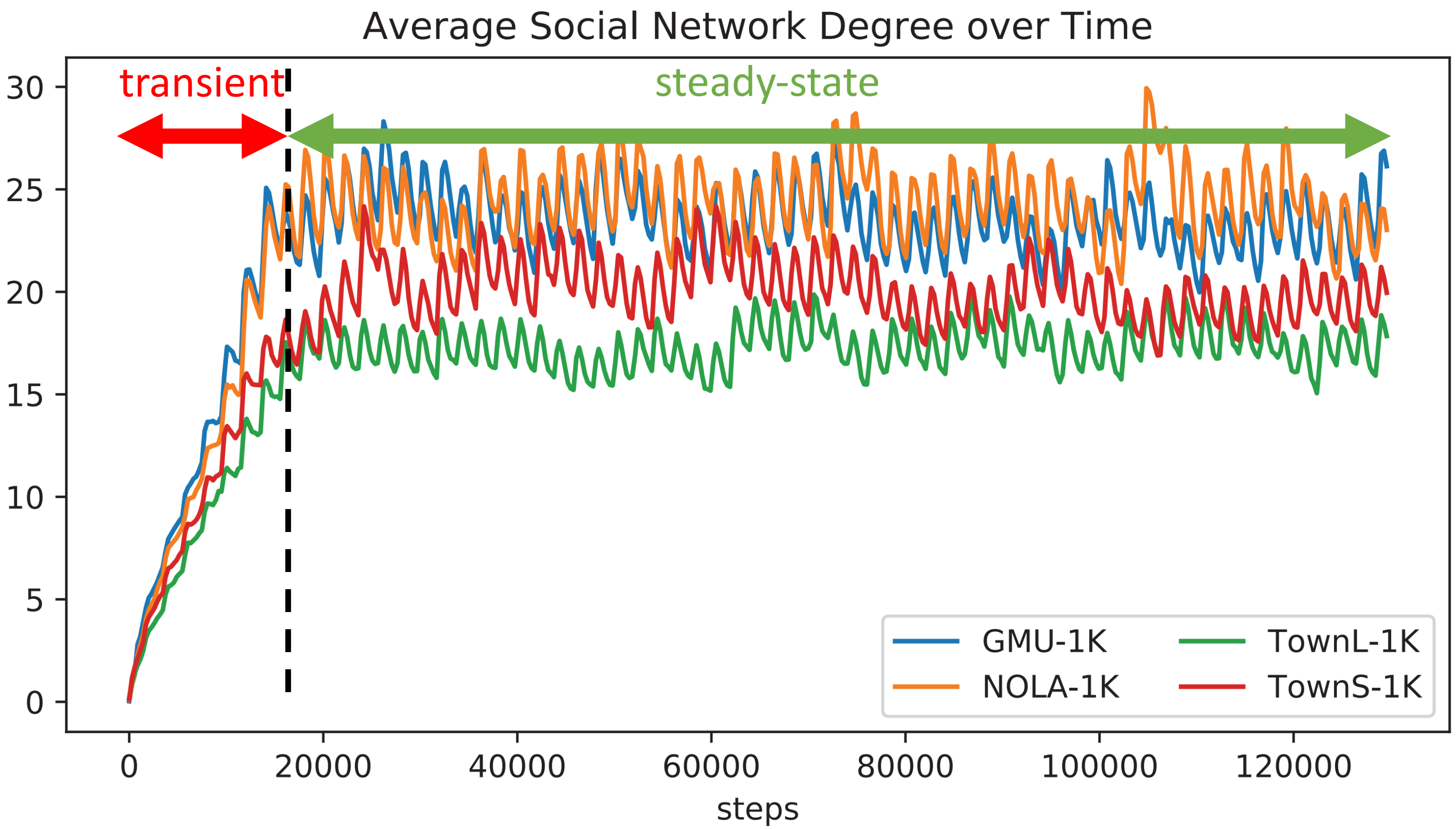
Recall the plot again

Can you show the **transient** and **steady-state** behavior?



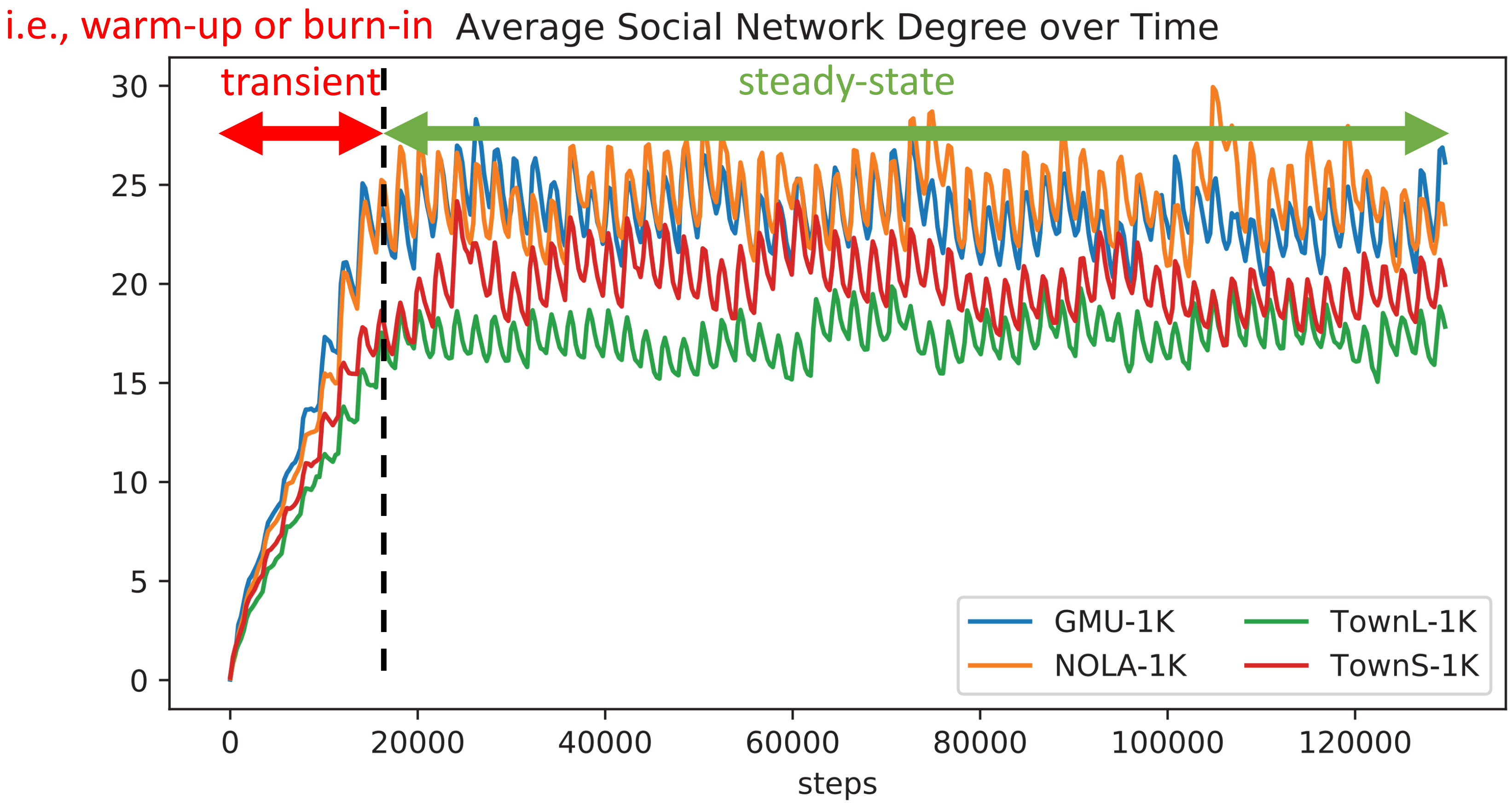
Kim, J. S., Jin, H., Kavak, H., Rouly, O. C., Crooks, A., Pfoser, D., ... & Züfle, A. (2020). Location-based Social Network Data Generation Based on Patterns of Life. In *IEEE International Conference on Mobile Data Management (MDM'20). IEEE*.

Recall the plot again



Kim, J. S., Jin, H., Kavak, H., Rouly, O. C., Crooks, A., Pfoser, D., ... & Züfle, A. (2020). Location-based Social Network Data Generation Based on Patterns of Life. In *IEEE International Conference on Mobile Data Management (MDM'20). IEEE*.

Recall the plot again



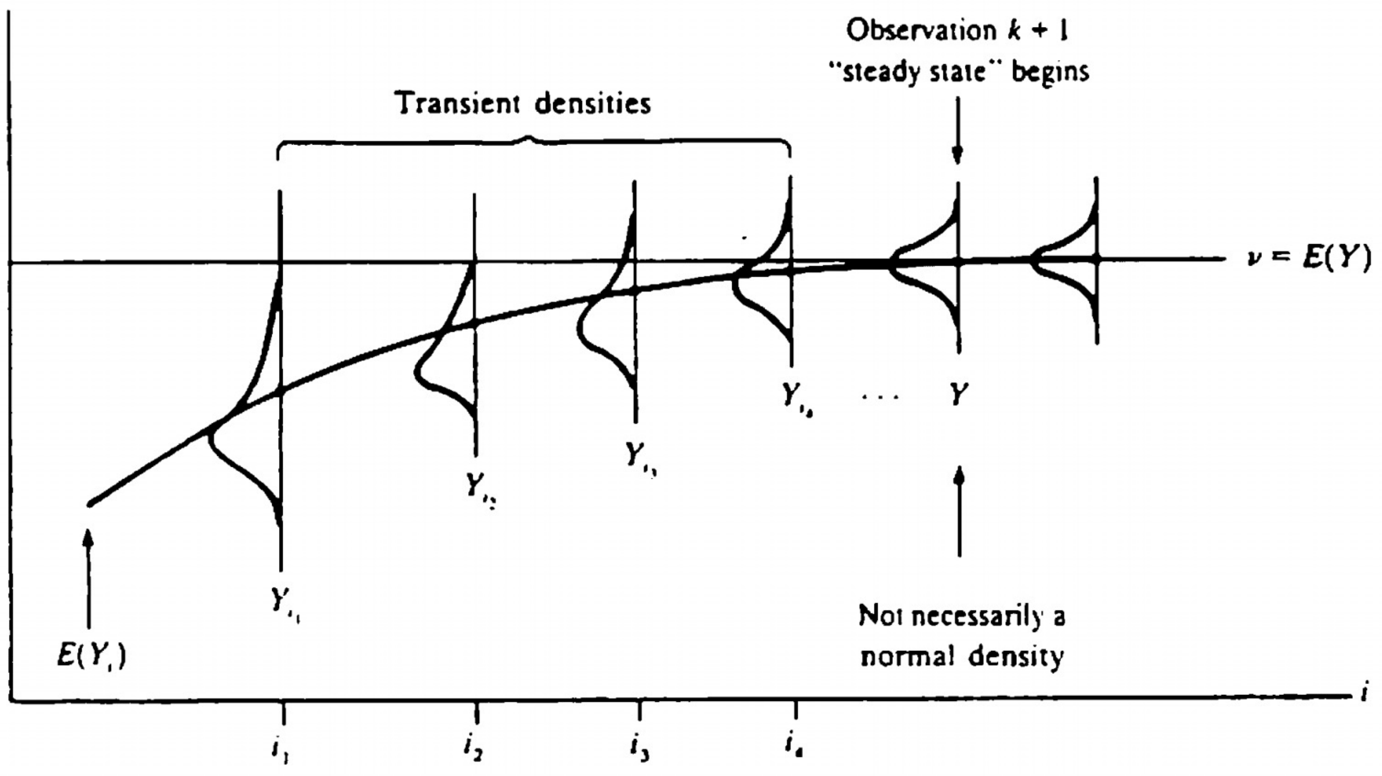
Kim, J. S., Jin, H., Kavak, H., Rouly, O. C., Crooks, A., Pfoser, D., ... & Züfle, A. (2020). Location-based Social Network Data Generation Based on Patterns of Life. In *IEEE International Conference on Mobile Data Management (MDM'20). IEEE*.

How to detect the transient state?

Mahajan & Ingalls (2004) summarizes several methods

- Graphical
- Statistical
- Heuristic
- Initialization Bias

Welch’s Graphical Method



Law (1990)

Conway Rule heuristic

- "Truncate a series of measurements until the first of the series is neither the maximum nor the minimum of the remaining set" (Conway, 1963).
- Gafarian et al. (1978)

1. Decide n and m the number of exploratory replications and the length of the exploratory replications.
2. Compute y^{+}_{jr} and y^{-}_{jr} using following formulae:

$$y^{+}_{jr} = \max(y_{jl} : l = r,\dots,m)\ j = 1,\dots n \qquad (6)$$

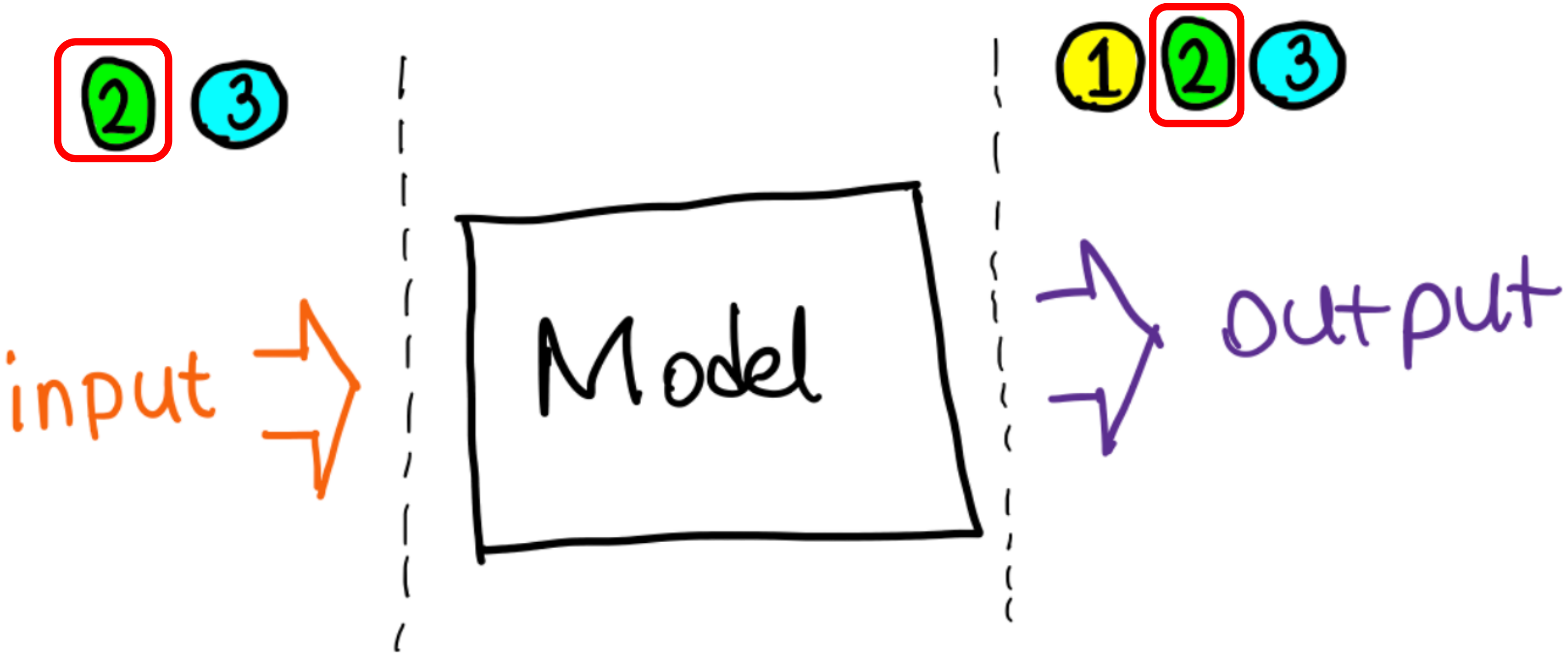
$$y^{-}_{jr} = \min(y_{jl} : l = r,\dots,m)\ j = 1,\dots n \qquad (7)$$

3. For $r = 1, 2, \dots, m$ determine t_j such that $t_{j=\min_r}\{Y^{-}_{jr} < Y_{jr} < Y^{+}_{jr}\}$ occurs for the first time.
4. Estimate of the truncation point t^* is given by $\max\{t_1, t_2, t_3, \dots, t_n\}$

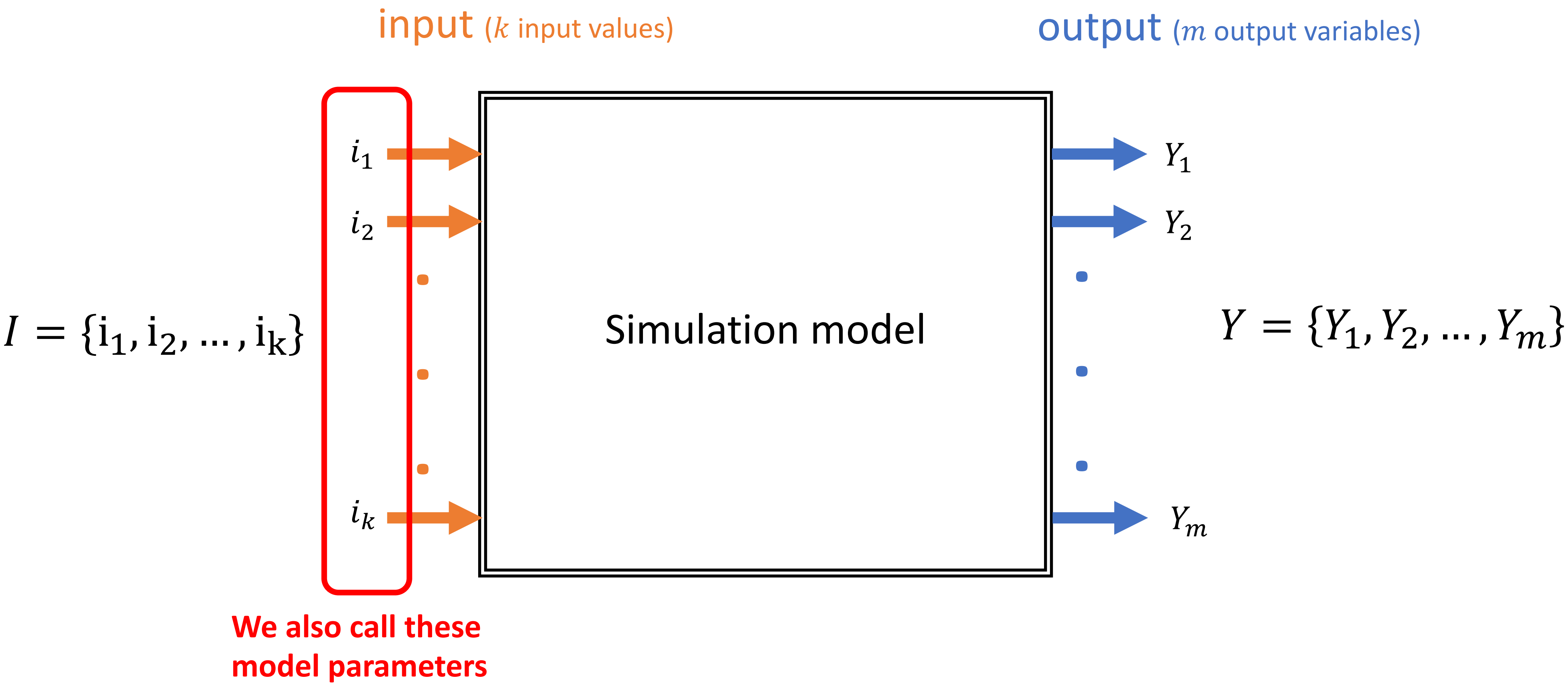
Why do we identify transient state, again?

- Transient state data should be removed from our analysis.
- Selecting inaccurate cut-off point (k) can lead to one of the two results
  - Including transient data in our analysis if $k \ll k_{actual}$
  - Removing meaningful simulation data in our analysis if $k \gg k_{actual}$
- Various techniques are introduced and compared in Mahajan and Ingalls (2004)

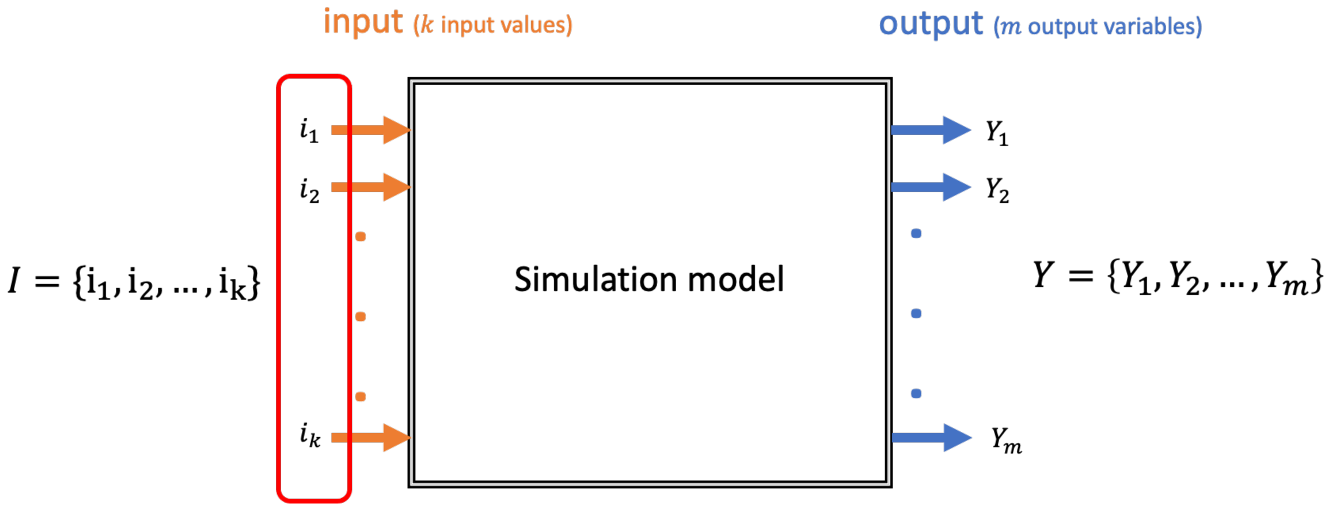
1. Non-terminating simulations
2. Model parameter value estimation
3. Model sensitivity



Simulation input/output formalization

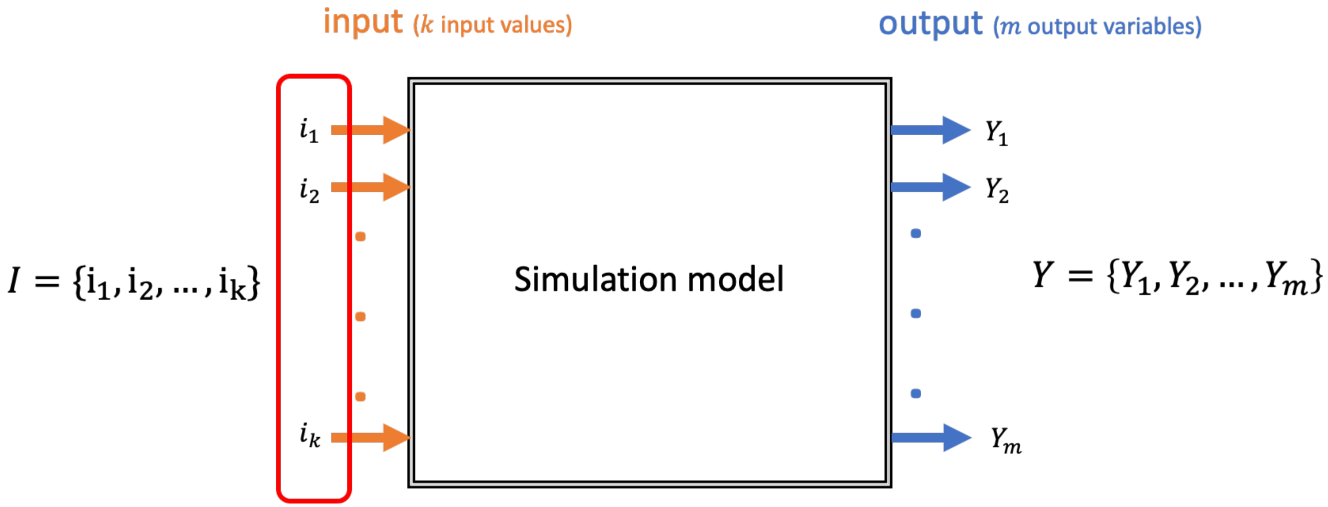


Model parameters



- These model parameter values (i.e., input values, I) are **critical**.
- If the values are not accurate, we cannot rely on simulation results.
  - Garbage in, garbage out

Model parameters



- In some cases, data for model parameter values are **directly available**. E.g.:
  - Population size in a social model
  - Number of cashiers in a specific grocery store model
  - Part assembling time in an assembly line model
  - Mass of a planet in a solar system model
  - …

Model parameters

- In some cases, data for model parameter values can be obtained using **secondary sources** if cannot be directly found.

Model parameters

- In some cases, data for model parameter values can be obtained using **secondary sources** if cannot be directly found.

Active data sources

- Surveys from a similar population
  - E.g.: a behavioral measure for a topic can be obtained from another geographic area with similar population characteristics.
- Crowd sourcing
  - E.g.: participants respond to decision-based questions in a smartphone app.
  - E.g.: participants use platforms like Amazon Mechanical Turk.

Model parameters

- In some cases, data for model parameter values can be obtained using **secondary sources** if cannot be directly found.

Active data sources

- Surveys from a similar population
  - E.g.: a behavioral measure for a topic can be obtained from another geographic area with similar population characteristics.
- Crowd sourcing
  - E.g.: participants respond to decision-based questions in a smartphone app.
  - E.g.: participants use platforms like Amazon Mechanical Turk.

Passive data sources

- Social media:
  - E.g.: Twitter (Padilla et al. 2016)
- Cell phone:
  - E.g.: call detail records characterizing movement patterns
  - E.g.: trip data providing traffic status (Uber Movement, etc.).
- Wi-Fi connections:
  - E.g.: Safegraph foot traffic data

Padilla, J. J., Diallo, S. Y., Kavak, et al. (2016). Semi-automated initialization of simulations: an application to healthcare. The Journal of Defense Modeling and Simulation, 13(2), 171-182.

Model parameters

- In other cases, data for model parameter values can be **unfeasible**, **unethical**, or **too costly** to obtain directly or via secondary sources.

Model parameters

- In other cases, data for model parameter values can be **unfeasible**, **unethical**, or **too costly to obtain** directly or via secondary sources.
- Examples
  - Constants that vary based on different environmental factors.
    - E.g.: *growth rates* in a predator-pray model
  - Behavior variables
    - E.g.: *driver's reaction time* and *speed acceptance* in a transportation model.
      - Ciuffo, B., Punzo, V., & Torrieri, V. (2008). Comparison of simulation-based and model-based calibrations of traffic-flow microsimulation models. *Transportation Research Record*, *2088*(1), 36-44.
  - Unobservable parameters
    - E.g.: *breast cancer natural history* parameters in a cancer epidemiology model.
      - Cevik, M., Ergun, M. A., Stout, N. K., Trentham-Dietz, A., Craven, M., & Alagoz, O. (2016). Using active learning for speeding up calibration in simulation models. *Medical Decision Making*, *36*(5), 581-593.

Simulation model calibration

- “Calibration is the activity of adjusting the unknown rate parameters until the outputs of the model fit the observed data” (Kennedy and O'Hagan, 2001).

Simulation model calibration

- "Calibration is the activity of adjusting the unknown rate parameters until the outputs of the model fit the observed data" (Kennedy and O'Hagan, 2001).
- Certain application domains use calibration more than others. What are the commonalities between these domains?
    - Traffic models
    - Economic models
    - Policy models

Simulation model calibration: preparation

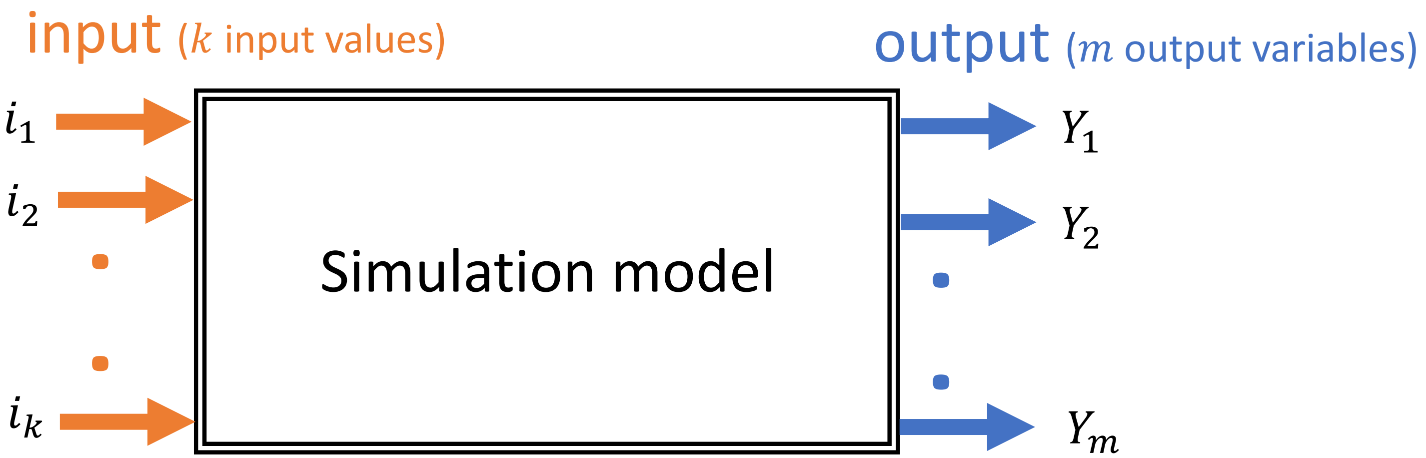
Preparation step

1) Identify a set of input parameters w/ unknown values (I_c).

2) Identify a set of output variables of interest (Y_c).

3) Get (real-world) data to measure the selected output variables ($Y_c^{(real)}$).

4) Set initial values for I_c.



Calibration in detail

Calibration step

1) Run the model and collect output data (Y_c).

2) Compare output data (Y_c) against real-world data ($Y_c^{(real)}$)

3) Is difference low enough?

No

Adjust parameter values (I_c)

Yes

Store (I_c)

End

Check the goodness of fit here

- Squared error
- Mean error
- Mean normalized error
- Mean absolute error
- Root mean square error
- …

Did we reach a reasonably low value?

Change the values in a way that it can improve the fit.

These three steps can be formulated as an optimization problem in which the objective function is to minimize the difference between Y_c and $Y_c^{(real)}$.

- Heuristics
- Dynamic programming
- …

Calibration example: projectile motion

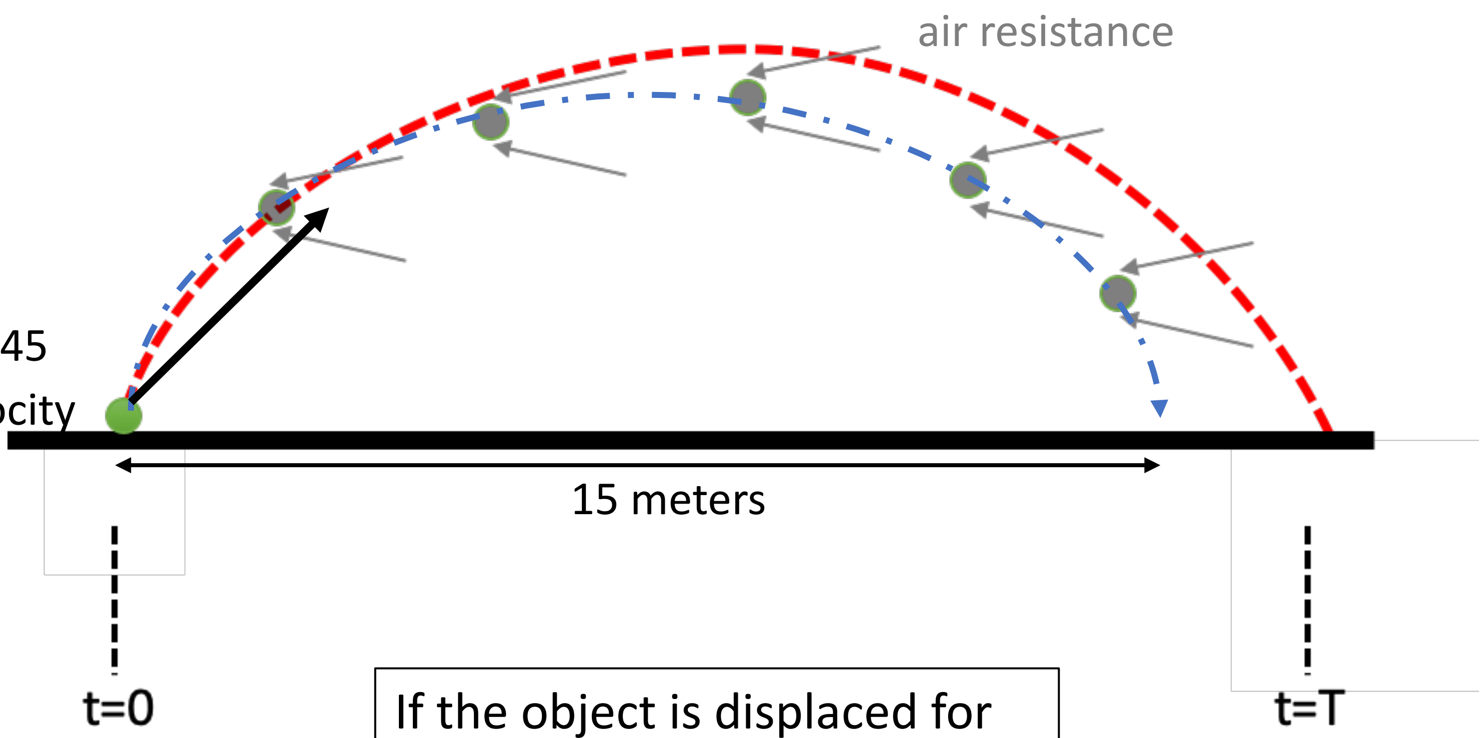
Parameters

$m = 1$

$g = 9.81$

$v_h = 10, v_v = 10$

The object is thrown at 45 degrees with $10\sqrt{2}$ velocity



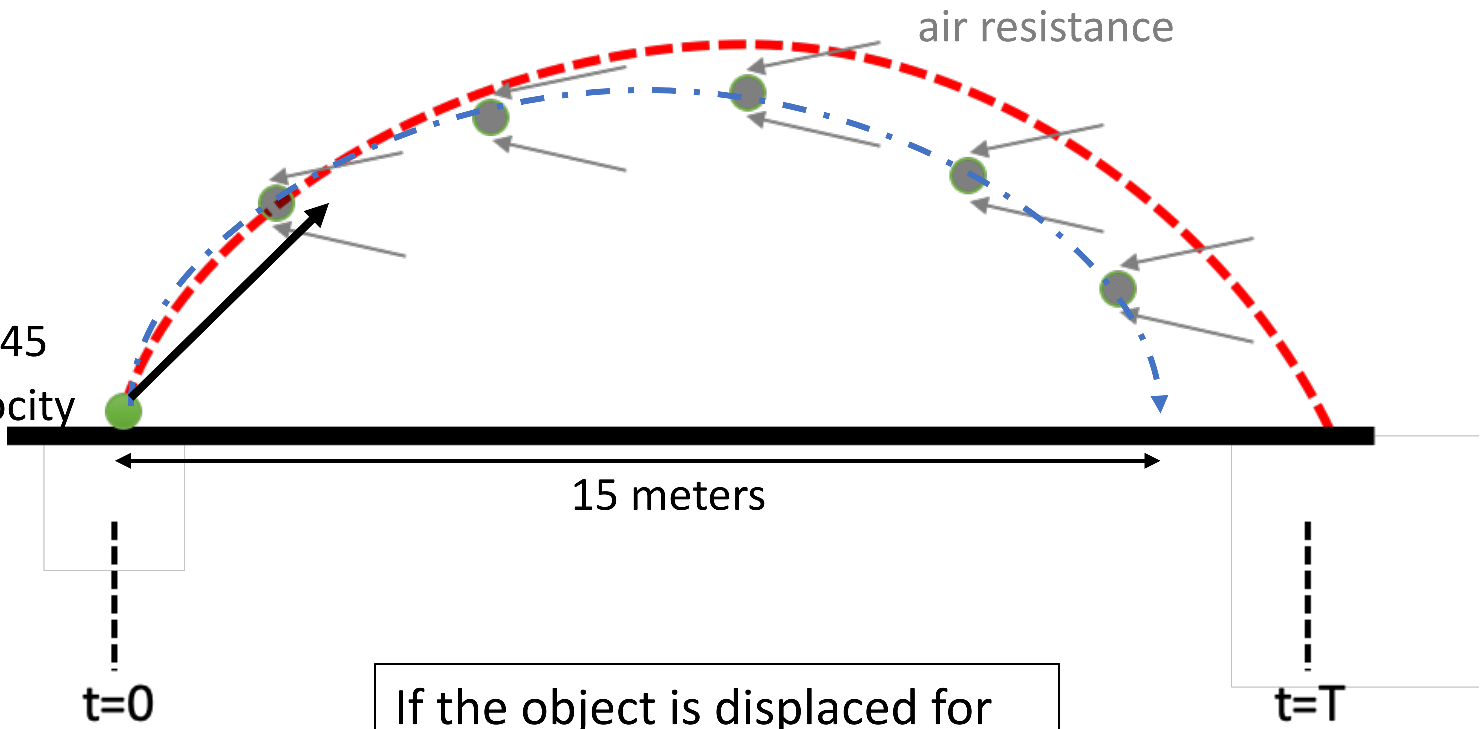
If the object is displaced for 15 meters when touches the ground, calibrate the model to find air resistance (k).

Calibration example: projectile motion

Parameters

$$m = 1$$
$$g = 9.81$$
$$v_h = 10, v_v = 10$$

The object is thrown at 45 degrees with $10\sqrt{2}$ velocity



If the object is displaced for 15 meters when touches the ground, calibrate the model to find air resistance (k).

Variables for projectile motion model

- System states
  - Position (x, y)
  - Speed (horizontal speed, vertical speed)
- Variables
  - Mass
  - Air resistance coefficient
  - Gravity

Dynamics for projectile motion

- $u = (x, y)$ - two dimensional position of our mass (m).
- Our mass is under the influence of two forces
  - Gravity mg or the vector of $(0, -9.81m)$.
  - Air resistance $F = -ku$'.
- Newton's Second Law of Motion (F=ma)
  - $F + mg = m \cdot u''$ can be rearranged by plugging $F = -ku'$ and leaving u'' alone.
    - $u'' = -\frac{k}{m}u' + g$ (second order ODE).
- Transform to a single-order ODE by plugging $\mathrm{v} = (u, u')$ thus $v' = (u', u'')$
  - $v' = (u', u'') = (u', -\frac{k}{m}u' + g)$ which can be represented as a function of v.

Parameters

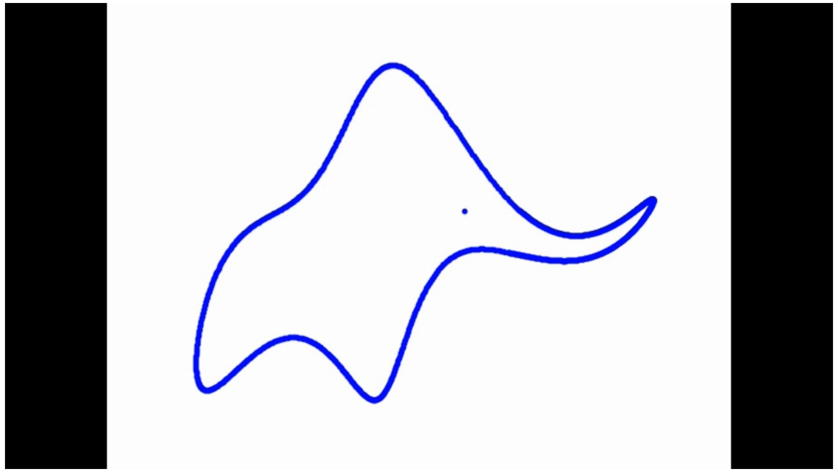
- Initial position (x, y): $(0, 0)$
- Final position: (15,0) – assume we collected this from the real-world
- Velocity $(horizontal, vertical)$: $(10, 10)$
- Parameter to be calibrated k which is set 0.1 initially.

- Goal find k which can predict the 15-meter disposition properly.

Calibration challenges

- Too many input parameters -> large search space
- Too many variables in the objective function
- Overfit (calibrating unrelated parameters)

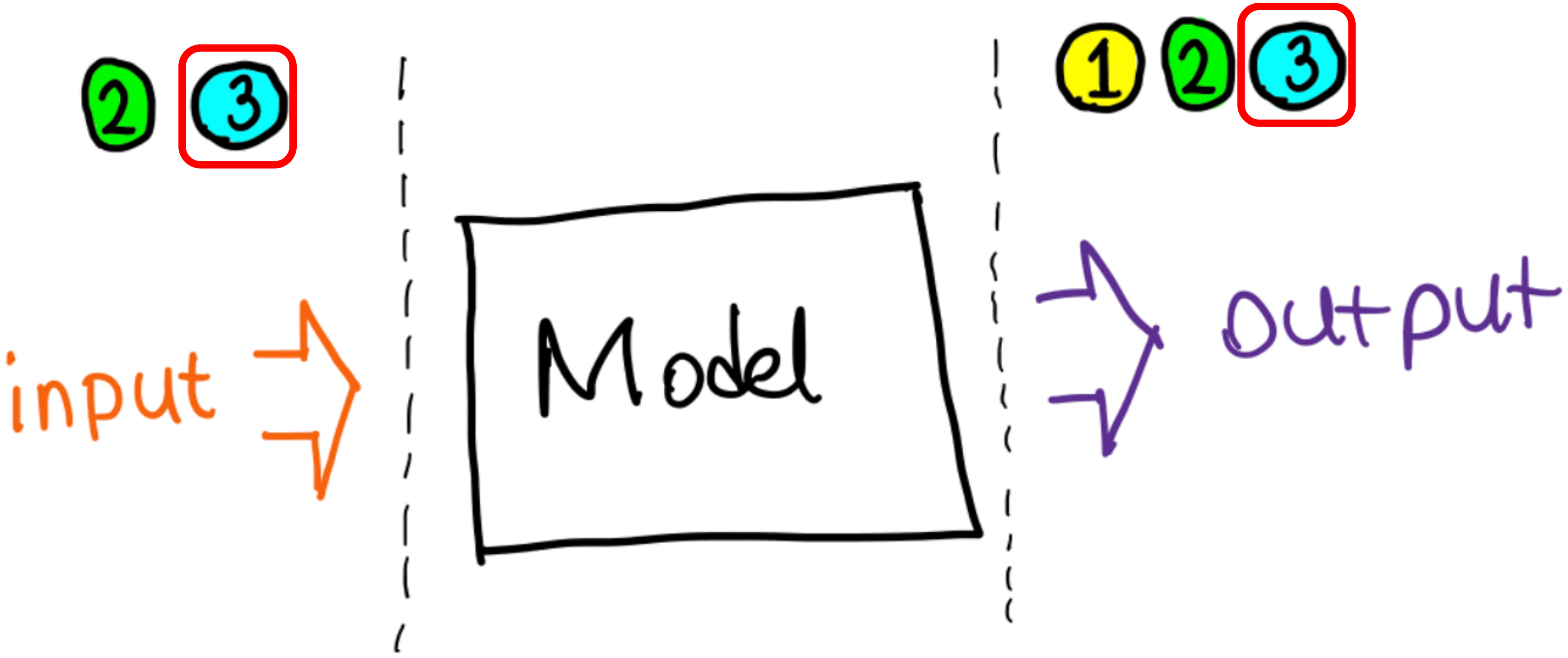
recall
"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."
– John von Neumann
check https://youtu.be/CVDYh5liQkA



1 Non-terminating simulations

2 Model parameter value estimation

3 Model sensitivity



Model sensitivity

- Ideally, models are tolerant to small changes in input parameter values.
- Tolerance to parameter value changes shows how robust the simulation model is.
- We study such tolerance cases using **sensitivity analysis**.

Hypothetical equation model

- Write a Python code for the following equation
  - $x_{n+1} = r\, x_n\, (1 - x_n)$
- Start with r=2, and $x_0 = 0.1$
- Compute for 100 iterations.
- Plot the results.
- What value does it settle on?

Hypothetical equation model

$$x_{n+1} = r\, x_n\, (1 - x_n)$$

- If the value of r is 2.5 what is the final value of the graph?
- If the value of r is 2.8 what is the final value of the graph?
- Before trying it, predict the value for r = 3.
- r=3.3?
- r=3.5?
- r=4.0?

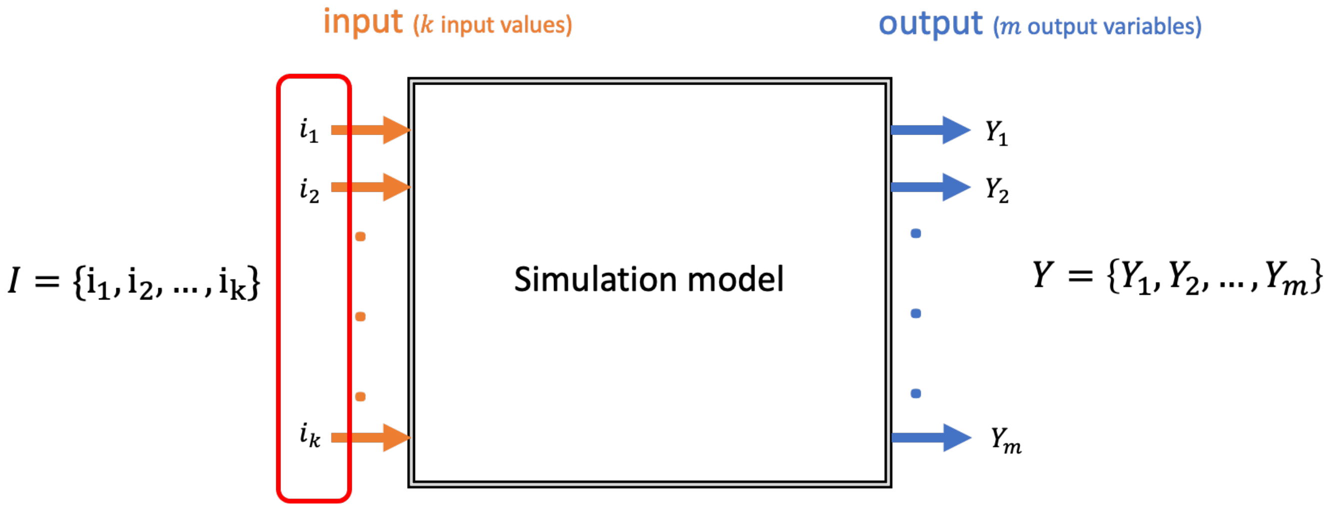
Sensitivity analysis

- "Sensitivity analysis is a validation method that compares magnitude and variability in simuland behavior to magnitude and variability in the model results" (Sokolowski & Banks, 2010).
- Thus, the challenge is to **systematically** vary input parameter values to observe output values.
  - Input-output relationship can be tested using "partial correlation" etc.
- If the simulation model is stochastic, we need to run multiple times to observe changes in the aggregated behavior (e.g., average).

How to vary input values?

- If you have one input to vary, it is a relatively simple problem.
  - Keep everything constant, sample from the min-max range of values
- If you have multiple inputs to vary
  1. Systematically
     - Design of experiments
       - Full factorial design
       - Selecting certain combinations. E.g., Latin Hypercube Sampling
  2. Randomized

Full Factorial Design



- Captures all possible combinations.
- If all k model parameters are binary, we will have 2^k combinations to run.
- If all model parameters are continuous, we will first need to identify discrete points within each range and then run all possible combinations.

Latin Hypercube Sampling (LHS)

- "is a way to sample the space of all combinations of the input variables" (Collins).
- Reduces the number of runs to a reasonable number.
- Your results should approximate to the case should you sampled the whole space.

LHS

- Step 1: Identify number of runs (n).
- Step 2: For each of k input variables
  - Identify range of possible values
  - Divide the range into n equally-probable bins.
  - Choose a value randomly from each bin.
- Step 3: Randomly combine these values from each bin w/out replacement. Meaning, make n combinations.
- Step 4: Run the model with all n combinations.
  - If the model is stochastic, you may want to run each combination multiple times.

LHS bin combinations

- $n = 2$

- $n = 3$

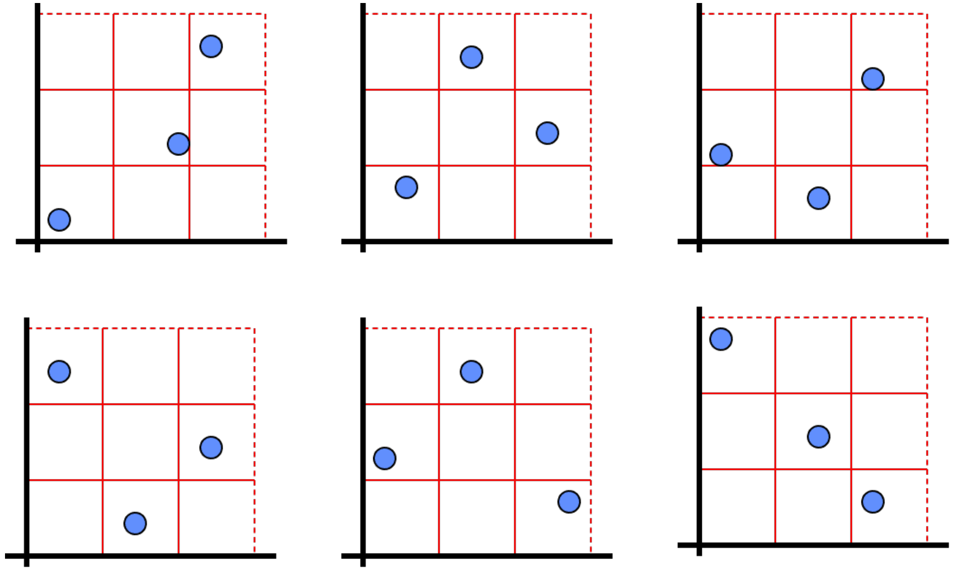
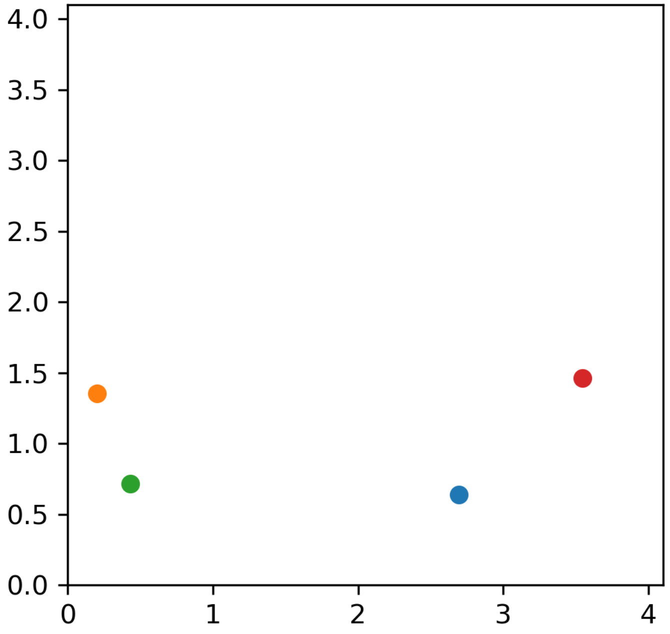
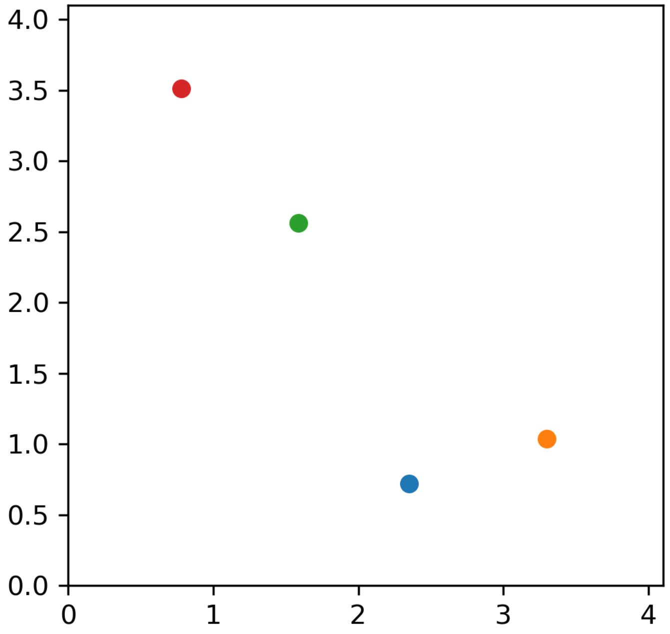


Figure source: Collins

LHS vs. random sampling

Value ranges: [0,4]
n = 4





Can you guess which one is which?

How many runs are needed for stochastic models per parameter configuration?

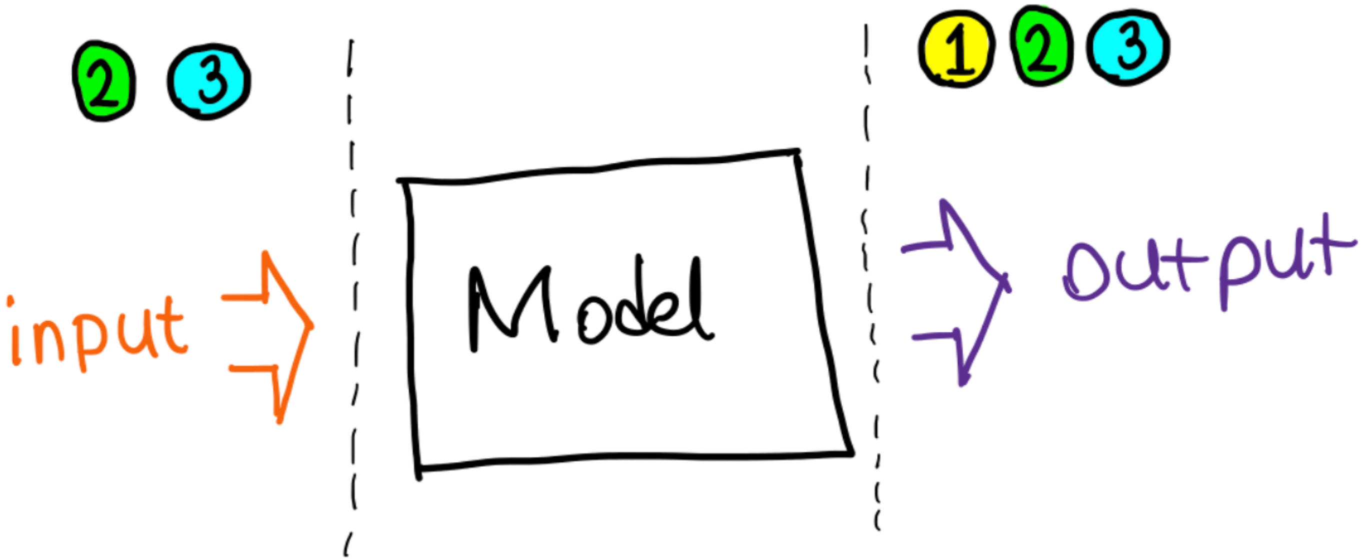
- There is no magical formula.
- Identify output variables of interest.
- Vary number of runs from lower to higher numbers
  - (e.g., [10, 100, 1000, 10k])
- Capture the variation in output variables.
- If two consecutive runs (e.g., 100 vs. 1000) have very similar variation, choose the lower number of runs (i.e., 100).

Sensitivity analysis challenges

- Large parameter space, leading to huge computational cost.
  - Imagine 10 input variables with 10 possible values each.
    - 10^{10} = 10,000,000,000 = 10 billion runs.
    - What if the variables are continuous.
  - Need ways to reduce computational time, especially for large-scale ABMs.
  - Stochastic simulation needs more runs.
- What is an ideal step-size?
  - What if a small change leads to a chaotic behavior but sensitivity analysis cannot capture it?

Wrap-up and questions

1. Non-terminating simulations
2. Model parameter value estimation
3. Model sensitivity



Sources

- Collins, A. MSIM 651 Lecture Slides. Old Dominion University.
- Conway, R. W. (1963). Some tactical problems in digital simulation. *Management science*, *10*(1), 47-61.
- Gafarian, A. V., Ancker Jr, C. J., & Morisaku, T. (1978). Evaluation of commonly used rules for detecting "steady state" in computer simulation. *Naval Research Logistics Quarterly*, *25*(3), 511-529.
- Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(3), 425-464.
- Law, A. M. (1990). *Design and analysis of simulation experiments for manufacturing applications*. Institute of Electrical and Electronics Engineers (IEEE).
- Mahajan, P. S., & Ingalls, R. G. (2004, December). Evaluation of methods used to detect warm-up period in steady state simulation. In *Proceedings of the 2004 Winter Simulation Conference, 2004.* (Vol. 1). IEEE.
- Mielke, R. MSIM 601 Lecture Slides. Old Dominion University.
- Nicol, D. M., Banks, J., Carson, J. S., Nelson, B. L. (2013). Discrete-Event System Simulation: Pearson New International Edition. United Kingdom: Pearson Education Limited. Padilla, J. J., Diallo, S. Y., Kavak, et al. (2016). Semi-automated initialization of simulations: an application to healthcare. The Journal of Defense Modeling and Simulation, 13(2), 171-182.
- Padilla, J. J., Diallo, S. Y., Kavak, et al. (2016). Semi-automated initialization of simulations: an application to healthcare. The Journal of Defense Modeling and Simulation, 13(2), 171-182.
- Sokolowski, J. A., & Banks, C. M. (2010). *Modeling and simulation fundamentals: theoretical underpinnings and practical domains*. John Wiley & Sons.