Introduction to Simulation

Reading: Law, Sections 1.1, 1.2, 1.8

Peter J. Haas

CS 590M: Simulation
Spring Semester 2020

How Can Computers Help Us Make Better Decisions Under Uncertainty?

A Gambling Game

THHHH

Is the following game a good bet over the long run?

- A fair coin is repeatedly flipped until $|\#\text{heads} - \#\text{tails}| = 3$
- Player receives \$8.99 at the end of the game but must pay \$1 for each coin flip

Approaches to answering the question:

- Try to compute the answer analytically (not easy)
- Play the game multiple times and use average reward to estimate expected reward (time-consuming)
- Use the power of the computer to experiment—Simulation!

Simulating the Gambling Game and Birds

Simulating coin flips on a computer: Pseudorandom numbers

- U "looks like" a uniform random number between 0 and 1
- To generate:
  - Python: `U = random.random()`
  - C: `U = (float)rand() / MAX_RAND`
  - Java: `U = Math.random()`
- Then "heads" if $0 \leq U \leq 0.5$ and "tails" if $0.5 < U \leq 1$

The need for careful simulation [Demo]

Simulation for science [NetLogo Demo]

Simulation: Definitions

Definition 1

A technique for studying real-world dynamical systems by imitating their behavior using a mathematical model of the system implemented on a digital computer

Definition 2

A controlled statistical sampling technique for stochastic systems

Q: Example of non-stochastic simulation?

Definition 3

A numerical technique for solving complicated probability models (analogous to numerical integration)

Monte Carlo methods

For static numerical problems

Example: Numerical integration with many dimensions

- WWII Manhattan Project: von Neumann, Teller, Turing

Will cover briefly in the course and homework

More on Simulation

Why simulation is awesome (mostly)

- Most frequently used tool of practitioners
- Interdisciplinary: spans Computer Science, Statistics, Probability, and Number Theory

Applications

traffic
financial risk
video games
disease modeling

biology (e.g. protein folding)
AI training
flight simulation
astronomy

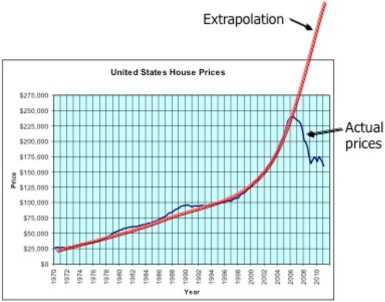
safety
military
business
telecom
healthcare

Advantages and disadvantages

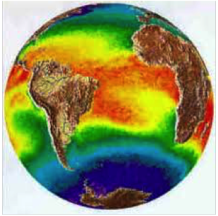
+ cheaper, faster, safer than dealing with real-world sys.
+ allows arbitrary model complexity
+ allows what-if analysis
+ can validate simpler models (analytic or simulation)

- only gives approximate answers
- can be expensive to create + costly to run (esp. if model is huge)
- need deep system knowledge
- subject to numerical issues

Simulation vs Machine Learning



Extrapolation of 1970-2006 median U.S. housing prices

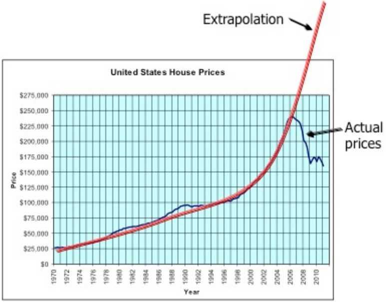


NCAR Community Atmosphere Model (CAM)

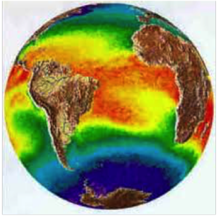
3.3 Eulerian Dynamical Core

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= \boldsymbol{k} \cdot \nabla \times (\boldsymbol{n}/\cos\phi) + F_{\zeta_H}, \\
\frac{\partial \delta}{\partial t} &= \nabla \cdot (\boldsymbol{n}/\cos\phi) - \nabla^2 (E + \Phi) + F_{\delta_H}, \\
\frac{\partial T}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial \lambda}(UT) + \cos\phi \frac{\partial}{\partial \phi}(VT)\right] + T\delta - \dot{\eta}\frac{\partial T}{\partial \eta} + \frac{R}{c_p^*} T_v \frac{\omega}{p} \\
&\quad + Q + F_{T_H} + F_{F_H}, \\
\frac{\partial q}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial \lambda}(Uq) + \cos\phi \frac{\partial}{\partial \phi}(Vq)\right] + q\delta - \dot{\eta}\frac{\partial q}{\partial \eta} + S, \\
\frac{\partial \pi}{\partial t} &= \int_1^{\eta_t} \boldsymbol{\nabla} \cdot \left(\frac{\partial p}{\partial \eta} \boldsymbol{V}\right) d\eta.
\end{aligned}
$$

Simulation vs Machine Learning



Extrapolation of 1970-2006 median U.S. housing prices



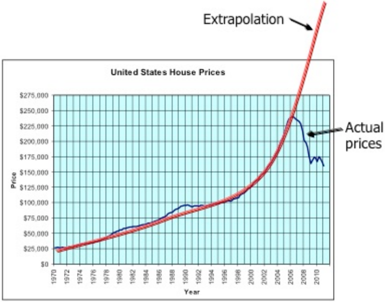
NCAR Community Atmosphere Model (CAM)

3.3 Eulerian Dynamical Core

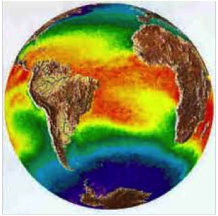
$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= \boldsymbol{k} \cdot \nabla \times (\boldsymbol{n}/\cos\phi) + F_{\zeta_H}, \\
\frac{\partial \delta}{\partial t} &= \nabla \cdot (\boldsymbol{n}/\cos\phi) - \nabla^2 (E + \Phi) + F_{\delta_H}, \\
\frac{\partial T}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial\lambda}(UT) + \cos\phi\frac{\partial}{\partial\phi}(VT)\right] + T\delta - \dot{\eta}\frac{\partial T}{\partial\eta} + \frac{R}{c_p^*}T_v\frac{\omega}{p} \\
&\quad + Q + F_{T_H} + F_{F_H}, \\
\frac{\partial q}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial\lambda}(Uq) + \cos\phi\frac{\partial}{\partial\phi}(Vq)\right] + q\delta - \dot{\eta}\frac{\partial q}{\partial\eta} + S, \\
\frac{\partial \pi}{\partial t} &= \int_1^{\eta_t} \boldsymbol{\nabla} \cdot \left(\frac{\partial p}{\partial\eta}\boldsymbol{V}\right) d\eta.
\end{aligned}
$$

Will the mechanism that generates data now generate it in the future?

Simulation vs Machine Learning



Extrapolation of 1970-2006
median U.S. housing prices



NCAR Community
Atmosphere Model (CAM)

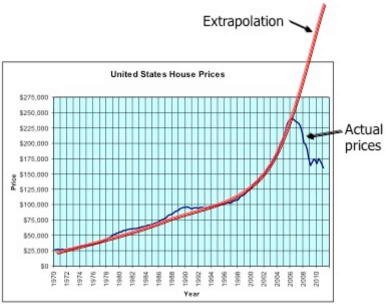
3.3 Eulerian Dynamical Core

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= \boldsymbol{k}\cdot\nabla\times(\boldsymbol{n}/\cos\phi) + F_{\zeta_H},\\
\frac{\partial \delta}{\partial t} &= \nabla\cdot(\boldsymbol{n}/\cos\phi) - \nabla^2(E+\Phi) + F_{\delta_H},\\
\frac{\partial T}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial\lambda}(UT) + \cos\phi\frac{\partial}{\partial\phi}(VT)\right] + T\delta - \dot{\eta}\frac{\partial T}{\partial\eta} + \frac{R}{c_p^*}T_v\frac{\omega}{p}\\
&\quad + Q + F_{T_H} + F_{F_H},\\
\frac{\partial q}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial\lambda}(Uq) + \cos\phi\frac{\partial}{\partial\phi}(Vq)\right] + q\delta - \dot{\eta}\frac{\partial q}{\partial\eta} + S,\\
\frac{\partial \pi}{\partial t} &= \int_1^{\eta_t}\boldsymbol{\nabla}\cdot\left(\frac{\partial p}{\partial\eta}\boldsymbol{V}\right)d\eta.
\end{aligned}
$$

Will the mechanism that generates data now generate it in the future?

(Not if I change the mechanism)

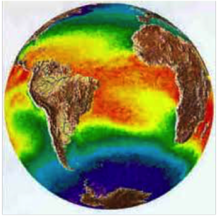
Simulation vs Machine Learning



Extrapolation of 1970-2006
median U.S. housing prices

Will the mechanism that generates data now generate it in the future?

(Not if I change the mechanism)



NCAR Community
Atmosphere Model (CAM)

3.3 Eulerian Dynamical Core

$$\frac{\partial \zeta}{\partial t} = \boldsymbol{k} \cdot \nabla \times (\boldsymbol{n}/\cos\phi) + F_{\zeta_H},$$

$$\frac{\partial \delta}{\partial t} = \nabla \cdot (\boldsymbol{n}/\cos\phi) - \nabla^2 (E + \Phi) + F_{\delta_H},$$

$$\frac{\partial T}{\partial t} = \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial\lambda}(UT) + \cos\phi\frac{\partial}{\partial\phi}(VT)\right] + T\delta - \dot{\eta}\frac{\partial T}{\partial\eta} + \frac{R}{c_p^*}T_v\frac{\omega}{p} + Q + F_{T_H} + F_{F_H},$$

$$\frac{\partial q}{\partial t} = \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial\lambda}(Uq) + \cos\phi\frac{\partial}{\partial\phi}(Vq)\right] + q\delta - \dot{\eta}\frac{\partial q}{\partial\eta} + S,$$

$$\frac{\partial \pi}{\partial t} = \int_1^{\eta_t} \boldsymbol{\nabla} \cdot \left(\frac{\partial p}{\partial \eta}\boldsymbol{V}\right) d\eta.$$

Allows What-If analyses

Simulation Resources

- TOMACS: ACM Transactions on Modeling and Computer Simulation
- OR/MS Today (biennial simulation software survey)
- INFORMS Simulation Society; see www.informs.org/Community/Simulation-Society
- Winter Simulation Conference proceedings; see http://informs-sim.org
  - Over 40 years of conference papers searchable by keyword
  - Introductory and advanced tutorials can be especially useful
- Society for Computer Simulation; see http://www.scs.org.
- ACM SIGSIM; see www.sigsim.org

See Sokolowski and Banks (Ch. 7) for extensive listing of simulation organizations and applications

Overview of Simulation Process

+

Real-world system
(existing or proposed)

Decision problem
(Choose design or operating policy)

modeling

Mathematical simulation model

states, events, clocks
state transitions

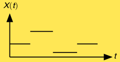
Input distributions
- Probability theory
- Fit distribution from data
  (maximum likelihood, Bayes)

$\{X(t) : t \geq 0\}$ or $\{X_n : n \geq 0\}$

Discrete-time Markov chain (DTMC)
Continuous-time Markov chain (CTMC)
Semi-Markov process (SMP)
Generalized semi-Markov process (GSMP)

Stochastic process definition

Sample path generation



Uniform random numbers
Non-uniform random numbers
- Inversion, accept-reject,
  composition, convolution,
  alias method

Time-advance mechanism
Event list management

Point estimates and confidence intervals
- Simple means (SLLN and CLT based)
- Nonlinear functions of means, quantiles
  (Taylor series, sectioning, jackknife, bootstrap)
- Steady-state quantities: time-avg limits, delays
  (regenerative, batch means, jackknifing)

Efficiency improvement
- Common random numbers, antithetic variates,
  conditional Monte Carlo, control variates,
  importance sampling

Experimental design
- Factor screening
- Sensitivity analysis
- Metamodeling

Optimization
- Continuous (Robbins -Monro)
- Ranking and selection
-Discrete optimization

Output analysis

Key Issues in Simulation

1. **What questions are we trying to answer?**
   - Complex, often dynamic
     (see Sawyer and Fuqua slides in Practitioner's Gallery)
   - Identify stakeholders and available resources
   - Continual interplay with stakeholders during project
   - See also Conway & McClain
     http://pubsonline.informs.org/doi/pdf/10.1287/ited.3.3.13
2. **How to model the system?**
   - State definition, random variables, etc.
   - Operational vs policy models: different levels of detail
   - "As simple as possible" vs model re-use

Example of Model Formulation: Gambling game

Outcome of ith toss: $H_i = \begin{cases} 1 & \text{if } U_i \le 0.5; \\ 0 & \text{if } U_i > 0.5 \end{cases}$ heads

# of heads in first n tosses: $S_n = \sum_{i=1}^{n} H_i$

# of tails in first n tosses: $n - \sum_{i=1}^{n} H_i$

heads - #tails: $2\sum_{i=1}^{n} H_i - n$

length of game: $L = \min\{n \ge 1 : |2\sum_{i=1}^{n} H_i - n| = 3\}$

reward for game: $X = 8.99 - L$

Goal: estimate $\mu = E[X]$

Key Issues, Continued

3. Is the quantity that we are trying to estimate well defined?

- Single-server queue with $\rho > 1$
- In gambling game, μ defined iff $P(L < \infty) = 1$ and $E[L] < \infty$
- Moral: do sanity checks!

4. How to generate run on a computer?

- Gambling game is easy, industrial strength models are hard
- In general, we will use low-level languages
  - Python, C/C++, Java versus Matlab, R
  - For deep understanding of foundational principles
  - Flexibility, low cost, fast execution
  - Programming ability strengthens your resume

Key Issues, Continued

5. How do we verify the simulation?

- Verification: Correctness of the computer implementation of the simulation model
- Good coding practices:
  - make debugging easy (e.g. use print statements)
  - write modular code (and unit-test it)
  - Lots of comments
  - Avoid too many global variables

Key Issues, Continued

6. How do we validate the simulation?

- Validation: Adequacy of the simulation model in capturing system of interest
- Beware of over-fitting: use, e.g., cross validation [Hastie et al., *Elements of Statistical Learning*, Sec. 7.10]
- Beware that good fit to current data $\not\Rightarrow$ good extrapolation
- Aim for *insights*: trends and comparisions
- Use sensitivity analysis to build credibility

Key Issues, Continued

7. **Number and length of simulation runs?**

8. **Can the simulation be made more efficient?**
   - Statistical and computational efficiency

9. **How do we use simulation to make decisions?**
   - Compare systems: ranking and selection
   - Set operating or design parameters: stochastic optimization
   - Set operating policies: reinforcement learning, Markov decision processes

Point Estimates & Strong Law of Large Numbers

Estimating expected reward in gambling game

- Replicate experiment (i.e., play game) n times to get $X_1, X_2, \ldots, X_n$
- Estimate expected reward by $\mu_n = \frac{1}{n} \sum_{i=1}^{n} X_i$
- Why is this a reasonable estimate?

Strong law of large numbers

- Suppose $X_1, X_2, \ldots$ are i.i.d. with finite mean μ
- Then, with probability 1,

$$\frac{1}{n} \sum_{i=1}^{n} X_i \to \mu \text{ as } n \to \infty$$

Confidence Intervals & Central Limit Theorem

How do we assess the error in our estimate?

- Need to distinguish true system differences from random fluctuations $\frac{\sqrt{n}}{\sigma}(\mu_n - \mu) \overset{D}{\approx} N(0,1) \rightarrow \mu_n - \mu \overset{D}{\approx} N(0, \frac{\sigma^2}{n})$

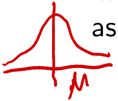
Central Limit Theorem $\rightarrow \mu_n \overset{D}{\approx} N(\mu, \frac{\sigma^2}{n})$

- Spose $X_1, X_2, \ldots$ are i.i.d., mean $\mu < \infty$ and variance $\sigma^2 < \infty$
- Then

$$\frac{\sqrt{n}}{\sigma}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) \Rightarrow N(0,1)$$

as $n \to \infty$, where $N(0,1)$ is a standard normal random variable and $\Rightarrow$ denotes convergence in distribution

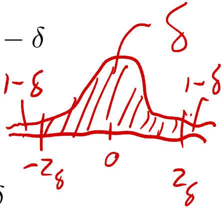
- Intuitively, the sample average μ_n is approximately distributed as $N(\mu, \sigma^2/n)$ when n is large (≥ 50)



Confidence Interval for Fixed Sample Size

To compute $100(1-\delta)\%$ **confidence interval:**

- Choose z_δ such that $P(-z_\delta \le N(0,1) \le z_\delta) = 1 - \delta$
  - Equivalently, $P(N(0,1) \le z_\delta) = 1 - \delta/2$
  - Can find in Table T1 (p. 716) in the textbook



- By CLT,

$$P\left\{-z_\delta \le \frac{\sqrt{n}\,(\mu_n - \mu)}{\sigma} \le z_\delta\right\} \approx 1 - \delta$$

or, after algebra,

$$P\left\{\mu_n - \frac{z_\delta \sigma}{\sqrt{n}} \le \quad \mu \quad \le \mu_n + \frac{z_\delta \sigma}{\sqrt{n}}\right\} \approx 1 - \delta$$

so random interval

$$\left[\mu_n - \frac{z_\delta \sigma}{\sqrt{n}}, \quad \mu_n + \frac{z_\delta \sigma}{\sqrt{n}}\right]$$

covers true value with probability $\approx 1 - \delta$

CI for Fixed Sample Size, Continued

$Var[X] = E[(X-\mu)^2]$

Problem: σ^2 is unknown

- Solution: Estimate σ^2 from data: $s_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mu_n)^2$

ensures that estimator is unbiased: $E[s_n^2] = \sigma^2$

Final $100(1-\delta)\%$ CI formula:

$$\left[\mu_n - \frac{z_\delta s_n}{\sqrt{n}}, \quad \mu_n + \frac{z_\delta s_n}{\sqrt{n}}\right]$$

The quantity $z_\delta s_n/\sqrt{n}$ is called the half-width of the CI

Questions:

- How, roughly, do I cut my error in half?
- What can go wrong if n is too small?

Choosing the Number of Simulation Runs

$\frac{\hat{s} z_\delta}{\sqrt{n}}$ $\frac{\hat{s} z_\delta}{\sqrt{n}}$

ϵ μ ϵ

Trial runs

- Generate $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_k$ (where $k \geq 50$)
- Compute $\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} \hat{X}_i$ and $\hat{s}^2 = \frac{1}{k-1} \sum_{i=1}^{k} \left(\hat{X}_i - \hat{\mu} \right)^2$
- Absolute precision intervals
  - Estimate μ to within $\pm\varepsilon$ with probabilty $100(1-\delta)\%$
  - Want to choose n so that $\frac{\sigma z_\delta}{\sqrt{n}} = \varepsilon$: $n = \frac{\hat{s}^2 {z_\delta}^2}{\varepsilon^2}$
- Relative precision intervals
  - Estimate μ to within $\pm 100\varepsilon\%$ with probabilty $100(1-\delta)\%$
  - Want to choose n so that $\frac{\sigma z_\delta}{\sqrt{n}} = \varepsilon\mu$: $n = \frac{\hat{s}^2 {z_\delta}^2}{\varepsilon^2 \hat{\mu}^2}$

Sequential estimation

- Simulate until interval is narrow enough
- Asymp. valid as $\varepsilon \to 0$ [Nadas, *Ann. Math Statist.*,1969]
- Danger: premature stopping

Numerical Issues: Computing the Sample Variance

The problem

- Sum and average : $S_n = x_1 + x_2 + \cdots + x_n$ and $\bar{X}_n = S_n/n$
- Goal: compute sample variance $V_n = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{X}_n\right)^2$

Two-pass method

- Compute $\bar{X}_n$ in first pass, V_n in second pass

Calculator method

- Based on fact that $\text{Var}[X] = E[X^2] - E^2[X]$
- Question: What can go wrong?

Numerically stable one-pass method

- Set $V_1 = 0$ and, for $k \geq 2$,

$$(k-1)V_k = (k-2)V_{k-1} + \left(\frac{S_{k-1} - (k-1)x_k}{k}\right)\left(\frac{S_{k-1} - (k-1)x_k}{k-1}\right)$$

Introduction to Simulation

More Complicated Systems: Discrete-Event Simulations

Discrete-event stochastic systems

- Make stochastic state transitions when events occur
- Events occur at a strictly increasing sequence of random times
- The main focus of the course

The naive approach

1. Learn about (proposed or existing) real world system
2. Write a complicated program
3. Run the program and generate reams of output
4. Return summary statistics
   (often without estimates of precision)

Discrete-Event Simulations, Continued

The modern (stochastic process) approach

1. Learn about real world system and questions of interest
2. Develop conceptual simulation model (system elements, random variables)
   - Input distributions based on theory and data fitting
   - Sim. models also called "stochastic" or "probability" models
3. Define "state of the system at time t", e.g. $X(t)$, or "state of the system at the nth observation", e.g. X_n
   - Should be as simple as possible for efficiency reasons
   - Must contain enough info to estimate characteristics of interest
   - Must permit simulation of system
   - Sometimes task can be eased via modeling frameworks: networks of queues, stochastic Petri nets, etc.
4. Specify the underlying stochastic process $\{ X(t) : t \geq 0 \}$ or $\{ X_n : n \geq 0 \}$

Discrete-Event Simulations, Continued

5. Define system characteristics of interest in terms of underlying stochastic process
   - Ex: Suppose

$$X(t) = \begin{cases} 1 & \text{If machine operational at time } t; \\ 0 & \text{otherwise} \end{cases}$$

and $X(t) \Rightarrow X$

"Long-run frac. of time that machine operational" $=$

$\lim_{t \to \infty} \frac{1}{t} \int_0^t X(u)\,du$

"Steady-state prob. that machine is operational" $=$

$P(X=1) = E[X]$ $\quad E[X] = 1 \cdot P(X=1) + 0 \cdot P(X=0)$

$= P(X=1)$

- Show perf. meas. is well-defined via stochastic process theory

Discrete-Event Simulations, Continued

6. Use computer to generate sample paths (realizations) of underlying stochastic process
   - Generation of random numbers is essential
7. Compute estimates of system characteristics (and assessments of precision)
   - Via limit theorems for stochastic processes (SLLN and CLT)
8. Use well-founded statistical procedures for comparing alternative system designs, optimizing system parameters, etc.

Why Program from Scratch?

1. Simulation packages come and go
2. Simulation packages can fool you with fancy animations
3. Want deep understanding of underlying concepts, algorithms, statistical, and implementation issues
4. Concepts apply beyond simulation
5. A package won't always do what you want (so need to hack)
6. Packages can be expensive (Python is free)
7. Python ties in with other ML tools (& good for your resume)
8. Custom programing can give faster execution speeds

Course Goals

- Understand the basic principles and methods of Monte Carlo and discrete-event simulation
- Gain familiarity with the most commonly used stochastic models for discrete-event systems
- Become skilled at developing probabilistic models of a wide variety of real-world systems
- Become adept at designing, running, and analyzing simulations
- Appreciate the power and wide applicability of simulation techniques
- Be able to critique someone else's simulation results
- Become educated consumers of simulation software
  - Know the questions you should be asking about what goes on "under the hood"
  - We'll focus on skills that transferrable to any simulation package