Monte Carlo simulation: A world view, some myths, some common errors, and a few random thoughts

Bruce Schmeiser
Purdue University
bruce@purdue.edu

September 30, 2009

National University of Singapore

Monte Carlo simulation: A world view, some myths, some common errors, and a few random thoughts

Bruce Schmeiser
Purdue University
bruce@purdue.edu

September 30, 2009

National University of Singapore

Main Thought

- Monte Carlo experiments estimate
  the values of performance measures.

Main Thought (continued)

- Monte Carlo experiments estimate the values of performance measures.
- They require ideas from
  - probability (usually frequentist),
  - inferential statistics (usually estimation), and
  - computer science (including data structures, data bases, compiler design, and algorithm design).

Main Thought (continued)

- Monte Carlo experiments estimate the values of performance measures.
- They require ideas from
  - probability (usually frequentist),
  - inferential statistics (usually estimation), and
  - computer science (including data structures, data bases, compiler design, and algorithm design).

- As researchers, we tend to focus on the mindsets of these underlying disciplines, caught up in the nuances of sufficient conditions and the latest cool idea.

Main Thought (concluded)

- Monte Carlo experiments estimate the values of performance measures.
- They require ideas from
  - probability (usually frequentist),
  - inferential statistics (usually estimation), and
  - computer science (including data structures, data bases, compiler design, and algorithm design).
- As researchers, we tend to focus on the mindsets of these underlying disciplines, caught up in the nuances of sufficient conditions and the latest cool idea.

We need to remember our purpose,

which **to support practitioners**,

- who (because of lack of background and/or time)
- should be protected from everything
- other than stating the problem and its context.

Thoughts: Outline

Uses of simulation.

What we (should) do.

Our world view.

How we (should) do it.

Some uses of simulation

- Simulation as a thought experiment
  (do I understand the problem?)
- Simulation for closed-form probability
  (sample-path analysis)
- Simulation to verify probability analysis
- Simulation to find counter examples
- Simulation to find conjectures
- Simulation for analysis of complex systems

What we (should) do

Help practitioners

- to determine the value
  of the performance measures, θ
- by developing
  theory, methods, and algorithms
- that can be applied
  automatically
  within vender software

What we (should) do

Help practitioners

- to determine the value
  of the performance measures, θ
- by developing
  theory, methods, and algorithms
- that can be applied
  automatically
  within vender software

"Reality is a great source of problems;
just be careful." (Shane Henderson)

Probability world view

Input Model

Logic Model

$$X \longrightarrow Y \longrightarrow \theta$$

where

X: has known distribution

Y: a known function of X

θ: given performance measure(s)

Our world view

Input Model

Logic Model

$$G \longrightarrow U \longrightarrow X \longrightarrow Y \longrightarrow Z$$

Random-number generator

Point Estimator

where

G: random-number generator/seed

U: U(0,1) random numbers

X: random variates

Y: observations

Z: point estimator of θ

Our world view

$$G \longrightarrow U \longrightarrow X \longrightarrow Y \longrightarrow Z$$

where

- G: random-number generator/seed
- U: U(0,1) random numbers
- X: arrival and service times
- Y: customer wait times
- Z: Ybar estimates E(Y)
  S_Y^2 estimates V(Y)

Our world view: sources of error

$$G \longrightarrow U \longrightarrow X \longrightarrow Y \longrightarrow Z$$

1 random-number generator

2 random-variate generation

3 model not the real world

4 bug(s) in the code

5 computer arithmetic
is not real arithmetic

6 *Z* is random (finite sample size)

Classic research topics

$$G \longrightarrow U \longrightarrow X \longrightarrow Y \longrightarrow Z$$

- U(0,1) random numbers
- random-process generation
- input modeling
- logic modeling
- variance reduction
- steady-state warm-up
- run-length, standard-error estimation

Classic research topics

$$G \longrightarrow U \longrightarrow X(x) \longrightarrow Y(x) \longrightarrow Z(x)$$

- U(0,1) random numbers
- random-process generation
- input modeling
- logic modeling
- variance reduction
- steady-state warm-up
- run-length, standard-error estimation
- optimization (find good x value)

Some other research topics

$$G \longrightarrow U \longrightarrow X(x) \longrightarrow Y(x) \longrightarrow Z(x)$$

- gradient estimation (1980s)
- event graphs (1980s)
- reporting point estimators (1990s)
- estimating modeling error (2000s)
- smoothing rate functions (2000s)
- simulation as linear programming (2000s)

Myths and Common Errors

- Simulation as a last resort
- Input modeling
- Methods for random variables
- U(0,1) random numbers
- Logic modeling
- Analysis of output data
- Tactical issues
- Ill-posed research problems
- Optimization
- New experiment

Myths and Common Errors

- **S**imulation as a last resort
- **I**nput modeling
- **M**ethods for random variables
- **U**(0,1) random numbers
- **L**ogic modeling
- **A**nalysis of output data
- **T**actical issues
- **I**ll-posed research problems
- **O**ptimization
- **N**ew experiment

Myths and Common Errors

- **S**imulation as a last resort
- **I**nput modeling
- **M**ethods for random variables
- **U**(0,1) random numbers
- **L**ogic modeling
- **A**nalysis of output data
- **T**actical issues
- **I**ll-posed research problems
- **O**ptimization
- **N**ew experiment

Myths and Common Errors

- **S**imulation as a last resort
- **I**nput modeling
- **M**ethods for random variables
- **U**(0,1) random numbers
- **L**ogic modeling
- **A**nalysis of output data
- **T**actical issues
- **I**ll-posed research problems
- **O**ptimization
- **N**ew experiment

Myths and Common Errors

- **S** imulation as a last resort
- **I** nput modeling
- **M** ethods for random variables
- **U** (0,1) random numbers
- **L** ogic modeling
- **A** nalysis of output data
- **T** actical issues
- **I** ll-posed research problems
- **O** ptimization
- **N** ew experiment

How we do it
(the good news)

- Informs simulation community
  - high standards
  - friendly
- Many background disciplines
- Real numbers and computer numbers
- Mathematicians, engineers,
  and computer scientists

How we (should) do it

- Find topic, problem, hook
  - opportunistic
  - creative
  - helps a practitioner (???)
- State the problem (difficult!)
  - what is given?
  - what is to be found?
  - how to evaluate the solution quality?

How we (should) do it

State the problem (difficult!)

– identify the practitioner.
– what is given?
– what is to be found?
– how to evaluate the solution quality?
(for example: low mse,
low computing time, low computing memory,
easy to understand, easy to implement,
no magic parameters, numerically robust)

How we (should not) do it

Example problem statement (ouch)…

I am going to write a simulation model
of the U.S. kidney transplantation system
to reduce
the number of wasted kidneys.

Ill-Posed Research Problem #1

Initial-Transient Problem

- How to choose
  - How much warm-up data $Y_1, Y_2, \ldots, Y_d$
- Purpose:
  - To minimize bias: $| \mathrm{E}(Z) - \theta |$

Ill-Posed Research Problem #1

Initial-Transient Problem

- How to choose
  - How much warm-up data $Y_1, Y_2, \ldots, Y_d$
- Purpose:
  - To minimize bias: $|\,\mathrm{E}(Z) - \theta\,|$
- Then discard almost all data.

Ill-Posed Research Problem #2

Confidence-Interval Procedures

- How to bound θ with (L, U)?

- Purpose:

$$P(L \leq \theta \leq U) = 1 - \alpha \; ?$$

Ill-Posed Research Problem #2

Confidence-Interval Procedures

- How to bound θ with (L, U)?
- Purpose:

$$P(L \le \theta \le U) = 1 - \alpha \ ?$$

- Then use two batches.

Thoughts about naive users

- Should be our focus
- Goals
  - save time of humans
  - better input and logic model
  - better understand results and limitations
- Examples
  - how many digits of thetahat to believe
  - avoid magic parameters

Thought 1 about sample size

- often…

  there is no fundamental observation
- therefore…

  avoid starting an algorithm with

Stage 1: collect ten observations

Thought 2 about sample size

- Seldom use automatic sample-size selection
  - factor of 100
  - run time is free,
    so run until want/need the answer

Thought 3 about sample size

In optimization algorithms,
don't assume a fixed sample size.

- want constant variance across solutions?
- need smaller standard error
  when closer to the optimal solution?

Thought 4 about sample size

In optimization algorithms…

- (subjunctive) cheating:

  say that there exists an *m*
  so that the solution quality would have been
  good enough if *m* had been used.

- not cheating:

  start *m* small and let *m* go to infinity,
  stopping when the solution quality
  is adequate
  (or the computing budget is expended).

Thought 1 about algorithm design

Claim: Continuity is good.

- The algorithm result should be a continuous function of the data.
- Therefore, don't use hypothesis testing.

Bonus question.

Why does failing to reject H_0 (e.g., normality, independence) allow the algorithm to assume H_0?

Thought 2 about algorithm design

- Separate
  - algorithm code
  - application code

- If published, make the code available.

Thought 3 about algorithm design

- Algorithm should be a subprogram
- Input
  - model parameters
  - experiment parameters
- Input/Output
  - random-number seeds
- Output
  - algorithm result
  - precision of algorithm result

Thoughts about confidence intervals

Don't get me started…

Thoughts about confidence intervals

Disadvantages (you got me started…)

- Even *we* don't use them.
- Users misinterpret them.
- Nominal coverage is (usually) wrong.
- (L,U) is bulky to report.
- Need normality (and/or other assumptions)
- What to do for high-dimensional θ?
- What to do for non-mean θ?

Thoughts about proofs

- How to do proofs in computer arithmetic?
- Why do we assume, for example, finite 8th moment, when only finite values are represented in computer arithmetic?
- Why does, for example, a truncated Cauchy random variable in computer arithmetic act like it is not truncated?
- Why do we assume that functions are uniformly bounded when
  - so many real-world models aren't?
  - algorithms work well regardless?

Problem definition: 4 examples

- Input modeling
  (MA Wagner)
- Comparing confidence-interval procedures
  (Schmeiser and Yeh)
- Poisson process rate functions
  (Schmeiser)
- Reporting point estimators
  (Song and Schmeiser)

Problem definition: smoothing rate functions

Given

rate constants

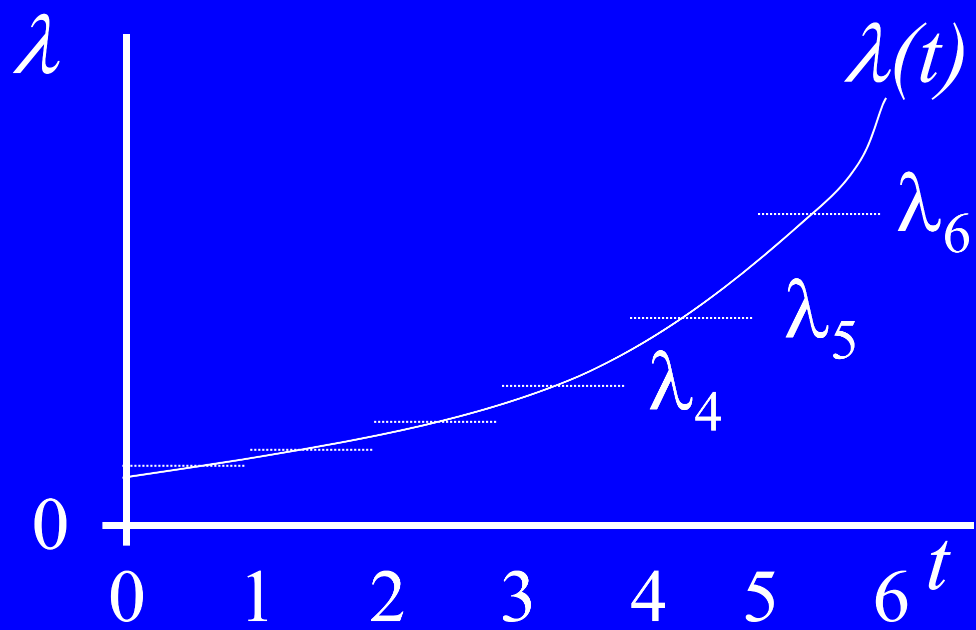
$\lambda_1, \lambda_2, \ldots, \lambda_k$

Determine rate

$\{\tau(t),\ 0 \le t \le k\}$

to approximate the true unknown continuous rate

$\{\lambda(t),\ 0 \le t \le k\}$



Criteria?

Problem Definition: defining "good" smoothing rate functions

Given

rate constants

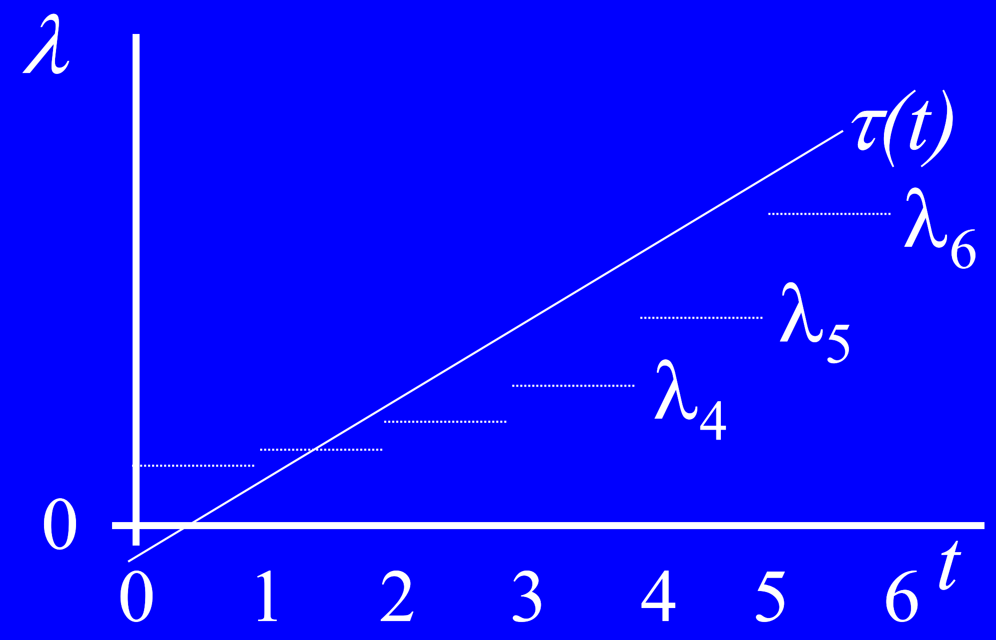
$\lambda_1, \lambda_2, \ldots, \lambda_k$

Determine rate

$\{\tau(t),\ 0 \le t \le k\}$

to approximate the true unknown continuous rate

$\{\lambda(t),\ 0 \le t \le k\}$



Nonnegative rates

Unchanged means

Reversible solution

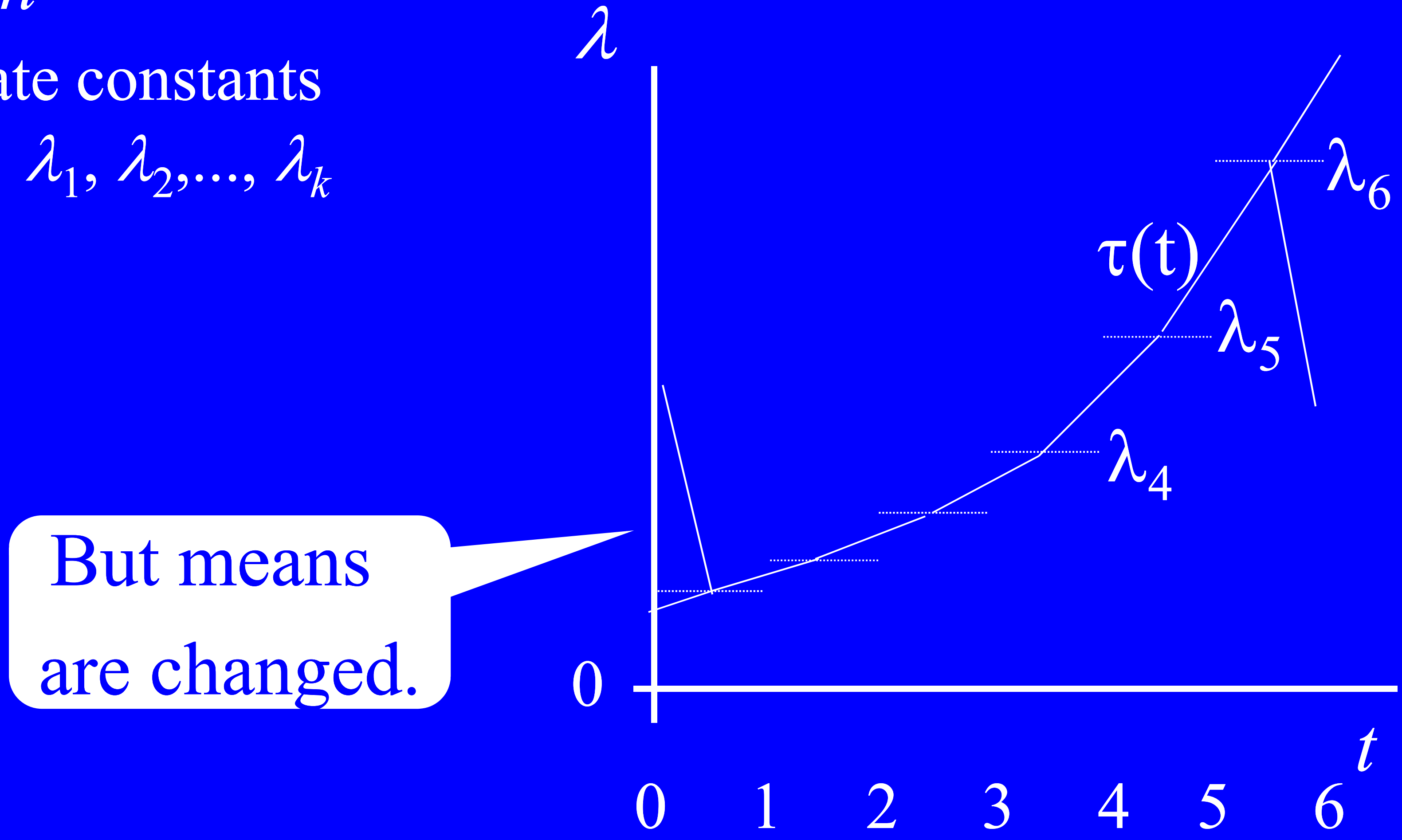
Easy random variates

A Solution: Connect Midpoints

Given

rate constants

$\lambda_1, \lambda_2, \ldots, \lambda_k$

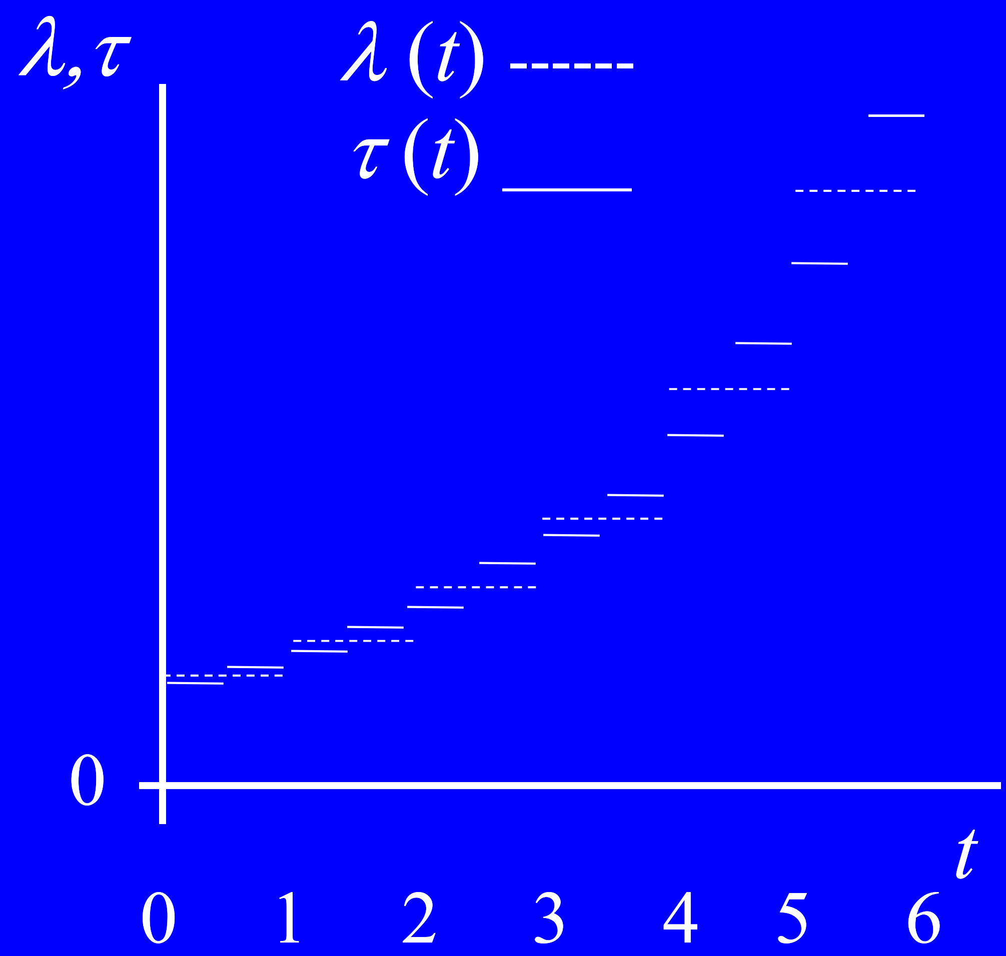


A Solution: Double Pieces

Given

rate constants

$\lambda_1, \lambda_2, \ldots, \lambda_k$



A critcism:
Don’t always mimic classical methods
Example: optimization

– Error: Assume that “observation” is a well-defined term.

– Error: Assume that the point estimator should have the same ste for every design point.

– Myth: A limit theorem can guarantee good algorithm performance.

– Myth: Lack of a limit theorem implies poor algorithm performance.

Technology transfer

- Few researchers are good at marketing.
- Mom said:
  - "Do good work and others will notice."
  - Mom was naïve.
- Technology transfer is difficult.

Thank you.

bruce@purdue.edu