CAS Public Loss Simulator and R

2013 CAS Spring Meeting

Presenter: You, Hai

VP Technology

http://www.goouon.com

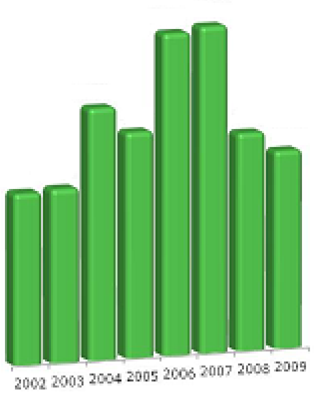
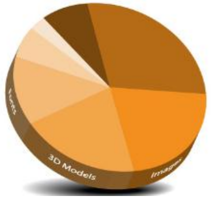


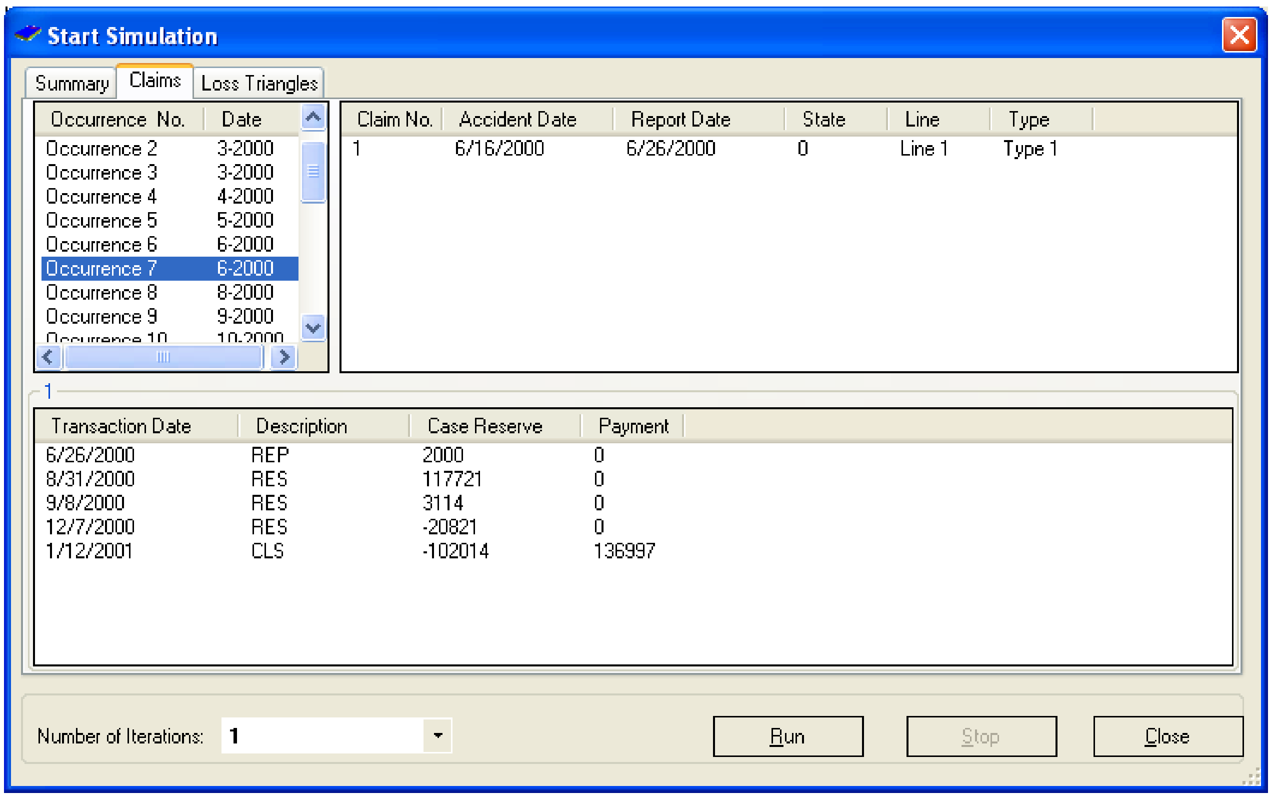
Table of Contents



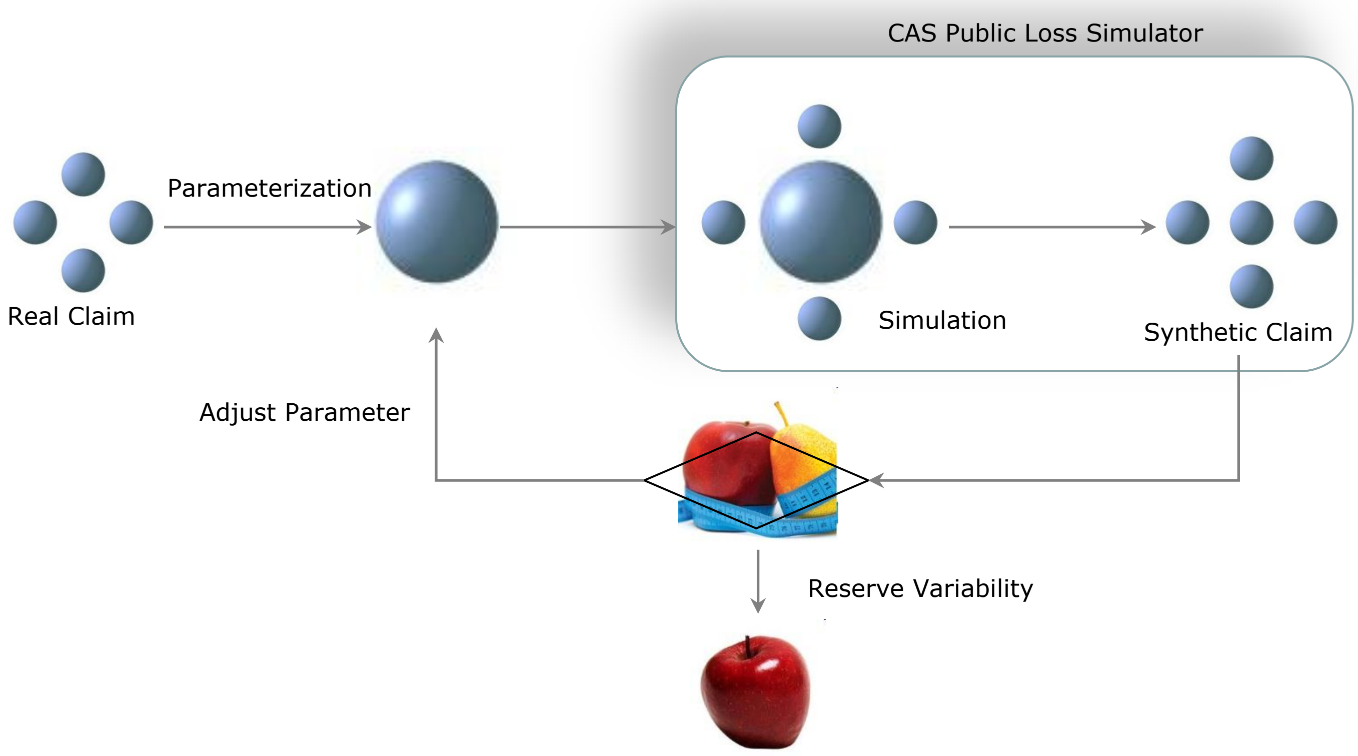
1. Public Loss Simulator

What can it do? Answer: Simulate Raw Claims

- Occurrences
- Claims
- Transactions (such as case reserve, payments, adjustments, etc)

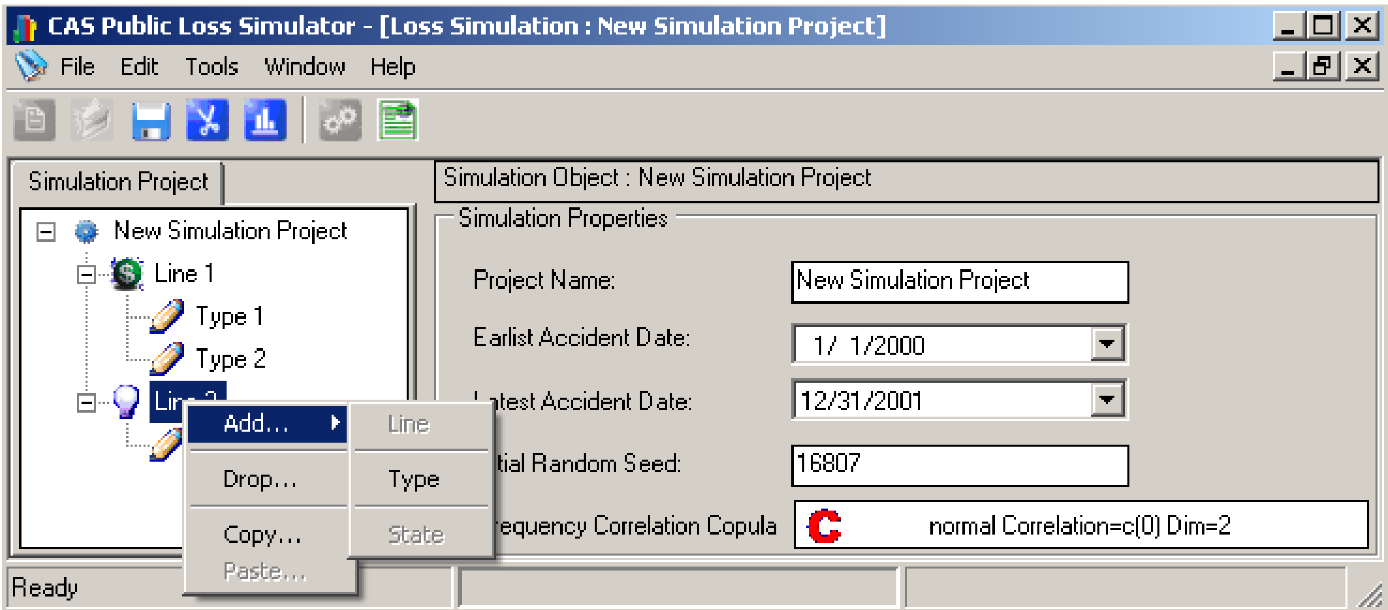


Where does it stand?



1. Public Loss Simulator

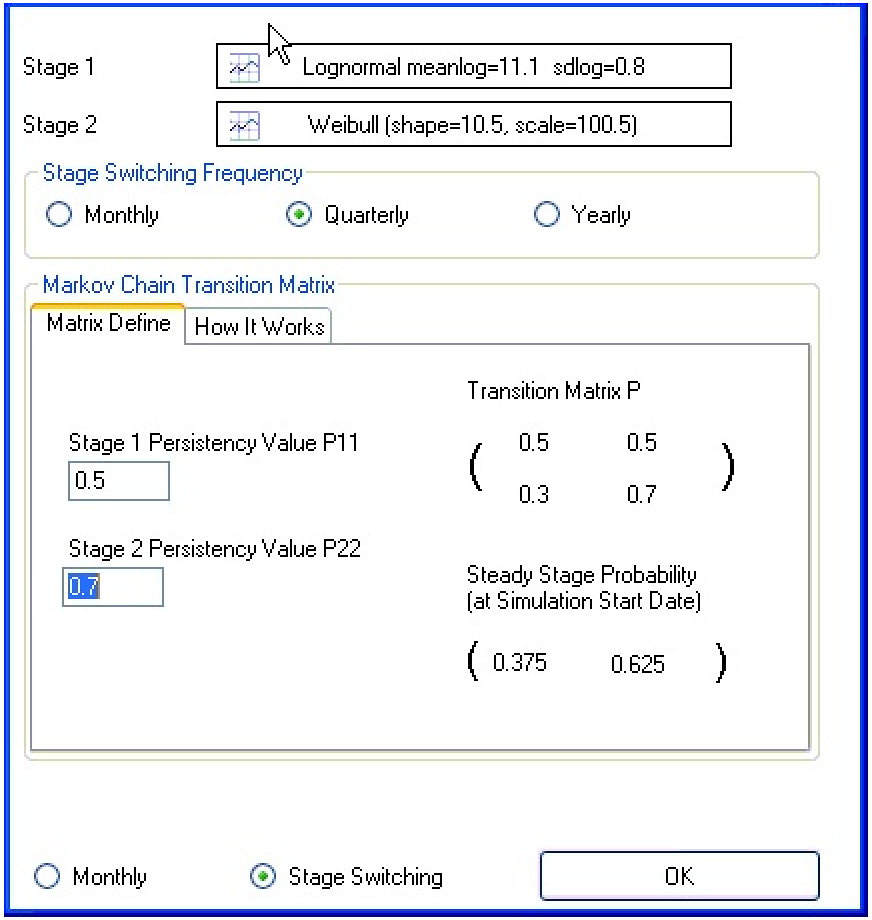
How do we run it?



- Windows Standard UI
- Tree Structure
- Accident Year Range
- Line Level Frequency Correlations from Copula
- etc, etc.

2. Enhancement in Public Loss Simulator

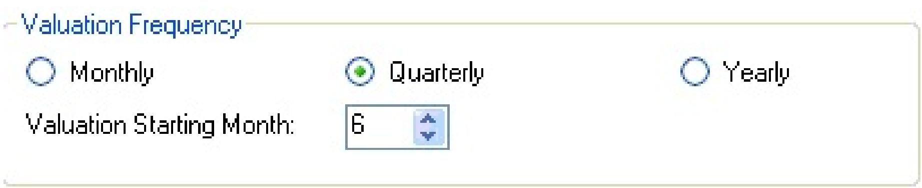
- Markov's Chain



2. Enhancement in Public Loss Simulator

- Case Reserve Interpolation by Incurred Loss

  Case Reserve Valuation Frequency



Example: Report Date: 2012-02-03, Payment Date: 2012-10-23

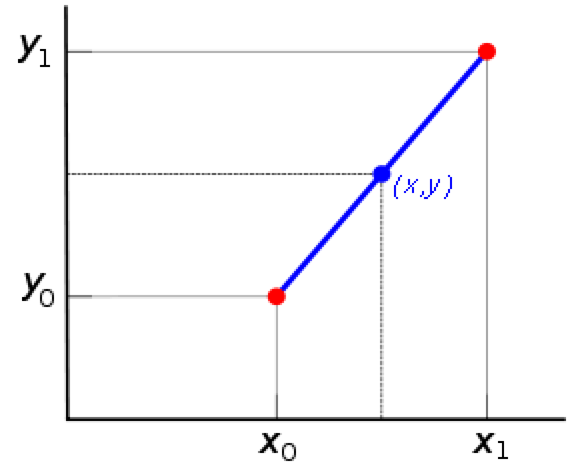
2012-02-03 (report date)
2012-02-29 (valuation date 1) *--month end will be chosen*
2012-03-31 (valuation date 2)
2012-04-30 (valuation date 3)
2012-05-18 (INT 40)
2012-05-31 (valuation date 4)
2012-06-30 (valuation date 5)
2012-07-31 (valuation date 6)
2012-08-05 (INT 70)
2012-08-31 (valuation date 7)
2012-09-26 (INT 90)
2012-09-30 (valuation date 8)
2012-10-23 (payment date, also the claim close date)

Payment Lag T=”2012-10-23” – “2012-02-03” =263 (days)
*0.40 date: 0.4 * 263 + “2012-02-03” = “2012-05-18"*
*0.70 date: 0.7 * 263 + “2012-02-03” = “2012-08-05"*
*0.90 date: 0.9 * 263 + “2012-02-03” = "2012-09-26"*

- Case Reserve Interpolation by Incurred Loss.
  Incurred Loss Interpolation

This excel sheet explains the whole story:

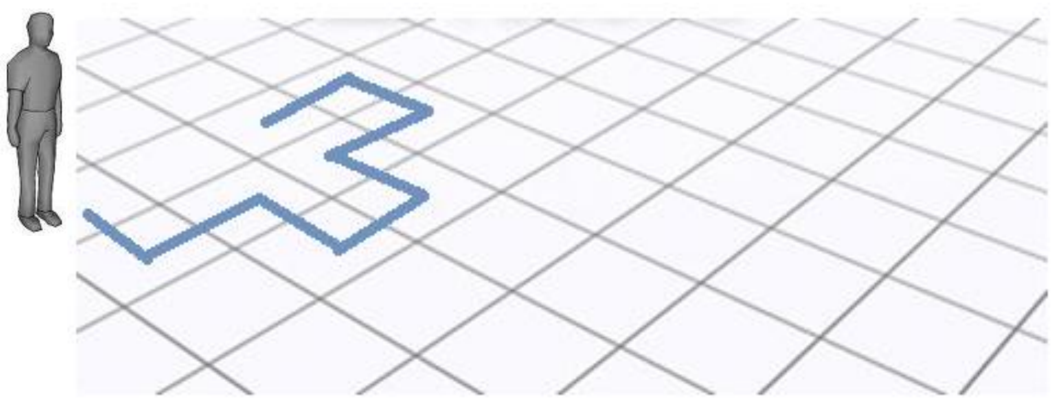
http://www.goouon.com/loss_simulator/doc/CaseReserve.xls



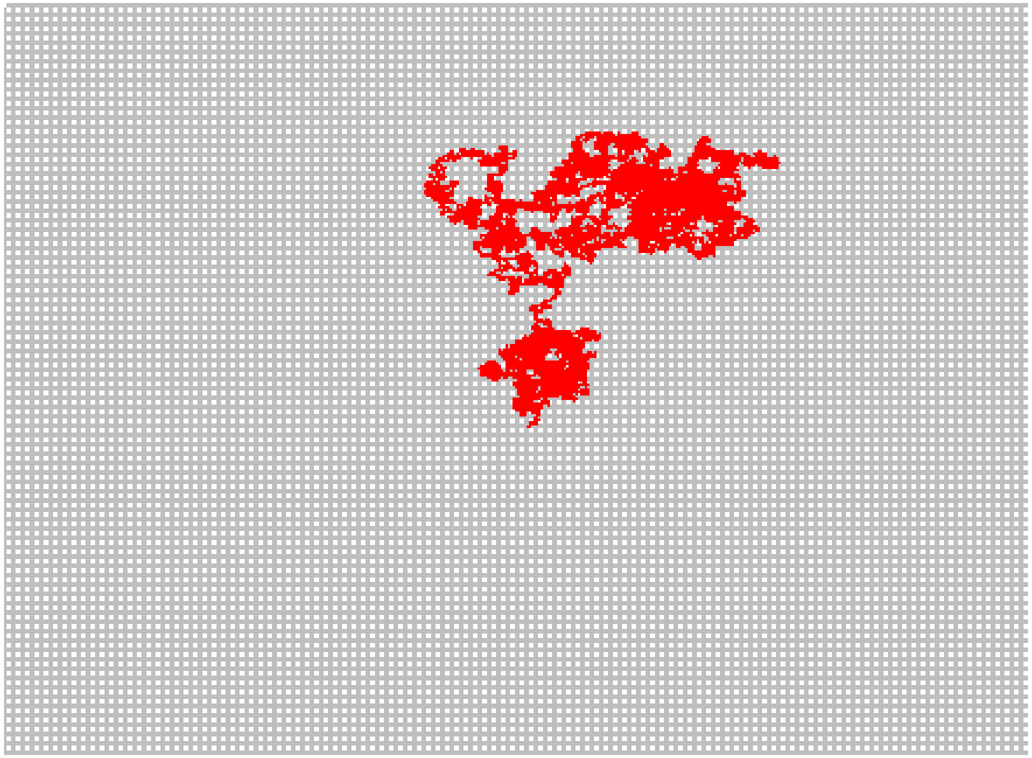
- What Is Markov's Chain

  We have a chain of *states, S* = {*s*1*, s*2*,… sn*}. The process starts in *s1* and moves to next state. Each move is called a *step.* If the chain is currently in state *si*, then it moves to state *sj* at the next step with a probability denoted by *pij* , which solely depends upon *si*.

  For example, let us define a simple Random Walk as a Markov's Chain (4 staged, left, right, forward, and backward, *pij*=0.25)

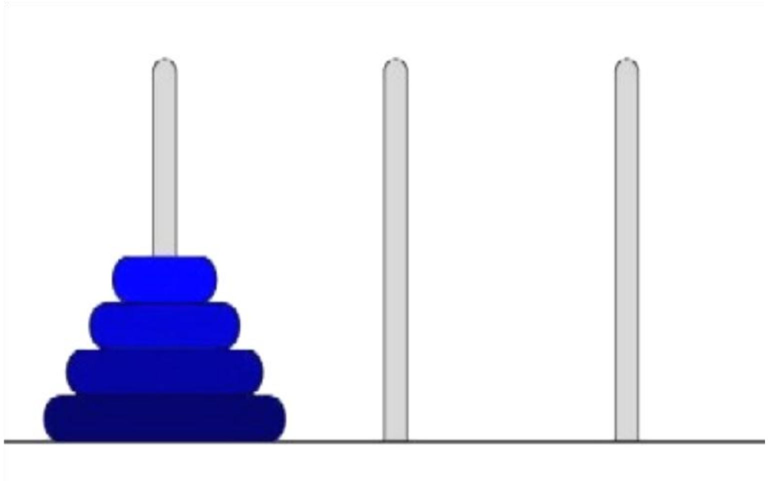


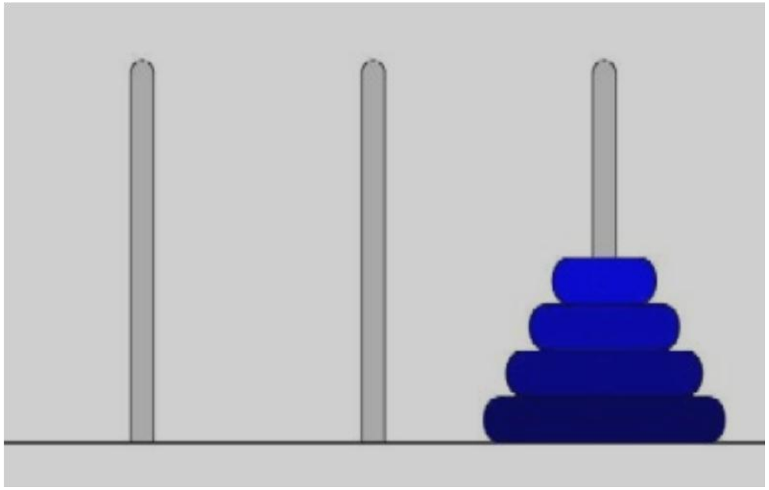
3. Markov's Chain (Random Walk) in R



You Can Download Code from http://www.goouon.com/downloads/randomWalk.R

4. Hanoi Tower (none actuarial related)



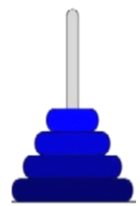


4. Hanoi Tower (Object Oriented Design)

Object 1: Block

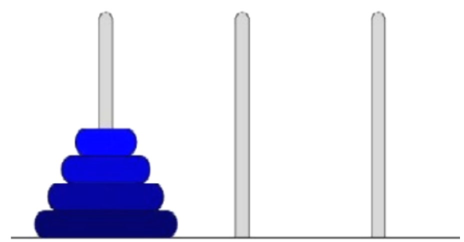
Dumb. Can only draw its self

Object 2: Tower (Stack)



Smart. Manage blocks in stack.

Object 3: Hanoi Tower



Wise. Do the Hanoi move, and manage the towers.

You can download code from http://www.goouon.com/downloads/hanoi.R

4. Define Triangle in R (actuarial related)

```
setClass("Triangle",
		representation (data="matrix", name="character", accumulated="logical",
		point="logical"),

		prototype(data=matrix(nrow=0, ncol=0), name="paid", accumulated=TRUE,
				point=FALSE)
)

################################################
#Methods: getYears, getAges
################################################
setGeneric("getYears", function(object, ...)  standardGeneric("getYears"))
setMethod("getYears", signature("Triangle"), function(object){
		return (dimnames(object@data)[[1]])
})

setGeneric("getAges", function(object, ...)  standardGeneric("getAges"))
setMethod("getAges", signature("Triangle"), function(object){
		return (dimnames(object@data)[[2]])
})
```