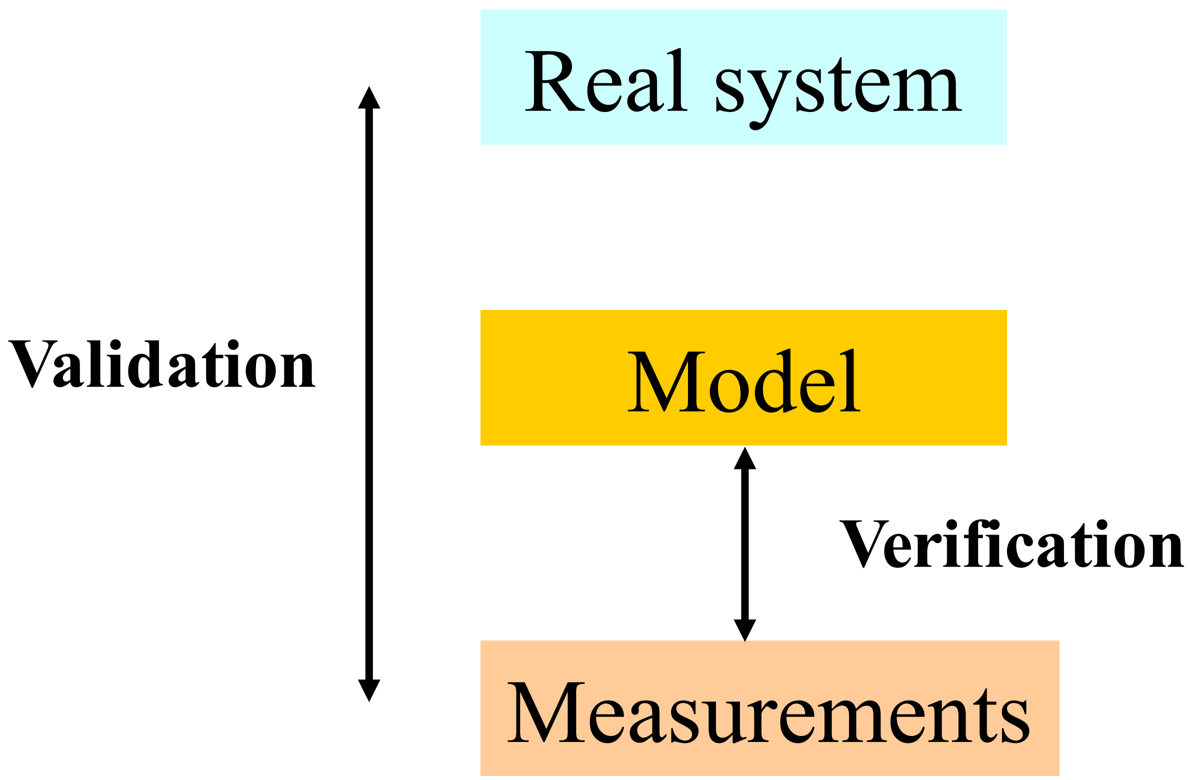
EITN95 Lecture 3, The artistic part and generating input

The creative process

- Make sure there is plenty of cheese and wine in the fridge…
- What do I want to know? What are the requirements?
- What do I know?, Which information / parts are interesting and what is noise?
- How can I simplify / what can I assume?
- Can I break the task down into manageable sub-tasks?
- Draw flow chart / state diagrams / architecture……..
- Plan the verification and validation processes

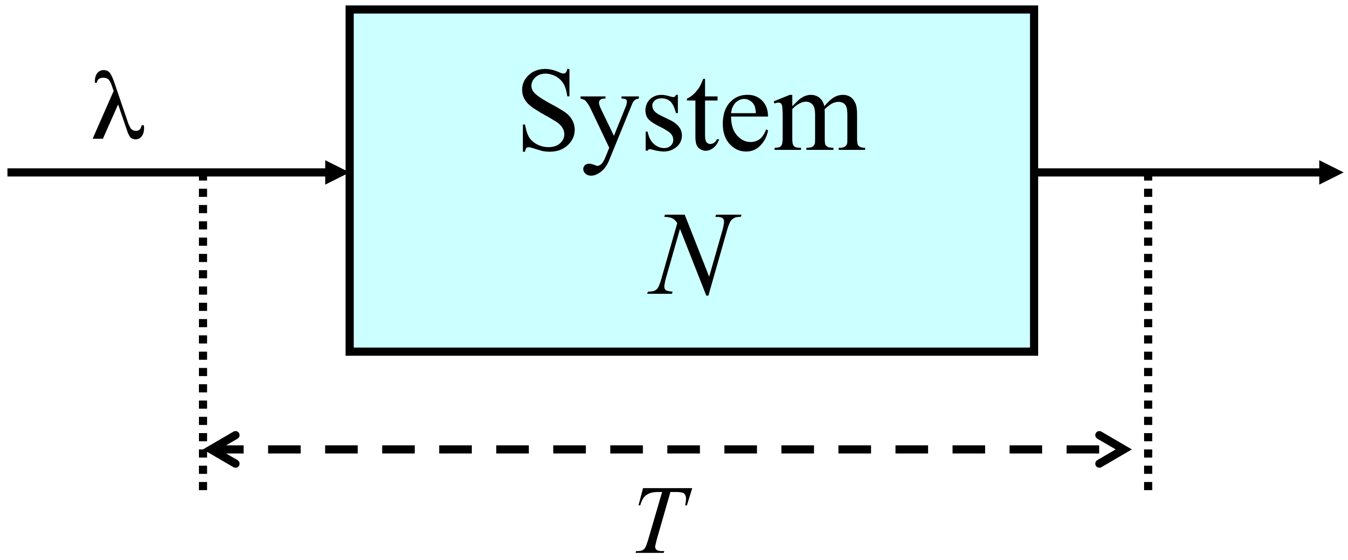
Verification and Validation



Verification

- Break model down into smaller bits and test each bit individually
- Examine the sanity of output, reasonable?
- Print parameters at end of program
- Have someone check the code
- Animate
- Compare with analytical results
- Little's theorem (for the queue example)

Little's theorem



$$N=\lambda T$$

A very general result!

Validation

- Often much more difficult than verification
- Talk to experts in the field
- Measure systems and compare results
- Can one find extremes, bounds?
  - Should the system tend to 0 or infinity when the input changes?

Generating Random Variables

Very different ways have been used

- Manual methods (rolling a die etc.)
- Using tables of random numbers
- Irrational numbers, decimals of π etc.
- Physical systems, radioactive decay, vacuum tubes
- Computer real-time clock

- All methods have drawbacks

Properties of a good method

- The routine should be fast
- The routine should be portable to different computers
- The routine should have a long cycle
- The random number should be replicable
- The random number should have the right distribution
- The random numbers should be uncorrelated

von Neuman, ca 1950

Midsquare method

- Take a number with 4 digits
  - Square the number
  - Take the 4 digits in the middle as new number
  - goto 1

- Very bad method, short sequences before repetition

Pseudo-Random Numbers

- Definition: A sequence of pseudo-random numbers is a deterministic sequence of numbers having the same relevant statistical properties as a sequence of random numbers.
- The most widely used method of generating pseudo-random numbers are the congruential generators:

$$X_i = (aX_{i-1} + c) \bmod M$$
$$U_i = X_i / M$$

- for a multiplier a, shift c, and modulus M, all integers

Pseudo-Random Numbers

- The sequence is periodic, with maximum period M.
- The values of a and c must be carefully chosen to maximise the period of the generator, and to ensure that the generator has good statistical properties, e.g.

M	a	c
2^{59}	13^{13}	0
2^{32}	69069	1
$2^{31}-1$	630360016	0
2^{32}	2147001325	715136305

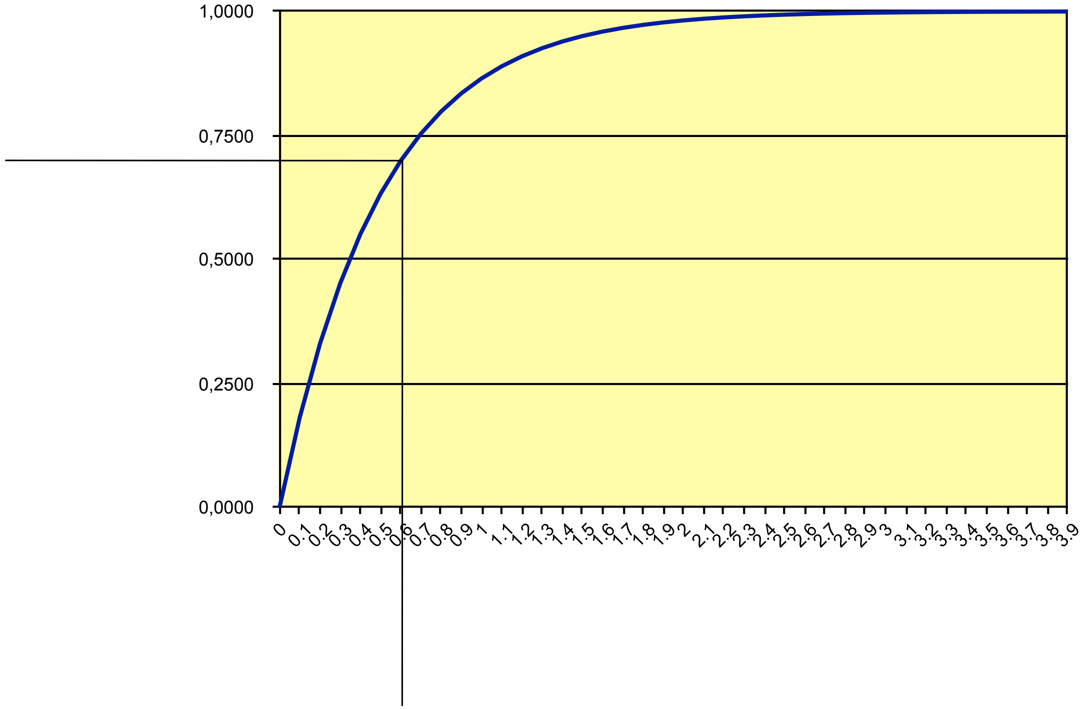
Generating Arbitrarily Distributed rvs

$$\text{Let } F(x) = \Pr\{X \leq x\}$$

- There is a theorem which says that, if $F(x)$ is continuous, then $F(X)$ is uniformly distributed on the interval (0,1).
- This can be used to generate random variables with other specified distributions.

CDF Inverse Method

Y~Uniform(0,1)



X~F(x)

Inverse Method Example

- Exponential Distribution ($1 - e^{-\lambda X}$):

$$X = F^{-1}(Y) = -\frac{\ln(1-Y)}{\lambda}$$

Gaussian Distribution

- No explicit inverse
- Approximate the inverse (Box and Muller 1958)

$$Q(x_p) = p$$

$$x_p = t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3} \qquad t = \sqrt{\ln \frac{1}{p^2}}$$

$$\begin{aligned} c_0 &= 2.515517 & d_1 &= 1.432788 \\ c_1 &= 0.802853 & d_2 &= 0.189269 \\ c_2 &= 0.010328 & d_3 &= 0.001308 \end{aligned}$$

- Sum of Uniforms (Central Limit Theorem, Irwin-Hall)

$X = \sum_{i=1}^{12} U_i - 6$ is approximately Gaussian....

General Discrete Distribution

- Assume that we want to generate discrete random variables from a specified distribution, e.g.

$$\{p_k; k = 0, 1, \cdots, N\}$$

The inversion method reduces to searching for an appropriate index in a table of probabilities:

$$\text{If } \sum_{k=0}^{j-1} p_k \leq U < \sum_{k=0}^{j} p_k \text{ then return } X = j$$

Example – Inverse Method for Discrete RV's

- Assume that the required distribution is
  $p0=0.5$
  $p1=0.3$
  $p2=0.2$
- Then
  if $0.0 \leq U < 0.5$ return $X=0$
  if $0.5 \leq U < 0.8$ return $X=1$
  if $0.8 \leq U \leq 1.0$ return $X=2$

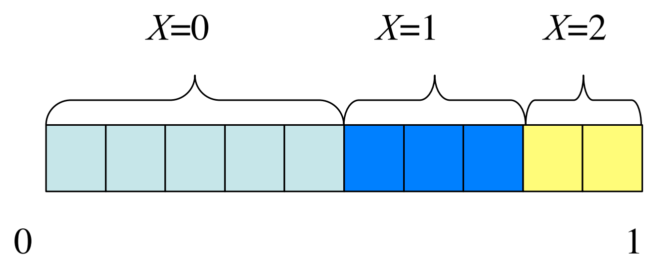


Table Method

- Generate a large array of size M, with
  - elements 1 to Mp_0 having the value 0,
  - elements $Mp_0 + 1$ to $Mp_0 + Mp_1$ having the value 1,
  - elements $Mp_0 + Mp_1 + 1$ to $Mp_0 + Mp_1 + Mp_2$ having the value 2,

etc.

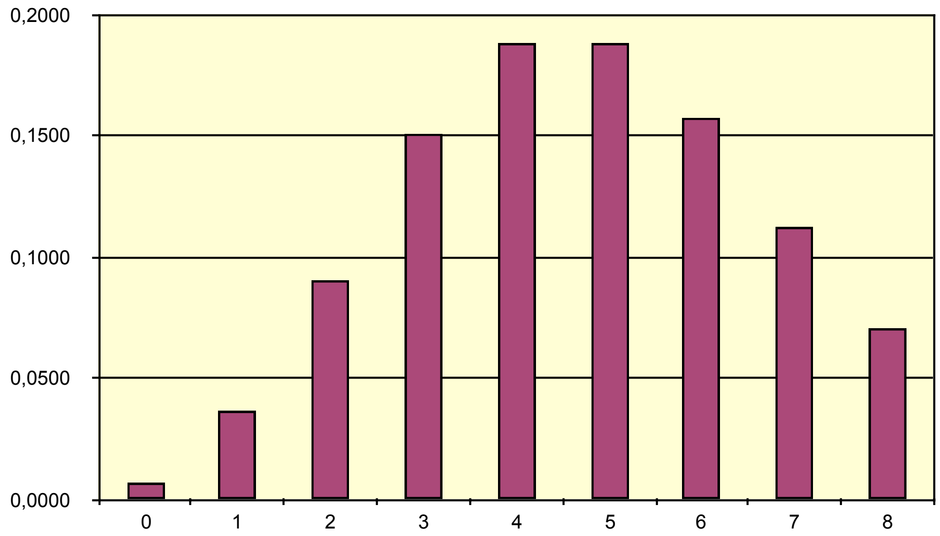
- Then generate a uniform integer rv X from 1 to M. The X*th* element of the array is a rv with the required distribution.
- This algorithm is very fast to run, but this is achieved at the expense of often requiring the allocation of large amounts of storage for the arrays.

Example: Table Method

- Assume that the required distribution is
  $p_0=0.5,\ p_1=0.3,\ p_2=0.2$
- Then construct an array of size 10 where 5 elements are zero, 3 are 1, and 2 are 2, i.e.
  [0,0,0,0,0,1,1,1,2,2]
- Now sample uniformly from this array. To do this, generate a uniform integer rv distributed from 1 to 10.
- If the generated rv is X, choose the X^{th} element of the array.

Example: Generating Truncated Poisson Distribution

$$p_k = \frac{\frac{A^k}{k!}}{\sum_{i=0}^{N} \frac{A^i}{i!}}$$



A=5
N=8

Example: Table Method

- Truncated Poisson distribution with *A=5* and *N=8*
  $\{p_k\}$={0.0072, 0.0362, 0.0904, 0.1506, 0.1883, 0.1883, 0.1569, 0.1121, 0.0700}
- Construct an array of size (say) 10,000 where
  - elements 1 to 72 have the value 0,
  - elements 73 to 434 have the value 1,
  - elements 435 to 1338 have the value 2,
  - elements 1339 to 2844 have the value 3,
  - elements 2845 to 4727 have the value 4,
  - elements 4728 to 6610 have the value 5,
  - elements 6611 to 8179 have the value 6,
  - elements 8180 to 9300 have the value 7,
  - elements 9301 to 10000 have the value 8.
- Now sample randomly from this array. The resulting rv will be approximately the required truncated Poisson.

Error analysis

- Almost all simulation models are approximations to some degree
  - Major question: How does this affect the validity of my results?
  - If I can't answer this question, what can I say about the results I get?

Boundary cases

- General strategy
  - Identify worst and best cases
  - Perhaps for sub-components of the large simulation
  - get results for these cases and compare
- Especially, test parts where you know you have approximated

Error propagation

- Before statistical analysis (which captures stochastic nature)
  – Have I got errors in input data?
  – Have I rounded, approximated if so how much?
  – Can I estimate the uncertainty in my model?

Example, estimate volume

- Use logarithmic transformation
- Old result derived from Taylor series yield:

$$\Delta f(x) = \frac{\delta f(x)}{dx} \times \Delta x$$

$$\Delta x << x$$

- from this we can derive rules for estimating error propagation

Some examples

- x = a+b-c

$$\Delta x = \sqrt{\Delta a^2 + \Delta b^2 + \Delta c^2}$$

- x = a x b/c

$$\Delta x = \sqrt{\left(\frac{\Delta a}{a}\right)^2 + \left(\frac{\Delta b}{b}\right)^2 + \left(\frac{\Delta c}{c}\right)^2}$$

- x = ln a

$$\Delta x = \frac{\Delta x}{x}$$

Estimating volume of a cylinder

- Measure radius and length, say:

$$r = 1cm, \Delta r = 0.1cm$$
$$l = 8cm, \Delta l = 0.05cm$$

$$V = \pi r^2 \times l = 25.13$$

$$\Delta V = \sqrt{\left(\frac{\Delta r}{r}\right)^2 + \left(\frac{\Delta r}{r}\right)^2 + \left(\frac{\Delta l}{l}\right)^2} =$$

$$\sqrt{\left(\frac{0.1}{1}\right)^2 + \left(\frac{0.1}{1}\right)^2 + \left(\frac{0.05}{8}\right)^2} =$$

$$= 0.02$$

Generalisation

- A simulation collects results from a specific scenario
  - What conclusions can we draw from this in the general case?
- Word of caution - don’t generalize in general
  - Think first, am I sure nothing will pop up I have not foreseen?

Next lecture

- Look at the statistical nature of measuring a **stochastic process**
- How can we estimate the error from this process
- How certain can we be that our result is an accurate reflection of reality?