

Stochastic model building and simulation

©Leif Gustafsson 2006-03-16

Contents:

Exercise 1. Generation of random numbers

(To build random number generators and testing them.)

Exercise 2. Monte Carlo simulation of a neutron shield

(Stochastics on a static model)

Exercise 3. A weather model

(Modelling temperature and precipitation over the year.)

Exercise 4. Stochastic modelling of radioactive decay

(Correct modelling or just adding noise.)

Exercise 5. An epidemic model

(Is the average of a stochastic model equal to that of a deterministic one?)

(Estimates of averages, variations and confidence intervals.)

Exercise 6. A logistic population model

(Warning for using a deterministic model structure. Start from the system when making your stochastic model!)

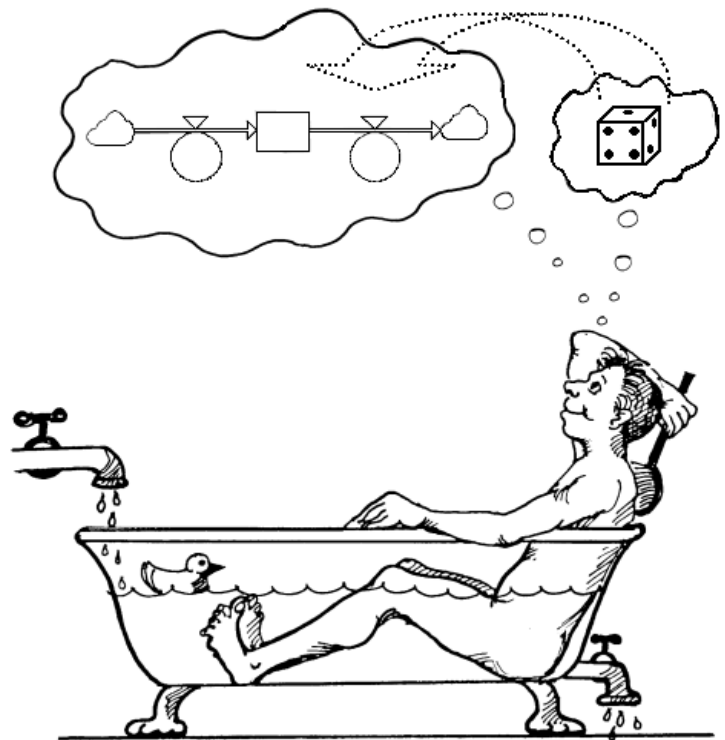
Exercise 7. A stochastic prey-predator model

(How stochastics excites dynamic variations and how new qualities are obtained.)

Exercise 8. Summary of your findings

(Sum up your findings in your own words.)

Purpose: *To practice stochastic model building and simulation and to find out important differences between stochastic and deterministic modelling of dynamic systems.*



Name:	Date:
Course:	Approved:

INTRODUCTION

This exercise is about stochastic modelling and consists of seven different studies and a summary to be written by you.

The first exercise is preferably done in Excel, while the following ones are done in Powersim.

Exercises 4-7 are about *population models*. A population consists of a *discrete* (i.e. *integer*) number of individuals. These individuals may be physical quantities like electrons, atoms or molecules or animals, plants or micro-organisms in biology or humans in e.g. epidemiology. It is then usually important to model the population as a number of individuals rather than as a continuous mass. For population models Poisson Simulation is a powerful technique. In these exercises you start by building deterministic, dynamic models. This is to be able to compare with the behaviour of a corresponding stochastic and dynamic model.

Use the *Euler* integration method for all simulations, but be sure to use a step-size that is proper *for each model*.

In this presentation we use lower case for deterministic variables (e.g. x, y) and upper case for stochastic ones (e.g. X, Y).

Exercise 1. Generation of random numbers

(To build random number generators and testing them.)

This exercise is preferably done in Excel.

1.1 A uniform R[0,1]-generator

Write a random number generator for uniformly distributed numbers between zero and one, a so called R[0,1]-generator (R stands for rectangular distribution).

A *linear congruential generator* has the algorithm:

```
rand=Seed
i=1
repeat
  rand = (a*rand+c) Mod M      (1)
  Ri=rand/M                  (2)
  i=i+1
until done
```

This algorithm generates a sequence $R_1, R_2, R_3, \dots, R_n, R_1, R_2, \dots$ of R[0,1] distributed random numbers. After a number of random numbers the sequence will start over again with R_1, R_2, \dots . The number of random numbers before it restarts is called the *period*.

Seed is a value you choose. The seed will exactly determine the sequence. This means that you can repeat the simulation by just using the same values of the seed.

The parameter m is called the modulus, a is called the multiplier and c is called the increment. They are all integer numbers.

Mod M means that you divide something (here $a*rand+c$) by M and keep only the remainder! Since you divide an integer number by M the remainder can only take values 0, 1, 2, ... $M-1$. Thus, M is the longest possible period of a sequence. Therefore, M should be a large number and a and c should be carefully chosen so that you get a generator of the full period length= m . In this example we only are interested in the principle and chose the relatively small values: $M=6075$, $a=106$, $c=1283$.

Open Excel and start with the following:

	A	B	C	D
1				
2	a	c	M	Seed
3	106	1283	6075	<i>your seed</i>
4				
5		rand	R(i)	
6	R1	<i>Equ 1</i>	<i>Equ 2</i>	
7	R2	<i>Equ 1</i>	<i>Equ 2</i>	
8	R3	<i>Equ 1</i>	<i>Equ 2</i>	

In *Equ 1* you have $rand=MOD(a*rand+c; M)$ where the first *rand* is 'your seed' and the next comes from the cell above. (If you have a Swedish version the MOD(number; divisor) function is called REST(tal; divisor). Note also that the delimiter ';' may be ',' instead depending on the National settings on your computer.)

Equ 2

calculates R by dividing *rand* by M . Thus Since *rand* has a value from 0 through $M-1$, R will get a value in the interval $0 \leq R < 1$.

Lock the cells with absolute addresses and copy so that you get 100 random numbers R1, R2,.. R100.

You should now get a series of 100 different random numbers between zero and one.

To test the generator, calculate the average and the standard deviation of the 100 numbers. Does it seem reasonable? (The theoretical standard deviation is: $(1-0)/\sqrt{12} \approx 0.289$.)

Answer: Average = Standard deviation =

Try some other seeds to see the changes!

Another of many possible tests is that the numbers should be rather evenly distributed. For example about the same fraction of random numbers should be in the intervals 0-0.1, 0.1-0.2, ... 0.9-1. To test this you may make a new column where you multiply R_i by 10 and round it down to an integer. (=ROUNDDOWN(10*C6;1). (Swedish: =RUNDA.NER(10*C6;1). You should then get a column of 100 numbers of values 0, 1 ... 9.

At the next column you may check how many of the 100 values that are equal to 0 using =COUNTIF(\$D\$6;\$D\$105,0). (Swedish: =ANTAL.OM(\$D\$6:\$D\$105;0).) Then do the same for 1, 2, ... 9.

Now mark the fields telling the number of occasions for 0, 1, 2, ...9, and make a histogram. Try some other seeds to see if there is a tendency to be more outcomes in a certain interval.

Answer:

1.2 An Exponential generator

Now we want exponential distributed random numbers of the form: $f(x)=(1/m) \cdot e^{-x/m}$. (This distribution is denoted Exp(m).) This is easily done by using the *inversion method*. First we need the cumulative

distribution function which is $F(x)=1-e^{-x/m}$. To obtain a random value x from an $R(0,1)$ distribution, we have to invert the relation $r=F(x)$ to get $x=F^{-1}(r)$.

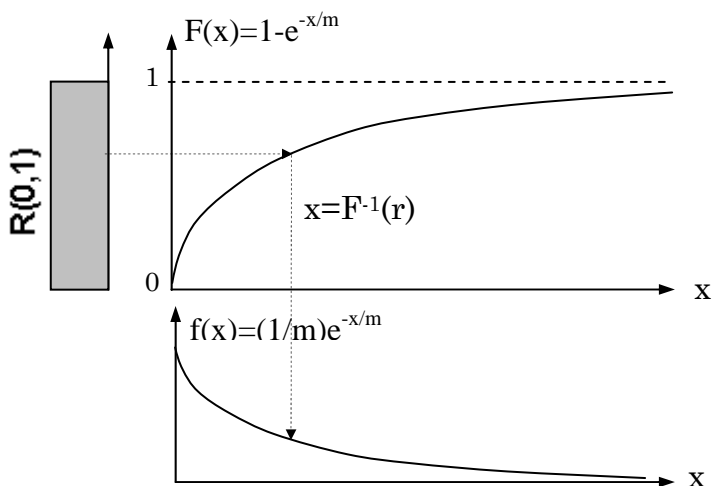


Figure 1. Map $r \in R[0,1]$ to x by using the inverse function F^{-1} . With our choice of F , x will be $\text{Exp}(m)$ distributed.

Thus $r=1-\exp(-x/m)$ i.e. $\exp(-x/m) = 1-r$. Taking the logarithm of both sides gives $-x/m = \ln(1-r)$ implying $x=-m*\ln(1-r)$. But since r has a uniform distribution between zero and one, the same is true for $1-r$ so we can replace $1-r$ by r giving: $x=-m*\ln(r)$. This is the inverse we use when mapping $r \in R[0,1]$ to the $x \in \text{Exp}(m)$ distribution.

Thus, to use an inverse mapping of $r \in R[0,1]$ -numbers it is practical to use a column somewhere to the right of the $R(i)$ column. Above this column write m and reserve a cell for its value, e.g. '2'. In this column use the formula: $=-G\$3*\text{LN}(C6)$; (if $G3$ is the cell containing the value of m).

Copy it down to get 100 cells with $\text{Exp}(m)$ distributed random numbers. Check the average and the standard deviations for a number of seeds. (Theoretically they should both be equal to m .)

Also make a histogram with 10 bars and try a number of seeds. (For $m=2$ the intervals 0-0.1, 0.1-0.2, ... 0.9-1 will do.) **[BÄTTRE: 0-1, 1-2, ...9-10!! Sven Smårs ??]**

What did you get?

Answer:

.....

1.3 A Poisson generator

A Poisson generator can be based on different methods. A simple and often used method makes use of the exponential distribution.

The underlying idea is that a Poisson process with intensity λ events/time unit can be regarded in two alternative ways:

- 1) The number of events during a time interval Δt is $\text{Po}[\Delta t \cdot \lambda]$ distributed.
- 2) The distance between events is exponentially $\text{Exp}[1/\lambda]$ distributed. (Its probability distribution function is $f(t) = \lambda e^{-\lambda t}$.)

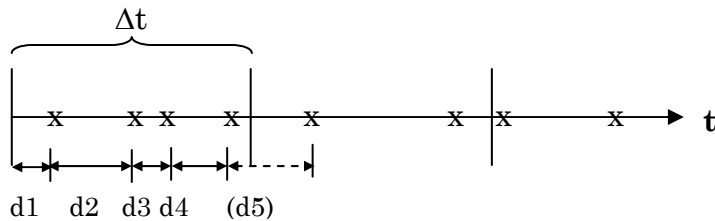


Figure 2. $Po[\Delta t \cdot \lambda]$ is the number of events during the time interval Δt when the intensity is λ . This number can be obtained by adding exponentially distributed distances $d_1+d_2+\dots$ until the sum exceeds Δt . In this figure you get four intervals $d_1+d_2+d_3+d_4 \leq \Delta t$ in the first time-step since $d_1+d_2+d_3+d_4+d_5 > \Delta t$.

The *direct algorithm* for Poisson distributed random numbers (where $\Delta t \cdot \lambda$ is called L) can be written:

```

Function Po(L)          (L=Δt·λ)
N=0                    (N counts the events)
D=0                    (D is a variable to sum up the distances)
Again: D=D+(-Ln(R[0,1])/L) (The inverse method. Add exp(L) distributed distances)
N=N+1                  (One more event)
IF D<1 THEN GoTo Again (Inside the interval of size 1? (Δt is included in L and therefore
also                  in D so compare with a unit interval))
Po=N-1                 (Don't include the last distance that goes beyond.)

```

As you see the Poisson algorithm includes the exponential algorithm which in turn calls the uniform $R[0,1]$ generator. If a *seed* is included it should be passed all the way back to the $R[0,1]$ generator. (If you look for the direct algorithm in the literature you will find that a number of tricks are used to make it faster. E.g. a multiplicative rather than additive mechanism is usually used.)

Although the $Po[L]$ algorithm is easily implemented in a program, it is a bit tricky to write it in a spreadsheet since you don't know a priori how many loops you will need for each Poisson distributed random number (unless you perform it in a macro: See the laboratory exercises: "Exercises with the spreadsheet Excel", Example 9).

To get a rough estimate, set $m=1/L=1/(\Delta t \cdot \lambda)$, e.g. $\Delta t=0.1$ and $\lambda=20$ gives $L=\Delta t \cdot \lambda=2$ and $m=1/L=0.5$. Then $Po(L)$ is a $Po(2)$ distribution. *In average* you expect to sum up 2 consecutive distances *before* the sum exceeds one. Set $m=0.5$ and start from the top of the $Exp(0.5)$ column and count the exponentially distributed numbers downwards like in the table.

In this example we got the numbers 1, 4, 2, 3, 2, 0, 3 ... of Poisson distributed random numbers for the seven first time-steps.

What did you get? Does it seem reasonable for your own figures?

Answer:

(Comment. There is a slight difference between the algorithm above where the "last distance" (making the sum above one) is discarded. In the table the "last distance" is reused for the next Δt . Both ways are correct. This is because of a unique property of the exponential distribution called memorylessness.)

Distamces	Exp(m)	Po(2)
d1	0.737799	1
d1	0.402327	
d2	0.239882	
d3	0.100582	
d4	0.05511	4
d1	0.952554	
d2	0.007714	2
d1	0.265108	
d2	0.263991	
d3	0.157778	3
d1	0.321454	
d2	0.029932	2
-	1.479682	0
d1	0.173016	
d2	0.792122	
d3	0.024543	3

Exercise 2. Monte Carlo simulation of a neutron shield

(Stochastics on a static model)

Monte Carlo simulation is a very primitive form of simulation where time is not explicitly present – and thus of course no dynamics (e.g. time derivatives). It only uses random numbers, usually uniform, to estimate the surface or volume of a body inscribed in another surface or body with a known area or volume. In an earlier exercise it was used to estimate the value of π . Another type of use is given here:

You have to protect yourself against a beam of neutrons by putting a wall of lead between you and a neutron ray. The wall is 5 units thick. Each neutron hits the wall at right angle and travels a unit distance before it collides with a lead atom. It then rebounds and continues another unit distance in a random direction and so on. After eight collisions all the neutron's energy is spent. What percentage of the neutrons will pass the lead shield?

(To make the model simple we make a number of simplifications: The neutron always travels a unit distance between collisions, all angles are equally probable, we only consider two dimensions etc. However, we could use random numbers for the distances and angles and make the calculations in three dimensions.)

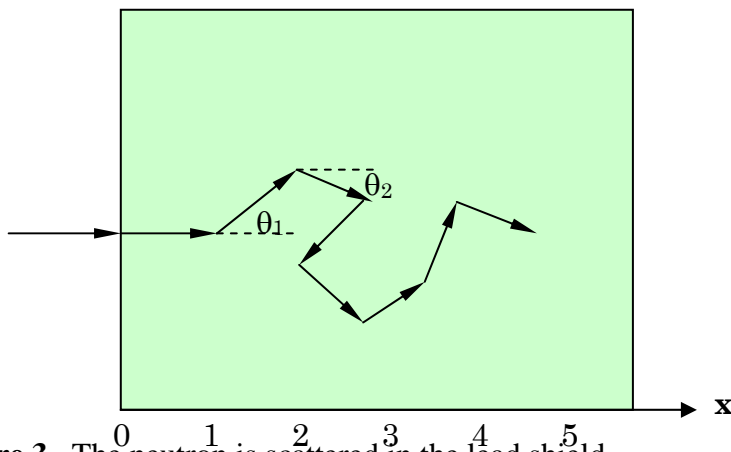


Figure 3. The neutron is scattered in the lead shield.

Let x be the distance through the wall the neutron reaches. Then it first reaches one unit before the first collision. The second distance will then be $\cos(\theta_1)$, where θ_1 is the scattered angle ($0 \leq \theta_1 < 2\pi$). the distance after eight distances therefore is:

$$x = 1 + \cos(\theta_1) + \cos(\theta_2) + \dots + \cos(\theta_7) = 1 + \cos(2\pi \cdot R_1) + \cos(2\pi \cdot R_2) + \dots + \cos(2\pi \cdot R_7)$$

The factor 2π is included because $0 \leq R[0,1] < 1$ is not enough for the full turn.

Note that time is not really present. It is only a sequence of a first distance plus seven distances in random angles that are added. Nor is there any dynamics involved.

The simple model above can easily be implemented in any programming language, in a spreadsheet or in a simulation language like Powersim. In Powersim the model may look like in Figure 4.

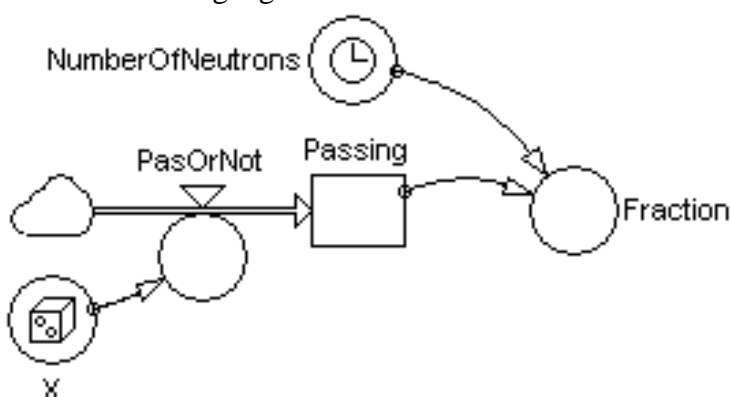


Figure 4. The Monte Carlo model implemented in Powersim.

Make a number of replications (=simulation runs). Do you get the same result every time?

Answer:

What percentage of the neutrons will pass the lead shield?

Answer: %

How to make the simulation reproducible

As seen we get a slightly different result for each new replication. That is good because if you had done several experiments on the real system you would also get somewhat different results. But if you for some reason want to repeat the model experiment exactly – just to demonstrate it or because you want to compare just this experiment with a neutron that behaves “stochastically the same” but where you have a thicker lead shield – then it would be nice to be able to control the stochastics.

This can be done by using the seed option in the random function. For example:

```
EXPRND(Mean [, Seed])  
NORMAL(Mean, Deviation [, Seed] )  
POISSON(Mean [, Seed])  
RANDOM(Min, Max [, Seed])
```

are the formats of the four random number functions (exponential, normal, Poisson and uniform distributions in Powersim). The last parameter ‘Seed’ within the brackets is optional. If it is left out you have no control of the random sequence. Specifying the seed (which should be different for the seven terms in x above) will give you the same result for every replication. Changing the seed values gives you a new result.

Include seven different seed values in the right hand side of the equation for x and do three replications. Did you get the same results?

Answer:

Exercise 3. A weather model

(Modelling temperature and precipitation over the year.)

3.1 Introduction

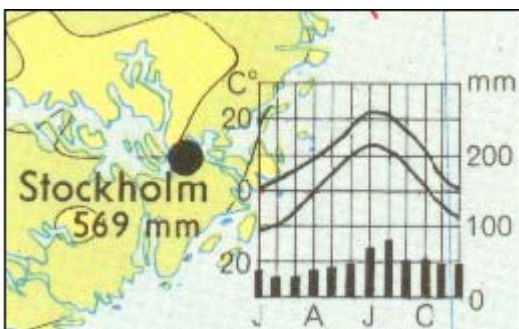
Weather is a complex concept of temperature, precipitation (Sw. nederbörd), humidity, solar radiation, wind, pressure etc. Different aspects of the weather are often crucial parts of a simulation model in ecology, agriculture, hydrology, system of hydroelectric power stations etc. For example a growing crop is dependent on radiation from the sun, temperature and water. The harvesting may require that the fields are not to wet. A cold winter may reduce a population. A wet summer may cause the spread of some disease. A hydroelectric power station has to control the water level upstreams and is dependent on the amount of water collected.

Depending on the *purpose* with the model it may be important to include more or less exact data about the weather over time. This may be done in two different ways:

1. As time-series data. Such data may e.g. be given in table form in Powersim or as data in a spreadsheet (and perhaps imported to Powersim). The advantage of this approach is that you get data from real measurements so that statistical averages, variations, correlations etc. are authentic. Disadvantages are that data can be lacking, it is time consuming if you need large amounts of data, you can't easily adjust data to a hypothetical situation to see what happens. (Say that you want to test what happens when the number of rainy days doubles. How do you implement that and what is the connection to other quantities like temperature and solar radiation?)
2. Build a sub-model describing the important aspects of the weather. In this case you should certainly not try to include huge meteorological models requiring a super-computer. Instead you should focus on those aspects that are important according to your purpose; say, temperature and precipitation. Also in this case you need real data from the location under study to fit your parameters. Depending on your purpose and how exact you must be the sub-model will be more or less complex. You will probably not reach the realism of time-series data with this approach. On the other hand the model may be good enough and easy to adjust in form of a number of parameters.

Below we will study how you can build a simple sub-model for temperature and precipitation over the year in say Stockholm.

3.2 Temperature and Precipitation over the year in Stockholm



Additional information: The number of days with precipitation is in average 42 per year.

Figure 5. A diagram of temperature and precipitation over the year for Stockholm. From Nordisk Skolatlas 1964.

The two temperature curves in Figure 5 are average daily max and min temperatures. Although not exactly the same, we take the average of these two curves as the *mean temperature* and the distance from this mean to max and min as \pm the *standard deviation*.

We also see that the shape of the mean temperature over the year curve is close to sinusoidal. If we, with regard to the purpose, decide that a sinusoidal description is acceptable it's fine, otherwise we use a table look-up function to describe it in detail.

The precipitation bars in the figure have min in February and max about half a year latter. Also these variations seems close to sinusoidal.

After the construction of the model the information from Figure 5 will be used to tune the model parameters ("model fitting").

3.3 The weather model

We will now construct a stochastic model of the weather in terms of temperature and precipitation. If the temperature is above zero the precipitation will be in form of rain, else it will come as snow. In the latter case it will stay frozen until its melt because of higher temperature.

The units will be **Day** (i.e. 24 hours), **°C** and **mm H₂O**.

The time horizon is one year, but since the weather is cyclical the simulation may run for any number of years.

3.3.1 Temperature

To describe the temperature we start with the mean annual temperature and add the seasonal variations. To this we add the daily fluctuations. For the seasonal variations we use a sinus or cosinus function. (cos is the same as sin but translated so that its peak occurs at time zero. Zero is a nicer offset to refer to when we later moves the peak.) Thus:

$$Temp = a + b \cdot \cos(\phi) + r$$

with

$a = T_YearMean$	[Mean annual temperature]
$b = T_YearAmp$	[Amplitude of annual variations]
$\phi = 2 \cdot \text{Pi} \cdot (Time - T_PeakDay) / 365$	[Location of temperature peak: $2\pi \cdot Time / 365$ gives $Time$ in days and $(Time - T_PeakDay)$ shifts the curve to have the peak at $T_PeakDay$.]
$r = N(0, T_DayVar)$	[Daily variations round $a + b \cdot \cos(\phi)$. A Normal distribution with mean 0 and T_DayVar as Standard deviation is used.]

With this sub-model the daily random variations around an annually fluctuating average temperature can be simulated.

Now adjust the parameters $T_YearMean$, $T_YearAmp$, $T_PeakDay$ and T_DayVar in accordance to Figure 5 and test the model.

Comments:

3.3.2 Precipitation

Precipitation also has a typical annual cycle. During the winter it is relatively dry and in late summer considerably more rain will fall. To model the annual precipitation cycle a cosin function, *YearCycle*, will be used. (Again, a table look-up function would be still more accurate.)

To decide if there will be any precipitation at a certain day we use a uniform random number generator, *RANDOM(0,1)*, and test if the random sample is smaller than a certain number *Risk_Precip*. In that case the amount of precipitation, *Amount*, will fall. Thus:

$$\begin{aligned} \text{Precipitation} &= \text{IF}(\text{RANDOM} < \text{Risk_Precip}, \text{Amount}, 0) && \text{[Test if precipitation today.]} \\ \text{Risk_Precip} &= \dots && \text{[See Figure 5.]} \\ \text{Amount} &= \text{EXPRND}(\text{Day_Av_If_Precip}) * \text{YearCycle} && \text{[YearCycle modifies Amount.]} \end{aligned}$$

The *Amount* draws a random number from an exponential distribution with average *Day_Av_If_Precip*. The exponential distribution is chosen because the amount is usually small but can at rare occasions become very large. [If you think the distribution is crucial you should collect statistics and build a more accurate random mechanism. You may then replace the exponential random number generator in *Amount* by an own made function where you map *R(0,1)* random numbers through a table look-up function to achieve a distribution more like the real distribution of the amount of precipitation during a day.]

Finally, *Amount* of precipitation is also modified by the sinusoidally varying *YearCycle*. The *YearCycle* varies sinusoidally over the year around a mean value equal to *one* with an amplitude, *Amp*. Again we use $2\pi \cdot \text{Time} / 365$ to get *Time* in days and $(\text{Time} - P_PeakDay)$ shifts the curve to have the peak at *P_PeakDay*. Thus:

$$\text{YearCycle} = 1 + \text{Amp} * \text{COS}(2 \cdot \text{Pi} \cdot (\text{Time} - P_PeakDay) / 365)$$

The ratio of max to min is denoted *MaxToMin* and will be recalculated into the amplitude, *Amp*, of the cosin function. *Amp* then gets the form:

$$\text{Amp} = (\text{MaxToMin} - 1) / (\text{MaxToMin} + 1)$$

Now, tune the constants: *Risk_Precip*, *Av_IF_Precip*, *MaxToMin* and *P_PeakDay* in accordance to Figure 5. [Note that since the annual average of *YearCycle* is one, the expected annual precipitation will be: $365 * \text{Risk_Precip} * \text{Day_Av_If_Precip}$. This makes it easier to fit the parameters to give correct average precipitation. This is why we put extra effort to give *YearCycle* a mean equal to one.]

When the temperature goes above zero the precipitation comes as rain, else it comes as snow:

$$\begin{aligned} \text{Rain} &= \text{IF}(\text{Temp} > 0, \text{Precipitation}, 0) \\ \text{Snow} &= \text{IF}(\text{Temp} \leq 0, \text{Precipitation}, 0) \end{aligned}$$

As long as the temperature stays below zero the precipitation is accumulated in form of snow and ice.

$$\text{SnowBuffer}(\text{time} + \Delta t) = \text{SnowBuffer}(\text{time}) + \Delta t * \text{Snow}(\text{Time}) - \Delta t * \text{Melt}(\text{Time}).$$

When the temperature then rises above zero the melting will start and its rate is a function of temperature, the size of the accumulated snow buffer and of a constant, *Melt_const*:

$$Melt = IF(Temp > 0, Melt_const * Temp * SnowBuffer, 0).$$

(This is *not* a very accurate formula. Further, the solar radiation should also play a role.)

The model is calibrated by the rate of melting given in the constant: *Melt_const*. We assign the value 0.3 to this parameter.

A possible layout of the weather sub-model is shown in Figure 6.

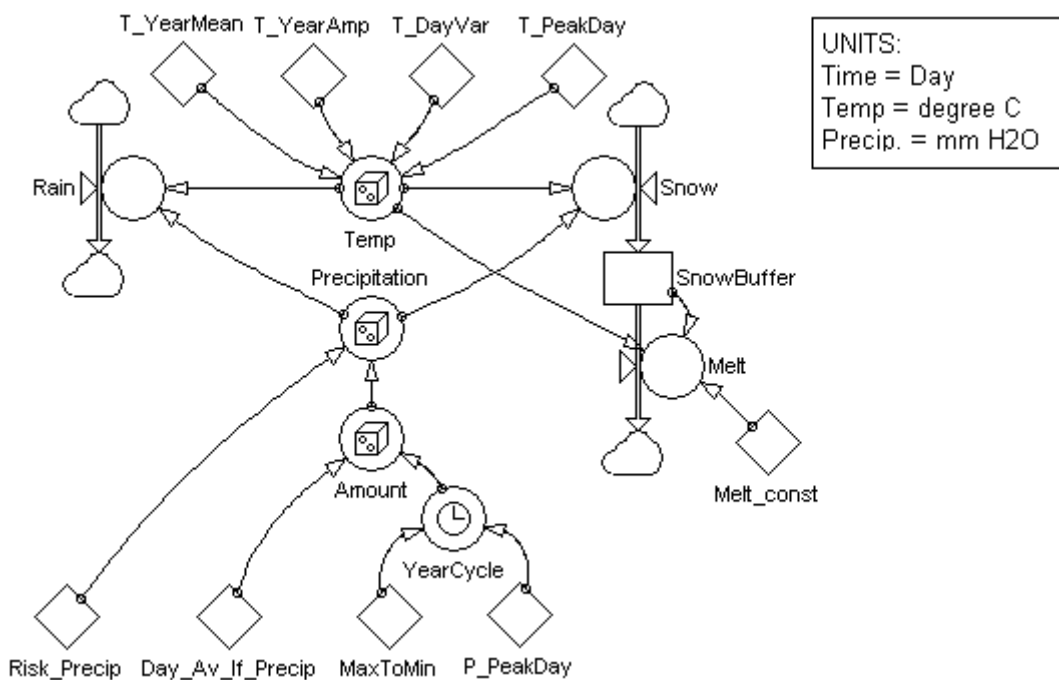


Figure 6. A possible layout of the weather sub-model. There are nine parameters to fit to the studied weather system in this case. (For a practical use rain and melting water may for example drain the humus from the soil or fill the water reservoirs for the power stations or affect the crop.)

3.3.3 Random numbers are affected by the time-step chosen

So far we have made the model for a step-size $\Delta t=1$. Usually, Δt is freely adjustable to a small enough value. But there is one exception: **Random numbers are step-size dependent!** The problem is that a new random sample is drawn for every Δt . This will not produce any errors with regard to the averages, but to the variations. For example if there is 12.5 per cent risk per day of precipitation, the statement: $Precipitation = IF(RANDOM < Risk_Precip, Amount, 0)$ will give precipitation about each eighth day for $\Delta t=1$. But changing Δt to 0.1 would give precipitation most of the days (although with the amount $\Delta t * Amount$ per time-step.) This may be OK if you run the hydroelectric water station but not if you are a tourist – so the purpose of your study decides if you have to deal with it.

Another problem comes when you cumulate a random variable like the snow into a snow-buffer. The accumulation with $\Delta t=0.1$ will result in ten times more snowfalls, each with a tenth of the content. But the variations in the snow-buffer will lose most of its variations because it represents an average based on ten times more samples. (“The law of large numbers”.)

One way to eliminate such problems is to sample the random functions once a day. This sample will then be used for the whole day independent of your time-step (as long as it is not larger than one). In principle you should write e.g.: $SAMPLE(RANDOM(0,1), FirstTime, Interval)$ with $FirstTime=0$ and

Interval=1 if the time unit is one day. (Unfortunately, you may not nest functions like RANDOM, IF etc. inside the SAMPLE function, so you need to include extra auxiliaries – one holding SAMPLE and the other holding the random function.)

With these *very important comments* we leave this subject for now and stick to $\Delta t=1$ in this exercise.

3.4 Testing the weather model

Simulation starts in January 1 and proceeds day per day with 365 days per year. If you want you may assign an initial value (other than zero) to the *SnowBuffer* if you think there typically is snow on the ground at January 1.

To test the model add a number of diagrams for temperature, rain and snow. Also cumulate the precipitation to see if you get a reasonable amount per year.

Simulate the model in five replications of one year each. Does the results seem to be in accordance to those in Figure 5 and to what you know about weather?

Depending on what you want to use the model for, you should test it by asking relevant questions. These could for example be:

- What was the highest and lowest temperature you got? Answer:
- Does the variations in temperature seem reasonable? Answer:
- Is the annual precipitation reasonable? Answer:
- During which part of the year did you get most precipitation? Answer:
- Which months did you get snow on the ground? Answer:
- How many days per year with snow on the ground did you get? Answer:
- According to the model, on how many days a year can you go skiing? (Temperature below zero and at least 10 cm of snow (=10 mm in *SnowBuffer*).

Answer: days/year.

Comments on what to improve on:

.....

3.5 Simplifications made

This model focuses on how *mean* temperature and precipitation changes around the year and on stochastic daily temperature and precipitation *variations*.

Such a simple model will necessarily contain a number of simplifications and a number of factors not included. For example the melting is also an effect of solar radiation which is not included. Further we have assumed the same risk of precipitation every day and let the amount vary over the year. Perhaps we should instead varied the risk of precipitation? Nor have we included correlations. For example it is probable that precipitation and low temperature are correlated. Further, low pressures tend to pass in

sequences of three to five in a row. No autocorrelation for this is included. If these simplifications are serious or not, depends on what is important according to the purpose of your study. (If you use real data, the nature will take care of all these problems!)

WARNING: Don't use this as a standard model. Always start with your own purpose formulated in operative terms to see what data about the weather you need and how accurate they have to be. Then find time-series or build an appropriate model. The exercise above may be a good help for ideas about how to proceed.

Acknowledgement: This model is a slight modification of a model by Tomas Thierfelder in "Dynamisk simulering av karakteristisk säsongsvariation i myrars humusbuffert".

Exercises 4-7 are about *population models*. A population consists of a *discrete* (i.e. *integer*) number of individuals. These individuals may be physical quantities like electrons, atoms or molecules or animals, plants or micro-organisms in biology or humans in e.g. epidemiology. It is then usually important to model the population as a number of individuals rather than as a continuous mass. The Poisson distribution plays a central role in population models. Further, the Poisson distribution is time scalable so that the step-size does not affect the model statistically!

Exercise 4. Stochastic modelling of radioactive decay

(Correct modelling or just adding noise.)

A deterministic model of radioactive decay is:

$$\begin{aligned} dx/dt &= -x/T && \text{where } T=10 \text{ time units is the time constant for the decay.} \\ x(0) &= x_0 && \text{where } x_0 \text{ is initial number of atoms.} \end{aligned}$$

If the initial number of atoms is large this is a good description of the process. But assume that we study a very small sample of $x_0=100$ atoms. Then stochastic variations have a considerable impact. How to include that?

What you often see is that the modeller just adds noise to make the result ‘look nicer’. This is a very stupid idea! If randomness should be included, it must be modelled in a correct way – otherwise you just distort the model and its behaviour. To see what happens we just add noise e.g. uniform or normally distributed. That is:

$$\begin{aligned} x(t+\Delta t) &= x(t) + F(x,t) \\ F(x,t) &= -x(t)/T + \text{Noise} && \text{where e.g. Noise=RANDOM}[-a, a] \text{ or Noise=Normal}[0, SD]. \end{aligned}$$

It is of course important that the noise has the expected value of zero, since you just want to include fluctuations – not adding a positive or negative inflow. The size of a or SD (for standard deviation) will determine the size of the random variations.

In Powersim make 1) a deterministic model and 2) a stochastic model where you just add noise and plot the results in a common diagram. If you haven’t added further errors the fluctuations of the stochastic solutions should roughly be around the deterministic one.

Now make a number of simulations. What peculiarities do you see? Name at least 4 impossible artifacts!

- Answer: 1)
- 2)
- 3)
- 4)

Finally, change Δt by a factor of ten. What happens?

Answer:

.....

Poisson Simulation

Poisson Simulation is a method to model stochastics within Continuous System Simulation for *population models*. A population model is a model where the state consists of an integer number of individuals, like 3 men, 12 customers, 16 plants, 74, rabbits, 17 ships, 100 atoms, 50 infected people. Flows to or from such a state may then only add or subtract integer number of individuals.

An individual arriving to or departing from a state is called an *event*. When events in a flow occur randomly, independently and singly, the flow is a Poisson process defined by the single intensity parameter, λ . Often the intensity of this flow varies over time implying a non-stationary Poisson process where $\lambda=\lambda(t)$. That λ varies causes no problem because $\lambda(t)$ can be stepwise constant during the short time interval, Δt , – just like other quantities in CSS. Stochastically, the probability of an event during the finite time interval, Δt , then becomes proportional to the length of the interval. The number of entities during the time interval, Δt , then becomes Poisson distributed and described by $Po[\lambda*\Delta t]$. In the case of discrete entities, it is thus natural to base the stochastics on the Poisson distribution.

How stochastics should be implemented

In the stochastic case the *expected* outflow is still $F=X/T$ per time unit or $\Delta t \cdot F = \Delta t \cdot X/T$ during the time interval Δt . Since the properties of single events and independency are fulfilled, the number of events during Δt should be Poisson distributed with the intensity $\lambda=X/T$. Thus, the outflow during the time interval Δt has a Poisson distributed variation denoted $Po[\Delta t \cdot X/T]$. The flow rate then becomes: $Po[\Delta t \cdot X/T]/\Delta t$ decays per time unit. Therefore, the model is reformulated as:

$$X = X + \Delta t \cdot (-F)$$

$$F = Po[\Delta t \cdot X/T] / \Delta t \quad [The\ decay\ is\ now\ stochastic.\ The\ rest\ is\ unchanged.]$$

$Po[\sim]$ means that for each time interval, Δt , a random number is sampled from a Poisson distribution with the actual parameter value specified in the expression within the brackets.

One advantage with this mechanism to introduce randomness is that the time interval Δt can be changed to handle the dynamics properly without distorting the stochastic behaviour of the model.

A special feature of the Poisson Simulation model is that if the state variable is initiated to an integer value it will stay an integer, as opposed to a deterministic CSS model where the state can take any real value.

In Figure 7, the deterministic model, the stochastic model where noise was added and the Poisson Simulation model of radioactive decay described above are shown as Forrester diagrams.

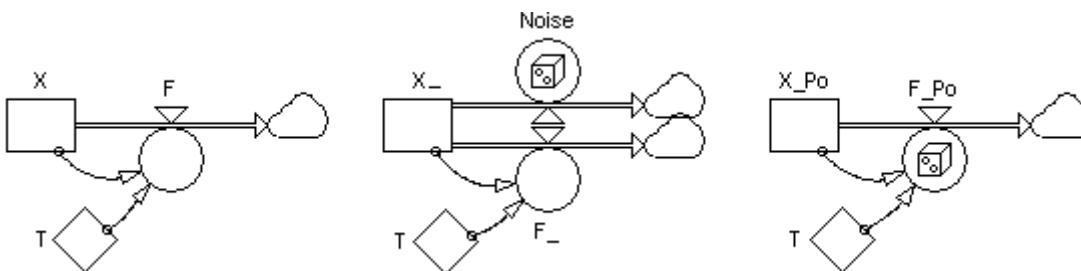


Figure 7. Three radioactive decay models.

Perform a number of replications with the Poisson Simulation model.

Which of the problems of the 'noise model' do you find here?

(Do you get integer values of the number of atoms?, can the number of atoms suddenly increase?, Can you get a negative number of atoms?, Are the stochastic variations the same for say 100 and 10 atoms left?, Do you still get variations after that all atoms have decayed?)

Answer;

.....

What happens when you decrease the step-size by a factor of ten or a hundred? Compare with what happens for the 'noise-model'.

Answer:

Exercise 5. An epidemic model

(Is the average of a stochastic model equal to that of a deterministic one? Estimates of averages, variations and confidence intervals.)

Here, we will return to a an epidemic model similar to what you have studied earlier. We study how an infectious disease hits a group of people (animals or plants) that are susceptible to an infection. The individuals that get infected become infectious (and sick) and will recover after some time to be become immune to the disease.

We can divide the population in three different stages of individuals: the **susceptible (S)**, the **infectious (I)** and the **recovered (R)** stage. Such models, therefore, are called SIR models.

The susceptible stage is the individuals that are not infected but have the potential of catching the infection. The infectious stage consists of people that have become sick. The recovered stage consists of the individuals that have been sick but are now recovered. See Figure 8.

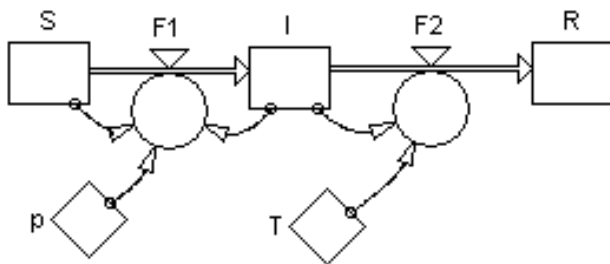


Figure 8. A deterministic SIR models of an infectious disease.

The epidemic development is dependent on the following rules:

- 1) The total size of the population is constant during the time period we are studying. At the start of the study the susceptible population has the size 1000 individuals, and one person has just become infected ($I(0) = 1$).
- 2) The number of individuals that catch the infection per time unit is proportional to the number of susceptible individuals (S) and to the number of infectious individuals (I). The proportional constant (the spread factor for the disease) is $p=0.0003$ per day and person.
- 3) The time for the infectious stage is typically about $T=4$ days, and then the sick people become immune.

Select a step-size and test if it is small enough. What step-size will you use?

Answer: $\Delta t = \dots\dots\dots$ time units. How many individuals become sick? Answer: $\dots\dots\dots$ persons

According to the deterministic theory, if an epidemic will occur depends on the so called *Reproduction number*: $R_0=p*S*T$ that tells how many individuals that are infected by each infectious person. If $R_0>0$ an epidemic will occur! What is R_0 in this case? Answer: $R_0= \dots\dots\dots$

A stochastic SIR model

Now make the model stochastic by implementing the Poisson mechanism in F1 and F2 of the deterministic model.

Perform a number of replications to see the number of individuals hit by the disease. Write down the numbers from 10 replications.

Answer:

Do you always get an epidemic? Does it look like you in average get the same results as you got from the deterministic model?

Answer:

As you notice, it is quite tedious to perform, say, 1000 replications in this way and thereafter calculate average, variations, confidence intervals etc. The 1000 replications will also take some time. Therefore, we first try to reduce the execution time. Firstly, don't use a too small Δt . ($\Delta t=0.1$ is enough.) Then, since it often happens that the one and only infectious individual recovers before he has infected anyone the epidemic can sometimes be over very quickly. Simulating until, say, $\text{Time}=1000$ will then be a waste of time. Instead make an auxiliary and call it BREAK for example. Open BREAK and use the function `STOPRUNIF(I < 0.5)` that stops the replication when I becomes zero. ($I < 0.5$ is better than $I=0$ since the binary representation may represent zero by e.g. 10^{-17} .) Also add another auxiliary denoted DURATION and define it as TIME to see the time when the epidemic is over. Test the model to see that it works well and save the model.

Now we open the supervisory program StocRes (for StOChastic RESults) that you find on the desktop. You will then get an interface like that in Figure 9.

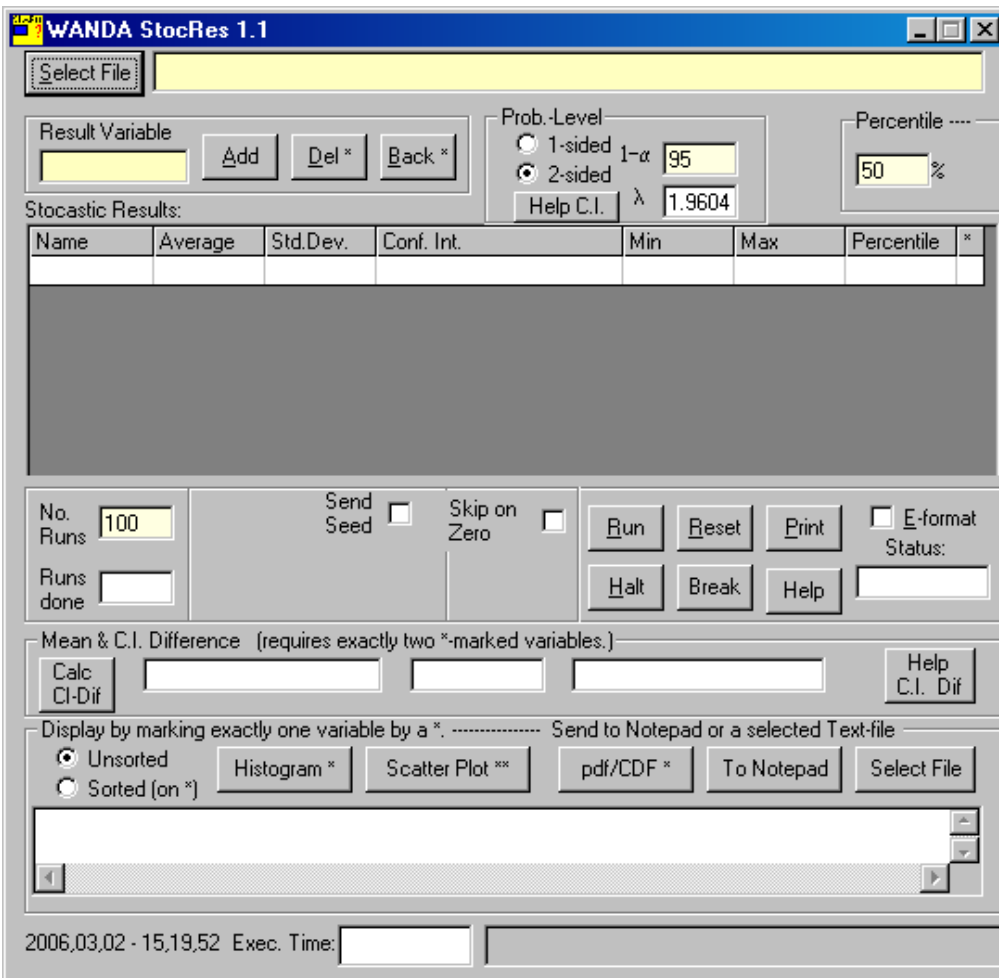


Figure 9. The StocRes interface for calculating statistics from many replications of a model.

- Start clicking [Select File] to specify and connect to your model.

- In the “Result Variable” field write the name of the Resistant state variable and click [Add]. Then write DURATION and add that variable.
- Set “No. Runs” to 1000.
- Click [RUN] and let StocRes do the job.

When the 1000 replications are done you see the statistics in form of Average, Standard deviation, Confidence interval, Min, Max etc for each studied quantity.

You may [Print] the results on paper. There is also possible to mark one or two of the variables by a star (*) to make a histogram, Scatter plot or to dump the results from each simulation onto a file for further analysis.

What was the average number of individuals that were hit by the disease and its confidence interval?

Answer: Average[R]=..... (C.I.= $\bar{\quad}$ ) individuals.

What was the average duration of the epidemic and its confidence interval?

Answer: Average[Duration]=..... (C.I.= $\bar{\quad}$ ) time units.

With how many percent differs these results from that of the deterministic model?

Answer:

*Does the reproduction number $R_0=p*S*T > 1$ grant an epidemic?*

Answer:

*Can there be an epidemic if $R_0=p*S*T < 1$? (Test it!)*

Answer:

Exercise 6. A logistic population model

(Warning for using a deterministic model structure. Start from the system when making your stochastic model!)

[Start from the system when making your stochastic model! Adjusting the model structure to fit observed stochastic variations - A logistic model. Warning for using a deterministic model structure.)

The deterministic case

A deterministic, logistic model has the form: $dx/dt = ax - bx - cx^2$; where a is the fertility rate, b the mortality rate and c is the mortality rate because of competition. The change in the state value x thus increases proportionally to x (if $a > b$) and decreases because of competition proportionally to the term x^2 (meaning that each of the x individuals competes with all the other).

But we can also write $dx/dt = a' \cdot x - cx^2 = (a-b)x - cx^2$ where $a' = a-b$. The term $(a-b)x$ is then the net flow. Further, how does the competition work? Does it result in an additional outflow of dying individuals or does it hamper the growth process (or is it involved in both)?

Depending whether we model ax and bx as separate flows or as a net flow $(a-b)x$ and depending on whether we assume that competition is a cause of additional death or a cause of reduced fertility we can come up with the four different models in Figure 10.

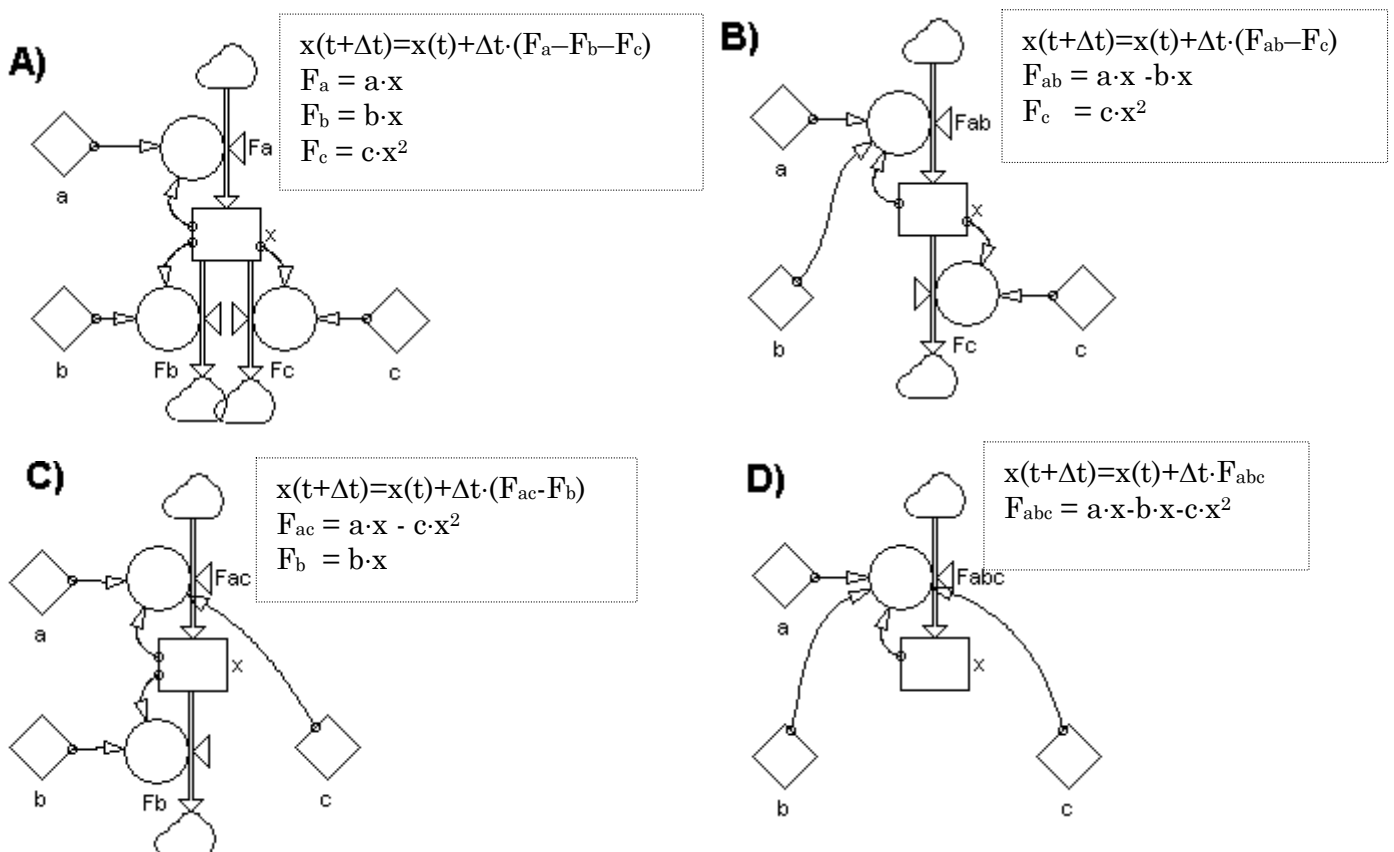


Figure 10. **A)** ax and bx as separate flows & competition as an outflow. **B)** $(a-b)x$ as a net flow & competition as an outflow. **C)** ax and bx as separate flows & competition reducing an inflow. **D)** $(a-b)x$ as a net flow & competition reducing an inflow.

In case A) we have assumed that the three flows F_a , F_b and F_c all are independent. This means that the number of births in F_a only depends on the parameter a and the value of X , and not on what happens in the other flows. A similar statements are true for F_b and F_c .

In case B) we have assumed that a *net* increase is because of $(a-b)$ and the value of X . Implicitly we have then assumed that no other death will occur except for that from competition. Does this seems reasonable?

In case C) the underlying assumption is that the effect of competition causes a reduced increase of the population.

In case D) the assumption is that no death whatsoever can occur, bX and cX^2 only will reduce the increase.

Mathematically this four models have all the differential equation: $dx/dt=ax-bx-cx^2$. They should, therefore, give exactly the same results. In deterministic modelling there is no reason to discuss the realism of the different structures because they gave all the same results so we couldn't decide which behaved more similar to data from a studied system. (Although we could use biological knowledge to at least rule out case D) as biologically infeasible.)

To check the models, set $a=0.2$, $b=0.1$ and $c=0.01$ and simulate the four models for 100 time units with a small step, e.g. $\Delta t = 0.05$. What number of individuals did you get?

Answer: A) B) C) D)

Poisson Simulation models

Now we will consider that the increase or decrease in the flows are individuals that at some points in time enter or departure from the state x . Assume that the events causing an increase or a decrease of individuals happen independently of each other, then the flows are Poisson processes that can be modelled as $PO[\Delta t \cdot \lambda] / \Delta t$.

Transform the four deterministic model into Poisson Simulation models and study their behaviours.

Note: In deterministic simulation a way to grant accurate calculations is to keep Δt small so that the change of the state(s) is small during Δt . For stochastic models making Δt small is also crucial – but here a number of events could happen within Δt even when it is short. In some models you can accept it while in other you may pass into a forbidden situation. For example assume that we in case A) have: $X(t+\Delta t)=X(t)+Po[\Delta t \cdot a \cdot X] - Po[\Delta t \cdot b \cdot X] - Po[\Delta t \cdot c \cdot X^2]$ and $X=1$. It is then possible that the first Po -term becomes zero and the second and third both become equal to one. $X(t+\Delta t)$ then gets the value -1 . In the following $Po[\Delta t \cdot a \cdot X]$ and $Po[\Delta t \cdot b \cdot X]$ will be zero because the $Po[\text{Neg number}]=0$. But $\Delta t \cdot c \cdot X^2$ is a positive number because of the square so $Po[\Delta t \cdot c \cdot X^2]$ will continue to produce departures in an increasing rate and X can even get large (negative) enough to produce overflow in the computer calculations. A reduced Δt usually is a good (but not absolutely sure) cure, because if you reduce Δt to, say, a tenth it is improbable that both departure events will happen in the same time-step. And if they don't the first event will take X to zero and there will be no second as long as X remains at zero.

[If you want to be *absolutely* sure you can guard against outflows making X negative. E.g.: $\Delta t \cdot F_{bc} = \text{MIN}(Po[\Delta t \cdot b \cdot X] + Po[\Delta t \cdot c \cdot X^2], X)$. In this case the combined output is limited to the value of X .]

Now, make a diagram so you can follow the development of the states by eye. Save the model and Start StocRes. Connect to the stochastic model, Study the states for the stochastic four models (XA,

XB, XC and XD) and run 500 replications. (If you get problems of the type described above so decrease the step-size.

Model	Average	Standard deviation	Confidence interval	Min	Max
A)					
B)					
C)					
D)					

Now its time to analyse the results. Firstly, in deterministic modelling there were no reason to discuss the realism of the different structures because they gave all the same results so we couldn't decide which behaved more similar to data from a studied system.

In the stochastic case, however, we see that we get different results for the four cases. This is because the four structures behave statistically different. And dynamics and stochastics interact!

Why did you get considerably smaller values of $X(t=100)$ in the stochastic case (for cases A, B and C)?

Answer:

Why did you in case D almost always end up at $XD=10$?

Answer:

Which model do you think is most realistic?

Answer:

As you probably have noticed, making the model stochastic is a considerably tougher task. But if you study populations that are not very large, you usually get wrong estimates and conclusions from a deterministic model.

Further, the stochastics that you obtain from data from system under study contains information that should not be thrown away by smoothing. This information you can take care of by fitting the three parameters a, b and c (the deterministic model had only two independent parameters $a'=(a-b)$ and c).

Say that you have concluded that A is a reasonable model.

The average of a stochastic variable X is defined by: $m[X]=E[X]=\sum x_i/N$ for the N data x_1, x_2, \dots, x_N .

The variance of a stochastic variable X is defined by: $V[X]=E[(X-m)^2]$ where m is the average. One way to calculate the variance is $V[X]=E[X^2] - (E[X])^2$. The variance $V[X]$ is a measure of the variations around the average and is meaningful if we have a rather constant average. However, here we are only interested in estimating the parameter values of a, b and c so that the behaviour of the system and the model become about the same.

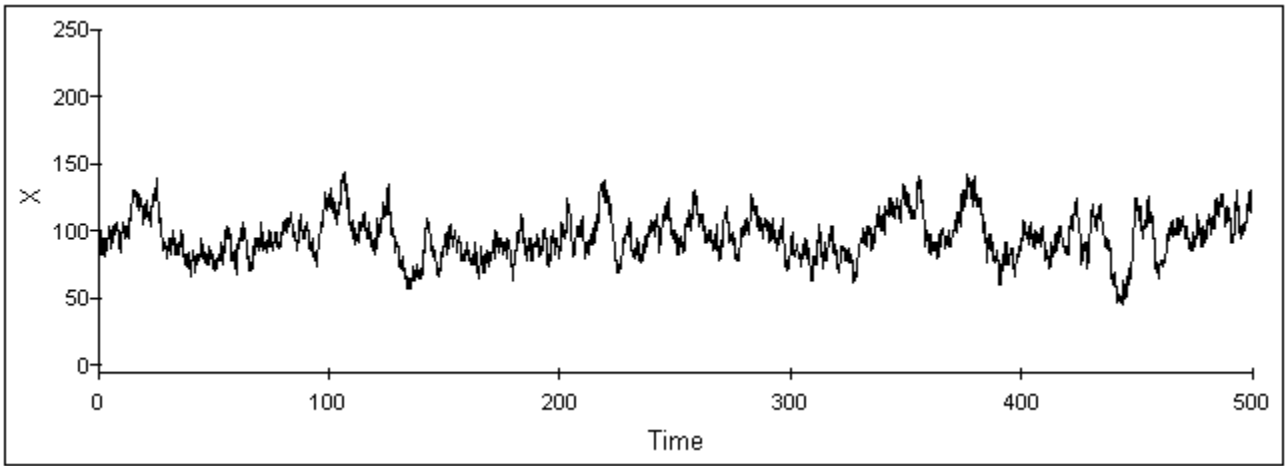


Figure 11. Data from the studied system. What are the average and the variance?

Analysing the data from the studied system (in e.g. Excel) gives $E[X]=96.8$ and $\text{Var}[X]=257.7$ ($\text{SD}[X]=16.1$; $\text{SD}=\sqrt{\text{Var}}$).

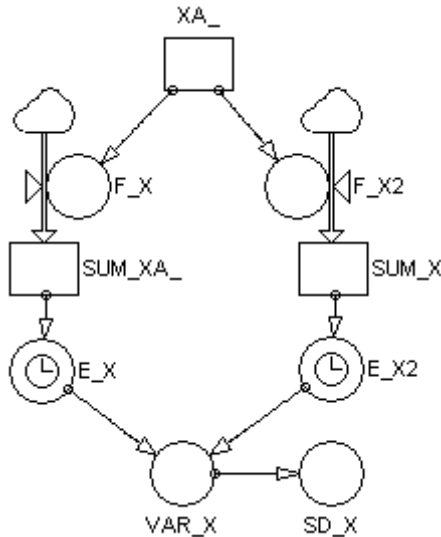


Figure 12. A simple device to measure average ($E[X]$) and variance ($\text{Var}[X]$) or Standard Deviation ($\text{SD}[X]$).

Start with fitting the model so you get a reasonable average: that is (a-b) and c needs proper values. When this is done increase or decrease a and b equally. This will not affect the average, but the variations become larger when a and b are increased. (Be careful to initiate X to a proper value!)

The estimates were: a =, b = and c =

Exercise 7. A stochastic prey-predator model

(How stochasticity excites dynamic variations and how new qualities are obtained.)

The Lotka-Volterra equations describe a prey-predator system for two species e.g. Rabbits (x) and Foxes (y) by differential equations. The rabbits breed at a rate proportional to their number x . They die because of encounters with the foxes, which is proportional to $x \cdot y$. Also, there is competition among the rabbits, where each rabbit competes with all the others. Competition, therefore, is proportional to x^2 . The encounters with rabbits give the foxes energy to breed so they increase in proportion to $x \cdot y$. Finally, the fox death rate is proportional to the number of foxes, y . The Lotka-Volterra model therefore has the form:

$$\begin{cases} dx/dt = ax - bxy - kx^2 \\ dy/dt = cxy - dy \end{cases}$$

where a and b are fertility constants, c and d mortality constants, and k is a proportionality constant for competition.

By setting the derivatives dx/dt and dy/dt to zero and solving for x and y we obtain three possible stationary solutions:

- 1) $x=0$ and $y=0$, 2) $x=a/k$ and $y=0$, 3) $x=d/c$ and $y=(a-kd/c)/b$.

Setting e.g. $a=0.2$, $b=0.005$, $c=0.005$, $d=0.3$ and $k=0.001$ gives in the second case (foxes become extinct) $x=200$ and $y=0$, and in the third case (both species survive) $x=60$ and $y=28$.

To treat the system numerically we rewrite the equations in Euler's form:

$$\begin{cases} x(t+\Delta t) = x(t) + \Delta t \cdot (F_1 - F_2 - F_3) \\ F_1 = ax \\ F_2 = bxy \\ F_3 = kx^2 \\ y(t+\Delta t) = y(t) + \Delta t \cdot (F_4 - F_5) \\ F_4 = cxy \\ F_5 = dy \end{cases}$$

Initialising this simulation in steady state with $x(0)=60$ and $y(0)=28$ give the trivial results of two constant lines for x and y . But even if we disturb the system to generate variations, these will die out.

Build the model in Powersim and initialise away from the equilibrium, e.g. $x(0)=150$ and $y(0)=10$.

Run the model and sketch the result in Figure 13.

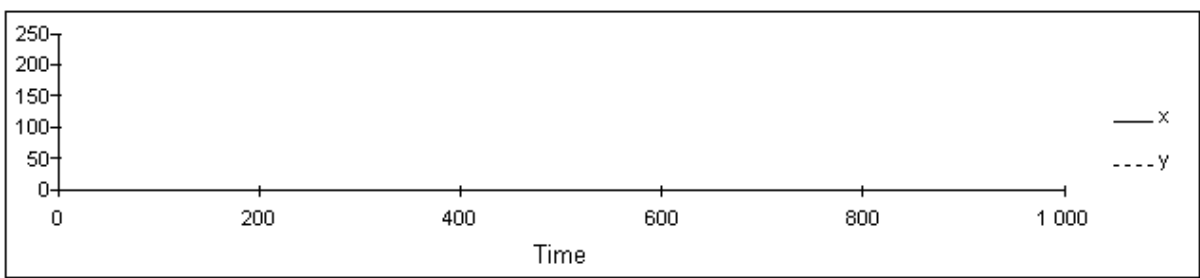


Figure 13. The deterministic system is damped by the competition and then stays in an equilibrium.

Now assume that the flows of births deaths and competition follow the Poisson requirements. Then each flow F_x is transferred into the form: $\text{Poisson}(\Delta t \cdot F_x) / \Delta t$. The model, with stochastic flows are shown in Figure 14.

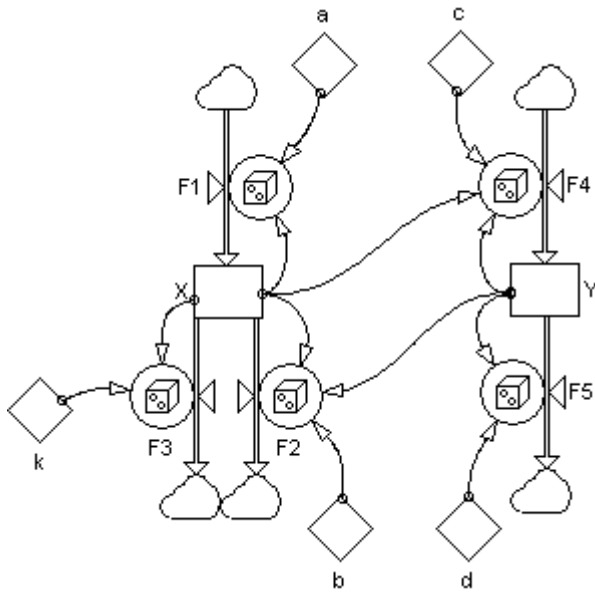


Figure 14. The structure of Volterra's equations. Randomness is introduced in each flow.

Start the simulation *from the equilibrium* values $X(0)=d/c=60$ and $Y(0)=(a-kd/c)/b=28$. Repeat the simulation a number of times and look for possible extinctions of preys or predators. Sketch the behaviour of a replication where extinction of a species happens in Figure 15.

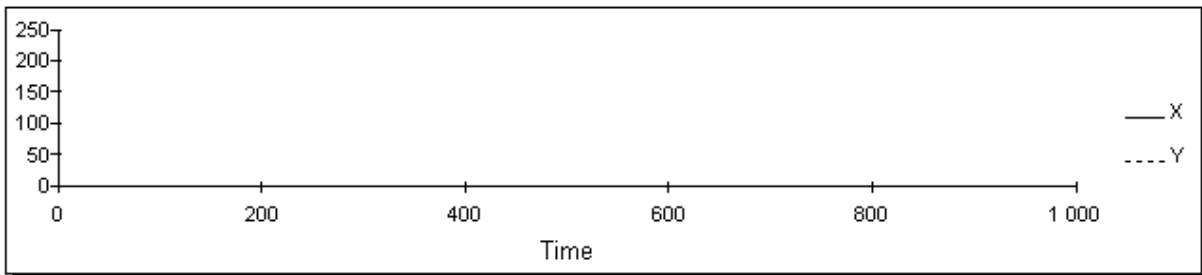


Figure 15. An example of the behaviour of the Poisson model starting at equilibrium.

In the Poisson simulation the inherent dynamics of the system is excited by the stochastic fluctuations (because we started at equilibrium). We also see that this dynamics not only causes the numbers of rabbits and foxes to vary. It also displays the periodical pattern and its typical period length of the deterministic model.

Also note that in some replications all foxes may starve to death, making the rabbits increase and fluctuate around the steady state $X=a/k=200$. (If rabbits get extinct, there will be no food for the foxes so they will soon follow.)

This example demonstrates several qualities. Firstly, it illustrates that dynamics and stochastics may generate continued oscillations also when starting at an equilibrium and even though a corresponding deterministic model would damp them out.

Secondly, the model may flip to a mode where foxes get extinct - a quality that could not happen in a deterministic model.

Finally

Also in this model we could have chosen different designs of the model that are deterministically equal but stochastically different. For example if we eliminate the predators and only keep the preys (X) (and F₂ from bxy to bx) we are back to the logistic model studied in the previous example.

Another issue that could be discussed is whether we should include an auxiliary “ENCOUNTERS” with the value $x \cdot y$. Again this would not affect the deterministic model – but in the stochastic model we would have $Po[\Delta t \cdot X \cdot Y] / \Delta t$ randomizing the number of possible meetings at only one place in the model instead of two ($F_2 = b \cdot \text{ENCOUNTERS}$ and $F_4 = c \cdot \text{ENCOUNTERS}$). This change would not affect the expected values of F_2 and F_4 but it would affect their variances. Further, this change would “synchronize” F_2 and F_4 so they become fully correlated (when F_2 is large/ small so is F_4). In the original setting F_2 and F_4 were uncorrelated. Which design is the best can only be determined from how the studied system behaves.

As seen stochastic modelling is much more tricky than deterministic, but it also gives you better possibilities to make a realistic model and to get better estimates and better conclusions and understanding.

Exercise 8. Summary of your findings

Go through your exercises and sum up your findings in your own words. Especially, differences between deterministic and stochastic modelling, simulation, model behaviour etc. are important.

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-