Differential Equations

Thomas Breuel

UniKL

Bacterial Reproduction

Bacterial reproduction

- bacteria sitting around in a petri dish
- each bacterium has some probability of dividing at a random point in time
- bacterial division is a Poisson process

Poisson process

- for very small time intervals, Δt , there is a constant probability that some event occurs
- if $N(0) = 0$, the $N(t)$ is Poisson distributed with parameter λt
- λ is average number of events per unit time
- waiting time distribution is an *exponential distribution*

Distributions

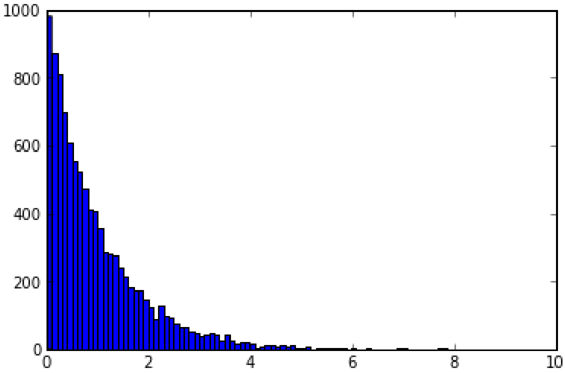
Poisson distribution:

$$p(k;\lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

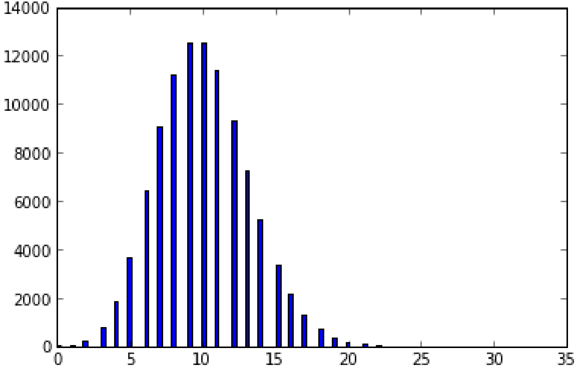
Exponential distribution:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

```
_=hist(expon.rvs(scale=1.0,size=10000),bins=100)
```



```
_=hist(poisson.rvs(10,size=100000),bins=100)
```



Direct Simulation of Bacterial Populations

Let's start with a direct, discrete event simulation of bacterial populations.

We start with a population of 100 cells.

For each cell, we just keep the time (usually, we might have an entire object representing each population member).

```
cell_division_time = list(expon.rvs(scale=1.0,size=100))
```

We keep an *agenda*, a list of objects by update times. The usual data structure for this is a *heap* or a *priority queue*.

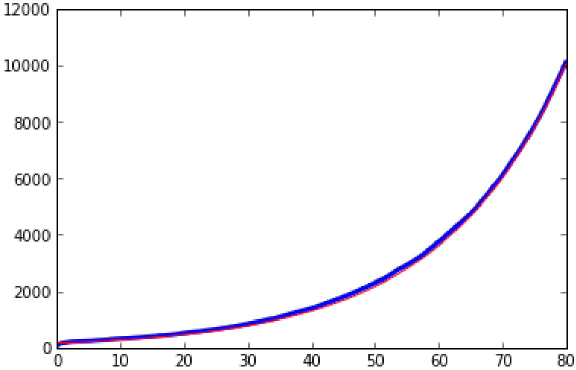
```
from heapq import *
heapify(cell_division_time)
```

The main simulation consists of taking the next item off the agenda, simulating the processs, and then pushing new events onto the agend.

```
growth = []
for i in range(10000):
    t = heappop(cell_division_time)
    growth.append((t,len(cell_division_time)+1))
    t1 = t+expon.rvs(scale=20.0)
    t2 = t+expon.rvs(scale=20)
    heappush(cell_division_time,t1)
    heappush(cell_division_time,t2)
```

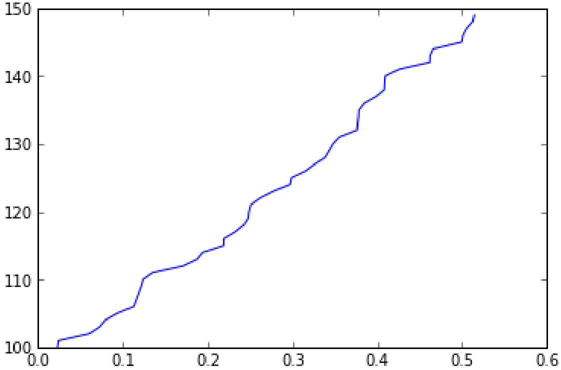
We an now see the exponential growth in action. The parameters of the exponential are related to the initial population and the probability of reproduction per unit time.

```
growth = array(growth); t = growth[:,0]; p = growth[:,1]; plot(t,p,
    linewidth=3)
plot(t,exp(0.051*t+log(170)),color='red')
```



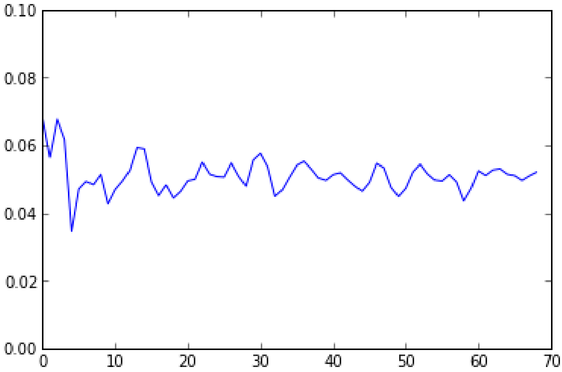
Note that on a small scale, population growth is a stochastic process. In this case, it is an instance of a *birth process*.

```
plot(t[:50],p[:50])
```



During any unit time, the increase in population is proportional to the size of the population, since each member of the population will reproduce with a certain (small) probability.

```
pops = measurements.mean(growth[:,1],labels=array(growth[:,0],'i'),
    index=arange(amax(growth[:,0])))
pops = pops[5:-5]; ylim(0,0.1); plot((pops[1:]-pops[:-1]) / pops
    [:-1])
```



Stochastic Difference Equation

In the discrete event model, we simulated each individual population increase as a Poisson process on each population member.

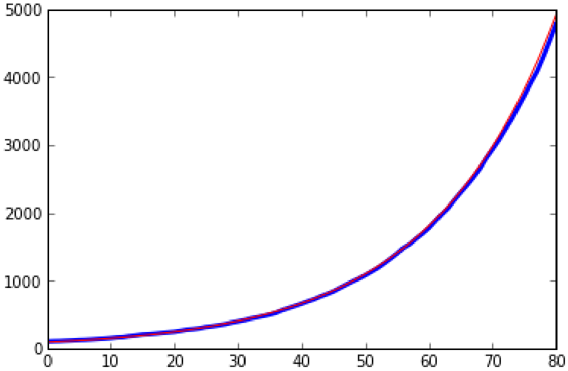
We can also ask the question: assuming we have a population of $N(t)$ individuals, what is the expected increase during the interval $[t, t + \Delta t]$.

This increase is given by the Poisson distribution. However, we are ignoring the births during the time interval itself, so this is a good approximation only for small Δt .

```
population = 100
times = []
p = []
for t in linspace(0,80,81):
    population = population + poisson.rvs(0.05 * population)
    times.append(t)
    p.append(population)
```

Again, we get exponential growth.

```
times = array(times)
plot(times,p,linewidth=3)
plot(times,exp(0.05*times+log(90)),color='red')
```

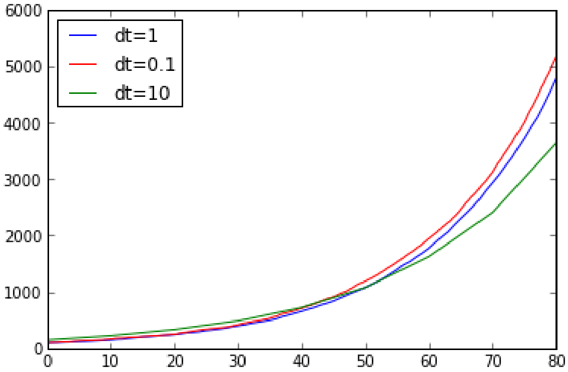


Let's repeat this with different time steps.

```
population = 100
times2 = []
p2 = []
for t in linspace(0,80,801):
    population = population + poisson.rvs(0.1 * 0.05 * population)
    times2.append(t)
    p2.append(population)
```

```
population = 100
times3 = []
p3 = []
for t in linspace(0,80,9):
    population = population + poisson.rvs(10.0 * 0.05 * population)
    times3.append(t)
    p3.append(population)
```

```
_p,=plot(times,p,color='blue')
_p2,=plot(times2,p2,color='red')
_p3,=plot(times3,p3,color='green')
legend([_p,_p2,_p3],"dt=1 dt=0.1 dt=10".split(),loc=2)
```



Poisson Distribution in the Limit

In the difference equation, we have been incrementing the population by the *expected* growth during each time period.

The actual update is a random variable drawn from a Poisson distribution.

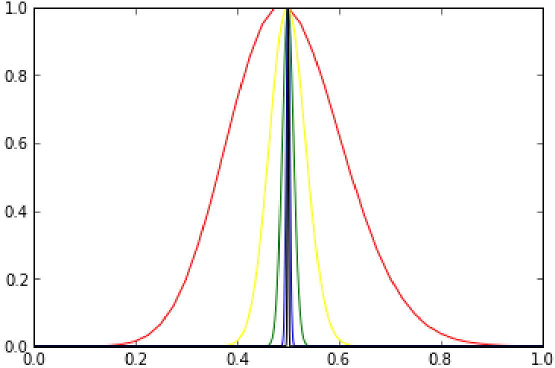
For large increments, the expected value is a good relative approximation to the actual update.

For smaller increments, the actual update is normally distributed.

For larger and larger k

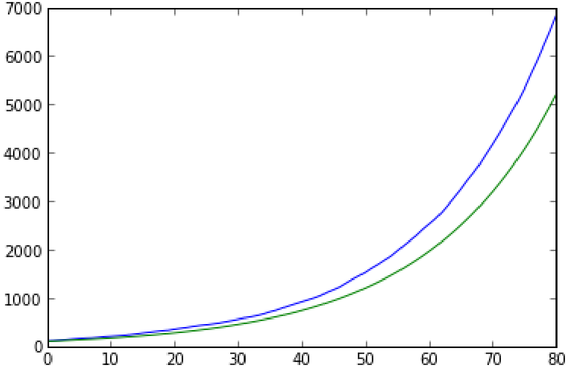
- the Poisson distribution approximates a normal distribution
- the Poisson distribution approximates a delta distribution when normalized for k

```
for n,c in zip([20,200,2000,20000,200000],['red','yellow','green','blue','black']):
    xs = arange(2*n)/2.0/n
    ys = poisson(n).pmf(arange(2*n))
    ys /= amax(ys)
    plot(xs,ys,color=c)
```



```
population = 100
population2 = 100
times = []
p = []
p2 = []
for t in linspace(0,80,81):
    population = population + poisson.rvs(0.05 * population)
    population2 = population2 + 0.05 * population2
    times.append(t)
    p.append(population)
    p2.append(population2)
```

```
plot(times,p)
plot(times,p2)
```



Continuum Limit

Continuum limit

- very large populations in absolute terms
- very large population increases in absolute terms even on small time scales

Over wide range of time scales:

$$\Delta N \approx N\lambda\Delta t$$

Express in terms of large population $N = xN_0$:

$$\Delta x \approx x\lambda\Delta t$$

Differential equation

Finite equation:

$$\frac{\Delta x}{\Delta t} \approx \lambda x$$

Forming the limit as $\Delta t \to 0$:

$$\frac{dx}{dt} = \lambda x$$

Stochastic differential equation

Finite equation, $E(\Delta t)$ is a random variable depending on the time difference:

$$\frac{\Delta x}{\Delta t} \approx \lambda E(\Delta t)$$

Forming the limit as $\Delta t \to 0$:

$$\frac{dx}{dt} = \lambda \eta(x)$$

Here, $\eta(x)$ is a kind of generalized function that encapsulates the notion of small random increments.

Summary

Concepts

- population: discrete / continuous
- time: discrete / continuous, synchronous / asynchronous
- growth: constant / stochastic

Types of simulations

- discrete event simulation (discrete population, continuous time, asynchronous updates, stochastic growth)
- stochastic difference equation (discrete or continuous population, discrete time, synchronous updates, stochastic growth)
- difference equation (discrete or continuous population, discrete time, synchronous updates, deterministic growth)
- differential equation (continuous population, continuous time, deterministic growth)
- stochastic differential equation (continuous population, continuous time, stochastic growth)

Exponential growth as differential equation

$$\frac{dx}{dt} = \lambda x$$

Shorthand for:

- growth proportional to population size
- large population, approximated with continuous variable
- growth rate λ related to Poisson process

Exponential growth

Difference equation:

$$x_{t+1} - x_t = \lambda x_t$$

Stochastic increments:

$$x_{t+1} - x_t = \Lambda_{\lambda,\sigma} x_t$$

Differential equation:

$$\frac{dx}{dt} = \lambda x$$

Note that the growth rates λ are slightly different in the three cases if we want the curves to line up.

Growth Patterns and Singularities

Exponential growth (growth proportional to population size)

$$\frac{dx}{dt} = x$$

$$x = e^t$$

Hyperbolic growth (growth proportional to square of population):

$$\frac{dx}{dt} = -x^2$$

$$x = \frac{1}{t}$$

Notes:

- exponential growth has no singularity, hyperbolic growth does
- difference equations do not have singularities