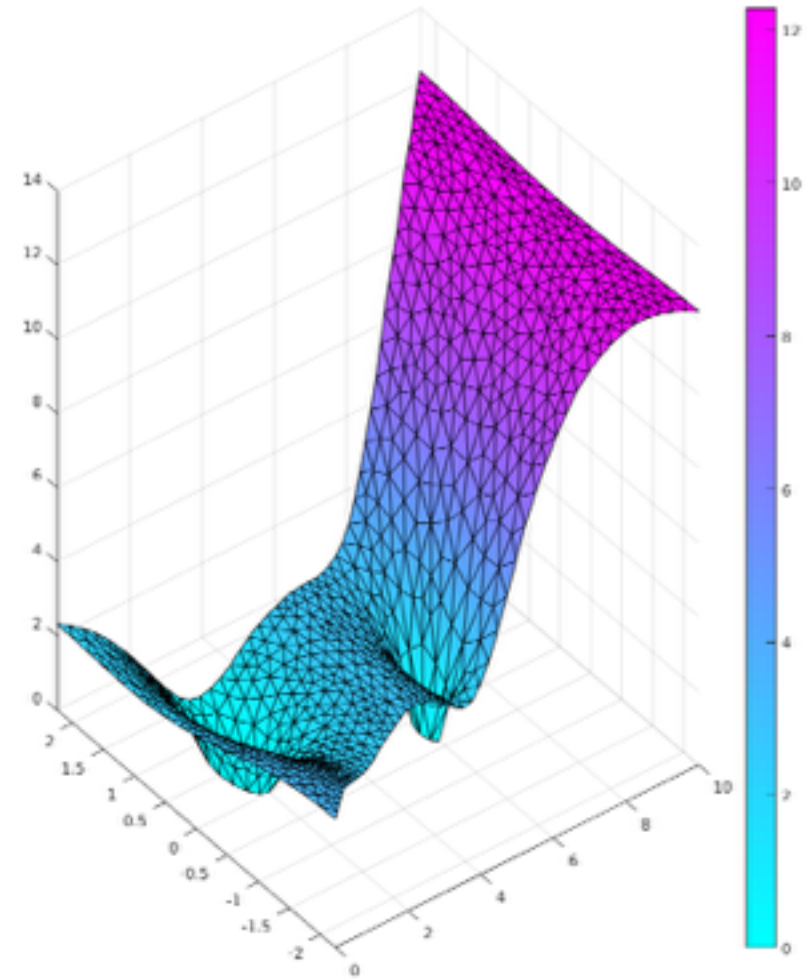
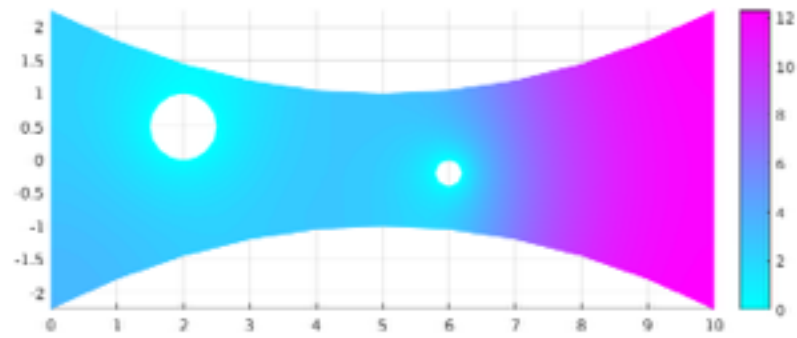
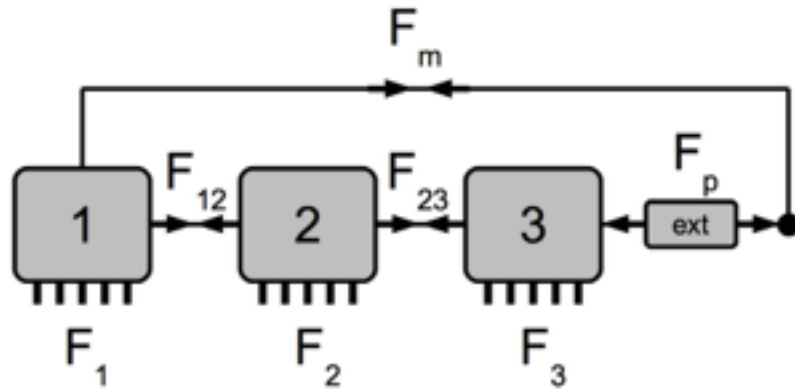


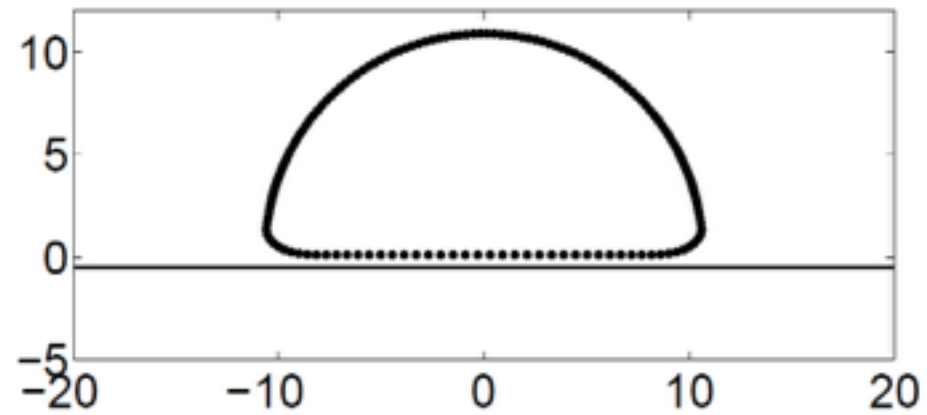
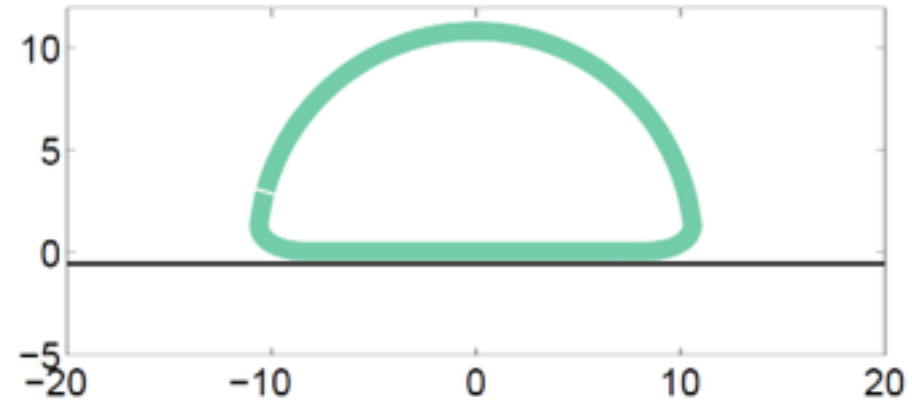
(-) Advection-reaction diffusion (in Matlab) with 3 different boundary conditions: Dirichlet, no flux, and Robin. The code uses a first-order finite difference scheme.



(-) Springs and nodes cell representation with elastic springs cell-surface binding (in C)



Spring and nodes cell structure



A bit about diffusion:

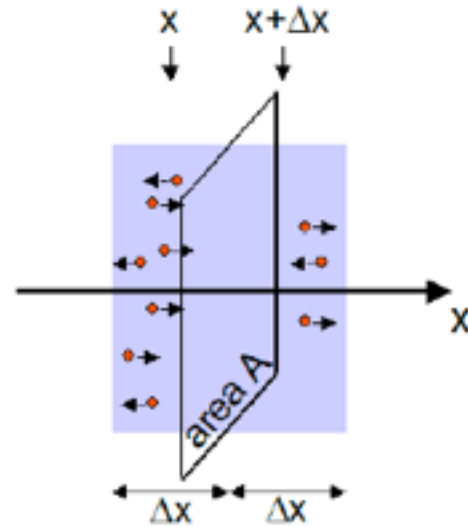


Figure Particles randomly moving along one-dimension crossing an area A

Consider Fig. illustrating particles moving along one dimension x . The particles are randomly moving. Assume there are $N(x)$ particles in the gray region on the left of area A and $N(x+\Delta x)$ particles in the gray region on the right of area A . How many particles will cross the area A to the right? Since the probability to travel to right or the left is identical $0.5N(x)$ particles will travel to the right. However $0.5N(x+\Delta x)$ will travel to the left and cross area A . Therefore the net number of crossing to the right is:

$-\frac{1}{2}(N(x+\Delta x) - N(x))$. The flux of molecules J through the area A during a short time

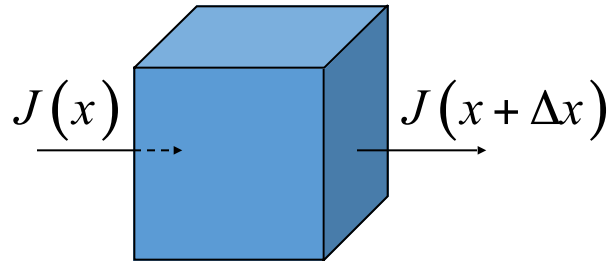
interval τ is defined as:

$$J = \frac{-\frac{1}{2}(N(x+\Delta x) - N(x))}{A\tau}$$

$$J = \frac{-\frac{1}{2}(N(x + \Delta x) - N(x))}{A\tau} \quad C(x) = \frac{N(x)}{A\Delta x}$$

$$J = \frac{-A\Delta x (C(x + \Delta x) - C(x))}{2A\tau} = -\frac{\Delta x^2}{2\tau} \frac{C(x + \Delta x) - C(x)}{\Delta x}$$

flux \rightarrow $J = -D \frac{C(x + \Delta x) - C(x)}{\Delta x} \rightarrow -D \frac{\partial C}{\partial x}$ \leftarrow concentration gradient



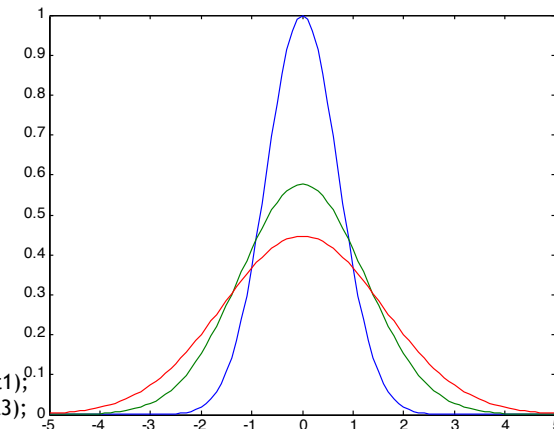
$$\frac{\Delta N}{\Delta t} = \frac{A\Delta x \Delta C}{\Delta t} = A [J(x) - J(x + \Delta x)]$$

$$\frac{\Delta C}{\Delta t} = -\frac{J(x + \Delta x) - J(x)}{\Delta x} \rightarrow \boxed{\frac{\partial C}{\partial t} = -\frac{\partial J}{\partial x}}$$

$$\boxed{\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2}}$$

$$C = \frac{C_0}{\sqrt{4Dt}} e^{-x^2/4Dt}$$

$$\boxed{\langle x \rangle = \sqrt{2Dt}}$$

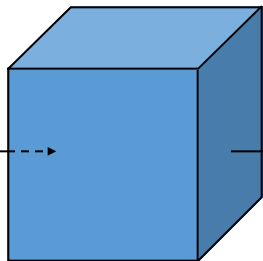


(read SolDifEq.bmp from the class web site)

```
>> x=(-5:0.1:5);t1=1;t2=3;t3=5;c1=exp(-x.^2./t1)./sqrt(t1);
>> c2=exp(-x.^2./t2)./sqrt(t2);c3=exp(-x.^2./t3)./sqrt(t3);
>> plot(x,c1,x,c2,x,c3)
```

Drift, reaction-drift, diffusion-drift, and reaction-diffusion-drift equations in cell biology

Remember the 1st Fick's law (conservation of the number of particles)?:



$$\frac{\partial C}{\partial t} = -\frac{\partial J}{\partial x}$$

Before, we considered the diffusive flux: $J = -D \frac{\partial C}{\partial x}$
drifting to the right with velocity $V(x)$. Then:

Now, let us consider particles

$$J = \frac{N(x)}{A\tau} \quad \text{Where} \quad N(x) = C(x)A\Delta x, \Delta x = V\tau$$

(particles from a box of length $V\tau$ can cross the boundary at x), so:

$$J = V(x)C(x) \text{ - drift flux}$$

Remember that minus corresponds to the right direction. If $V(x) = \text{const}$, then:

$$\frac{\partial C}{\partial t} = -\frac{\partial}{\partial x}(VC) \text{ - drift equation}$$

$$\frac{\partial C}{\partial t} = -V \frac{\partial C}{\partial x}$$

Solution to $\frac{\partial C}{\partial t} = -V \frac{\partial C}{\partial x}$ $C = C_{init}(x - Vt)$

Other possible cases: diffusion-drift

$$\frac{\partial C}{\partial T} = -V \frac{\partial C}{\partial X} + D \frac{\partial^2 C}{\partial X^2}$$

No flux at the boundaries:

$$J = VC - D \frac{\partial C}{\partial X} = 0, X = 0, L$$

L - scale of length

L/V - scale of time

Arbitrary S - scale
of concentration

$$\frac{\partial c}{\partial t} = -\frac{\partial c}{\partial x} + \mathcal{D} \frac{\partial^2 c}{\partial x^2}$$

$$\mathcal{D} = \frac{D}{LV}$$

$$\mathcal{D} = \frac{(L/V)}{(L^2/D)}$$

How much faster diffusion is than drift

The transients die out in L/V or in L^2/D , whichever is longer;
Number of motors is conserved (integrate and check). In the steady state, flux is zero everywhere. Integrate the steady state equation once:

$$-c + \mathcal{D} \frac{dc}{dx} = -j = 0$$

$$c = \exp[x/\mathcal{D}] \rightarrow C = A \exp[VX/D]$$

Can estimate either D or V if either V or D is known and $\log(C)/X$ is measured

Turing Instability

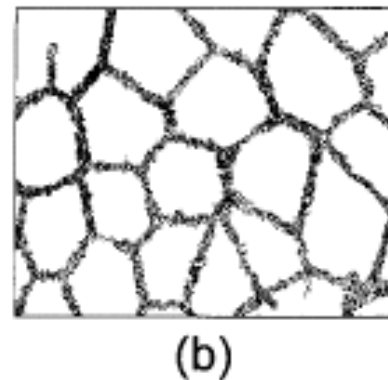
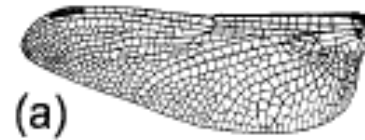
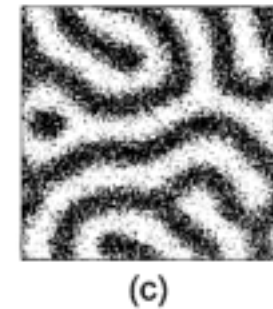
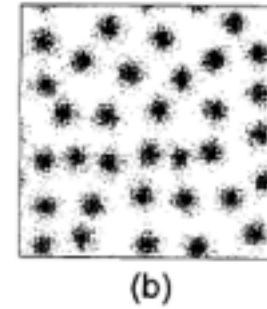
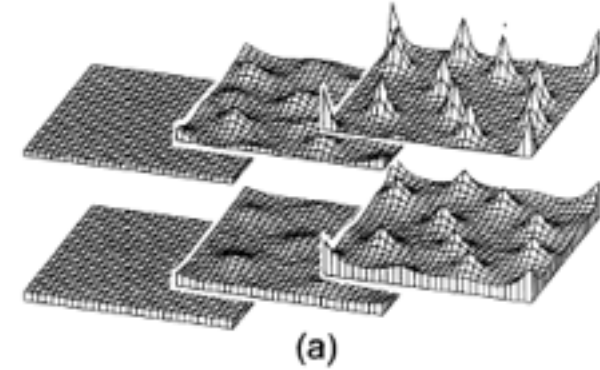


$$\frac{\partial a}{\partial t} = D_a \Delta a + \rho_a \left(\frac{a^2}{h} - a \right)$$

$$\frac{\partial h}{\partial t} = D_h \Delta h + \rho_h \left(a^2 - h \right) .$$

	a	h
a	+	-
h	+	-

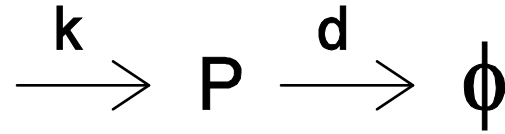
activator-inhibitor system



Turing machine, algorithm,
computer, Bletchley park,
Enigma, AI - Turing test;
RD eqs - morphogenesis

A. M. Turing (1952).
The Chemical Basis of Morphogenesis.
Phil Trans Royal Soc London,
volume B 237, pages 37-72.

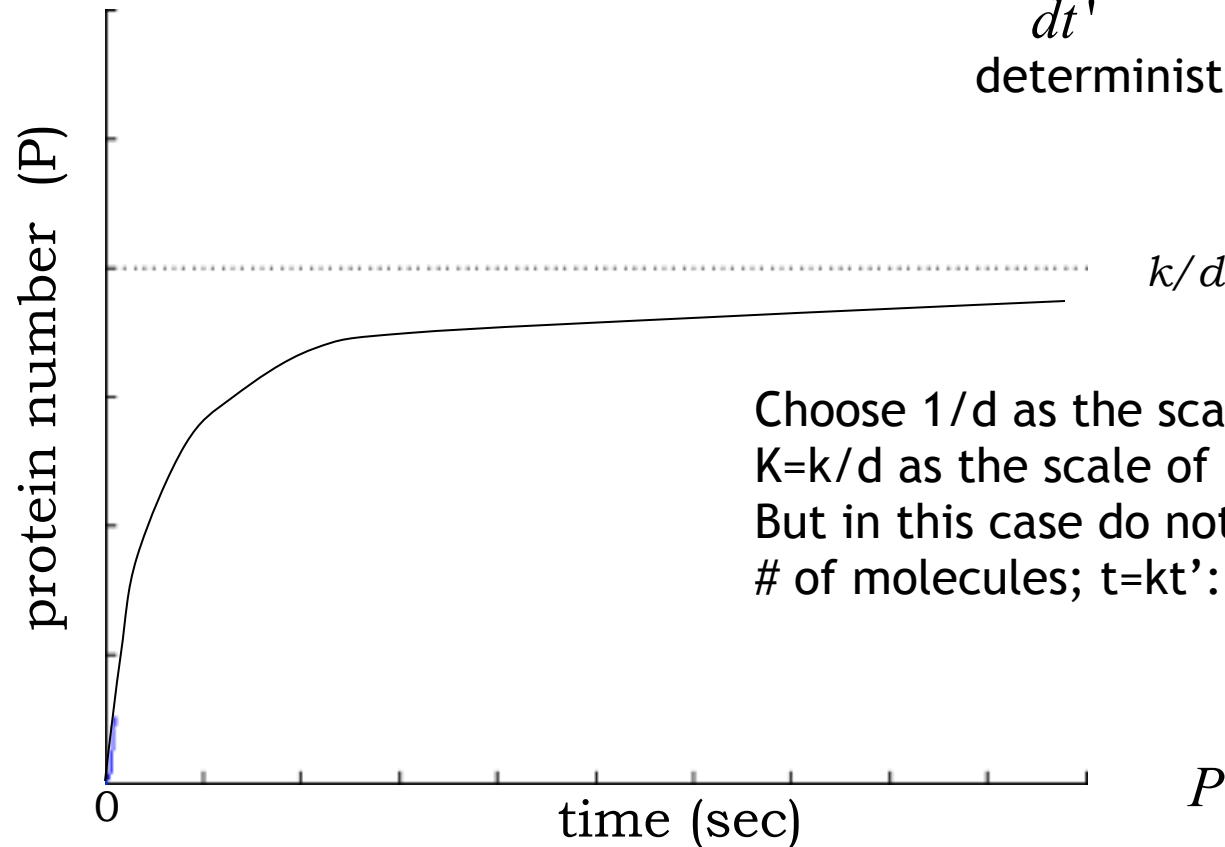
Any chemical reaction is stochastic.



$$\frac{dP}{dt'} = k - d \times P$$

$\begin{matrix} \# \\ \downarrow \\ \frac{dP}{dt'} \end{matrix} \quad \begin{matrix} \# / s \\ \swarrow \\ k \end{matrix} \quad \begin{matrix} 1 / s \\ \swarrow \\ d \end{matrix}$

deterministic equation

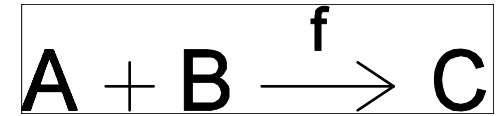


Choose $1/d$ as the scale of time;
 $K=k/d$ as the scale of the # of molecules;
 But in this case do not rescale the
 # of molecules; $t=kt'$:

$$\frac{dP}{dt} = K - P$$

$$P = K (1 - \exp[-t])$$

Why are chemical reactions stochastic?



1. Reactants diffuse to find each other in solution.
2. They must overcome the energy barrier of the reaction.

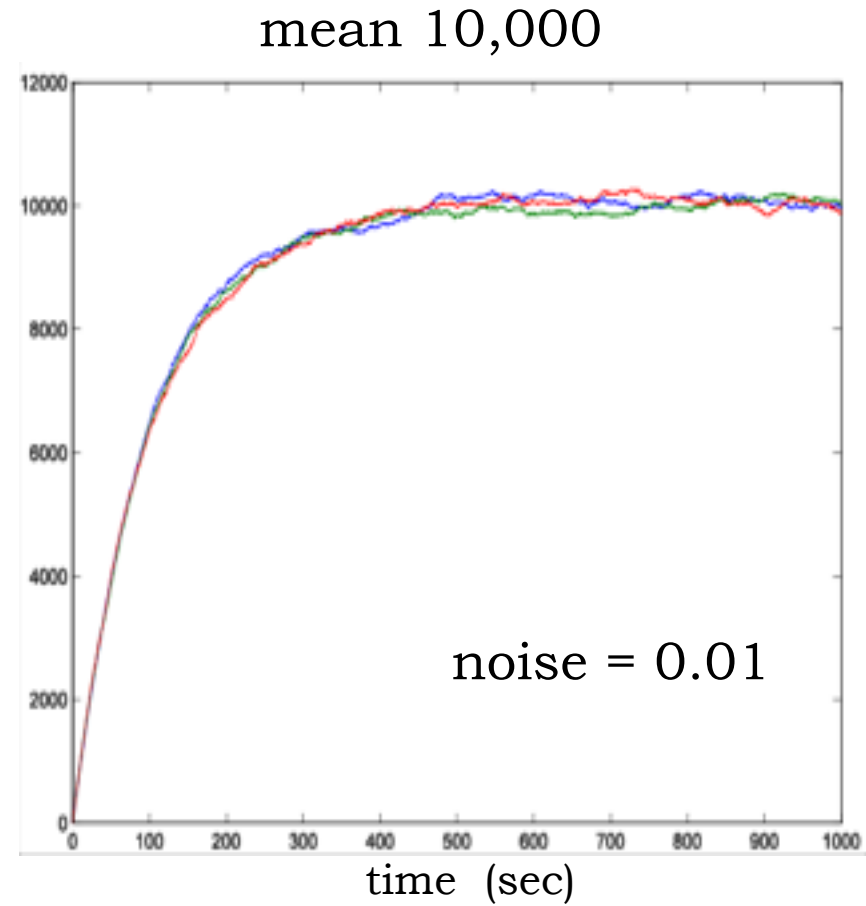
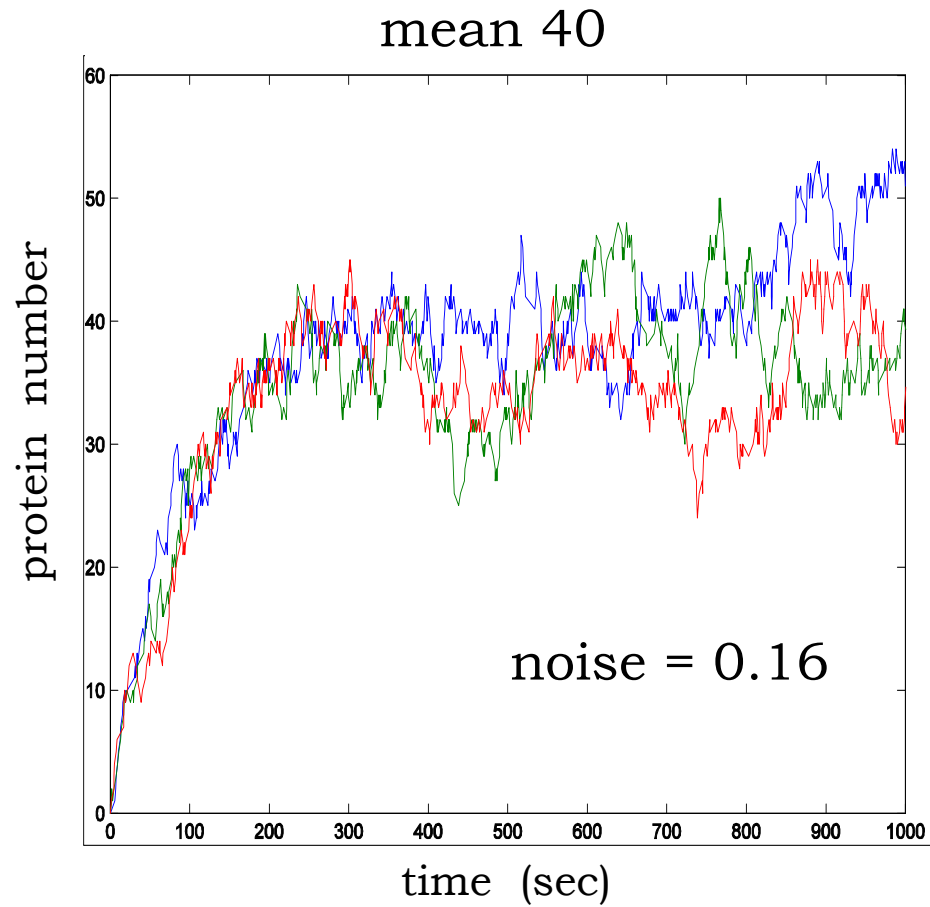
Both events are randomly affected by thermal fluctuations:
collisions with other (solvent) molecules

When are chemical reactions significantly stochastic?

As reaction only increases or decreases the number of molecules by one or two, it is only when numbers of molecules are small that random timing of individual reactions will matter.

$$\text{noise} = \frac{\text{standard deviation}}{\text{mean}}$$

Noise depends on low numbers.



Stochastic model for this simple chemical reaction:

- 1) The rate of production is K , so over time dt we expect $K*dt$ new molecules. If $K*dt \ll 1$, then most likely no new molecules will appear, or at most one, and most certainly not 2,3,... One new molecule will appear with probability $Pr+ = K*dt$.
- 2) The rate of degradation is P , so over time dt we expect $P*dt$ existing molecules to disappear. If $P*dt \ll 1$, then most likely no molecules disappear, or at most one, and most certainly not 2,3,... One new molecule will disappear with probability $Pr- = P*dt$.

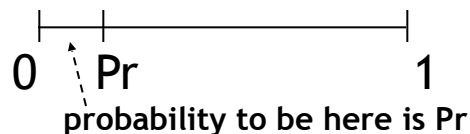
Direct Monte Carlo method:

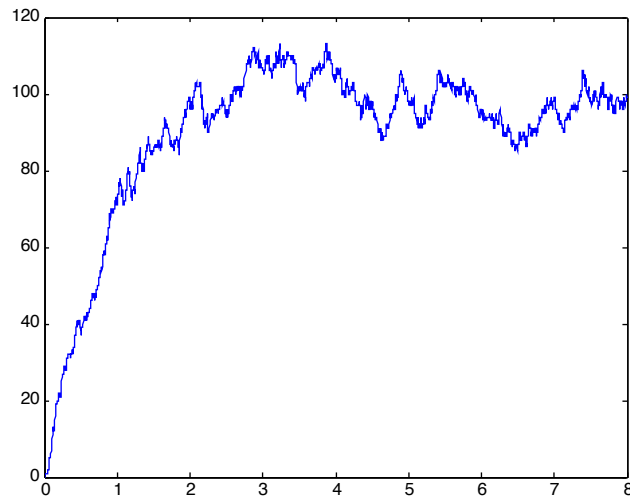
If the current number of molecules is P , (a) generate a random number equal to 1 with probability $Pr+ = K*dt$ and to 0 with probability $(1 - Pr+)$, and add this number to P ; (b) generate a random number equal to 1 with probability $Pr- = P*dt$ and to 0 with probability $(1 - Pr-)$, and subtract this number from P ; (c) repeat this step for a few time units.

How to generate a random number r equal to 1 with given probability Pr and to zero otherwise?

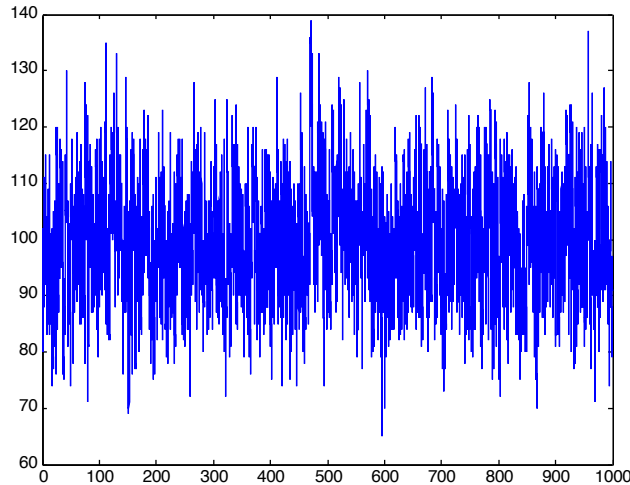
In Matlab: **rand**

$r = 0.5*(1+\text{sign}(Pr - \text{rand}));$





```
K=100; dt=0.001; tend=8;
t=(0:dt:tend); P=zeros(size(t));
for s=1:(tend/dt)
    P(s+1)=P(s)+0.5*(1+sign(K*dt-rand))-0.5*(1+sign(P(s)*dt-rand));
end
plot(t,P)
```



← Start with K

$$Mean(x) = \langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i$$

$$Var(x) = \langle x^2 \rangle - \langle x \rangle^2$$

$$std(x) = \sqrt{Var(x)}$$

```
>> mean(P)
```

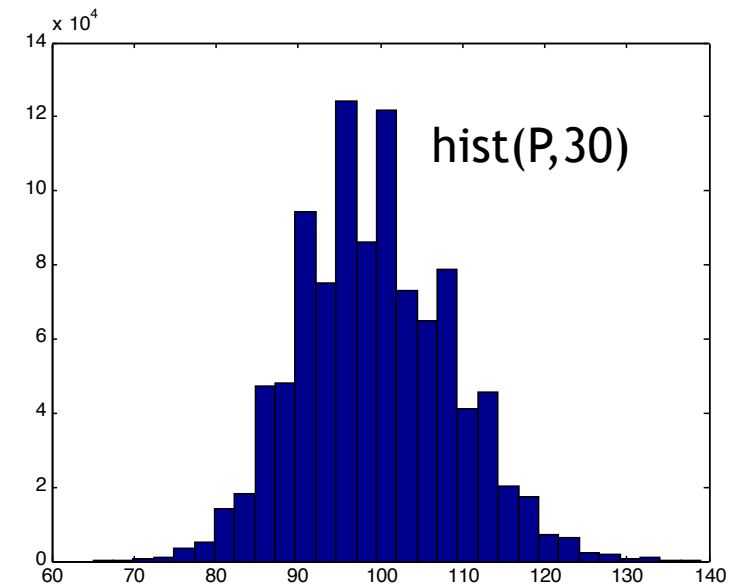
```
ans =
```

```
99.5100
```

```
>> std(P)
```

```
ans =
```

```
9.2659
```



Gaussian with average 100 and std-10

K	10	36	100	400	900
std	3.18	5.82	9.27	16.01	27.75

So, we expect:

$$N \approx K \pm \sqrt{K}$$

Try simulating K=900 with dt=0.001 and see what your mean and std is. Think about it!

Note about dt: $K*dt$ and $P(s)*dt \ll 1$; otherwise, one of the main assumptions is violated. But this means that during most of the steps nothing happens so we waste a lot of computer time generating random numbers and computing during 'empty' steps.

Another way to simulate this chemical reaction (without decay for now) stochastically:

$$P_n(t + \Delta t) = \text{Prob}(N(t) = n \text{ and no event in } [t, t + \Delta t]) = P_n(t) \times (1 - K \times \Delta t)$$

$$P_n(t + \Delta t) - P_n(t) = -KP_n(t) \times \Delta t \quad \rightarrow \quad \frac{dP_n}{dt} = -KP_n \quad \rightarrow \quad P_n(t) = \exp[-Kt]$$

$$\text{Prob}(\text{waiting time for a reaction event } T < t) = 1 - P_n(t) = 1 - \exp[-Kt]$$

$$\text{Prob}(\text{waiting time for a reaction event } T < t] = 1 - \exp[-Kt]$$

Let us prove that if we generate the random waiting time

$$T = \frac{-1}{K} \ln R, R = \text{rand}$$

then its probability distribution will be as that for the reaction event:

$$T < t \rightarrow \frac{-1}{K} \ln R < t \rightarrow \ln R < -Kt \rightarrow R > \exp[-Kt]$$

$$\text{Prob}(T < t] = 1 - \exp[-Kt] = \text{Prob}(R > \exp[-Kt]) = 1 - \exp[-Kt]$$

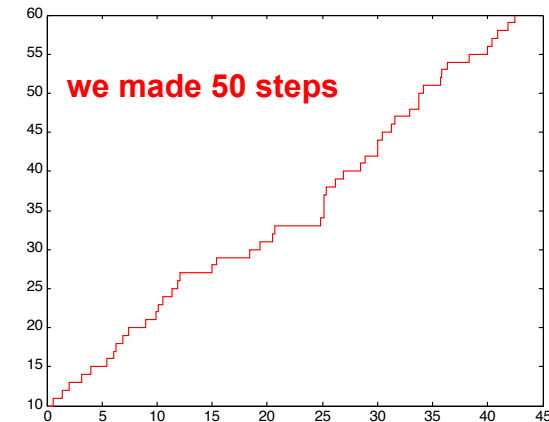
Gillespie algorithm

Gillespie algorithm

Algorithm: if you have a stochastic process happening with rate K , then you generate an exponentially distributed time (average time $1/K$), and increase the number of molecules at this time.

The method is exact; it also saves you time when you have to make intermittent small and large time steps. The downside (there's always one) is difficult code writing if there are too many simultaneous reactions

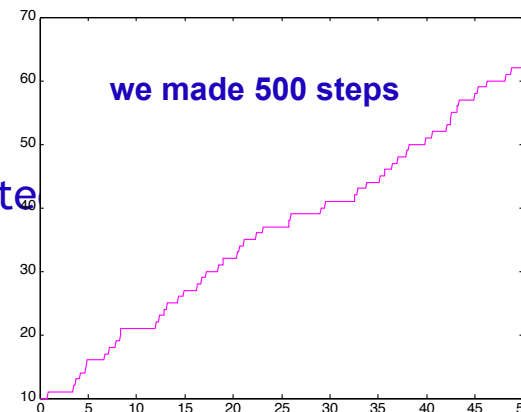
```
K=1; T=50; n=10;  
time=zeros(size(T)); nn=n*ones(size(T));  
for t=1:T % number of events  
    tt=-(1/K)*log(rand); time(t+1)=time(t)+tt;  
    % time for an event, time is updated  
    nn(t+1)=nn(t)+1; % molecule number is updated  
end  
stairs(time,nn,'r')
```



Direct Monte Carlo

```
K=1; dt=0.1; T=50; n=10;  
time=(0:dt:T); nn=n*ones(size(time));
```

```
for t=1:length(time) % number of steps  
    r = 0.5*(1+sign(K*dt - rand));  
    % random increment for molecule number  
    nn(t+1)=nn(t)+r; % molecule number is updated  
end  
plot(time,nn,'m')
```



How do we simulate a little bit more complex reaction (include degradation):

Algorithm: if you have 2 possible processes happening with rates r_1 and r_2 , then an event (any of the two processes) take place with the rate $(r_1 + r_2)$. So, we first generate an exponentially distributed time (average time $1/(r_1+r_2)$), and then with probability $r_1/(r_1+r_2)$ make the first event happen at this time, and with probability $r_2/(r_1+r_2)$ make the second event happen at this time:

$r_1 = K$ and $r_2 = \text{molecule \#}$

```
K=1; T=50; n=10;  
time=zeros(size(T)); nn=n*ones(size(T));  
  
for t=1:T % number of events  
    T=-(1/(K+nn(t)))*log(rand); time(t+1)=time(t)+T;  
    % time for an event, time is updated  
    r=rand; if r<K/(K+nn(t)) n=n+1; % production  
    else n=n-1; end % degradation  
    nn(t+1)=n; % length is updated  
end  
stairs(time,nn,'r')
```

