Notes for Math 450
Stochastic Petri nets and reactions

Renato Feres

1 Petri nets

Petri nets are a special class of networks, introduced in 1962 by Carl Adam Petri, that provide a convenient language and graphical representation for many kinds of processes in a variety of areas of science and engineering. Some of the terms used in our discussion will come from chemical kinetics, an area where Petri nets are widely employed, but it will be clear that their applicability extends well beyond chemistry. I will often call them simply *nets*, for short.

The first special feature of a Petri net is that the underlying graph is *bipartite*. We begin by defining this and other basic terms concerning graphs and networks. A *directed graph*, or *digraph*, is a pair $\mathcal{G} = (V, E)$ where V is a set whose elements are called *vertices*, or *nodes*, and E is a subset of the product $V \times V$. Elements of E are called *edges*, or *arcs*. An edge is thus simply an ordered pair of vertices, (u, v). We sometimes write $u \to v$ if (u, v) belongs to E, and say that this edge *connects* u to v.

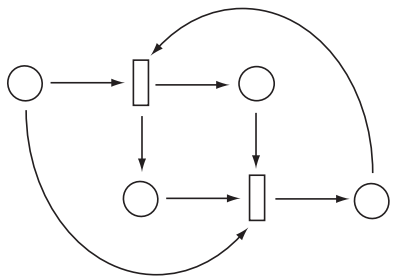

Figure 1: An example of a bipartite graph.

A graph is said to be *bipartite* if the set V of vertices decomposes into a disjoint union of two sets, $V = P \cup R$, and each element of E has the form (p, r) or (r, p), where r belongs to R and p belongs to P. In other words, an edge can only connect vertices of different types. Vertices from P are called (in the Petri nets literature) *places*, and will be represented by circles in the graph diagram. Vertices from R are usually called *transitions*, and will be represented

by rectangles. We will also use the terms *(molecular) species* and *reactions*, respectively. Figure 1 shows an example of a bipartite graph.

The *state* of a net is defined as an assignment of a non-negative integer, $M(p)$, to each place $p \in P$. We call $M(p)$ the *population size* of the species p. We also use, alternatively, the term *number of tokens*, which is more standard in the net literature.

In addition to the above basic structure, the edges of a net are assigned weights, defined by functions Pre and Post. To an edge (p, r) (going from a place to a transtition) we associate a non-negative integer $\mathrm{Pre}(p, r)$, and to each edge (r, p), from a transition to a place, we associate a non-negative integer $\mathrm{Post}(r, p)$.

I wish to think of each place as representing a molecular species and the number of tokens $X(p)$ as the number of molecules of species p at a given time. (Time will be brought into this general set up explicitly once we define a Markov process subordinate to a given Petri net. This will be done shortly when we introduce *stochastic* Petri nets.) In the same spirit, I wish to view a transition r as a reaction. The number $\mathrm{Pre}(p, r)$ indicates how many molecules of type p are consumed in the reaction r, and $\mathrm{Post}(r, p)$ indicates how many molecules of type p are produced by r. For example, consider the graph of figure 2.

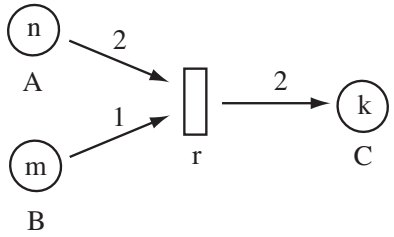

Figure 2: The Petri net of the reaction represented in ordinary chemistry notation by $r : 2A + B \to 2C$.

The net with the above diagram and weights $\mathrm{Pre}(A, r) = 2$, $\mathrm{Pre}(B, r) = 1$, and $\mathrm{Post}(r, C) = 2$ (indicated over the arrows), can be viewed as a representation of the reaction $r : 2A + B \to 2C$ in standard chemical notation. The symbols inside the circles are the number of tokens for each place. They are variables $n = X(A)$, $m = X(B)$ and $k = X(C)$, which we view as the number of molecules of each species at a given time. A reaction event changes the population numbers according to Pre and Post, as follows: n is updated to $n - 2$, m to $m - 1$, and k to $k + 2$ every time the reaction event t "occurs."

It is useful to consider reactions in which tokens of a given species are created or destroyed. We will indicate such transitions by

$$r : \emptyset \to A \quad \text{or} \quad r : A \to \emptyset.$$

This may represent, for example, migration or transport between separate populations.

In actual chemical processes, reactions usually come together with their reversed form. This is usually the case when the reaction represents an elementary step in an overall complicated reaction mechanism. Given a reaction r, we denote its reversed form by $-r$. This is the reaction in which all the arrows of r are reversed while the weights attached to each arrow are kept. In other words, $\mathrm{Post}(-r, A) = \mathrm{Pre}(A, r)$. (Later we will be regarding reactions as vectors in an appropriate vector space, and the sign will have a more standard mathematical meaning.) It may be convenient to indicate a forward-backward reaction pair by some short-hand convention so as not to crowd the diagram too much. I will do this by drawing a double harpoon such as $\rightleftharpoons$ on top of the reaction box. For example, consider the following set of reactions:

$$\begin{aligned} \pm r_1 &: \emptyset \rightleftharpoons f \\ \pm r_2 &: \emptyset \rightleftharpoons m \\ \pm r_3 &: f + m \rightleftharpoons p \\ r_4 &: p \to \emptyset \end{aligned}$$

One may think of this system of reactions as a model for a dating service: the reversible pair of reactions $\pm r_1$ represents one woman enrolling in or leaving (still unmatched) the service; the reaction pair $\pm r_2$ represents the same for a man. The reaction r_3 represents the formation of one matched pair, and the reverse reaction $-r_3$ represents the dissociation of the pair and return to the unmatched f and m populations (possibly due to "bad chemistry.") Reaction r_4 represents a successfully matched pair leaving the service. It is not meaningful to consider the reverse reaction $-r_4$ since a successful pair will not re-enter the service as a pair. (They might subsequently break up and re-enter the system individually through reactions r_1 and r_2.) We assume the functions Pre and Post take value 1 on all edges.

The diagram of figure 3 shows the collection of reactions represented above in standard chemical notation.

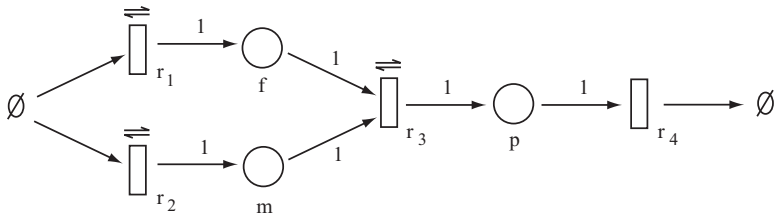

Figure 3: A Petri net representation of the reaction mechanism described above in standard notation. The number of tokens are not indicated.

We wish to think of the individual reaction events as taking place along a succession of time steps with each reaction proceeding independently of the other ones. This can be imagined as the work of several agents acting without coordination, following their own individual clocks, each responsible for the execution of one of the reactions in the overall mechanism. An *event* in the process consists of one execution of a given reaction by its agent. Whenever an event occurs, the number count of the populations of species participating in the corresponding reaction changes according to the functions Pre and Post. I.e., the number $X(A)$ is reduced by $\mathrm{Pre}(A, r)$, or increase by $\mathrm{Post}(r, A)$ depending on whether A is a reactant or a product of reaction r. We will translate this picture into a dynamical (stochastic) mechanism in the next section.

Generally, we omit the reaction rectangles if there is no more than one reactant and no more than one product species. Also, for small number of tokens, it is customary to indicate them by dots. The next example can be regarded as a transition diagram for a Markov chain (with transition rates not yet indicated). In this case we assume that the number of tokens at each place is either 0 or 1 and the total number of tokens is 1, the occupied place representing the present state of the Markov chain. State transitions of Markov chains can be regarded as reactions of type $A \to B$. In such cases we omit the reaction rectangle.

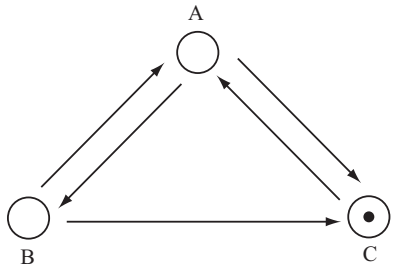

Figure 4: The transition diagram for a Markov chain is a special case of a Petri net. Here we omit the reaction rectangles. The token represents the present state.

2 State transitions diagram

Recall that the *state* of a net is defined by a function X on the set of places: $X(p)$ indicates the number of tokens occupying a place p. Let S denote the set of all states that a given net can attain. The *state transition graph* is a graph with vertex set S and edges (s, s') where s' is a state that can follow s after the occurrence of one reaction event.

Consider for example the Petri net of figure 5.

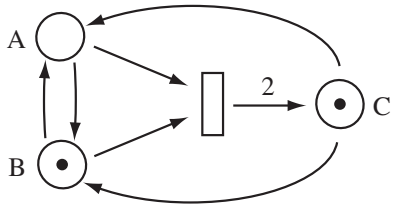

Figure 5: A Petri net with a total of two tokens.

Its set of reactions is:

$$\begin{aligned} \pm R_1 &: A \rightleftharpoons B \\ R_2 &: A + B \to 2C \\ R_3 &: C \to A \\ R_4 &: C \to B \end{aligned}$$

To obtain the transition diagram we first enumerate the states. Each state is represented by a vector (a, b, c) giving the number of tokens in places A, B and C. They are:

$$\begin{aligned} s_1 &= (2 \quad 0 \quad 0) \\ s_2 &= (1 \quad 1 \quad 0) \\ s_3 &= (1 \quad 0 \quad 1) \\ s_4 &= (0 \quad 2 \quad 0) \\ s_5 &= (0 \quad 1 \quad 1) \\ s_6 &= (0 \quad 0 \quad 2) \end{aligned}$$

The transitions diagram for the Petri net of figure 5 is given in figure 6. Note, for example, that the arrow from s_6 to s_5 is due to reaction $C \to B$, and the arrow from s_2 to s_6 is due to reaction $A + B \to 2C$.

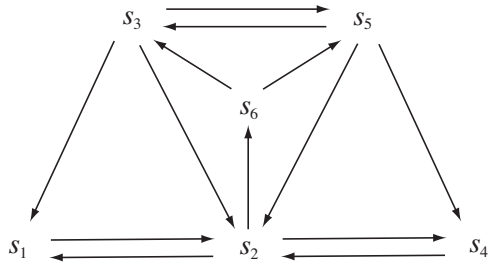

Figure 6: State transitions diagram for the Petri net of figure 5.

3 Stochastic Petri nets

We wish to introduce now a dynamical process that governs the way in which the states of a Petri net change in time. We can define processes that are either deterministic or stochastic. We begin with stochastic evolution model that incorporates the property that the various reactions operate in an uncoordinated, asynchronous way, and the reaction events happen at exponential random times.

The additional data needed to specify the stochastic dynamics are the *reaction rates*, $h(r, X)$. These are functions of the reaction r and of the state X of a Petri net. The reaction rates give the transition probability rates for the state transitions diagram associated to the Petri net, making them transition diagrams for a continuous-time Markov chain.

Note that the way in which the state of the process changes in time has the following general form:

$$X_{t_1}(p) = X_{t_0}(p) + \sum_{r \in R} \#_r([t_0, t_1])(\mathrm{Post}(r, p) - \mathrm{Pre}(p, r)), \tag{1}$$

where $\#_r([a, b])$ represents the number of reaction events of type r that occurred during the time interval $[a, b]$. With this in mind, the evolution process of the net can be described as follows:

1. The system is initially (time $t = 0$) in state X_0. This may be a random variable with probability distribution λ on the set of states.

2. Let t be the time of occurrence of the last reaction and $X = X_t$ the present state.

3. For each reaction $r \in R$ let S_r be an exponential random variable with parameter $h(r, X)$. Let S be the minimum among the S_r and r the random reaction that gives the minimum, i.e., $S = S_r$. Now set the new time $t' = t + S$ and new state

$$X'(p) = X(p) + \mathrm{Post}(r, p) - \mathrm{Pre}(p, r).$$

4. Rename the present time t and present state X and if $t < T_{\max}$, return to step 2. Otherwise, stop.

Using the general properties of exponential distributions, we can restate the third item above as follows: Let X be the present state and t the time of last transition. Define

$$h(X) = \sum_{r \in R} h(r, X)$$

and probabilities

$$p_r = h(r, X)/h(X).$$

Obtain S, an exponential random variable with rate $h(X)$, and new reaction r chosen from R with probabilities p_r. Now set the new time $t + S$ and new state X' as in the above algorithm.

Natural choices of the reaction rates will ensure that the rate for a reaction r will decrease as the population size for a reactant involved in r decreases, so that the states do not take on negative values.

What we have described is, essentially, what is called *Gillespie method* of stochastic simulation. Before implementing this method, we stop to consider a class of rate functions coming from chemical kinetics known as *mass-action law*.

3.1 Mass-action kinetics

We specialize now to a family of rate functions defining what is called *mass-action kinetics*. First of all, the *order* of a reaction r is defined as the number of reactant species,

$$d = \sum_{p \in P} \mathrm{Pre}(r, p)$$

In situations where a reaction event requires individuals of different species randomly to come together in close proximity at about the same time (a multiple collision), reactions of high order are rare. This is the case in chemistry. In fact, it is often the case in chemistry that reactions of order greater than 2 describe the overall effect of a sequence of elementary reactions of order 2 or less. Because of that, I will only write the precise expression of the mass-action laws up to order 2. This is shown in the next table.

order	reaction r	rate function $h(r, X)$
0	$\emptyset \to *$	c
1	$p \to *$	$cX(p)$
2	$p_1 + p_2 \to *$	$cX(p_1)X(p_2)$
2	$2p \to *$	$cX(p)(X(p) - 1)/2$

For example, consider the following stochastic version of a classical ecological model of predator-prey interaction known as the Lotka-Volterra system. Here a_1 represents the prey species and a_2 the predator species.

$$\begin{aligned} a_1 &\to 2a_1 \\ a_1 + a_2 &\to 2a_2 \\ a_2 &\to \emptyset \end{aligned}$$

The first reaction represents the reproduction of the prey species. The second indicates the reproduction of predators require consumption and death of preys, and the last reaction represents predator death. The Petri diagram is shown in figure 7.

The mass-action rate functions are, respectively:

$$\begin{aligned} h(r_1, X) &= c_1 X(a_1) \\ h(r_2, X) &= c_2 X(a_1) X(a_2) \\ h(r_3, X) &= c_3 X(a_2). \end{aligned}$$

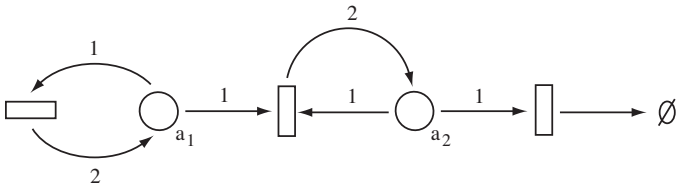

Figure 7: A net diagram for the predator-prey reaction model.

For large population sizes, the reaction rate function for the second order equation $2A \to *$ is approximated by $cX(A)^2$. More generally, the mass-action rate for the reaction

$$r : n_1 p_1 + \cdots + n_k p_k \to *$$

is taken to be

$$h(r, X) = cX(p_1)^{n_1} X(p_2)^{n_2} \ldots X(p_k)^{n_k}$$

if the population sizes are large. Each term of the form x^n in the product can be regarded as an approximation of the binomial coefficient $C(x, n)$ (x-choose-n).

For ordinary chemical mixtures these quantities are typically measured in number of moles, where one mole equals 6.03×10^{23} molecules. This number is known as the Avogadro number. If we are considering reacting substances inside a well-mixed container of fixed volume, these quantities are usually expressed as molar concentrations, i.e., as the number of moles per unit volume.

3.2 Simulation of the stochastic Petri net process

The full specification of a stochastic Petri net at any given time involves the data $\mathcal{N} = (P, R, \mathrm{Pre}, \mathrm{Post}, h, X)$, where P is the set of places; R the set of transitions (or reactions); Pre gives the multiplicities of place-transition edges; Post gives the multiplicities of transition-place edges; h is the probability rates function, which we often choose according to the mass-action law; and X is represents the state of the process at a given time. Recall that X is a function on P that gives the number of tokens, $X(p)$, of each place $p \in P$. We consider the edge-multiplicity functions Pre and Post as defined on all pairs (p, r) and (r, p) respectively, taking the value 0 if a pair is not an element in the edge set E. Thus the set E, which we didn't include in the Petri net data above, is already specified by the edge-multiplicity functions. Also recall that the function $h(r, X)$ is completely specified by the rate constants c_r under the mass-action law.

Before giving a more detailed algorithm for simulating the stochastic Petri net, it is convenient to represent the information in $\mathcal{N}$ in matrix form based on some numbering of the sets P and R. We write $P = \{p_1, p_2, \ldots, p_m\}$, $R = \{r_1, r_2, \ldots, r_n\}$. (In realistic reaction mechanisms in chemistry, often the number n of reactions is much greater than the number m of molecular species.) Define $h_j(X) = h(r_j, X)$, $j = 1, \ldots, n$, and write the state of the system as a

column vector with coordinates $X_i = X(p_i)$. Thus we write $X = (X_1, \ldots, X_m)^t$, where the upper-script indicates matrix transpose. When we need to write the state vector X as a function of time, we write it X_t or $X(t)$, depending on convenience of notation. We write the individual coordinates of X_t as $X_i(t)$.

Denote by $U = (u_{ij})$ the m-by-n matrix such that

$$u_{ij} = \mathrm{Post}(r_j, p_i) - \mathrm{Pre}(p_i, r_j).$$

This is sometimes called the *stoichiometric matrix* of the net. For example, in the predator-prey reaction model, U has the form

$$U = U^+ - U^- = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}.$$

The number of reaction events for each r_j over the interval of time $[a, b]$ will be denoted by $\#_i([a, b])$ and the column vector with these components will be written $\#([a, b])$. Formula 1 now takes the form

$$X_{t_1} = X_{t_0} + U\#([t_0, t_1]), \tag{2}$$

where $U\#([t_0, t_1])$ is the matrix product of U by the column vector $\#([t_0, t_1])$. We note that if only one reaction event took place over a given time interval and the reaction was r_j, then

$$X_{t_1} = X_{t_0} + U^{(j)},$$

where $U^{(j)}$ is the j-th colum of U.

A more explicit description of Gillespie's algorithm, in the form of a hold-and-jump process, can now be given in the following way:

1. Initialize the state vector $X = X_0$;
2. for each $i = 1, 2, \ldots, n$, calculate $h_i(X)$ based on the current state X;
3. calculate the combined reaction rate $h(X) = h_1(X) + \cdots + h_n(X)$;
4. simulate a sample value of an exponential random variable with rate constant $h(X)$ and call the value s. This is the holding time till the next reaction event;
5. set the current time to $t + s$ and call it t;
6. simulate a sample value of index k with probabilities $h_k(X)/h(X)$ and call it j;
7. set the new state of the system as $X + U^{(j)}$, where $U^{(j)}$ denotes the j-th column of the stoichiometric matrix U;
8. output X and t;
9. if t is less than a preassigned maximum time (or the number of steps is less than a preassigned value), return to step 2. Otherwise, stop.

3.3 A Matlab program for the Gillespie method

We implement here Gillespie method for the special case of reactions satisfying the mass-action law.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [t y]=gillespie(U_pre, U_post, c, x_0, N)
%Simulate a reaction system using the Gillespie
%algorithm.
%Inputs: U_pre is the m-by-n matrix of
%place-to-reaction multiplicities (there are
%m places and n reactions);
%U_post is the m-by-n matrix of reaction-to-place
%multiplicities; c=[c_1 ... c_n] are the reaction
%constants; x_0 is the initial state written as
%a row vector of length m; and N-1 the
%maximum number of reaction steps.
%Outputs: t is a vector of length N of time events;
%y is an m-by-N matrix whose k-th column gives the state
%of the system at time step k. I.e., the system is
%at state y(k) during the interval from times
%t(k) to t(k+1), for each k.
[m n]=size(U_pre); %n=number of reactions; m=number of species
U=U_post-U_pre; %stoichiometric matrix
t=zeros(1,N);
y=zeros(m,N);
y(:,1)=x_0;
for k=1:N-1
    h=c.*prod(repmat(y(:,k),1,n).^U_pre); %reaction rate vector
    h_sum=sum(h);
    pi=h/h_sum; %jump probabilities
    s=-log(rand)/h_sum; %holding time
    t(k+1)=t(k)+s; %time of next reaction event
    %simulate next reaction with probability pi:
    a=rand;
    e=0;
    for i=1:n
        e=e+i*(sum(pi(1:i-1))<=a & a<sum(pi(1:i)));
    end
    y(:,k+1)=y(:,k)+U(:,e);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

We apply this program to the predator-prey model. The following commands can be used to obtain the graph of the population sizes as functions of time . Here we have used reaction constants $c_1 = 1$, $c_2 = 0.005$, and $c_3 = 0.6$. The initial population sizes are 50 preys and 100 predators.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U_pre =[1 1 0; 0 1 1];
U_post=[2 0 0; 0 2 0];
N=10000;
c=[1 0.005 0.6];
x_0=[50 100];
[t y]=gillespie(U_pre,U_post,c,x_0,N);
plot(t,y(1,:))
hold on
plot(t,y(2,:),'--')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

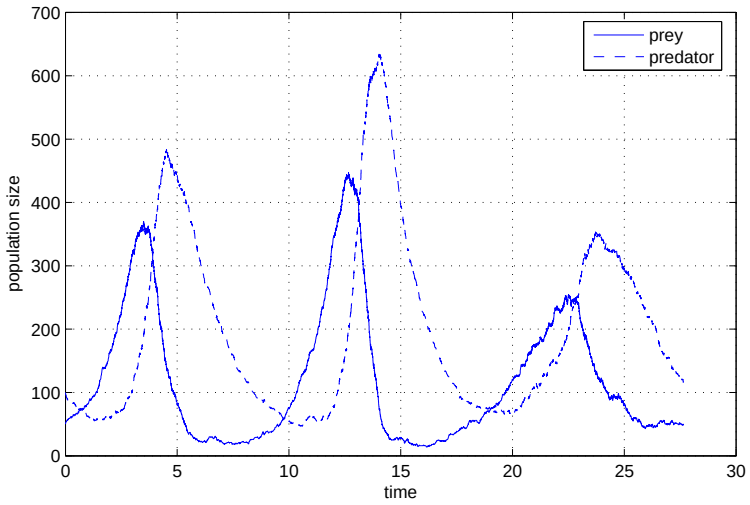

Figure 8: Changing populations of predator and prey in the stochastic Lotka-Volterra model.

Exercise 3.1 The `gillespie` program, as given, does not take into account the possibility that rate function h may become zero and the reaction network reaches an absorbing state. For example, the species populations may go "extinct." Modify `gillespie` so that the program will exit when reaching an absorbing state, returning the sample path of the process up to that moment.

3.4 Improvements of `gillespie`

The above program suffers from a few drawbacks. First, it outputs data of every single event that occurs in the simulation of the reaction network. This is often

not desirable for systems with relatively large size and complexity, for which the number of simulated events becomes very large. In this case all that we may want to keep is the state of the system on a sufficiently fine grid of time points rather than the full sequence of states at every event time. Also the program does not take into account the possibility that one species population can go extinct and the reactions cannot continue.

The following modification of the `gillespie` program takes care of these shortcomings.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [t y]=gillespied(U_pre, U_post, c, x_0, T, dt)
%Simulate a reaction system using the Gillespie
%algorithm only recording states on a regular grid
%of time points. Adapted from D. Wilkinson, "Stochastic
%Modelling for Systems Biology" (1st edition, page 155).
%Inputs: U_pre is the m-by-n matrix of
%place-to-reaction multiplicities (there are
%m places and n reactions);
%U_post is the m-by-n matrix of reaction-to-place
%multiplicities; c=[c_1 ... c_n] are the reaction
%constants; x_0 is the initial state written as
%a row vector of length m; and T is the
%final time (the initial time being 0).
%Outputs: t is a regular grid of time points with
%step size dt;
%y is an m-by-N matrix whose k-th column gives the state
%of the system at time step k. I.e., the system is
%at state y(k) during the interval from times
%t(k) to t(k+1), for each k.
[m n]=size(U_pre);
U=U_post-U_pre;
N=floor(T/dt);
y=zeros(m,N);
y(:,1)=x_0';
t=0:dt:(N-1)*dt;
ylast=x_0';
tt=0;
target=dt;
k=1;
while k<N
    h=c.*prod(repmat(ylast,1,n).^U_pre);
    h_sum=sum(h);
    if h_sum<10^(-10)
        tt=10^(99);
    else
        tt=tt-log(rand)/h_sum;
```

```
        pi=h/h_sum;
    end
    while tt>=target
        y(:,k+1)=ylast;
        k=k+1;
        target=target+dt;
        if k>=N
            return
        end
    end
    a=rand;
    e=0;
    for i=1:n
        e=e+i*(sum(pi(1:i-1))<=a & a<sum(pi(1:i)));
    end
    ylast=ylast+U(:,e);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

As an example, we run `gillespied` for the same Lotka-Volterra model and same initial conditions as in the first numerical example, taking $T = 100$ and $dt = 0.05$. The result is shown in figure 9.

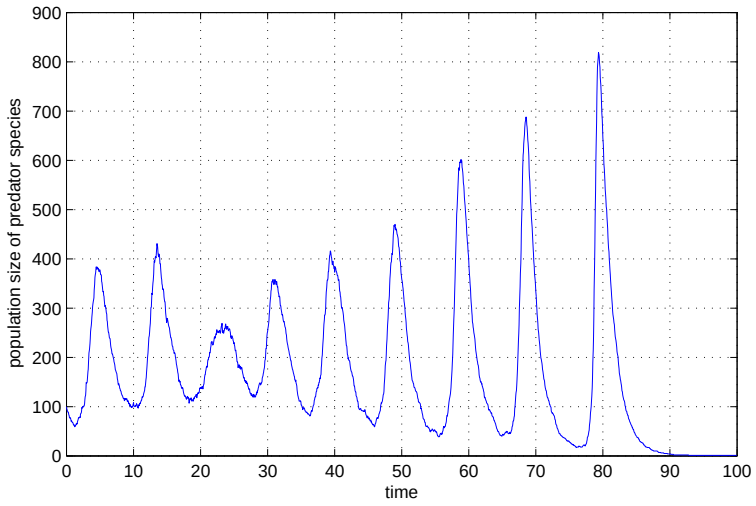

Figure 9: Changing predator population in the stochastic Lotka-Volterra model, showing extinction.

4 Homework 10

We modify the stochastic Lotka-Volterra model to allow migration of both predator and prey between two geographic sites. A diagram is shown in figure 10.

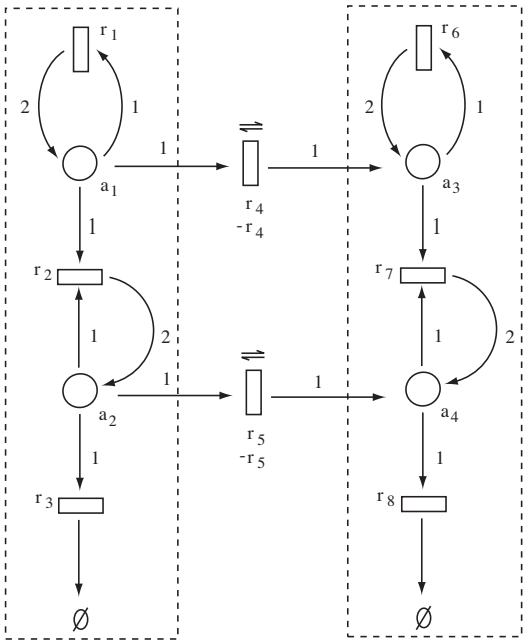

Figure 10: Predator-prey model with migration.

Note that the prey population is divided into two Petri net places, a_1 and a_3. The places corresponding to the two predator populations are a_2 and a_4. Let us assume that the interactions inside a given site have the same rate constants as in the one-site Lotka-Volterra model we examined before. Somewhat arbitrarily, let us set the migration rate for prey equal to $c_{r_4} = c_{-r_4} = 0.01$ and for predator $c_{r_5} = c_{-r_5} = 0.1$. Thus the full set of reaction rates is:

reaction	r_1	r_2	r_3	r_4	$-r_4$	r_5	$-r_5$	r_6	r_7	r_8
rate constant	1	0.005	0.6	0.01	0.01	0.1	0.1	1	0.005	0.6

1. Write down the matrices U_{pre}, U_{post} and U. (Assume that the reactions are ordered as in the above table of reaction rates and that places are ordered as in the graph a_1, a_2, a_3, a_4.)

2. Assume that initially the population sizes are 100 at a_1, 50 at a_2, and zero at the other places. Do a simulation of the process using `gillespied` for

$T = 150$, $dt = 0.05$, and obtain a graph of the predator population at the site that was initially not populated. I.e., plot the population size of a_4 as a function of time. (This took the `gillespied` program about one minute on my laptop.)

References

[Wilk] Darren J. Wilkinson. *Stochastic Modelling for Systems Biology*, Chapman and Hall/CRC, 2006.