Homework 1: question 4

Table of Contents

Starting with the implementation of the birth process, implement the τ -leaping algorithm, and the approximation of τ -leaping using increments chosen from a normal distribution. Run 10,000 realizations of the Yule process with each algorithm and compare the distribution of states after some fixed time T.

Wei-Ting Lin

psID: 1192656

clean workspace

```
clear; clc
```

The Yule process

Model parameter

```
    lambda = 1; % per capita birth rate
% Simulation parameters
    n0 = 10 % initial value
    tau= 0.05; % "tau"
    T = 6;  % total time
        points =T/tau; % number of time points in the simulation
% Variable
    N =nan(1, points); % state, population size
%
    it = 100; % iteration
    NendPois = nan(1, it);    % end points of the Poisson simulation
    NendNorm = nan(1, it); % end points of the Normal
```

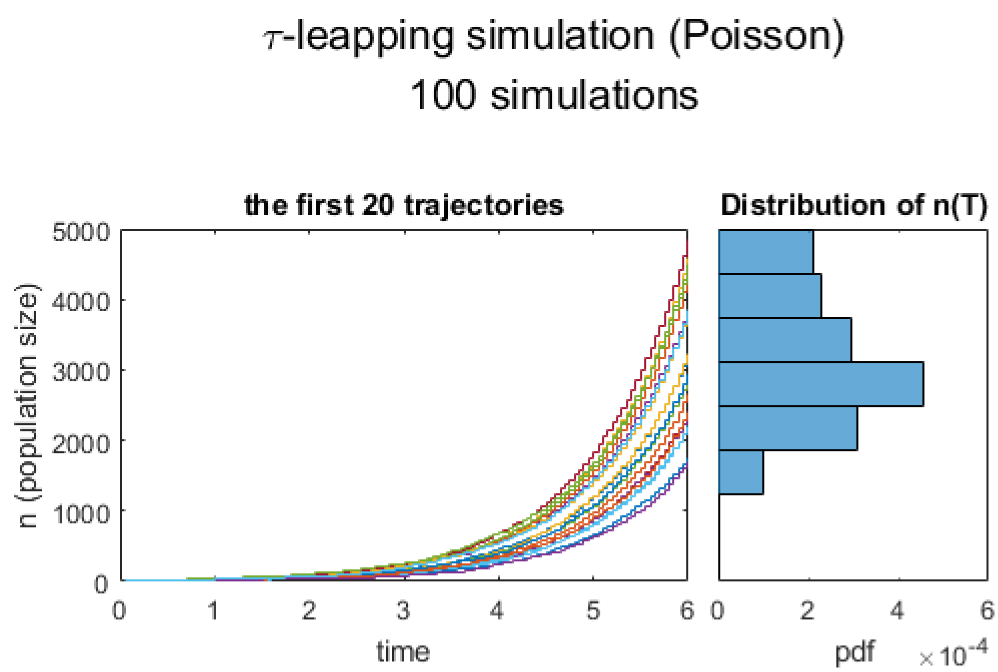
```
n0 =

    10
```

Poisson algorism

```
rng(123)
% make major title with my plot
mysubplot(1,3, 0, {'\tau-leapping simulation (Poisson)',
 [num2str(it) ' simulations']})
tic
```

```
mysubplot(1,3,1:2, [], [], 0.1,0.1)
for i = 1: it
    N(1) = n0;
    for p = 1: (points-1)
        n = N(p); % N at the begining of interval
        events = random('Poisson', tau * lambda * n); % number of
 events
        N(p+1) = n + events;
    end
    if i <= 20
            stairs((1:points) * tau,N); hold on
    end
    vaxis = ylim;
    NendPois(i) = N(end);
end
ylabel('n (population size)'); xlabel ('time'); title('the first 20
 trajectories')
toc
% Plot distribution of states N(T)
    mysubplot(1,3,3, [], [], 0.1,0.1)
    histogram(NendPois,'Orientation','horizontal','BinLimits', vaxis,
  'Normalization','pdf')
    xlabel('pdf'); title('Distribution of n(T)')
    set(gca, 'ytick', [])
```

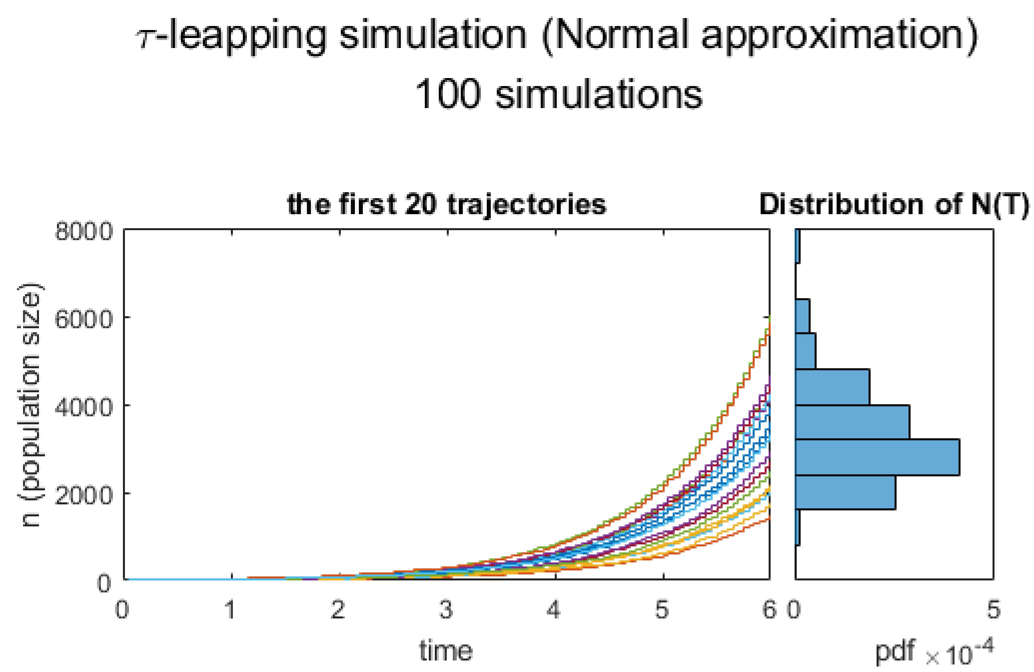
Elapsed time is 0.931794 seconds.



Normal algorism

```
rng(123) % set seed

figure
% make major title
mysubplot(1,4, 0,{ '\tau-leapping simulation (Normal approximation)',
 [num2str(it) ' simulations']})
tic
mysubplot(1,4,1:3, [], [], 0.1,0.1)
for i = 1: it
    N(1) = n0;
    for p = 1: (points-1)
        n = N(p); % N at the begining of interval
        rate = tau * lambda * n;
        events = randn*sqrt(rate) + rate; % number of events
                           % std,              mean
        N(p+1) = n + events;
        if   N(p+1) < 0 % normal random variable could give
 unreasonable negative numbers
              N(p+1) =0;
        end
    end
    if i <= 20
            stairs((1:points) * tau,N); hold on
    end
    vaxis = ylim;
    NendNorm(i) = N(end);
end
ylabel('n (population size)'); xlabel ('time'); title('the first 20
 trajectories')
toc
% Plot distribution of states N(T)
    mysubplot(1,4,4, [], [], 0.1,0.1)
    histogram(NendNorm,'Orientation','horizontal','BinLimits', vaxis,
  'Normalization','pdf')
    xlabel('pdf'); title('Distribution of N(T)')
    set(gca, 'ytick', [])
```

Elapsed time is 0.151080 seconds.



Published with MATLAB® R2015b