Numerical and simulation methods for solving integral equations in branching processes

Plamen Trayanov

Sofia University, Faculty of Mathematics and Informatics, Department of Probability, Operational Research and Statistics

Badajoz 2018, 10-13 April, 2018

The research is supported by the National Fund for Scientific Research at the Ministry of Education and Science of Bulgaria, grant No DFNI-I02/17 and partially supported by the financial funds allocated to the Sofia University "St. Kl. Ohridski", grant No 80-10-146/21.04.2017.

Problem Formulation

The general equation

$$Z(t) = z(t) + \int_0^t f(y, Z(t-y))\, \mathrm{d}G(y). \qquad (1.1)$$

Equations of this type could be seen for example for Galon-Watson BP, Sevastyanov BP, Bellman-Harris BP, Crump-Mode-Jagers BP (see [Sevastyanov, 1971, Mitov and Yanev, 985a, Jagers, 1975, Haccou et al., 2005, Crump and Mode, 1968, Crump and Mode, 1969, Sagitov and Serra, 2009, Serra, 2006, Serra and Haccou, 2007].

Properties of the solution (1)

Lemma 1 (Lemma)

Let the function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is continuous, function $z : \mathbb{R} \to \mathbb{R}$ – differentiable and the function $G : [0, \infty] \to \mathbb{R}$ has a continuous second derivative. Then the solution to the following equation

$$Z(t) = z(t) + \int_0^t f(y, Z(t-y))\, \mathrm{d}G(y) \tag{2.1}$$

is also differentiable and thus also continuous. If in addition G has continuous second derivative and z continuous first derivative, then Z has continuous derivative.

In the lemma above the condition for smoothness of $G(t)$ and $z(t)$ can be relaxed to smooth or continuous functions almost everywhere, except finite number of points.

Properties of the solution (2)

Lemma 2 (Lemma)

Let the assumptions of smoothness in Lemma 2 hold, except at point t_0, where we relax the conditions to have a jump discontinuity. Then

a) If G' is discontinuous at t_0, then Z' is also discontinuous at t_0. The jump of Z' at t_0 is $\delta'_Z(t_0) = \delta_G(t_0) \cdot f(Z(0))$;

b) If G is discontinuous at t_0, then Z is also discontinuous at t_0. The jump of Z at t_0 is $\delta_Z(t_0) = f(Z(0))[G_+(t_0) - G_-(t_0)]$;

c) If z' is discontinuous at t_0, then Z' is also discontinuous at t_0; The jump of Z' at t_0 is $\delta'_Z(t_0) = \delta'_z(t_0)$.

d) If z is discontinuous at t_0, then Z is also discontinuous at t_0; The jump of Z at t_0 is $\delta_Z(t_0) = \delta_z(t_0)$.

Note: If a) and c) are both satisfied at the same time, then the jump of Z' is the sum of the jumps in a) and c). If b) and d) are both satisfied at the same time, then the jump of Z is the sum of the jumps in b) and d).

Note: Even though Lemma 2 considers the case of a single jump discontinuity in G and z, a corresponding result obviously holds for the case of finite number of such jump points.

Numerical approximation (1)

Theorem 3

Let the function $f : [0,\infty]x[0,\infty] \to \mathbb{R}$ has continuous partial derivatives and the function $G : [0,\infty] \to \mathbb{R}$ has a continuous second derivative. Let the function $z : [0,\infty] \to \mathbb{R}$ is differentiable. Then the solution to the equation $Z(t) = z(t) + \int_0^h f(y, Z(t-y))\,\mathrm{d}G(y)$ can be approximated by the recurrence equation

$$\hat{Z}(kh) = z(kh) + f(h, \hat{Z}((k-1)h)) \cdot [G(h) - G(0)] + \dots$$
$$+ f(kh, \hat{Z}(0)) \cdot [G(kh) - G(kh-h)]; \quad (3.1)$$
$$\hat{Z}(0) = Z(0) = z(0),$$

where the approximation error is $E_{kh} = Z(kh) - \hat{Z}(kh) = O(h)$, for every $k = 1, \dots, t/h$. The approximation result continue to hold if the functions G, G', z and z' have finite number of jump discontinuities.

Notice that if f is the identity function, equation (2.1) turns into renewal equation. So the numerical method is also suitable for solving renewal equations.

Numerical approximation (2)

Theorem 4

Let the conditions of the Theorem 3 hold. The solution to the equation (2.1) can also be approximated by

$$\hat{Z}(kh) = z(kh) + f(h, \hat{Z}((k-1)h)) \cdot G'(h)h + \ldots + f(kh, \hat{Z}(0)) \cdot G'(kh)h;$$
$$\hat{Z}(0) = Z(0) = z(0), \tag{3.2}$$

where the approximation error is $E_{kh} = Z(kh) - \hat{Z}(kh) = O(h)$, *for every* $k = 1, \ldots, t/h$.

Theorem 4 can be used when we have a model for $G'(t)$, where Theorem 3 is used when we have a model for G, but not its derivative. The latter could happen for example if we model G with smoothing splines and L_2 roughness penalty. Then we have a reliable model for G, but the derivative of the model is not that stable.

Renewal equation case (1)

Theorem 5 (Renewal equation approximation)

If we consider the special case $f(u, Z(t-u)) = Z(t-u)$, then Equation (2.1) is a renewal equation. Let ω is such that $z(s) > 0$ for all $s \in [0, \omega)$ and $z(s) = 0$, for all $s \geq \omega$. Then the numerical approximation also has a matrix formulation:

$$Z_{kh} = z(0) \cdot \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}_{\mathbf{1} \times \frac{\omega}{h}} \cdot \mathbf{A}^k \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{\frac{\omega}{h} \times \mathbf{1}}$$

for every $k = 1, \dots, t/h$.

Previous numerical methods include [Xie, 1989, From, 2001, Bartholomew, 1963, Mitov and Omey, 2014]

Renewal equation case (2)

Theorem 6 (Renewal equation approximation)

The approximation matrix $\mathbf{A}$ *is defined as*

$$\begin{bmatrix} \frac{G(h)-G(\mathbf{0})}{z(\mathbf{0})}z(0) & \frac{G(2h)-G(h)}{z(h)}z(0) & \dots & \frac{G(\omega-h)-G(\omega-2h)}{z(\omega-2h)}z(0) & \frac{G(\omega)-G(\omega-h)}{z(\omega-h)}z(0) \\ \frac{z(h)}{z(\mathbf{0})} & 0 & \dots & 0 & 0 \\ 0 & \frac{z(2h)}{z(h)} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{z(\omega-h)}{z(\omega-2h)} & 0 \end{bmatrix}_{\frac{\omega}{h}\times\frac{\omega}{h}}.$$

Note that if $t \leq \omega$ then we can use a truncated matrix, consisting of the first $(t/h)+1$ rows and columns and we will get the same approximation. Also, if $z(s)>0$ for all s, then $\omega=+\infty$.

The Leslie matrix

For a human population $G(0) = 0$ and $z(0) = 1$. Lets consider $h = 1$ and denote $q_k = z(k+1)/z(k)$ to be the probability a woman aged k to survive to $k+1$ and $p_k = (G(k+1) - G(k))/q_k$ to be the probability a woman aged k to give a birth, $k = 0, \dots \omega - 1$. Then the approximation matrix $\mathbf{A}$ coincides with the Leslie matrix in demography ([Leslie, 1945, Leslie, 1948]):

$$\mathbf{A} = \begin{bmatrix} p_0 & p_1 & \dots & p_{\omega-1} & p_\omega \\ q_0 & 0 & \dots & 0 & 0 \\ 0 & q_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & q_{\omega-1} & 0 \end{bmatrix}^k_{\omega\times\omega}.$$

The solution of the integral equation, $Z(k)$, is then the expected future population count in k years.

Example. Virtues and restrictions (1)

Virtues

1. One code to simulate Galton-Watson, Bellman-Harris and General Branching Processes;
2. Considers **multi-type processes** with mutations between them;
3. Mutation probabilities could be constant, varying or random;
4. Considers constant, varying or random **environment**;
5. Considers constant, varying or random **immigration**;
6. Considers constant, varying or random **initial number of particles** on constant, varying or random ages;

Simulation Code. Matlab

```matlab
function [Z, Z_types]=simulate_BP(sim_num, T, h, draw_S, draw_H, draw_U, draw_Z_0, draw_mu, draw_Im, approx_limit)
if(isnumeric(draw_S) && ismatrix(draw_S)), draw_S=@()(repmat(draw_S,[1,1,T/h+1])); end
if(isnumeric(draw_H) && ismatrix(draw_H)), draw_H=@()(repmat(draw_H,[1,1,T/h+1])); end
if(isnumeric(draw_U) && ismatrix(draw_U)), draw_U=@()(repmat(draw_U,[1,1,T/h+1])); end
if(isnumeric(draw_Z_0) && ismatrix(draw_Z_0)), draw_Z_0=@()(draw_Z_0); end
if(~isempty(draw_mu) && isnumeric(draw_mu) && ismatrix(draw_mu)), draw_mu=@()(repmat(draw_mu,[1,1,T/h+1])); end
if(~isempty(draw_Im) && isnumeric(draw_Im) && ismatrix(draw_Im)), draw_Im=@()(repmat(draw_Im,[1,1,T/h+1])); end
n_types=size(draw_S(),2);     % the number of types
Z_types=zeros(sim_num, n_types, T/h+1);     % the branching process by types - sum of all individuals by types at time [0, T].
parfor ind=1:sim_num
  S=draw_S(); H=draw_H(); U=draw_U(); Z_0=draw_Z_0();
  if ~isempty(draw_mu), mu=draw_mu(); end;
  if ~isempty(draw_Im), Im=draw_Im(); end;
  N=zeros(size(S));     % population structure - N(age, type, time)
  if(size(Z_0,1)~=size(S,1)), N(1,:,1)=Z_0; else N(:,:,1)=Z_0; end
  for t=1:T/h       % go through time
    if ~isempty(draw_Im)  % add immigration
      N(:,:,t)=N(:,:,t)+Im(:,:,t);
    end
    if any(any(N(:,:,t))) || ~isempty(draw_Im)  % if there is at least one particle of any type at time t, simulate for t+h
      for i=1:size(N,1)-1    % for each age
        for j=1:size(N,2)  % for each type of particle
          if N(i,j,t)~=0 && S(i,j,t)~=0 % if no particles of that type are alive, no need to simulate their deaths
            deaths=N(i,j,t)-binornd_large(N(i,j,t), S(i+1,j,t)/S(i,j,t), approx_limit);
            if isempty(draw_mu) % assumes the particle splits at its death if mu is unspecified
              births=mnrnd_large(deaths, H(:,j,t)',1, approx_limit)*(0:(length(H(:,j,t))-1))';
            else
              births=mnrnd_large(binornd_large(N(i,j,t), mu(i,j,t)*h, approx_limit), H(:,j,t)',1, approx_limit)*(0:(length(H(:,j,t))-1))';
            end
            N(i+1,j,t+1)=N(i,j,t)-deaths;   % the ones that survived get to get older by h
            N(1,:,t+1)=N(1,:,t+1) + mnrnd_large(births, U(:,j,t)',1, approx_limit);
          end
        end
      end
    else       % if the population count is 0 at t, and we have no immigration,
      break;   % no need to simulate any further
    end
  end
  Z_types(ind, :, :)=sum(N,1);  % sum of all individuals by type
end
Z=squeeze(sum(Z_types, 2));     % sum of all types
end
```

Example Code. Defining the simulation parameters

```
%% EXAMPLE CODE: multi-type GBP in random environment, with mutations, immigration and
% random initial age structure
sim_num=100; % number of simulations to perform
T=250;    % simulate the branching process in the interval [0, T].
h=0.1;  % time discretization
omega=110;    % the maximum lifelength until time T.

% Example for time invariant H:
H=[1, 0; 0, 1]';     % set to be time invariant

% Example for H in random environment:
function H=draw_H(T, h)
H=zeros(3, 2, T/h+1);
    for t=1:T/h
        p_step=-0.2/(T/h);
        p=binornd(1,1:p_step:(0.8-p_step)); % changing and random environment
        H(:,:,t)=p*[1, 0, 0; 0, 1, 0]'+(1-p)*[1, 0, 0; 0, 0.5, 0.5]';
    end
end

U=[1, 0; 0.55, 0.45]'; % set to be time invariant (could be also set to random)
Z_0=@()([zeros(omega/h+1, 1), mnrnd(1+binornd(10,0.5), ones(1,omega/h+1)./(omega/h+1))']);
[Z, Z_types]=simulate_BP(sim_num, T, h, @()(draw_S(T,h,omega)), @()(draw_H(T,h)), U, Z_0, ...
    @()(draw_mu(T,h,omega)), @()(draw_Im(T,h,omega)), 20);
```

```
function S=draw_S(T, h, omega) % generate random path for S
S=zeros(omega/h+1, 2, T/h+1);
for t=1:T/h
    mean_w=unifrnd(70, 74);
    std_w=unifrnd(8,12);
    mean_m=unifrnd(74, 78);
    std_m=unifrnd(9,11);
    S(:,:,t)=[(1-normcdf(0:h:omega, mean_w, std_w)')./(1-normcdf(0,mean_w, std_w)), ...
        (1-normcdf(0:h:omega,mean_m, std_m)')./(1-normcdf(0,mean_m, std_m))];
    S(end,:,t)=0;
end
end
function mu=draw_mu(T, h, omega)
% generate the path of function mu
% mean is increasing
mu_mean=linspace(28,35,T/h+1);
mu=zeros(omega/h+1, 2, T/h+1);
for t=1:T/h
    m=unifrnd(1.5, 2);
    mu_women_pdf=normpdf(0:h:omega, mu_mean(t), 5)';
    mu_women_pdf([1:12/h, 50/h:end])=0;
    mu(:,:,t)=[zeros(size(mu_women_pdf)), mu_women_pdf*m/(sum(mu_women_pdf)*h)];
end
end
function Im=draw_Im(T, h, omega)
% generate the path of function mu
% mean is increasing
Im=zeros(omega/h+1, 2, T/h+1);     % no immigration at time 0
for t=1:T/h+1
    Im(:,:,t)=[zeros(omega/h+1, 1), mnrnd(binornd(1, h*0.01), ones(1,omega/h+1)./(omega/h+1))'];
end
end
```

Simulations. Virtues and restrictions (1)

Virtues

1. One code to simulate Galton-Watson, Bellman-Harris and General Branching Processes;
2. Considers **multi-type processes** with mutations between them;
3. Mutation probabilities could be constant, varying or random;
4. Considers constant, varying or random **environment**;
5. Considers constant, varying or random **immigration**;
6. Considers constant, varying or random **initial number of particles** on constant, varying or random ages;

Simulations. Virtues and restrictions (2)

Other Virtues

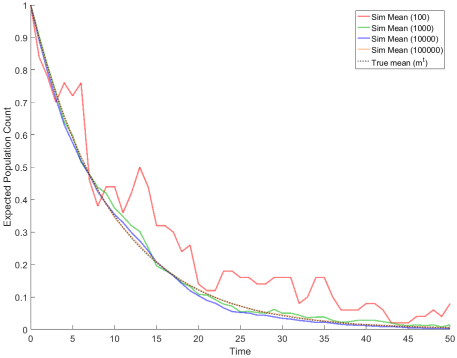
1. It presents a **simple short code** (only 43 lines) that simulates branching processes;
2. It is capable of **simulating VERY large number** of particles (as an example: 10^{250}) in the branching processes, without requiring a lot of RAM (by using normal approximation of binomial distribution when possible);
3. The simulation is actually faster for large populations due to the normal approximation;
4. It produces not only the total population count, but also the **population count by age and type at each moment of time**. Returning the population count by age as an output however may require a lot of computer memory (more than a 100 GB RAM in some cases);
5. It could be extended to include controlled branching processes. However, this would most probably require the code to be customized only for the specific process as the possibilities for such theoretical dependences between population size, birth, death and migration laws could be quite large;
6. Uses all computer cores for faster calculation.

Simulations. Virtues and restrictions (3)

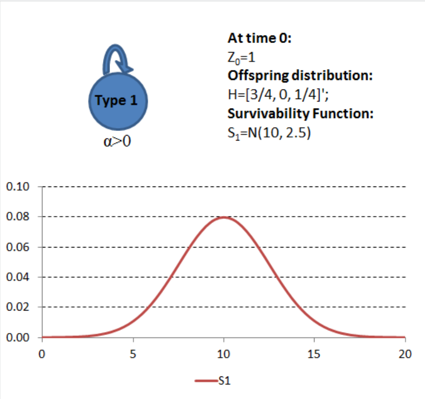
Restrictions

1. Particles cannot die/give birth at the beginning of their life. Probability of that event is considered zero;
2. Birth and death densities must be smooth functions with exception of finite jump type discontinuities;
3. The birth, death distributions, mutation probabilities and immigration are independent with each other and with the branching process itself. I.e. the simulation does not include the class of "controlled branching processes although it could be extended to suit the specific needs. Depending on the type of controlled process and the dependence on the age structure at precious times, this could require a lot of memory;
4. It considers only immigration, as emigration is naturally dependent on the population count and age structure and could be modelled in variety of ways.

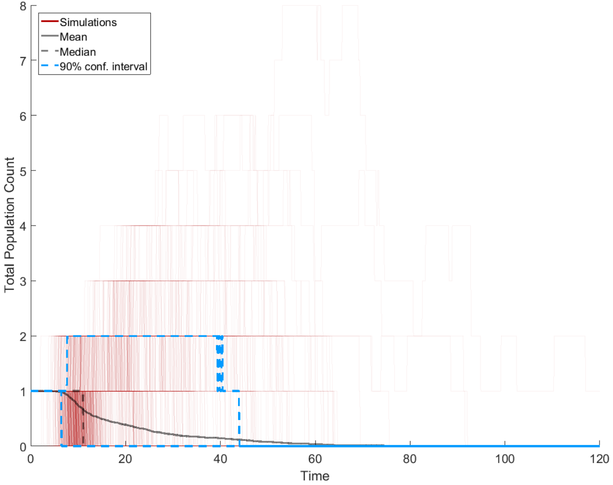
Example 1. The classical Galton-Watson



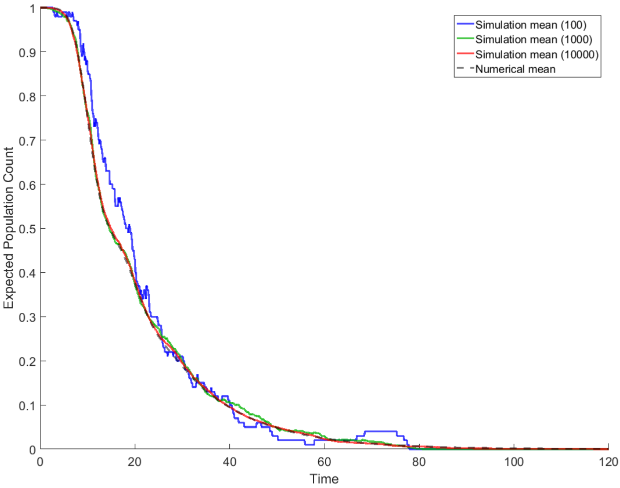
Example 2. Bellman-Harris (1)



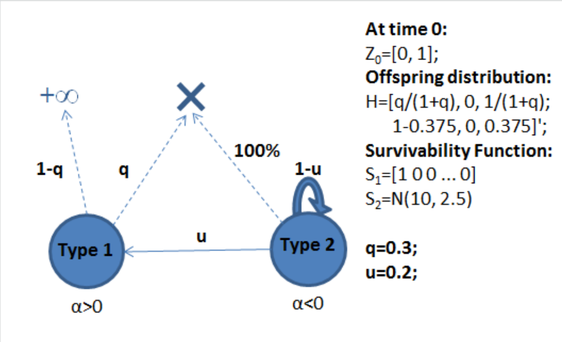
Example 2. Bellman-Harris (2)



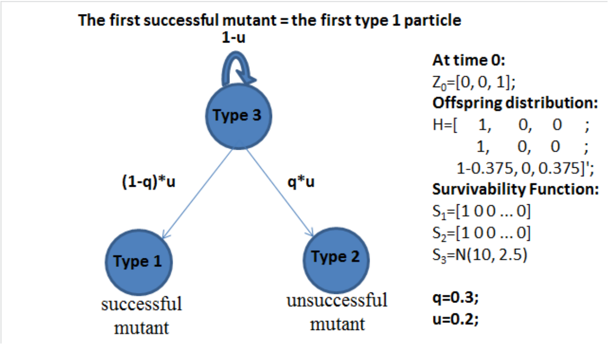
Example 2. Bellman-Harris (3)



Example 3. Bellman-Harris (1)



Example 3. Bellman-Harris (2)



Example 3. Bellman-Harris (3)

T - waiting time for 'successful mutant' appearance

Theorem 7

The probability $\mathbb{P}(T > t)$ that successful mutant has not been born yet satisfies the following integral equation:

$$\mathbb{P}(T > t) = 1 - G_1(t) + \int_0^t f_1(uq_0 + (1-u)\mathbb{P}(T > t-y))\,\mathrm{d}G_1(y). \qquad (5.1)$$

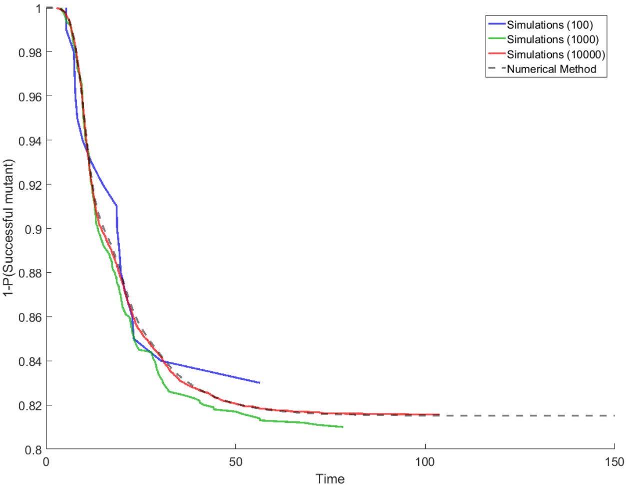
Example 1. Bellman-Harris (4)

Theorem 8

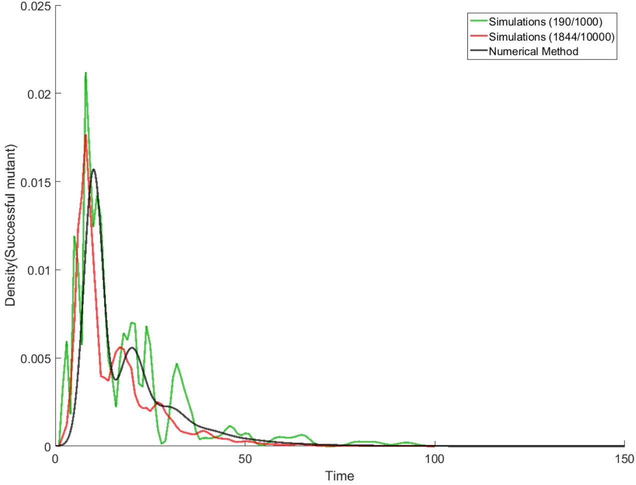
The joint probability that successful mutant has not been born yet and we do not have cells of type 1 (with subcritical reproduction, $m_1 < 1$) satisfies the following integral equation:

$$\mathbb{P}(T > t, Z^1(t) = 0) = \int_0^t f_1(uq_0 + (1-u)\mathbb{P}(T > t-y, Z^1(t-y) = 0))\, \mathrm{d}G_1(y). \tag{5.2}$$

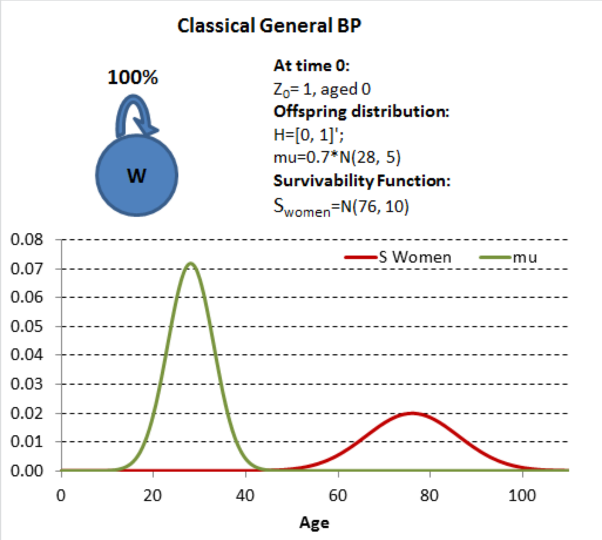
Example 3. Bellman-Harris (5)



Example 3. Bellman-Harris (6)



Example 4. General BP (1)



Example 4. General BP (2)

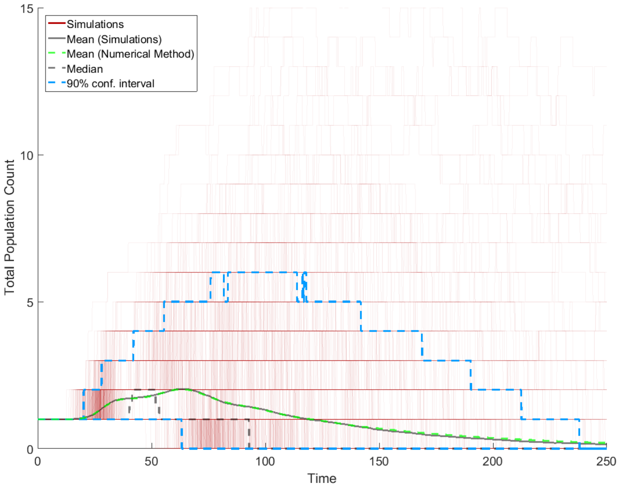
In GBP, the expected population count satisfies an integral equation.

Theorem 9

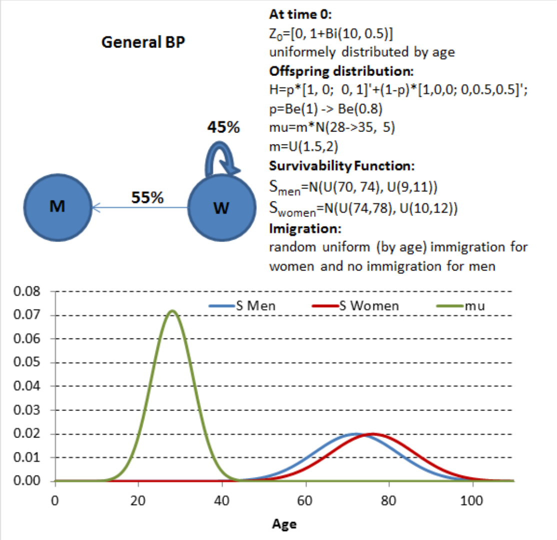
Let $\mu(t) = \mu([0,t]) < \infty$ *for* $t > 0$. *Let* $L(t) = \mathbb{P}(\lambda_x \le t)$ *and* $S(t) = 1 - L(t)$ *is the survival probability function. Then* $m_t = \mathbb{E}(z_t) < \infty, \forall t$ *and* $m_t^a = \mathbb{E}(z_t^a)$ *satisfies*

$$m_t^a = 1_{[0,a]}(t)\{1 - L(t)\} + \int_0^t m_{t-u}^a \, \mu(\mathrm{d}u). \tag{5.3}$$

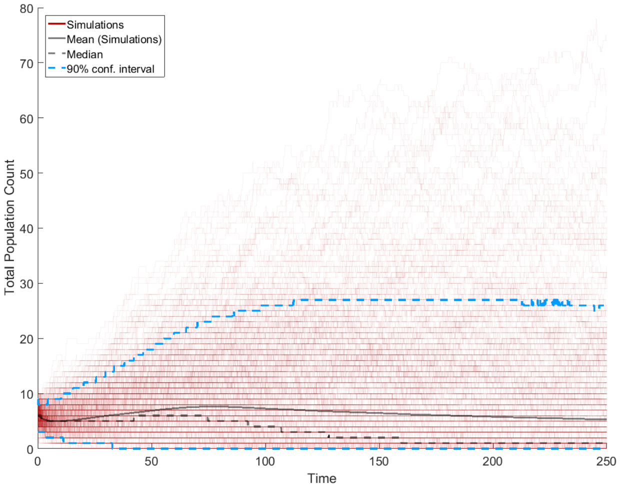
Example 4. General BP (3)



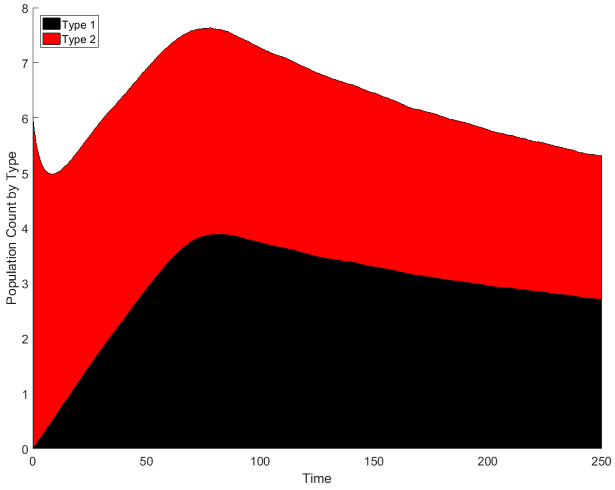
Example 5. General BP (1)



Example 5. General BP (2)



Example 5. General BP (3)



THANK YOU!

References I

Bartholomew, D. (1963).
An approximate solution to the integral equation of renewal theory.
J. Royal Statist. Soc., B 25(2):432–441.

Crump, K. and Mode, C. (1968).
A general age-dependent branching process I.
Journal of Mathematical Analysis and Applications, 24(3):494–508.

Crump, K. and Mode, C. (1969).
A general age-dependent branching process II.
Journal of Mathematical Analysis and Applications, 25(1):8–17.

From, S. (2001).
Some new approximations for the renewal function.
Commun. Statist.-Simula., 30(1):113–128.

Haccou, P., Jagers, P., and Vatutin, V. (2005).
Branching Processes: Variation, Growth, and Extinction of Populations.
Cambridge University Press.

References II

Jagers, P. (1975).
Branching Processes with Biological Applications.
John Wiley & Sons Ltd, first edition.

Leslie, P. (1945).
On the use of matrices in certain population mathematics.
Biometrika, 33(3):183–212.

Leslie, P. (1948).
Some further notes on the use of matrices in population mathematics.
Biometrika, 35:213–245.

Mitov, K. and Omey, E. (2014).
Intuitive approximations for the renewal function.
Statistics and Probability Letters, 84:72–80.

Mitov, K. and Yanev, N. (1985a).
Bellman–Harris branching processes with state-dependent immigration.
J. Appl. Prob., 22:757–765.

References III

Sagitov, S. and Serra, M. (2009).
Multitype bienayme–galton–watson processes escaping extinction.
Adv. Appl. Prob., 41:225–246.

Serra, M. (2006).
On waiting time to escape.
J. Appl. Prob., 43:296–302.

Serra, M. and Haccou, P. (2007).
Dynamics of escape mutants.
Theor. Popul. Biol., 72:167–178.

Sevastyanov, B. (1971).
Branching Processes.
Nauka.

Xie, M. (1989).
On the solution of renewal-type integral equations.
Communications in Statistics - Simulation and Computation, 18(1):281–293.