




Efficient simulation of individual-based population models

The R package IBMPopSim

ISSN 2824-7795

Daphné Giorgi Laboratoire de Probabilités, Statistique et Modélisation, Sorbonne Université, CNRS
Vincent Lemaire ¹ Laboratoire de Probabilités, Statistique et Modélisation, Sorbonne Université, CNRS
Sarah Kaakai² Laboratoire Manceau de Mathématiques, Le Mans Université, CNRS, Institut du Risque et de l'Assurance

Date published: 2024-03-16 Last modified: 2024-03-16

Abstract

The R Package IBMPopSim aims to simulate the random evolution of heterogeneous populations using stochastic Individual-Based Models (IBMs). The package enables users to simulate population evolution, in which individuals are characterized by their age and some characteristics, and the population is modified by different types of events, including births/arrivals, death/exit events, or changes of characteristics. The frequency at which an event can occur to an individual can depend on their age and characteristics, but also on the characteristics of other individuals (interactions). Such models have a wide range of applications in fields including actuarial science, biology, ecology or epidemiology. IBMPopSim overcomes the limitations of time-consuming IBMs simulations by implementing new efficient algorithms based on thinning methods, which are compiled using the Rcpp package while providing a user-friendly interface.

Keywords: Individual-based models, stochastic simulation, population dynamics, Poisson measures, thinning method, actuarial science, insurance portfolio simulation

Contents

1	Contents	
2	1 Introduction	2
3	2 Stochastic Individual-Based Models (IBMs) in IBMPopSim	3
4	2.1 Brief package overview	4
5	2.1.1 Model creation	4
6	2.1.2 Simulation	5
7	2.2 Population	5
8	2.2.1 Notations	5
9	2.2.2 Individuals	6
10	2.2.3 Population process	6
11	2.3 Events	6
12	2.4 Events intensity	8
13	2.4.1 Events intensity bounds	9
14	2.5 Pathwise representation	10
15	2.6 Thinning of Poisson measure	11
16	References	11

¹Corresponding author: vincent.lemaire@sorbonne-universite.fr

²Corresponding author: sarah.kaakai@univ-lemans.fr

1 Introduction

In various fields, advances in probability have contributed to the development of a new mathematical framework for so-called individual-based stochastic population dynamics, also called stochastic Individual-Based Models (IBMs).

Stochastic IBMs allow the modeling in continuous time of populations dynamics structured by age and/or characteristics. In the field of mathematical biology and ecology, a large community has used this formalism for the study of the evolution of structured populations (see e.g. (Ferrière and Tran 2009; Collet, Méléard, and Metz 2013; Bansaye and Méléard 2015; Costa et al. 2016; Billiard et al. 2016; Lavallée et al. 2019; Méléard, Rera, and Roget 2019; Calvez et al. 2020)), after the pioneer works (Fournier and Méléard 2004; Champagnat, Ferrière, and Méléard 2006; Tran 2008).

IBMs are also useful in demography and actuarial sciences, for the modeling of human populations dynamics (see e.g. (Bensusan 2010; Boumezoued 2016; El Karoui, Hadji, and Kaakai 2021)). They allow the modeling of heterogeneous and complex population dynamics, which can be used to compute demographic indicators or simulate the evolution of insurance portfolios in order to study the basis risk, compute cash flows for annuity products or pension schemes, or for a fine assessment of mortality models (Barrieu et al. 2012). There are other domains in which stochastic IBMs can be used, for example in epidemiology with stochastic compartmental models, neurosciences, cyber risk, or Agent-Based Models (ABMs) in economy and social sciences, which can be seen as IBMs. Many mathematical results have been obtained in the literature cited above, for quantifying the limit behaviors of IBMs in long time or in large population. In particular, pathwise representations of IBMs have been introduced in (Fournier and Méléard 2004) (and extended to age-structured populations in (Tran 2008)), as measure-valued pure jumps Markov processes, solutions of SDEs driven by Poisson measures. These pathwise representations are based on the *thinning* and projection of Poisson random measures defined on extended spaces. However, the simulation of large and interacting populations is often referred as computationally expensive.

The aim of the R package `IBMPopSim` is to meet the needs of the various communities for efficient tools in order to simulate the evolution of stochastic IBMs. `IBMPopSim` provides a general framework for the simulation of a wide class of IBMs, where individuals are characterized by their age and/or a set of characteristics. Different types of events can be included in the modeling by users, depending on their needs: births, deaths, entry or exit in/to the population and changes of characteristics (swap events). Furthermore, the various events that can happen to individuals in the population can occur at a non-stationary frequency, depending on the individuals' characteristics and time, and also including potential interactions between individuals.

We introduce a unified mathematical and simulation framework for this class of IBMs, generalizing the pathwise representation of IBMs by thinning of Poisson measures, as well as the associated population simulation algorithm, based on an acceptance/rejection procedure. In particular, we provide general sufficient conditions on the event intensities under which the simulation of a particular model is possible.

We opted to implement the algorithms of the `IBMPopSim` package using the `Rcpp` package, a tool facilitating the seamless integration of high-performance C++ code into easily callable R functions (Eddelbuettel and Francois 2011). With just a few lines of C++ code, `IBMPopSim` offers user-friendly R functions for defining IBMs. Once events and their associated intensities are specified, an automated procedure creates the model. This involves integrating the user's source code into the primary C++ code using a template mechanism. Subsequently, `Rcpp` is invoked to compile the model and integrate

it into the R session. Following this process, the model becomes callable with varying parameters, enabling the generation of diverse population evolution scenarios. Combined with the design of the simulation algorithms, the package structure yields very competitive simulation runtimes for IBMs, while staying user-friendly for R users. Several outputs function are also implemented in IBMPopSim. For instance the package allows the construction and visualization of age pyramids, as well as the construction of death and exposures table from the censored individual data, compatible with R packages concerned with mortality modelling, such as (Hyndman et al. 2023) or (Villegas, Millosovich, and Kaishev Hyndman 2018). Several examples are provided in the form of R vignettes on the [website](#), and in recent works of (El Karoui, Hadji, and Kaakai 2021) and (Roget et al. 2022).

Designed for applications in social sciences, the R package MicSim (Zinn 2014) can be used for continuous time microsimulation. In continuous-time microsimulation, individual life-courses are usually specified by sequences of state transitions (events) and the time spans between these transitions. The state space is usually discrete and finite, which is not necessarily the case in IBMPopSim, where individuals can have continuous characteristics. But most importantly, microsimulation does not allow for interactions between individuals. Indeed, microsimulation produces separately the life courses of all individuals in the populations, based on the computation of the distribution functions of the waiting times in the distinct states of the state space, for each individual (Zinn 2014). This can be slow in comparison to the simulation by thinning of event times occurring in the population, which is based on selecting event times among some competing proposed event times. Finally, MicSim simplifies the Mic-Core microsimulation tool implemented in Java (Zinn et al. 2009). However, the implementation in R of simulation algorithms yields longer simulation run times than when using Rcpp. To the best of our knowledge, there are no other R packages currently available addressing the issue of IBMs efficient simulation.

In Section 2, we introduce the mathematical framework that characterizes the class of Stochastic Individual-Based Models (IBMs) that can be implemented in the IBMPopSim package. In particular, a general pathwise representation of IBMs is presented. The population dynamics is obtained as the solution of an SDE driven by Poisson measures, for which we obtain existence and uniqueness results in Theorem 1. Additionally, a succinct overview of the package is provided. In `?@sec-simulation` the two main algorithms for simulating the population evolution of an IBM across the interval $[0, T]$ are detailed. In `?@sec-package` we present the main functions of the IBMPopSim package, which allow for the definition of events and their intensities, the creation of a model, and the simulation of scenarios. Two examples are detailed in `?@sec-insurance-portfolio` and `sec-example-interaction`, featuring applications involving an heterogeneous insurance portfolio characterized by entry and exit events, and an age and size-structured population with intricate interactions.

2 Stochastic Individual-Based Models (IBMs) in IBMPopSim

Stochastic Individual-Based Models (IBMs) represent a broad class of random population dynamics models, allowing the description of populations evolution on a microscopic scale. Informally, an IBM can be summarized by the description of the individuals constituting the population, the various types of events that can occur to these individuals, along with their respective frequencies. In IBMPopSim, individuals can be characterized by their age and/or a collection of discrete or continuous characteristics. Moreover, the package enables users to simulate efficiently populations in which one or more of the following event types may occur:

- **Birth event:** addition of an individual of age 0 to the population.
- **Death event:** removal of an individual from the population.
- **Entry event:** arrival of an individual in the population.
- **Exit (emigration) event:** exit from the population (other than death).

- **Swap event:** an individual changes characteristics.

Each event type is linked to an associated event kernel, describing how the population is modified following the occurrence of the event. For some event types, the event kernel requires explicit specification. This is the case for entry events when a new individual joins the population. Then, the model should specify how the age and characteristics of this new individual are chosen. For instance, the characteristics of a new individual in the population can be chosen uniformly in the space of all characteristics, or can depend on the distribution of his parents or those of the other individuals composing the population.

The last component of an IBM are the event intensities. Informally, an event intensity is a function $\lambda_t^e(I, Z)$ describing the frequency at which an event e can occur to an individual I in a population Z at a time t . Given a history of the population (\mathcal{F}_t), the probability of event e occurring to individual I during a small interval of time $(t, t + dt]$ is proportional to $\lambda^e(I, t)$:

$$P(\text{event } e \text{ occurring to } I \text{ during } (t, t + dt] | \mathcal{F}_t) \simeq \lambda_t^e(I, Z) dt.$$

The intensity function λ^e can include dependency on the individual's I age and characteristics, the time t , or the population composition Z in the presence of interactions.

2.1 Brief package overview

Prior to providing a detailed description of an Individual-Based Model (IBM), we present a simple model of birth and death in an age-structured *human* population. We assume no interactions between individuals, and individuals are characterized by their gender, in addition to their age. In this simple model, all individuals, regardless of gender, can give birth when their age falls between 15 and 40 years, with a constant birth rate of 0.05. The death intensity is assumed to follow a Gompertz-type intensity depending on age. The birth and death intensities are then given by

$$\lambda^b(t, I) = 0.05 \times \mathbf{1}_{[15,40]}(a(I, t)), \quad \lambda^d(t, I) = \alpha \exp(\beta a(I, t)),$$

with $a(I, t)$ the age of individual I at time t . Birth events are also characterized with a kernel determining the gender of the newborn, who is male with probability p_{male} .

2.1.1 Model creation

To implement this model in IBMPopSim, it is necessary to individually define each event type. In this example, the `mk_event_individual` function is used. The creation of an event involves a few lines of cpp instructions defining the intensity and, if applicable, the kernel of the event. For a more in depth description of the event creation step and its parameters, we refer to `?@sec-package_events`.

The events of this simple model are for example defined through the following calls.

```
birth_event <- mk_event_individual(
  type = "birth",
  intensity_code = "result = birth_rate(I.age(t));",
  kernel_code = "newI.male = CUnif(0,1) < p_male;")

death_event <- mk_event_individual(
  type = "death",
  intensity_code = "result = alpha * exp(beta * I.age(t));")
```

In the cpp codes, the names `birth_rate`, `p_male`, `alpha` and `beta` refer to the model parameters defined in the following list.

```

params <- list(
  "alpha" = 0.008, "beta" = 0.02,
  "p_male" = 0.51,
  "birth_rate" = stepfun(c(15, 40), c(0, 0.05, 0)))

```

In a second step, the model is created by calling the function `mk_model`. A cpp source code is automatically created through a template mechanism based on the events and parameters, subsequently compiled using the `sourceCpp` function from the `Rcpp` package.

```

birth_death_model <- mk_model(
  characteristics = c("male" = "bool"),
  events = list(death_event, birth_event),
  parameters = params)

```

2.1.2 Simulation

Once the model is created and compiled, the `popsim` function is called to simulate the evolution of a population according to this model. To achieve this, an initial population must be defined. In this example, we extract a population from a dataset specified in the package (a sample of 100 000 individuals based on the population of England and Wales in 2014). It is also necessary to set bounds for the events intensities. In this example, they are obtained by assuming that the maximum age for an individual is 115 years.

```

a_max <- 115
events_bounds = c(
  "death" = params$alpha * exp(params$beta * a_max),
  "birth" = max(params$birth_rate))

```

The function `popsim` can now be called to simulate the population starting from the initial population `population(EW_pop_14$sample)` up to time $T = 30$.

```

sim_out <- popsim(
  birth_death_model,
  population(EW_pop_14$sample),
  events_bounds,
  parameters = params, age_max = a_max,
  time = 30)

```

The data frame `sim_out$population` contains the information (birth, death, gender) on individuals who lived in the population over the period $[0, 30]$. Functions of the package allows to provide aggregated information on the population.

In the remainder of this section, we define rigorously the class of IBMs that can be simulated in `IBMPopSim`, along with the assumptions that are required in order for the population to be simulatable. The representation of age-structured IBMs based on measure-valued processes, as introduced in (Tran 2008), is generalized to a wider class of abstract population dynamics. The modeling differs slightly here, since individuals are *kept in the population* after their death (or exit), by including the death/exit date as an individual trait.

2.2 Population

2.2.1 Notations

In the remainder of the paper, the filtered probability space is denoted by $(\Omega, \{\mathcal{F}_t\}, \mathbb{P})$, under the usual assumptions. All processes are assumed to be càdlàg and adapted to the filtration $\{\mathcal{F}_t\}$ (for

instance the history of the population) on a time interval $[0, T]$. For a càdlàg process X , we denote
 $X_{t-} := \lim_{s \uparrow t} X_s$.

2.2.2 Individuals

An individual is represented by a triplet $I = (\tau^b, \tau^d, x) \in \mathcal{I} = \mathbb{R} \times \bar{\mathbb{R}} \times \mathcal{X}$ with:

- $\tau^b \in \mathbb{R}$ the date of birth,
- $\tau^d \in \bar{\mathbb{R}}$ the death date, with $\tau^d = \infty$ if the individual is still alive,
- a collection $x \in \mathcal{X}$ of characteristics where \mathcal{X} is the space of characteristics.

Note that in IBMs, individuals are usually characterized by their age $a(t) = t - \tau^b$ instead of their date of birth τ^b . However, using the latter is actually easier for the simulation, as it remains constant over time.

2.2.3 Population process

The population at a given time t is a random set

$$Z_t = \{I_k \in \mathcal{I}; k = 1, \dots, N_t\},$$

composed of all individuals (alive or dead) who have lived in the population before time t . As a random set, Z_t can be represented by a random counting measure on \mathcal{I} , that is an integer-valued measure $Z : \Omega \times \mathcal{I} \rightarrow \bar{\mathbb{N}}$ where for $A \in \mathcal{I}$, $Z(A)$ is the (random) number of individuals I in the subset A . With this representation:

$$Z_t(d\tau^b, d\tau^d, dx) = \sum_{k=1}^{N_t} \delta_{I_k}(\tau^b, \tau^d, x),$$

$$\text{with } \int_{\mathcal{I}} f(\tau^b, \tau^d, x) Z_t(d\tau^b, d\tau^d, dx) = \sum_{k=1}^{N_t} f(I_k).$$

The number of individuals present in the population *before time* t is obtained by taking $f \equiv 1$:

$$N_t = \int_{\mathcal{I}} Z_t(d\tau^b, d\tau^d, dx) = \sum_{k=1}^{N_t} \mathbf{1}_{\mathcal{I}}(I_k).$$

Note that $(N_t)_{t \geq 0}$ is an increasing process since dead/exited individuals are kept in the population Z .

The number of alive individuals in the population at time t is:

$$N_t^a = \int_{\mathcal{I}} \mathbf{1}_{\{\tau^d > t\}} Z_t(d\tau^b, d\tau^d, dx) = \sum_{k=1}^{N_t} \mathbf{1}_{\{\tau_k^d > t\}}. \quad (1)$$

Another example is the number of alive individuals of age over a is

$$N_t([a, +\infty)) := \int_{\mathcal{I}} \mathbf{1}_{[a, +\infty)}(t - \tau^b) \mathbf{1}_{]t, \infty]}(\tau^d) Z_t(d\tau^b, d\tau^d, dx) = \sum_{k=1}^{N_t} \mathbf{1}_{\{t - \tau_k^b \geq a\}} \mathbf{1}_{\{\tau_k^d \geq t\}}.$$

2.3 Events

The population composition changes at random dates following different types of events. IBMPopSim allows the simulation of IBMs with the following events types:

- A **birth** event at time t is the addition of a new individual $I' = (t, \infty, X)$ of age 0 to the population. Their date of birth is $\tau^b = t$, and characteristics is X , a random variable of distribution defined by the birth kernel $k^b(t, I, dx)$ on \mathcal{X} , depending on t and its parent I . The population size becomes $N_t = N_{t-} + 1$, and the population composition after the event is

$$Z_t = Z_{t-} + \delta_{(t, \infty, X)}.$$

- An **entry** event at time t is also the addition of an individual I' in the population. However, this individual is not of age 0. The date of birth and characteristics of the new individual $I' = (\tau^b, \infty, X)$ are random variables of probability distribution defined by the entry kernel $k^{en}(t, ds, dx)$ on $\mathbb{R} \times \mathcal{X}$. The population size becomes $N_t = N_{t-} + 1$, and the population composition after the event is:

$$Z_t = Z_{t-} + \delta_{(\tau^b, \infty, X)}.$$

- A **death** or **exit** event of an individual $I = (\tau^b, \infty, x) \in Z_{t-}$ at time t is the modification of its death date τ^d from $+\infty$ to t . This event results in the simultaneous addition of the individual (τ^b, t, x) and removal of the individual I from the population. The population size is not modified, and the population composition after the event is

$$Z_t = Z_{t-} + \delta_{(\tau^b, t, x)} - \delta_I.$$

- A **swap** event (change of characteristics) results in the simultaneous addition and removal of an individual. If an individual $I = (\tau^b, \infty, x) \in Z_{t-}$ changes of characteristics at time t , then it is removed from the population and replaced by $I' = (\tau^b, \infty, X)$. The new characteristics X is a random variable of distribution $k^s(t, I, dx)$ on \mathcal{X} , depending on time, the individual's age and previous characteristics x . In this case, the population size is not modified and the population becomes:

$$Z_t = Z_{t-} + \delta_{(\tau^b, \infty, X)} - \delta_{(\tau^b, \infty, x)}.$$

To summarize, the space of event types is $E = \{b, en, d, s\}$, and the jump $\Delta Z_t = Z_t - Z_{t-}$ (change in the population composition) generated by an event of type $e \in \{b, en, d, s\}$ is denoted by $\phi^e(t, I)$. We thus have the following rules summarized in the table Table 1.

Table 1: Action in the population for a given event name

Event	Type	$\phi^e(t, I)$	New individual
Birth	b	$\delta_{(t, \infty, X)}$	$\tau^b = t, X \sim k^b(t, I, dx)$
Entry	en	$\delta_{(\tau^b, \infty, X)}$	$(\tau^b, X) \sim k^{en}(t, ds, dx)$
Death/Exit	d	$\delta_{(\tau^b, t, x)} - \delta_{(\tau^b, \infty, x)}$	$\tau^d = t$
Swap	s	$\delta_{(\tau^b, \infty, X)} - \delta_{(\tau^b, \infty, x)}$	$X \sim k^s(t, I, dx)$

Remark 2.1 (Composition of the population).

- At time T , the population Z_T contains all individuals who lived in the population before T , including dead/exited individuals. If there are no swap events, or entries, the population state Z_t for any time $t \leq T$ can be obtained from Z_T . Indeed, if $Z_T = \sum_{k=1}^{N_T} \delta_{I_k}$, then the population at time $t \leq T$ is simply composed of the individuals born before t :

$$Z_t = \sum_{k=1}^{N_T} \mathbf{1}_{\{\tau_k^b \leq t\}} \delta_{I_k}.$$

- In the presence of entries (open population), a characteristic x can track the individuals' entry dates. Then, the previous equation can be easily modified in order to obtain the population Z_t at time $t \leq T$ from Z_T .

2.4 Events intensity

Once the different event types have been defined in the population model, the frequency at which each event occur in the population e have to be specified. Informally, the intensity $\Lambda_t^e(Z_t)$ at which an event e can occur is defined by

$$\mathbb{P}(\text{event } e \text{ occurs in the population } Z_t \in (t, t + dt] | \mathcal{F}_t) \simeq \Lambda_t^e(Z_t) dt.$$

For a more formal definition of stochastic intensities, we refer to (Brémaud 1981) or (Kaakai and El Karoui 2023). The form of the intensity function ($\Lambda_t^e(Z_t)$) determines the population simulation algorithm in IBMPopSim:

- When the event intensity does not depend on the population state,

$$(\Lambda_t^e(Z_t))_{t \in [0, T]} = (\mu^e(t))_{t \in [0, T]}, \quad (2)$$

with μ^e a deterministic function, the events of type e occur at the jump times of an inhomogeneous Poisson process of intensity function $(\mu^e(t))_{t \in [0, T]}$. When such an event occurs, the individual to whom the event happens to is drawn uniformly among alive individuals in the population. In a given model, the set of events $e \in E$ with Poisson intensities will be denoted by \mathcal{P} .

- Otherwise, we assume that the global intensity $\Lambda_t^e(Z_t)$ at which the events of type e occur in the population can be written as the sum of individual intensities $\lambda_t^e(I, Z_t)$:

$$\Lambda_t^e(Z_t) = \sum_{k=1}^{N_t} \lambda_t^e(I_k, Z_t),$$

$$\text{with } \mathbb{P}(\text{event } e \text{ occurs to an individual } I \in (t, t + dt] | \mathcal{F}_t) \simeq \lambda_t^e(I, Z_t) dt.$$

Obviously, nothing can happen to dead or exited individuals, i.e. individuals $I = (\tau^b, \tau^d, x)$ with $\tau^d \leq t$. Thus, individual event intensities are assumed to be null for dead/exited individuals:

$$\lambda_t^e(I, Z_t) = 0, \text{ if } \tau^d \leq t, \text{ so that } \Lambda_t^e(Z_t) = \sum_{k=1}^{N_t^a} \lambda_t^e(I_k, Z_t),$$

with N_t^a the number of alive individuals at time t .

The event's individual intensity $\lambda_t^e(I, Z_t)$ can depend on time (for instance when there is a mortality reduction over time), on the individual's age $t - \tau^b$ and characteristics, but also on the population composition Z_t . The dependence of λ^e on the population Z models interactions between individuals in the populations. Hence, two types of individual intensity functions can be implemented in IBMPopSim:

1. *No interactions*: The intensity function λ^e does not depend on the population composition. The intensity at which the event of type e occur to an individual I only depends on its date of birth and characteristics:

$$\lambda_t^e(I, Z_t) = \lambda^e(t, I), \quad (3)$$

where $\lambda^e : \mathbb{R}_+ \times \mathcal{I} \rightarrow \mathbb{R}^+$ is a deterministic function. In a given model, we denote by \mathcal{E} the set of event types with individual intensity Equation 3.

2. “*Quadratic*” interactions: The intensity at which an event of type e occurs to an individual I depends on I and on the population composition, through an interaction function W^e . The quantity $W^e(t, I, J)$ describes the intensity of interactions between two alive individuals I and J at time t , for instance in the presence of competition or cooperation. In this case, we have

$$\lambda_t^e(I, Z_t) = \sum_{j=1}^{N_t} W^e(t, I, I_j) = \int_{\mathcal{I}} W^e(t, I, (\tau^b, \tau^d, x)) Z_t(d\tau^b, d\tau^d, dx), \quad (4)$$

where $W^e(t, I, (\tau^b, \tau^d, x)) = 0$ if the individual $J = (\tau^b, \tau^d, x)$ is dead, i.e. $\tau^d \leq t$. In a given model, we denote by \mathcal{E}_W the set of event types with individual intensity Equation 4.

To summarize, an individual intensity in IBMPopSim can be written as:

$$\lambda_t^e(I, Z_t) = \lambda^e(t, I) \mathbf{1}_{\{e \in \mathcal{E}\}} + \left(\sum_{j=1}^{N_t} W^e(t, I, I_j) \right) \mathbf{1}_{\{e \in \mathcal{E}_W\}}. \quad (5)$$

Example 2.1.

1. An example of death intensity without interaction for an individual $I = (\tau^b, \tau^d, x)$ alive at time t , $t < \tau^d$, is:

$$\lambda^d(t, I) = \alpha_x \exp(\beta_x a(I, t)), \text{ where } a(I, t) = t - \tau^b$$

is the age of the individual I at time t . In this classical case, the death rate of an individual I is an exponential (Gompertz) function of the individual’s age, with coefficients depending on the individual’s characteristics x .

2. In the presence of competition between individuals, the death intensity of an individual I also depend on other individuals J in the population. For example, if $I = (\tau^b, \tau^d, x)$, with x its size, then we can have:

$$W^d(t, I, J) = (x_J - x)^+ \mathbf{1}_{\{\tau_J^d > t\}}, \quad \forall J = (\tau_J^b, \tau_J^d, x_J). \quad (6)$$

This can be interpreted as follows: if the individual I meets randomly an individual J alive at time t , and of bigger size $x_J > x$, then he can die at the intensity $x_J - x$. If J is smaller than I , then he cannot kill I . The bigger is the size x of I , the lower is his death intensity $\lambda_t^d(I, Z_t)$ defined by

$$\lambda_t^d(I, Z_t) = \sum_{\substack{J \in Z_t \\ x_J > x}} (x_J - x) \mathbf{1}_{\{\tau_J^d > t\}}.$$

3. IBMPopSim can simulate IBMs that include intensities expressed as a sum of Poisson intensities and individual intensities of the form $\Lambda^e(Z_t) = \mu_t^e + \sum_{k=1}^{N_t} \lambda^e(I_k, Z_t)$. Other examples are provided in `?@sec-insurance-portfolio` and `?@sec-example-interaction`. Finally, the global intensity at which an event can occur in the population is defined by:

$$\Lambda_t(Z_t) = \sum_{e \in \mathcal{P}} \mu^e(t) + \sum_{e \in \mathcal{E}} \left(\sum_{k=1}^{N_t} \lambda^e(t, I_k) \right) + \sum_{e \in \mathcal{E}_W} \left(\sum_{k=1}^{N_t} \sum_{j=1}^{N_t} W^e(t, I_k, I_j) \right). \quad (7)$$

An important point is that for events $e \in \mathcal{E}$ without interactions, the global event intensity $\Lambda_t^e(Z_t) = \sum_{k=1}^{N_t} \lambda^e(t, I_k)$ is of order N_t^a defined in Equation 1 (number of alive individuals at time t). On the other hand, for events $e \in \mathcal{E}_W$ with interactions, $\Lambda_t^e(Z_t) = \sum_{k=1}^{N_t} \sum_{j=1}^{N_t} W^e(t, I_k, I_j)$ is of order $(N_t^a)^2$. Informally, this means that when the population size increases, events with interaction are more costly to simulate. Furthermore, the numerous computations of the interaction kernel W^e can also be quite costly. The randomized Algorithm [algo::rzndomized], detailed in Section 2.3, allows us to overcome these limitations.

2.4.1 Events intensity bounds

The simulation algorithms implemented in IBMPopSim are based on an acceptance/rejection procedure, which requires to specify bounds for the various events intensities $\Lambda_t^e(Z_t)$. These bounds are defined differently depending on the expression of the intensity.

Assumption 2.1. For all events $e \in \mathcal{P}$ with Poisson intensity Equation 2, the intensity is assumed to be bounded on $[0, T]$:

$$\forall t \in [0, T], \quad \Lambda_t^e(Z_t) = \mu^e(t) \leq \bar{\mu}^e.$$

When $e \in \mathcal{E} \cup \mathcal{E}_W$, $\Lambda_t^e(Z_t) = \sum_{k=1}^{N_t} \lambda_t^e(I_k, Z_t)$, assuming that $\Lambda_t^e(Z_t)$ is uniformly bounded is too restrictive since the event intensity depends on the population size. In this case, the assumption is made on the individual intensity λ^e or on the interaction function W^e , depending on the situation.

Assumption 2.2. For all event types $e \in \mathcal{E}$, the associated individual event intensity λ^e with no interactions, i.e. λ^e verifies Equation 3, is assumed to be uniformly bounded:

$$\lambda^e(t, I) \leq \bar{\lambda}^e, \quad \forall t \in [0, T], \quad I \in \mathcal{I}.$$

In particular,

$$\forall t \in [0, T], \quad \Lambda_t^e(Z_t) = \sum_{k=1}^{N_t} \lambda^e(t, I_k) \leq \bar{\lambda}^e N_t. \quad (8)$$

Assumption 2.3. For all event types $e \in \mathcal{E}_W$, the associated interaction function W^e is assumed to be uniformly bounded:

$$W^e(t, I, J) \leq \bar{W}^e, \quad \forall t \in [0, T], \quad I, J \in \mathcal{I}.$$

In particular, $\forall t \in [0, T]$,

$$\lambda_t^e(I, Z_t) = \sum_{j=1}^{N_t} W^e(t, I, I_j) \leq \bar{W}^e N_t, \quad \text{and} \quad \Lambda_t^e(Z_t) \leq \bar{W}^e (N_t)^2.$$

Assumption 2.1, Assumption 2.2 and Assumption 2.3 yield that events in the population occur with the global event intensity $\Lambda_t(Z_t)$, given in Equation 7, which is dominated by a polynomial function in the population size:

$$\Lambda_t(Z_t) \leq \bar{\Lambda}(N_t), \quad \text{with} \quad \bar{\Lambda}(n) = \sum_{e \in \mathcal{P}} \bar{\mu}^e + \sum_{e \in \mathcal{E}} \bar{\lambda}^e n + \sum_{e \in \mathcal{E}_W} \bar{W}^e n^2. \quad (9)$$

This bound is linear in the population size if there are no interactions, and quadratic if there at least is an event including interactions. This assumption is the key to the algorithms implemented in IBMPopSim. Before presenting the simulation algorithm, we close this section with a rigorous definition of an IBM, based on the pathwise representation of its dynamics a Stochastic Differential Equation (SDE) driven by Poisson random measures.

2.5 Pathwise representation

Since the seminal paper of (Fournier and Méléard 2004), it has been shown in many examples that a stochastic IBM dynamics can be defined rigorously as the unique solution of an SDE driven by Poisson measures, under reasonable non explosion conditions. In the following, we introduce a unified framework for the pathwise representation of the class of stochastic IBMs introduced above. Some recalls on Poisson random measures are presented in the Appendix ?@sec-recall-poisson, and for more details on these representations on particular examples, we refer to the abundant literature on the subject.

In the following we consider an individual-based stochastic population $(Z_t)_{t \in [0, T]}$, keeping the notations introduced in Section 2.3 and Section 2.4 for the events and their intensities. In particular, the set of events types that define the population evolution is denoted by $\mathcal{P} \cup \mathcal{E} \cup \mathcal{E}_W \subset E$, with \mathcal{P} the set of events types with Poisson intensity verifying Assumption 2.1, \mathcal{E} the set of events types with individual intensity and no interaction, verifying Assumption 2.2 and finally \mathcal{E}_W the set of event types with interactions, verifying Assumption 2.3.

2.6 Thinning of Poisson measure

References

- Bansaye, Vincent, and Sylvie Méléard. 2015. *Stochastic Models for Structured Populations*. Springer International Publishing.
- Barrieu, Pauline, Harry Bensusan, Nicole El Karoui, Caroline Hillairet, Stéphane Loisel, Claudia Ravanelli, and Yahia Salhi. 2012. “Understanding, Modelling and Managing Longevity Risk: Key Issues and Main Challenges.” *Scandinavian Actuarial Journal* 2012 (3): 203–31.
- Bensusan, Harry. 2010. “Interest rate and longevity risk: dynamic model and applications to derivative products and life insurance.” Theses, Ecole Polytechnique X.
- Billiard, Sylvain, Pierre Collet, Régis Ferrière, Sylvie Méléard, and Viet Chi Tran. 2016. “The Effect of Competition and Horizontal Trait Inheritance on Invasion, Fixation, and Polymorphism.” *Journal of Theoretical Biology* 411: 48–58.
- Boumezoued, Alexandre. 2016. “Micro-macro analysis of heterogenous age-structured populations dynamics. Application to self-exciting processes and demography.” Theses, Université Pierre et Marie Curie.
- Brémaud, Pierre. 1981. *Point Processes and Queues: Martingale Dynamics*. Vol. 66. Springer.
- Calvez, Vincent, Susely Figueroa Iglesias, Hélène Hivert, Sylvie Méléard, Anna Melnykova, and Samuel Nordmann. 2020. “Horizontal Gene Transfer: Numerical Comparison Between Stochastic and Deterministic Approaches.” *ESAIM: Proceedings and Surveys* 67: 135–60.
- Champagnat, Nicolas, Régis Ferrière, and Sylvie Méléard. 2006. “Unifying Evolutionary Dynamics: From Individual Stochastic Processes to Macroscopic Models.” *Theoretical Population Biology* 69 (3): 297–321.
- Collet, Pierre, Sylvie Méléard, and Johan AJ Metz. 2013. “A Rigorous Model Study of the Adaptive Dynamics of Mendelian Diploids.” *Journal of Mathematical Biology* 67: 569–607.
- Costa, Manon, Céline Hauzy, Nicolas Loeuille, and Sylvie Méléard. 2016. “Stochastic Eco-Evolutionary Model of a Prey-Predator Community.” *Journal of Mathematical Biology* 72: 573–622.
- Eddelbuettel, Dirk, and Romain Francois. 2011. “Rcpp: Seamless r and c++ Integration.” *Journal of Statistical Software* 40 (8): 1–18. <https://doi.org/10.18637/jss.v040.i08>.
- El Karoui, Nicole, Kaouther Hadji, and Sarah Kaakai. 2021. “Simulating Long-Term Impacts of Mortality Shocks: Learning from the Cholera Pandemic.” *arXiv Preprint arXiv:2111.08338*.
- Ferrière, Régis, and Viet Chi Tran. 2009. “Stochastic and Deterministic Models for Age-Structured Populations with Genetically Variable Traits.” In, 27:289–310. *ESAIM Proc. EDP Sci., Les Ulis*.
- Fournier, Nicolas, and Sylvie Méléard. 2004. “A Microscopic Probabilistic Description of a Locally Regulated Population and Macroscopic Approximations.” *Ann. Appl. Probab.* 14 (4): 1880–1919.
- Hyndman, Rob, Heather Booth Booth, Leonie Tickle Tickle, John Maindonald, Simon Wood Wood, and R Core Team. 2023. *demography: Forecasting Mortality, Fertility, Migration and Population Data*. <https://cran.r-project.org/package=demography>.
- Kaakai, Sarah, and Nicole El Karoui. 2023. “Birth Death Swap Population in Random Environment and Aggregation with Two Timescales.” *Stochastic Processes and Their Applications* 162: 218–48. <https://doi.org/https://doi.org/10.1016/j.spa.2023.04.017>.
- Lavallée, François, Charline Smadi, Isabelle Alvarez, Björn Reineking, François-Marie Martin, Fanny

- Dommanget, and Sophie Martin. 2019. “A Stochastic Individual-Based Model for the Growth of a Stand of Japanese Knotweed Including Mowing as a Management Technique.” *Ecological Modelling* 413: 108828.
- Méléard, Sylvie, Michael Rera, and Tristan Roget. 2019. “A Birth–Death Model of Ageing: From Individual-Based Dynamics to Evolutive Differential Inclusions.” *Journal of Mathematical Biology* 79: 901–39.
- Roget, Tristan, Pierre Jolivet, Sylvie Méléard, and Michael Rera. 2022. “Positive Selection of Senescence Through Increased Evolvability: Ageing Is Not a by-Product of Evolution.” *bioRxiv*, 2022–03.
- Tran, Viet Chi. 2008. “Large Population Limit and Time Behaviour of a Stochastic Particle Model Describing an Age-Structured Population.” *ESAIM: Probability and Statistics* 12: 345–86. <https://doi.org/10.1051/ps:2007052>.
- Villegas, Andres, Pietro Millossovich, and Vladimir Kaishev Hyndman. 2018. *StMoMo: Stochastic Mortality Modelling*. <https://cran.r-project.org/package=StMoMo>.
- Zinn, Sabine. 2014. “The MicSim package of R: an entry-level toolkit for continuous-time microsimulation.” *International Journal of Microsimulation* 7 (3): 3–32.
- Zinn, Sabine, Jutta Gampe, Jan Himmelsbach, and Adelinde M Uhrmacher. 2009. “MIC-CORE: A Tool for Microsimulation.” In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 992–1002. IEEE.

Session information

`sessionInfo()`