`mpmsim`: An R package for simulating matrix population models

Owen R. Jones[1*]

[1] Department of Biology, University of Southern Denmark, Campusvej 55, Odense M, Denmark

* Corresponding author. Email: jones@biology.sdu.dk

Abstract

1. Matrix population models (MPMs) are widely used in ecology and evolution to explore population dynamics, including assessing management impacts and extinction risk. In comparative studies, MPMs can be used to test life history theory or investigate macro-evolutionary patterns in demographic traits.
2. Simulated MPMs can help researchers explore the effects of life cycle structure, vital rate trajectories, and uncertainty in transition rates due to sampling error. They are also valuable teaching tools.
3. The **mpmsim** R package enables users to simulate random or semi-random Lefkovitch and Leslie MPMs based on life history archetypes or mortality and fecundity patterns. It also allows the exploration of sampling error effects and uses parametric bootstrapping to calculate confidence intervals for matrix-derived estimates.
4. **mpmsim** provides a convenient toolset for addressing questions about MPMs and life history, with full documentation and user-friendly vignettes.

Keywords: simulation, sampling error, bias, teaching tools, mortality trajectory, fertility trajectory, Leslie matrix, Lefkovitch matrix, life history archetypes

Introduction

Matrix population models (MPMs) are a versatile tool in population biology and evolution (Caswell, 2001), first introduced by Leslie (1945) to study age-structured population dynamics. Lefkovitch (1965) expanded the approach to stage-classified life histories, followed by developments like stochastic (Cohen, Christensen & Goodyear, 1969) and density-dependent MPMs (Pennycuick, 1969), broadening their applications. Analytical methods such as elasticity and perturbation analysis, life table response experiments (LTRE), and Markov chain methods (Caswell, 2001) have further enhanced their utility.

MPMs describe a population's demography at a specific time and place by modelling individuals categorised by (st)age over a discrete projection interval (Caswell, 2001). At the core is the projection matrix (**A**), representing transitions between stages through survival, growth, and reproduction, which can be split into submatrices **U** (growth/survival), **F** (sexual reproduction), and **C** (clonal reproduction), such that $\mathbf{A} = \mathbf{U} + \mathbf{F} + \mathbf{C}$. MPMs provide biologically meaningful outputs to estimate population growth rates, extinction risk, responses to vital rate perturbations, transient dynamics, effective population size, and life history traits. Consequently, MPMs have been pivotal in advancing population biology and life history theory (Caswell, 2001; Crone et al., 2011). Supporting this, the COMPADRE and COMADRE databases (Salguero-Gomez et al., 2015, 2016) provide >12,000 MPMs for >1,100 species, ranging from annual plants to whales, which address topics from population management to evolutionary theory.

Several R packages support MPM analysis (e.g., `popbio` (Stubben & Milligan, 2007), `popdemo` (Stott et al., 2012), `Rage` (Jones et al., 2022), `exactLTRE` (Hernández et al., 2023)), but none provide broad scope for simulating MPMs with specific characteristics. This limits researchers' ability to explore population dynamics beyond empirical data constraints (Römer et al., 2024). To address this, I introduce `mpmsim`, an R package designed to simulate MPMs with defined characteristics, enabling

users to explore life history and population dynamics. The core functions are `make_leslie_mpm` and its wrapper `rand_leslie_set` for Leslie matrices, and `rand_lefko_mpm` and `rand_lefko_set` for Lefkovitch matrices. The functions `compute_ci` and `compute_ci_U` calculate confidence intervals via parametric bootstrapping, and `add_mpm_error` allows users to simulate MPMs with sampling error. Together, these functions enable simulations of diverse life histories and the assessment of the impact of sampling error on inferences.

Illustrating use of `mpmsim`

To demonstrate `mpmsim`'s versatility, I provide three examples. First, I show how to generate Leslie matrices based on mortality and fertility trajectories. Second, I generate Lefkovitch models using defined life cycle characteristics. Finally, I calculate confidence intervals for MPM-derived estimates. These are brief demonstrations rather than exhaustive analyses; more extensive vignettes are available on the package website (https://jonesor.github.io/mpmsim/).

Example 1: Generating Leslie matrices

Leslie MPMs model age-classified populations, with survival probabilities (p_x) in the subdiagonal representing survival probabilities from age x to $x+1$, and fecundity (f_x) in the first row representing reproduction. Matrix **A** combines survival/growth (**U**) and sexual reproduction (**F**), such that **A** = **U** + **F**. For clonal organisms, a third submatrix **C** can be added (**A** = **U** + **F** + **C**).

The main function for generating Leslie MPMs in the package is `rand_leslie_set`, which creates MPMs based on randomly drawn parameters from specified mortality and fertility models. Users define the model types (`mortality_model`, `fertility_model`), parameter distributions (`mortality_params`, `fertility_params`, `fertility_maturity_params`), and the number of MPMs (`n_models`). The function outputs MPMs either as a `list` or a `compadreDB`

object (Jones et al., 2021), depending on the output argument. The underlying functions `model_mortality` and `model_fertility` calculate age-specific survival (p_x) and fertility (f_x) using standard functional forms.

The available mortality models include Siler, Exponential, Gompertz, Gompertz-Makeham, Weibull, and Weibull-Makeham (Table 1). The `model_mortality` function calculates survival probabilities for each age based on age-specific hazard rates. It first calculates hazard rates (h_x) from the chosen mortality model using the model argument and a vector of parameters (`params`). The cumulative hazard (H_x) is then computed by integrating h_x up to each age, giving total mortality risk. Survivorship (l_x) is determined as $exp(-H_x)$, and the age-specific survival probability (p_x) is the ratio of survivorship values at $x+1$ and x. The function outputs a life table as a data frame that extends by default until l_x drops below 0.01.

The `model_fertility` function calculates age-specific fertility (f_x), the average number of offspring produced at age x, using canonical models such as logistic, step-function, von Bertalanffy, Hadwiger, and Normal (Table 1). Key arguments include the model type (`model`), parameters (`params`), and a vector of ages (`age`), and the output is a vector of fertility values corresponding to input ages.

Code Box 1 shows how to create 500 Leslie MPMs with Gompertz-Makeham mortality and step-function fertility, with ages at maturity varying from one to four, and parameter values drawn from uniform distributions. The code runs in 4.01 seconds (SD = 0.92; 100 runs) on a MacBook with macOS Sonoma 14.6.1, an Intel Core i5 2.40 GHz CPU, 16 GB of RAM, and R version 4.1.1.

Code box 1: Leslie matrix models

```
# Load package
library(mpmsim)

# Define mortality model parameters
```

```r
# min/max values in Gompertz-Makeham model
mortParams <- data.frame(
  minVal = c(0, 0.01, 0.1),
  maxVal = c(0.05, 0.15, 0.2)
)
# Define fertility model parameters
# min/max values in step model
fertParams <- data.frame(
  minVal = 2,
  maxVal = 10
)
# Define age-at-maturity
# min/max values
maturityParam <- c(1, 4)

# Produce 500 MPMs
# Gompertz-Makeham mortality model and step function fertility
# Parameters drawn from uniform distribution
# Output compadreDB
myMatrices <- rand_leslie_set(
  n_models = 500,
  mortality_model = "GompertzMakeham",
  fertility_model = "step",
  mortality_params = mortParams,
  fertility_params = fertParams,
  fertility_maturity_params = maturityParam,
  dist_type = "uniform",
  output = "Type1"
)
```

Example 2: Generating Lefkovitch matrices

Lefkovitch MPMs are stage-based, making them ideal when age data is unavailable or less relevant, such as in life cycles governed by developmental stages (e.g., juvenile, adult). A key advantage is their adaptability to various life cycles, including transitions like retrogression or dormancy. This flexibility is essential for studying species with non-age-based life cycles. While these models accommodate diverse life cycles, some rules apply: survival probabilities cannot exceed 1, and reproduction cannot be negative. Thus, transition probabilities in **U** range from 0 to 1, with column sums constrained to ≤ 1, while reproduction in **F** has a lower limit of zero.

In `mpmsim`, the function `rand_lefko_set` generates sets of Lefkovitch MPMs and is a wrapper for `rand_lefko_mpm`, which creates individual MPMs. These functions model the **U** matrix by drawing values from a random Dirichlet distribution, ensuring survival probabilities for each stage are ≤ 1. Users can generate various life cycle structures using the `archetype` argument, based on Takada *et al.*'s (2018) four archetypes. In Archetype 1, individuals can move freely between stages, either progressing or retrogressing, with no constraints on the transition rate. Archetype 2 adds the assumption that survival improves with stage progression. Archetype 3 allows only forward progression, mimicking species with slow development. Archetype 4, similar to Archetype 3, includes improved survival with stage progression but without retrogression.

In Takada *et al.*'s models, fecundity was placed in the top-right of the matrix, restricting reproduction to the final life cycle stage. In `mpmsim`, this constraint is relaxed through the fecundity argument, which offers four options: (1) a single value representing reproduction in the most developed stage, (2) a vector matching the number of stages to assign stage-specific fecundity across the top row, (3) a matrix defining fecundity for each element, or (4) a list of two matrices setting upper and lower fecundity limits, with values drawn from a uniform distribution. This flexibility captures various reproductive strategies, accommodating diverse life histories.

In addition to generating matrices based on selected archetypes, outputs can be fine-tuned using the `constraint` argument, allowing users to set limits based on any metric derived from the **A** matrix, such as population growth rates within a defined range, to ensure viable life cycles. This enables tailored simulations for specific ecological or evolutionary scenarios.

Code Box 2 generates 500 Lefkovitch models in a `compadreDB` object for Archetype 4, constrained to have a λ between 0.9 and 1.1. Fecundity is set to 0 for the first two stages, and 8 and 14 for the last two. The code runs in 1.22 seconds (SD = 0.12; 100 runs).

Code box 2: Lefkovitch matrix models

```
# Load packages
library(popdemo)
library(mpmsim)

# Define constraints
# Lambda between 0.9 and 1.1
constrain_df <- data.frame(
  fun = "eigs", arg = "lambda", lower = 0.9, upper = 1.1
)

# Produce 500 MPMs, 3 stages, Archetype 4
# Set fecundity of 8 for stage 3 and 14 for stage 4.
# Output as compadreDB object
myMatrices <- rand_lefko_set(
  n = 500, n_stages = 4, fecundity = c(0,0,8,14),
  constraint = constrain_df, archetype = 4, output = "Type1")
```

Example 3: Calculating confidence intervals

MPMs are parameterised in various ways, often by estimating transition rates from repeated surveys of stage classes. Typically, only a population sample is used, and sample sizes may vary. For instance, juveniles might be common, resulting in larger sample sizes for estimating juvenile survival, while adults may be rarer, leading to smaller samples. As a result, sampling error varies across the matrix and between years in multi-year studies, potentially influencing life history or population dynamics analyses. I will illustrate how to explore these effects on inferences.

I start with the matrix model, **A**, the sum of the **U** and **F** submatrices. For example, $\mathbf{A} = \begin{bmatrix} 0.1 & 5.0 \\ 0.2 & 0.4 \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0.4 \end{bmatrix}$ and $\mathbf{F} = \begin{bmatrix} 0 & 5.0 \\ 0 & 0 \end{bmatrix}$. Transition rates are assumed to follow specific statistical distributions: sexual reproduction follows a Poisson process, while growth/survival transitions follows a binomial process. The well-known properties of these statistical processes enable the estimation of expected distributions for each matrix element, based on their average values and sample sizes. In `mpmsim`, the `compute_ci` function uses parametric resampling, simulating multiple models by resampling matrix elements from their respective distributions to generate a distribution of derived trait

estimates. For tractability, this method assumes no covariance between rates, though trade-offs among elements may introduce nuances. **Code Box 3** shows how to calculate confidence intervals for generation time and λ, with sample sizes of 15 for adult fecundity and 40 for the survival/growth transitions.

Code box 3 – Confidence Intervals

```
# Load packages
library(popbio)
library(mpmsim)

# Define U matrix
matU <- matrix(c(
  0.1, 0.0,
  0.2, 0.4
), byrow = TRUE, nrow = 2)

# Define F matrix
matF <- matrix(c(
  0.0, 5.0,
  0.0, 0.0
), byrow = TRUE, nrow = 2)

# Combine matrices to create A matrix
matA <- matU + matF

# Define sample sizes for F
mat_F_ss <- matrix(c(
  0.0, 15,
  0.0, 0.0
), byrow = TRUE, nrow = 2)

# Define the sample sizes for U
mat_U_ss <- matrix(c(
  40, 40,
  40, 40
), byrow = TRUE, nrow = 2)

# Combine sample sizes into list
sampleSizes <- list(mat_U_ss = mat_U_ss, mat_F_ss = mat_F_ss)

# Calculate lambda
lambda(matA)
#> [1] 1.261187

# Calculate CI for lambda
compute_ci(
  mat_U = matU, mat_F = matF, sample_size = sampleSizes,
  FUN = lambda
```

```
)
#>      2.5%     97.5%
#> 0.9033941 1.6022154

# Calculate generation time
generation.time(matA)
#> [1] 2.65536

# Calculate CI for generation time
compute_ci(
  mat_U = matU, mat_F = matF, sample_size = sampleSizes,
  FUN = popbio::generation.time
)
#>     2.5%    97.5%
#> 2.371819 3.106800
```

Conclusion

These examples highlight `mpmsim` as a valuable tool for simulating MPMs, ideal for research on population dynamics and life history evolution, as well as for teaching. Detailed examples are available in the GitHub vignettes (http://www.github.org/jonesor/mpmsim). The package can be installed via CRAN with `install.packages("mpmsim")` or from GitHub for the development version. Overall, `mpmsim` provides user-friendly tools for MPM simulation, making it a powerful resource for population ecology and evolutionary biology research.

Acknowledgements

I thank Rob Salguero-Gómez, Samuel Gascoigne, Emily Simmonds, Patrick Barks and Iain Stott for discussion on the functionality of the package at various stages of development.

References

Cochran, M. E., & Ellner, S. (1992). Simple methods for calculating age-based life history parameters for stage-structured populations. Ecological Monographs, 62(3), 345–364.

Cohen, J. E., Christensen, S. W., & Goodyear, C. P. (1983). An age-structured fish population model with random survival of eggs: calculation of asymptotic growth rates and application to Potomac River striped river bass. Journal of Fisheries and Aquatic Science, 40, 2170–2183.

Colchero, F., Jones, O. R., & Rebke, M. (2012). BaSTA: an R package for Bayesian estimation of age-specific survival from incomplete mark--recapture/recovery data with covariates. Methods in Ecology and Evolution, 3(3), 466–470.

Colchero, F., Jones, O. R., Conde, D. A., et al. (2019). The diversity of population responses to environmental change. Ecology Letters, 22(2), 342–353.

Cox, D.R. & Oakes, D. (1984) Analysis of Survival Data. Chapman and Hall, London, UK.

Crone, E. E., Menges, E. S., Ellis, M. M., et al. (2011). How do plant ecologists use matrix population models? *Ecology Letters*, *14*(1), 1–8.

Franco, M., & Silvertown, J. (1996). Life History Variation in Plants: An Exploration of the Fast-Slow Continuum Hypothesis. *Philosophical Transactions of the Royal Society of London*, 351(1345), 1341–1348.

Gompertz, B. (1825). XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In a letter to Francis Baily, Esq. F. R. S. &c. Philosophical Transactions of the Royal Society of London, 115(0), 513–583.

Hadwiger, H. (1940). Eine analytische Reproduktionsfunktion für biologische Gesamtheiten. Scandinavian actuarial journal, 1940(3–4), 101–113.

Hernández, C. M., Ellner, S. P., Adler, P. B., Hooker, G., & Snyder, R. E. (2023). An exact version of Life Table Response Experiment analysis, and the R package exactLTRE. Methods in Ecology & Evolution, 14(3), 939–951.

Jones, O. R., Barks, P., Stott, I., et al. (2022). Rcompadre and Rage—Two R packages to facilitate the use of the COMPADRE and COMADRE databases and calculation of life-history traits from matrix population models. Methods in Ecology & Evolution, 13(4), 770–781.

Lefkovitch, L. P. (1965). The study of population growth in organisms grouped by stages. Biometrics, 21(1), 1-18

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

Makeham, W. M. (1867). On the law of mortality. *Journal of the Institute of Actuaries (1866)*, *13*(06), 325–358.

Morris, W. F., & Doak, D. F. (2004). Buffering of life histories against environmental stochasticity: accounting for a spurious correlation between the variabilities of vital rates and their contributions to fitness. American Naturalist, 163(4), 579–590.

Pennycuick, L. (1969) A computer model of the Oxford great tit population. Journal of Theoretical Biology, 22, 381-400.

Peristera, P. & Kostaki, A. (2007) Modeling fertility in modern populations. Demographic Research. 16. Article 6, 141-194.

305 Pfister, C. A. (1998). Patterns of variance in stage-structured populations: evolutionary predictions and
306 ecological implications. Proceedings of the National Academy of Sciences of the United States of
307 America, 95(1), 213–218.

308 Pinder III, J.E., Wiener, J.G. & Smith, M.H. (1978) The Weibull distribution: a method of summarizing
309 survivorship data. Ecology, 59, 175–179.

310 Pletcher, S. (1999) Model fitting and hypothesis testing for age-specific mortality data. Journal of
311 Evolutionary Biology, 12, 430–439.

312 Ricklefs, R. E., & Scheuerlein, A. (2002). Biological implications of the Weibull and Gompertz models
313 of aging. Journal of Gerontology, 57(2), B69-76.

314 Römer, G., Dahlgren, J. P., Salguero-Gómez, R., Stott, I. M., & Jones, O. R. (2024). Plant demographic
315 knowledge is biased towards short-term studies of temperate-region herbaceous perennials. Oikos,
316 Article e10250

317 Sæther, B.-E., Coulson, T., Grøtan, V., et al. (2013). How Life History Influences Population Dynamics
318 in Fluctuating Environments. American Naturalist, 182(6), 743–759.

319 Salguero-Gómez, R., Jones, O. R., Archer, C. R., et al. (2015) The COMPADRE Plant Matrix
320 Database: an open online repository for plant demography. Journal of Ecology, 103, 202-218.

321 Salguero-Gómez, R., Jones, O. R., Archer, C. R., et al. (2016). COMADRE: a global database of animal
322 demography. Journal of Animal Ecology, 85(2), 371–384.

323 Siler, W. (1979) A competing-risk model for animal mortality. Ecology, 60, 750–757.

324 Stubben, C., Milligan, B., & Others. (2007). Estimating and analyzing demographic models using the
325 popbio package in R. Journal of Statistical Software, 22(11), 1–23.

326 Takada, T., Kawai, Y., & Salguero-Gómez, R. (2018). A cautionary note on elasticity analyses in a
327 ternary plot using randomly generated population matrices. Population Ecology, 60(1), 37–47.

328 Tuljapurkar, S. (1989). An uncertain life: demography in random environments. Theoretical Population
329 Biology, 35(3), 227–294.

330 Vaupel, J., Manton, K. & Stallard, E. (1979) The impact of heterogeneity in individual frailty on the
331 dynamics of mortality. Demography, 16, 439–454.

332 von Bertalanffy, L. (1957). Quantitative laws in metabolism and growth. The Quarterly Review of
333 Biology, 32(3), 217–231.Caswell, H. (2001). Matrix Population Models. Sinauer Associates.

334

Tables

Table 1: Mathematical expressions for mortality and fertility models in `mpmsim`. In mortality models, h_x is the hazard function at age x, with parameters a_0, a_1, b_0, b_1, and c reflecting factors like aging rate and baseline risk. In fertility models, f_x is the fertility rate at age x, with parameters A, k, x_m, m, x_0, μ, σ, a, b, and c determining peak fertility timing, distribution, and overall levels. See `?mortality_model` and `?fertility_model`.

Mortality models		
Gompertz	$h_x = b_0 e^{b_1 x}$	Gompertz (1825), Pletcher (1999), Ricklefs & Scheuerlein (2002), Colchero et al. (2012)
Gompertz-Makeham	$h_x = b_0 e^{b_1 x} + c$	Pletcher (1999), Colchero et al. (2012)
Exponential	$h_x = c$	Cox & Oakes (1984), Colchero et al. (2012)
Siler	$h_x = a_0 e^{-a_1 x} + c + b_0 e^{b_1 x}$	Siler (1979), Colchero et al. (2012)
Weibull	$h_x = b_0 b_1 (b_1 x)^{(b_0 - 1)}$	Pinder *et al.* (1978), Ricklefs & Scheuerlein

		(2002), Colchero et al. (2012)
Weibull-Makeham	$h_x = b_0 b_1 (b_1 x)^{(b_0 - 1)} + c$	Colchero et al. (2012)
Fertility models		
Logistic	$f_x = A / \left(1 + \exp\left(-k(x - x_m)\right)\right)$	
Step	$f_x = \begin{cases} A, x \geq m \\ A, x < m \end{cases}$	
Normal	$f_x = A \times \exp\left(-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right)$	
Hadwiger	$f_x = \frac{ab}{c}\left(\frac{c}{x}\right)^{\frac{3}{2}} \exp\left\{-b^2\left(\frac{c}{x} + \frac{x}{c} - 2\right)\right\}$	Hadwiger (1940) Peristera & Kostaki (2007)
von Bertalanffy	$f_x = A\left(1 - exp\left(-k(x - x_0)\right)\right)$	von Bertalanffy (1938)

342

343