

Hyperstate matrix models: extending demographic state spaces to higher dimensions

Gregory Roth* and Hal Caswell

Institute for Biodiversity and Ecosystem Dynamics, University of Amsterdam, 1090 GE Amsterdam, The Netherlands

Summary

1. Demographic models describe population dynamics in terms of the movement of individuals among states (e.g. size, age, developmental stage, parity, frailty, physiological condition). Matrix population models originally classified individuals by a single characteristic. This was enlarged to two characteristics in vec-permutation models and the closely related megamatrix models. However, it has been recognised that the interplay of more than two characteristics can affect the population dynamics.
2. Here, we present a framework, called *hyperstate matrix model*, in which individuals may be classified by any number of characteristics, using the generalisation of the vec-permutation approach to hypermatrices. These models are constructed from a simple block-diagonal matrix formulation of the movement of individuals among each dimension of the *i*-state.
3. This framework provides a step-by-step construction and makes available the usual demographic analysis developed for classical matrix models. In particular, we derive a general formula for the sensitivity of any output of the hyperstate matrix model, to any vector of parameters. In spite of the technicalities underlying these models, implementation is straightforward and we provide the MATLAB code to carry it out.
4. We apply this approach to a three-dimensional example in which individuals are classified by developmental stage, age and heterogeneity classes. The analysis of this model provides insights into how the heritability of the heterogeneity classes affects the long-term growth rate of the population.
5. As the questions in conservation biology become more sophisticated and data on threatened species become more detailed, multiple dimensions in demographic models will become increasingly important. Hyperstate matrix methods will make such analyses possible and directly applicable to conservation and population management.

Key-words: matrix models, modelling, population ecology, sensitivity analysis

Introduction

Demographic models describe population dynamics in terms of the states of the individuals making up the population (*i*-states in the terminology of Metz & Diekmann (1986)). The state of the population (the *p*-state) is a distribution over the set of possible *i*-states. Thus, age as an *i*-state leads to models of the age distribution, size as an *i*-state leads to models of the size distribution and so on. The model of the population is formulated in terms of the movement of individuals among *i*-states by survival, development and reproduction.

Multi-state models are those in which individuals are classified by two different characteristics (e.g. age and stage). Each of the characteristic is described by a finite set of classes; for example, age is described by a set of age classes. If the sets of classes are denoted by \mathcal{S}_1 and \mathcal{S}_2 , then the multistate model operates on the two-dimensional *i*-state space given by $\mathcal{S}_1 \times \mathcal{S}_2$. Multistate models originated in the work of Rogers (1967, 1975) on multiregional (or metapopulation) models and

have been applied by both ecologists and human demographers (e.g. Land & Rogers 1982; Andersen *et al.* 1993; Lebreton 1996; Willekens 2014) to many other multiple classifications.

A systematic approach to the construction and analysis of multistate models, using the vec-permutation matrix, was introduced by Hunter & Caswell (2005). The approach to multistate models has been applied to models in which individuals are classified by stage and location (Burns & Gear 2008; Tinker, Doak & Estes 2008; Ozgul *et al.* 2009; Ezard *et al.* 2010; Goldberg *et al.* 2010; Strasser *et al.* 2012; Flockhart *et al.* 2015; Warden *et al.* 2015), age and developmental stage (Caswell 2012; Caswell & Salguero-Gómez 2013), age and frailty (Caswell 2014b), age and parity (Caswell 2014a), stage and infection status (Klepac & Caswell 2011; Metcalf *et al.* 2012) and combinations of stage and time or environmental state (Caswell, 2006, 2011; Hernandez-Suarez, Rabinovich & Hernandez 2012; Perez-Heydrich, Oli & Brown 2012). One case of the vec-permutation model yields the widely used 'megamatrix' model (e.g. Pascarella & Horvitz 1998; Tuljapurkar, Horvitz & Pascarella 2003; Dostal 2007; Yule, Miller & Rudgers 2013).

*Correspondence author. E-mail: greg.roth51283@gmail.com

An obvious extension of the vec-permutation approach is to models including more than two individual characteristics. We refer to these as *hyperstate matrix models*. Whereas a multi-state matrix population model describes the state of the population by a matrix \mathcal{N} , a hyperstate matrix model describes the state of the population by a hypermatrix with more than two dimensions. Among the possible applications are models for migration including age, size and spatial location; models for reproductive performance including age, parity and duration; models for mortality including age, stage and frailty; and stochastic models including size, frailty and environmental state.

Section ‘The vec-permutation approach to multistate models’ describes the vec-permutation approach to multi-state matrix models and introduces the main necessary ingredients that will be generalised later on. In Section ‘The vec-permutation approach to hyperstate matrix models’, hyperstate matrix models are introduced and the vec-permutation approach is generalised step by step. The detailed mathematical justification for these generalisations is in the Appendix S2, Supporting Information; in the main text, we present only the essential tools for constructing a hyperstate matrix model. The only technical concept that we introduce in the main text is the *linear indexing* of a hypermatrix, which numerates its entries (see Section ‘From arrays to vectors: linear indexing and the vec operator’). This indexing allows us to transform, in a very simple way, a hypermatrix into a vector. In Section ‘Sensitivity analysis’, we provide a formula that permits the sensitivity analysis of a generic-dependent variable of the model. In Section ‘An example: stage, age, and frailty’, we apply the vec-permutation approach to a (hypothetical) three-dimensional hyperstate matrix model in which individuals are classified by developmental stage, chronological age and heterogeneity classes. The sensitivity analysis of the long-term population growth rate to the demographic parameters is presented in Section ‘Effect of heterogeneous frailty on population growth rate’.

Although the formalism of the vec-permutation is quite technical and abstract, it is easily implemented in MATLAB. Throughout the paper, we dedicate special boxes – called *Implementation* – to the implementation in MATLAB of the different key mathematical objects. Moreover, the Appendix S1 provides a step-by-step procedure to implement a generic hyperstate matrix model and the implementation in MATLAB of the example presented in Section ‘An example: stage, age, and frailty’. Because it is designed throughout for matrix manipulations, MATLAB is the most appropriate language to implement hyperstate matrix models. Of course, any calculation can be implemented in any general-purpose computer language, including hyperstate matrix models; this can be done by translating our code or by following the step-by-step procedure.

The vec-permutation approach to multistate models

Multistate models classify individuals by two characteristics. For convenience, we will call these *stages* and *groups*, with

$$\text{stage} \in \mathcal{S}_1 = \{1, \dots, s\} \quad \text{eqn 1}$$

$$\text{group} \in \mathcal{S}_2 = \{1, \dots, g\} \quad \text{eqn 2}$$

Within group i , the movement of individuals among stages is described by a $s \times s$ matrix \mathbf{B}_i . Within stage j , the movement of individuals among groups is described by a $g \times g$ matrix \mathbf{C}_j . The p -state is a density function over \mathcal{S} , which can be written as a two-dimensional array \mathcal{N} ,

$$\mathcal{N} = \begin{pmatrix} n_{11} & \cdots & n_{1g} \\ \vdots & \vdots & \vdots \\ n_{s1} & \cdots & n_{sg} \end{pmatrix} \quad \text{eqn 3}$$

where n_{ij} is the number of individuals in stage i and group j .

To project the density function \mathcal{N} using matrix multiplication, it must first be transformed into a population vector. The vector can be created in two ways,

$$\text{vec } \mathcal{N} = \begin{pmatrix} n_{11} \\ \vdots \\ n_{s1} \\ \vdots \\ n_{1g} \\ \vdots \\ n_{sg} \end{pmatrix} \quad \text{or} \quad \text{vec } (\mathcal{N}^\top) = \begin{pmatrix} n_{11} \\ \vdots \\ n_{1g} \\ \vdots \\ n_{s1} \\ \vdots \\ n_{sg} \end{pmatrix} \quad \text{eqn 4}$$

In the vector $\text{vec } \mathcal{N}$, stages are arranged within groups; in $\text{vec } (\mathcal{N}^\top)$, groups are arranged within stages. The two vectors are related by the vec-permutation matrix $\mathbf{K}_{s,g}$,

$$\text{vec}(\mathcal{N}^\top) = \mathbf{K}_{s,g} \text{vec } \mathcal{N} \quad \text{eqn 5}$$

where the vec-permutation matrix is indexed by the dimensions of the matrix \mathcal{N} to which it is applied (Magnus & Neudecker 1979; Henderson & Searle 1981). We choose to use the vector $\tilde{\mathbf{n}} = \text{vec } \mathcal{N}$ as the state variable for the multistate model.

The model is constructed from the matrices \mathbf{B}_i , $i = 1, \dots, g$ and the matrices \mathbf{C}_j , $j = 1, \dots, s$. Define \mathbb{B} and \mathbb{C} as the block-diagonal matrices with the \mathbf{B}_i and \mathbf{C}_j , respectively, on the diagonal:

$$\mathbb{B} = \begin{pmatrix} \mathbf{B}_1 & & \\ & \ddots & \\ & & \mathbf{B}_g \end{pmatrix} \quad \mathbb{C} = \begin{pmatrix} \mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_s \end{pmatrix} \quad \text{eqn 6}$$

The stage and the group dynamics operate sequentially (see Fig. 2). Because of the arrangement of stages within groups in $\tilde{\mathbf{n}}(t)$ (the population vector at time t), multiplication by \mathbb{B} moves individuals among stages, keeping groups fixed. The result is rearranged to correspond to $\text{vec}(\mathcal{N}^\top)$ and then multiplied by \mathbb{C} to move individuals among groups, keeping stages fixed, and the result of this multiplication is rearranged to obtain $\tilde{\mathbf{n}}(t+1)$,

$$\tilde{\mathbf{n}}(t+1) = \mathbf{K}_{s,g}^\top \mathbb{C} \mathbf{K}_{s,g} \mathbb{B} \tilde{\mathbf{n}}(t) \quad \text{eqn 7}$$

$$= \tilde{\mathbf{A}} \tilde{\mathbf{n}}(t) \quad \text{eqn 8}$$

In the multistate projection matrix $\tilde{\mathbf{A}}$ defined in (8), the vec-permutation matrix \mathbf{K} performs the task of rearranging the

population vector to permit multiplication by the block-diagonal matrices. The great advantage of this is that it keeps the processes determining stage and group transitions in well-defined locations, so that analyses – particularly sensitivity analyses – can operate on them directly (Caswell 2006, 2012 2014b; Caswell & Salguero-Gómez 2013).

The vec-permutation approach to hyperstate matrix models

In a multi-state matrix model, the two-dimensional array \mathcal{N} in (3) is transformed into a vector by the vec operator and rearranged by the vec-permutation matrix so that the block-diagonal matrices \mathbb{B} and \mathbb{C} can be applied. This Section develops hyperstate matrix models as a step-by-step generalisation of the vec-permutation approach. In Section ‘Hypermatrix and i -state space’, we define the i -state space and the p -state of a hyperstate matrix model. In Section ‘From arrays to vectors: linear indexing and the vec operator’, we introduce the *linear indexing* of a hypermatrix and generalise the vec operator. In Section ‘Population vectors and projection matrices’, we first define the population vectors which are the generalisation of the vectors defined in eqn (4). Then, we build the block-diagonal projection matrices that operates on those vectors which are the generalisation of the matrices \mathbb{B} and \mathbb{C} defined in eqn (6). Finally, in Section ‘Vec-permutation matrices’, we define the vec-permutations matrices that connect the different population vectors with each other, and in Section ‘Sequential dynamics of $\tilde{\mathbf{n}}(t)$ ’, we write the model as a product of the block-diagonal projection matrices. Specifying these models requires some careful and (unfortunately) elaborate notation, which we begin to introduce in the next Section.

HYPERMATRIX AND i -STATE SPACE

In the familiar multistate model presented in the ‘The vec-permutation approach to multistate models’ Section, individuals are classified by two characteristics. The i -state space is a two-dimensional product space; one dimension identifies ‘stages’ and the other identifies ‘groups’. In the general case, individuals are classified by m characteristics, called 1-stages, 2-stages, ..., m -stages, with

$$\begin{aligned}
 &1 - \text{stage } i_1 \in \mathcal{S}_1 = \{1, \dots, s_1\} \\
 &\quad \vdots \\
 &m - \text{stage } i_m \in \mathcal{S}_m = \{1, \dots, s_m\}
 \end{aligned}$$

The i -state space is then the product space $\mathcal{S} = \mathcal{S}_1 \times \dots \times \mathcal{S}_m$ of dimension m . For any k , the k th dimension of \mathcal{S} identifies k -stages. The state of an individual is a vector $(i_1, \dots, i_m) \in \mathcal{S}$ that specifies its category according to each of the m characteristics. The space \mathcal{S} contains $s = s_1 \dots s_m$ possible i -states. In the example to be presented in Section ‘An example: stage, age, and frailty’, the number of dimensions is $m = 3$, the 1-stages are the stages, the 2-stages are the age classes, and the 3-stages are the frailty classes.

The state of the population, or p -state, is a density function over \mathcal{S} , which can be written as a m -dimensional hypermatrix of size $s_1 \times \dots \times s_m$

$$\mathcal{N} = (n_{i_1, \dots, i_m}) \quad 1 \leq i_1 \leq s_1, \dots, 1 \leq i_m \leq s_m \quad \text{eqn 9}$$

where n_{i_1, \dots, i_m} is the number of individuals in 1-stage i_1 , 2-stage i_2, \dots and m -stage i_m .

In the particular case $m = 3$, the hypermatrix \mathcal{N} can be visualised as a file drawer of matrices as illustrate in Fig. 1. The reader should refer to this diagram and the three-dimensional example of Section ‘An example: stage, age, and frailty’ to help visualise the definitions stated in next sections.

FROM ARRAYS TO VECTORS: LINEAR INDEXING AND THE VEC OPERATOR

A hypermatrix is an m -dimensional array, the entries of which are identified by their subscripts (i_1, \dots, i_m) . The key element in transforming a hypermatrix into a vector is the numeration of its entries, which associates a single index, $i = 1, \dots, s$, with each entry of the hypermatrix. Given a rule for this numeration, the k th entry of the resulting vector is unambiguously identified with a specific entry of the hypermatrix, that is the entry associated with the single index k .

Defining such a rule is equivalent to assigning, for each entry i_1, \dots, i_m of the hypermatrix \mathcal{N} , a number $f(i_1, \dots, i_m)$, and conversely, assigning for each number k an entry $g(k) = (i_1, \dots, i_m)$ of the hypermatrix. Formally, f is a one-to-one function

$$f: \mathcal{S}_1 \times \dots \times \mathcal{S}_m \longrightarrow \{1, \dots, s\}, \quad \text{eqn 10}$$

that maps hypermatrix subscripts to a vector indices, and g is the inverse function of f

$$g: \{1, \dots, s\} \longrightarrow \mathcal{S}_1 \times \dots \times \mathcal{S}_m \quad \text{eqn 11}$$

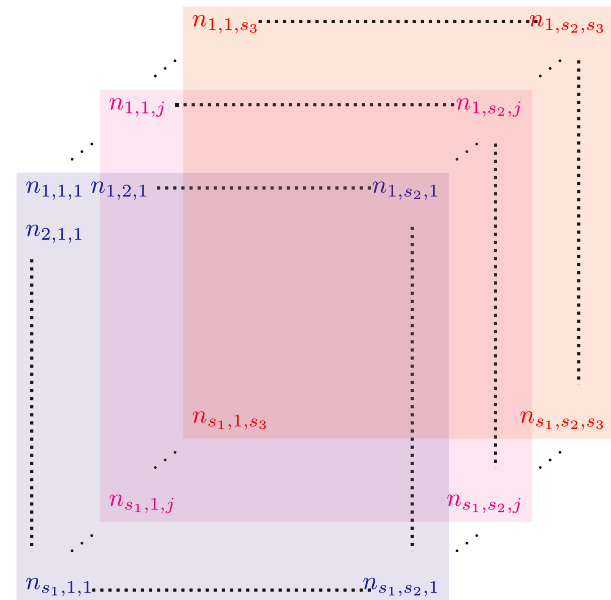


Fig. 1. Hypermatrix \mathcal{N} when $m = 3$.

that maps the vector indices to the hypermatrix subscripts.

The *linear indexing* of a hypermatrix is defined by a particular numeration of its entries. The resulting vector \mathbf{n} is the generalisation to higher dimension of the two-dimensional *vec* of a matrix. Those particular functions f and g , which depend on the sizes s_1, \dots, s_m of the hypermatrix, are called the *index functions* associated with the sizes s_1, \dots, s_m , and they are inverses of each other.

Before stating the general definition of the index function f , we consider a simple example. Consider a clock with dials that show the seconds ($s_1 = 60$), minutes ($s_2 = 60$), hours ($s_3 = 24$) and days ($s_4 = 365$) within a year, so that $m = 4$. On this clock, any second within the year can be identified by its subscript (i_1, i_2, i_3, i_4) or by its index, which tells which of the $60 \times 60 \times 24 \times 365 = 31\,536\,000$ s it is. Each revolution of the seconds dial moves the minutes dial by one position; each revolution of the minutes dial moves the hours dial by one position and so on. The number $f(i_1, \dots, i_4)$ assigned to the time (i_1, \dots, i_4) is the number of seconds required to go from the initial time $(1, 1, 1, 1)$ to the time (i_1, \dots, i_4) , that is

$$f(i_1, i_2, i_3, i_4) = i_1 + 60(i_2 - 1) + 60 \times 60(i_3 - 1) + 24 \times 60 \times 60(i_4 - 1). \quad \text{eqn 12}$$

Formally, the index function f is defined as

$$f(i_1, \dots, i_m) = i_1 + \sum_{k=2}^m (i_k - 1)s_{k-1}s_{k-2} \dots s_1 \quad \text{eqn 13}$$

The inverse function $g = f^{-1}$ is explicitly defined in Section 2.5 in Appendix S2.

The *vec* operator transforms the hypermatrix \mathcal{N} of size $s_1 \times \dots \times s_m$ into the $s \times 1$ column vector

$$\text{vec}(\mathcal{N}) := \sum_{i=1}^s n_{g(i)} \mathbf{e}_i = \begin{pmatrix} n_{g(1)} \\ n_{g(2)} \\ \vdots \\ n_{g(s)} \end{pmatrix}, \quad \text{eqn 14}$$

where g is the inverse of the index function f defined in eqn (13), and \mathbf{e}_i is the column vector of dimension s with one in its i th entry and zero elsewhere. In the two-dimensional case, the *vec* operator stacks the columns of the matrix one on top of the other. This fact passes to higher dimension; the *vec* operator defined in eqn (14) stacks the columns of the hypermatrix one on top of the other (see Theorem 1 in the Appendix S2).

Implementation 3.1 *Conveniently, in MATLAB a multidimensional array is stored as a column vector, although it is displayed as a hypermatrix. The index functions f and g allow one to go back and forth from the display and the storage of the array. The MATLAB functions that carry out the functions f and g are `sub2ind` and `ind2sub`, respectively. MATLAB's 'colon' notation produces the *vec* of a multidimensional array using this indexing. That is, if X is a MATLAB array, then $X(:)$ is the *vec* of X .*

POPULATION VECTORS AND PROJECTION MATRICES

To project the population by matrix multiplication, the hypermatrix \mathcal{N} is transformed into the *population vector*

$$\tilde{\mathbf{n}} = \text{vec}(\mathcal{N}). \quad \text{eqn 15}$$

The population vector is projected by the projection matrix $\tilde{\mathbf{A}}$ of size $s \times s$ containing the transition rates between all i -states,

$$\tilde{\mathbf{n}}(t+1) = \tilde{\mathbf{A}}\tilde{\mathbf{n}}(t),$$

from a specified initial population $\tilde{\mathbf{n}}(0) = \tilde{\mathbf{n}}_0$. The goal of the *vec*-permutation approach is to write the projection matrix $\tilde{\mathbf{A}}$ as a product of matrices, each of which projects one dimension of the i -state space.

To do this, we partition the i -state vector by identifying one dimension at a time as stage and allow the remaining dimensions to specify a group complementary to that stage. Formally, for each dimension k , we say that an individual in state $(i_1, \dots, i_m) \in \mathcal{S}$ is in:

$$k - \text{stage } i_k \in \mathcal{S}_k \quad \text{eqn 16}$$

$$k - \text{group } (i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m) \in \mathcal{G}_k = \mathcal{S}_1 \times \dots \times \mathcal{S}_{k-1} \times \mathcal{S}_{k+1} \times \dots \times \mathcal{S}_m \quad \text{eqn 17}$$

The k -group space \mathcal{G}_k is the product space over all the dimensions other than the one (k in this case) chosen to represent stages; it is of dimension $m-1$ and it contains s/s_k elements. In particular, the vector $\tilde{\mathbf{n}}$ has the 1-stages arranged within 1 groups as the *vec* operator stacks the columns of the hypermatrix one on top of the other.

For each k , the k -stage transition matrix

$$\mathbf{A}_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m}^k \quad \text{eqn 18}$$

projects the k -stages within the k -group $(i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m)$. The matrix in (18) is of size $s_k \times s_k$, and it operates on the $s_k \times 1$ vector, denoted by $\mathbf{n}_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m}^k$, describing the density of the population within the k -group $(i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m)$. This vector is extracted from the hypermatrix as follows

$$\mathbf{n}_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m}^k = \begin{pmatrix} n_{i_1, \dots, i_{k-1}, 1, i_{k+1}, \dots, i_m} \\ n_{i_1, \dots, i_{k-1}, 2, i_{k+1}, \dots, i_m} \\ \vdots \\ n_{i_1, \dots, i_{k-1}, s_k, i_{k+1}, \dots, i_m} \end{pmatrix}. \quad \text{eqn 19}$$

In the two-dimensional case (i.e. $m = 2$), the matrix \mathbf{A}_j^1 is the matrix \mathbf{B}_j , defined in eqn (6), operating on the j th column of the matrix \mathcal{N} , and the matrix \mathbf{A}_i^2 is the matrix \mathbf{C}_i operating on the i th row of the matrix \mathcal{N} .

Implementation 3.2 *The transition matrices $\{\mathbf{A}_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m}^k\}$ are the base of the hyperstate matrix model. They are assumed to be known by the person who intends to implement the model. Following the step-by-step procedure (see Appendix S1), these matrices are stored in a cell array \mathbf{a} .*

To simultaneously project the k -stages within all the k -groups, the matrices $\mathbf{A}_{i_1 \dots i_{k-1} i_{k+1} \dots i_m}^k$ are arranged in the $s \times s$ block-diagonal matrix \mathbb{A}_k . The order in which they are arranged in the diagonal has to be determined. Here, again the key element is the linear indexing. Denote by f_{-k} and g_{-k} the index functions associated with the sizes $s_1, \dots, s_{k-1}, s_{k+1}, \dots, s_m$. Each block $\mathbf{A}_{i_1 \dots i_{k-1} i_{k+1} \dots i_m}^k$ is identified by its subscript $(i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m)$, and the index function f_{-k} associates to this subscript the number $f_{-k}(i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m)$. Now, the blocks are numerated and they are arranged in the diagonal of \mathbb{A}_k in the ascending order,

$$\mathbb{A}_k = \sum_{i=1}^{s/s_k} \mathbf{E}_{ii} \otimes \mathbf{A}_{g_{-k}(i)}^k, \quad \text{eqn 20}$$

where $\mathbf{B} \otimes \mathbf{C}$ is the Kronecker product of the matrices \mathbf{B} and \mathbf{C} , and \mathbf{E}_{ii} is the $s/s_k \times s/s_k$ matrix with one in its i, j entry and zero elsewhere.

Implementation 3.3 *In the Supporting Information material, we provide the MATLAB code for the function `BD_proj_mat` that computes the block-diagonal projection matrix \mathbb{A}_k from the cell array `a` containing the transition matrices $\{\mathbf{A}_{i_1 \dots i_{k-1} i_{k+1} \dots i_m}^k\}$. More precisely, the function `BD_proj_mat` applied to `a{k}` returns the block-diagonal projection matrix \mathbb{A}_k .*

The matrix \mathbb{A}_k operates on a vector \mathbf{n}_k which is a rearrangement of the population vector $\tilde{\mathbf{n}}$,

$$\mathbf{n}_k = \sum_{i=1}^{s/s_k} \mathbf{e}_i \otimes \mathbf{n}_{g_{-k}(i)}^k = \begin{pmatrix} \mathbf{n}_{g_{-k}(1)}^k \\ \vdots \\ \mathbf{n}_{g_{-k}(s/s_k)}^k \end{pmatrix}, \quad \text{eqn 21}$$

where \mathbf{e}_i is the column vector of dimension s/s_k with one in its i th entry and zero elsewhere. Within the vector \mathbf{n}_k , the vectors $\mathbf{n}_{g_{-k}(i)}^k$ are stacked one on top of the other in the same order than the matrices fill the diagonal of \mathbb{A}_k . In other words, the k -stages are arranged within the k -group. The vector \mathbf{n}_k is called the k th population vector.

VEC-PERMUTATION MATRICES

The vec-permutation approach aims to project the dimensions of the i -state space one after the other, that is multiplying successively by the matrices $\mathbb{A}_1, \dots, \mathbb{A}_m$. However, each matrix applies to a different population vector (the projection matrix \mathbb{A}_k applies to the vector \mathbf{n}_k). Hence, at each step, the population vector must be rearranged – for matrix \mathbb{A}_k , k -stages must be arranged within k -groups. This rearrangement is made by multiplying the population vector by an appropriate permutation matrix. In the two-dimensional case, the population vectors defined in ((4) are related by the vec-permutation matrix $\mathbf{K}_{s,g}$ defined in (5). In the general case, there exists a set of matrices $\mathbf{K}_2, \dots, \mathbf{K}_m$ such that

$$\mathbf{n}_j = \mathbf{K}_j \mathbf{n}_{j-1} \quad \text{eqn 22}$$

for all $j = 2, \dots, m$. The population vectors \mathbf{n}_{j-1} and \mathbf{n}_j have the same entries but they are arranged in a different order. The matrix \mathbf{K}_j shuffles the entries of the vector \mathbf{n}_{j-1} to arrange them in the order of the vector \mathbf{n}_j . Each row and each column of the matrix \mathbf{K}_j has a unique entry equal to one and the others equal to zero. The matrices $\mathbf{K}_2, \dots, \mathbf{K}_m$ belong to the larger family called the *vec-permutation* matrices which relate the vec of a hypermatrix to its transposes (see Section 2.3 in Appendix S2 for more details). These vec-permutation matrices are orthogonal; that is, their inverse is their transpose. In particular, the inverse the matrix \mathbf{K}_j is its transpose

$$\mathbf{K}_j^{-1} = \mathbf{K}_j^\top. \quad \text{eqn 23}$$

Implementation 3.4 *In the Supplementary Information material, we provide the MATLAB code of the function `vecperm_hyp` that returns the vec-permutation matrix \mathbf{K}_j associated with a hypermatrix of size $s_1 \times \dots \times s_m$.*

In the remainder of this section, we explicitly describe the matrices $\mathbf{K}_2, \dots, \mathbf{K}_m$. For any integers $u, v > 0$, let $\mathbf{E}_{ij}(u, v)$ be the $u \times v$ matrix with a one in the (i, j) position and zeros elsewhere and define the $uv \times uv$ matrix

$$\mathbf{Q}(u, v) = \sum_{i=1}^u \sum_{j=1}^v \mathbf{E}_{ij}(u, v) \otimes \mathbf{E}_{ij}(u, v)^\top. \quad \text{eqn 24}$$

For any integers u_1, \dots, u_m and any $k \geq 0$, define the matrix

$$\mathbf{P}_k(u_1, \dots, u_m) = \begin{cases} \mathbf{I}_{u_{k+1} \dots u_m} \otimes \mathbf{Q}(u_1 \dots u_{k-1}, u_k) & \text{if } 2 \leq k \leq m-1 \\ \mathbf{I}_{u_1 \dots u_m} & \text{if } k = 1 \\ \mathbf{Q}(u_1 \dots u_{m-1}, u_m) & \text{if } k = m \end{cases} \quad \text{eqn 25}$$

where \mathbf{I}_d is the identity matrix of dimension d . Then, the vec-permutation matrices are given by

$$\mathbf{K}_j = \mathbf{P}_k(s_1, \dots, s_m) \mathbf{P}_{k-1}(s_1, \dots, s_m)^\top, \quad \text{eqn 26}$$

for $j = 1, \dots, m$. Note that the matrix \mathbf{K}_j depend on the sizes s_1, \dots, s_m . When it is needed, we explicitly write those sizes as parameters, $\mathbf{K}_j(s_1, \dots, s_m)$. For example, the matrix $\mathbf{K}_2(s, g)$ is the matrix $\mathbf{K}_{s,g}$ defined in eqn (5) for the two-dimensional model.

The mathematical difficulties of the formalism of the hyperstate matrix model is concentrated in the proof that the matrices $\mathbf{K}_2, \dots, \mathbf{K}_m$ satisfy eqn (22). This proof is presented in Appendix S2 (see Theorem 13).

SEQUENTIAL DYNAMICS OF $\tilde{\mathbf{n}}(t)$

Combining the ingredients introduced above gives the hyperstate projection matrix $\tilde{\mathbf{A}}$ as the product of matrices

$$\tilde{\mathbf{A}} = (\mathbf{K}_m \dots \mathbf{K}_2)^\top \mathbb{A}_m \mathbf{K}_m \mathbb{A}_{m-1} \mathbf{K}_{m-1} \dots \mathbb{A}_2 \mathbf{K}_2 \mathbb{A}_1 \quad \text{eqn 27}$$

and the dynamics of $\tilde{\mathbf{n}}(t)$

$$\tilde{\mathbf{n}}(t+1) = \tilde{\mathbf{A}} \tilde{\mathbf{n}}(t) \quad \text{eqn 28}$$

where $\tilde{\mathbf{n}}(0) = \text{vec}(\mathcal{N}(0))$ is the initial population vector.

Each of the matrices \mathbb{A}_k describes movement of the population along one dimension, which we have identified as a set of k - stages. Each of the matrices \mathbf{K}_j is a vec-permutation matrix that rearranges the population vector. The dynamics of each of the dimensions operate sequentially (Fig. 2). Equation (27) can be read from right to left as follows: the population vector is set up with 1-stages arranged within 1-groups, the population is projected within each of those 1-groups (by \mathbb{A}_1), the new population vector is shuffled to arrange the 2-stages within 2-groups (by \mathbf{K}_2) and so on until the population is projected within each of the m -groups (by \mathbb{A}_m); finally the population vector is shuffled back to its initial arrangement (by $(\mathbf{K}_m \cdots \mathbf{K}_2)^\top$).

Note that the order in which the matrices are applied matters, because matrix multiplication is not commutative. Cyclic permutations of the matrices (e.g. changing $1 \rightarrow 2 \rightarrow 3$ to $2 \rightarrow 3 \rightarrow 1$) correspond to observing the population at different points in the cyclic dynamics. Some properties (e.g. asymptotic population growth rates) are invariant under such permutations; others (e.g. eigenvectors, the net reproductive rate) are not invariant, but change in understandable ways (Caswell, 2001, 2009). The importance of the order in which the matrices are applied is also discussed in Section ‘Discussion’.

Implementation 3.5 *In Appendix S1, we provide a step-by-step procedure to generate the matrix $\tilde{\mathbf{A}}$ from the list of transition matrices $\{\mathbf{A}_{i_1 \dots i_{k-1} i_{k+1} \dots i_m}^k\}$. This makes it possible to carry out population growth calculations directly from the demographic processes operating in all m dimensions.*

SENSITIVITY ANALYSIS

One advantage of the vec-permutation approach to hyperstate matrix models is that it makes possible a systematic calculation of the sensitivity of demographic results to changes in

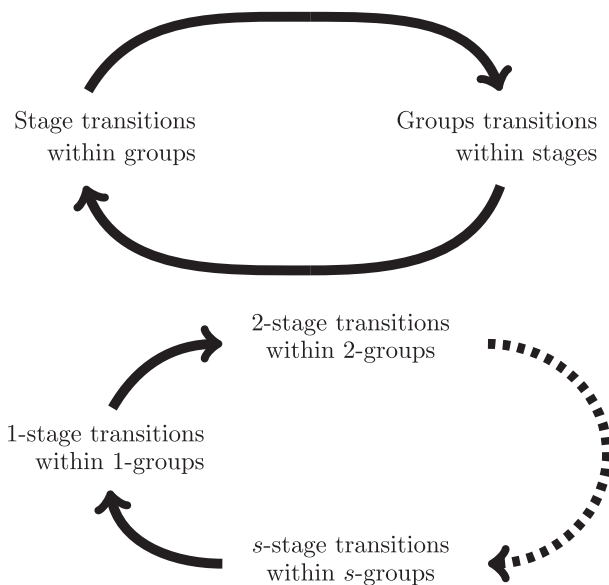


Fig. 2. The sequential operation of the transitions in a multistate (top) and hyperstate (bottom) matrix model.

parameters. The dynamics of the population are determined by the matrix $\tilde{\mathbf{A}}$, which in turn depends on the m block-diagonal matrices \mathbb{A}_k , as shown in (27). The matrix \mathbb{A}_k contains the matrices $\mathbf{A}_{g(i)}$, for $i = 1, \dots, s/s_k$, as shown in (20). And finally, depending on the parameters of interest, each of those matrices may depend in its own way on some vector $\boldsymbol{\theta}$ of parameters. Untangling all the causal links from the parameters to the model output is difficult, but the vec-permutation formulation permits a straightforward methodological approach using matrix calculus (Magnus & Neudecker (1985, 1999); see Caswell (2007, 2008, 2012); see Caswell & Shyu (2012); Caswell & Salguero-Gómez (2013) for ecological presentations).

Consider the sensitivity analysis of a generic-dependent variable (vector or scalar), which we will denote ξ , calculated from the projection matrix $\tilde{\mathbf{A}}$. Let $\boldsymbol{\theta}$ be a vector of parameters; these could be entries of the matrices, or lower level parameters determining those entries. The goal is to obtain the derivative of ξ with respect to $\boldsymbol{\theta}$. This derivative,

$$\frac{d\xi}{d\boldsymbol{\theta}^\top} = \left(\frac{d\xi_i}{d\theta_j} \right), \tag{eqn 29}$$

is a matrix whose i,j entry is the derivative of ξ_i with respect to θ_j .

By the chain rule of matrix calculus,

$$\frac{d\xi}{d\boldsymbol{\theta}^\top} = \frac{d\xi}{d\text{vec}^\top \tilde{\mathbf{A}}} \frac{d\text{vec} \tilde{\mathbf{A}}}{d\boldsymbol{\theta}^\top}. \tag{eqn 30}$$

The first term in (30) is specific to the choice of the dependent variable ξ and its calculation from $\tilde{\mathbf{A}}$. The second term is specific to the choice of the parameter vector $\boldsymbol{\theta}$ and the way that all the components of $\tilde{\mathbf{A}}$ depend on $\boldsymbol{\theta}$. In Section ‘Effect of heterogeneous frailty on population growth rate’, we present an example where ξ is the population growth rate λ and $\boldsymbol{\theta}$ is a measure of the accuracy of parent-to-offspring transmission of frailty.

The projection matrix $\tilde{\mathbf{A}}$ is defined by (27). For each value of k , we partition the product in (27) into components to the left and to the right of \mathbb{A}_k , writing

$$\mathbf{L}_k = (\mathbf{K}_2 \cdots \mathbf{K}_m)^\top \mathbb{A}_m \cdots \mathbf{K}_{k+2} \mathbb{A}_{k+1} \mathbf{K}_{k+1} \tag{eqn 31}$$

$$\mathbf{R}_k = \mathbf{K}_k \mathbb{A}_{k-1} \cdots \mathbf{K}_2 \mathbb{A}_1 \tag{eqn 32}$$

for $k = 1, \dots, m$.

Differentiating $\tilde{\mathbf{A}}$ gives

$$\frac{d\text{vec} \tilde{\mathbf{A}}}{d\boldsymbol{\theta}^\top} = \sum_{k=1}^m (\mathbf{R}_k^\top \otimes \mathbf{L}_k) \frac{d\text{vec} \mathbb{A}_k}{d\boldsymbol{\theta}^\top}. \tag{eqn 33}$$

This reduces the problem to that of calculating the derivatives of the matrices \mathbb{A}_k . Applying Theorem 11 in Magnus & Neudecker (1985), on the derivative of the Kronecker product, to (20) gives

$$\frac{d\text{vec} \mathbb{A}_k}{d\boldsymbol{\theta}^\top} = \sum_{i=1}^{s/s_k} (\mathbf{I}_{s/s_k} \otimes \mathbf{K}_2(s_k, s/s_k) \otimes \mathbf{I}_{s_k}) (\text{vec} \mathbf{E}_{ii} \otimes \mathbf{I}_{s_k^2}) \frac{d\text{vec} \mathbf{A}_{g(i)}^k}{d\boldsymbol{\theta}^\top}. \tag{eqn 34}$$

Everything in (34) is constant except for the derivatives of the $\mathbf{A}_{g(i)}^k$ with respect to $\boldsymbol{\theta}$. These derivatives contain all the biological content of the model. Substituting eqn (34) into eqn (33) and then substituting eqn (33) into eqn (30) yield the derivative

$$\frac{d\xi}{d\boldsymbol{\theta}^\top} = \frac{d\xi}{d\text{vec}^\top \tilde{\mathbf{A}}} \left[\sum_{k=1}^m (\mathbf{R}_k^\top \otimes \mathbf{L}_k) \left(\sum_{i=1}^{s/s_k} (\mathbf{I}_{s/s_k} \otimes \mathbf{K}_2(s_k, s/s_k) \otimes \mathbf{I}_{s_k}) (\text{vec} \mathbf{E}_{ii} \otimes \mathbf{I}_{s_k^2}) \frac{d\text{vec} \mathbf{A}_{g(i)}^k}{d\boldsymbol{\theta}^\top} \right) \right] \quad \text{eqn 35}$$

The formula (35) provides the sensitivity of a generic variable, ξ , to any set of parameters, $\boldsymbol{\theta}$. Moreover, each of the terms in the summation in eqn (35) provides the contribution to the sensitivity of ξ from parameter effects on the transition matrices of each of the groups along each of the dimensions.

An example: stage, age and frailty

In this Section, we construct and analyse a (hypothetical) three-dimensional (i.e. $m = 3$) hyperstate matrix model. We start from a multistate model of Caswell (2012), in which individuals are classified by age and developmental stage. We add a third dimension by assuming that heterogeneous *frailty* may affect mortality. Frailty is a concept introduced by human demographers Vaupel, Manton & Stallard (1979) to explain deviations in the age trajectories of mortality from simple parametric models. Since then, it has been extensively studied as a way to incorporate individual differences in susceptibility to mortality risks (e.g. Yashin, Iachine & Begun 2000; Wienke 2011). We include two developmental stages, *juvenile* and *adult*, and two frailty classes, *weak* and *strong*.

Thus, $S_1 = \{1, 2\}$ is the set of developmental stages, $S_2 = \{1, \dots, \omega\}$ is the set of age classes, and $S_3 = \{1, 2\}$ is the set of frailty classes. An individual in the i -state (i, j, k) is in developmental stage i , age class j and frailty class k . With this order, individuals first move among developmental stages and reproduce, as a function of their age and frailty. Then, ageing acts to advance individuals to the next age class, and finally, the frailty class of surviving individuals is updated. Then, the process repeats.

We make the following assumptions:

(H1) The mortality of juveniles is independent of their age and their frailty. The mortality of adults depends on frailty but is independent of age.

(H2) Mortality happens during the second phase of the dynamics when individuals advance to the next age class (this permits easy addition of an age-specific mortality component if desired).

(H3) Transitions of existing individuals among frailty classes depend on their stage but not on their age.

The transmission of frailty from parents to offspring is independent of the age and stage of the parents.

(H4) there is an interval of ages $[a_{\min}, a_{\max}]$ within which adults reproduce at rate $r \geq 0$.

Each of these could be replaced with different assumptions about the action of frailty, the timing of mortality, the dynamics of frailty and/or the schedule of reproduction. In Section ‘The hyperstate projection matrix’, we use H1–H4 to demonstrate the step-by-step construction of the hyperstate matrix model and its implementation in MATLAB. The resulting model can be analysed like any other matrix model (Caswell 2001) to obtain a complete set of demographic outputs, both at the population level (e.g. population growth rate, stable structure, reproductive value) and at the cohort level (e.g. longevity, net reproductive rate). In Section ‘Effect of heterogeneous frailty on population growth rate’, we illustrate this potential by analysing the effect of heterogeneous frailty on population growth rate.

THE HYPERSTATE PROJECTION MATRIX

It is convenient to decompose the projection matrix as $\mathbf{A} = \mathbf{U} + \mathbf{F}$, where \mathbf{U} describes the transitions and survival of existing individuals and \mathbf{F} describes the creation of new individuals by reproduction. Here, we do the same for the hyperstate projection matrix

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}} + \tilde{\mathbf{F}} \quad \text{eqn 36}$$

Both components, $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{F}}$, are built following the method presented in Section ‘The vec-permutation approach to hyperstate matrix models’.

Construction of the matrix $\tilde{\mathbf{U}}$

Following eqn (27) with $m = 3$, the hyperstate matrix $\tilde{\mathbf{U}}$ is the product of three block-diagonal matrices, which we denote here by $\mathbb{U}, \mathbb{D}^U, \mathbb{H}^U$, respectively. The elements in these block-diagonal matrices are described below; they appear in the hyperstate matrix

$$\tilde{\mathbf{U}} = (\mathbf{K}_3 \mathbf{K}_2)^\top \mathbb{H}^U \mathbf{K}_3 \mathbb{D}^U \mathbf{K}_2 \mathbb{U} \quad \text{eqn 37}$$

where the matrices \mathbf{K}_2 and \mathbf{K}_3 are the vec-permutation matrices defined in (26) and associated with the sizes 2, ω , 2.

Implementation 4.1 *Following Implementation 3.4, the vec-permutation matrices \mathbf{K}_2 and \mathbf{K}_3 are given by `vecperm_hyp(2,[2, ω , 2])` and `vecperm_hyp(3,[2, w , 2])`, respectively.*

Reading from right to left within the matrix product $(\mathbf{K}_3 \mathbf{K}_2)^\top \mathbb{H}^U \mathbf{K}_3 \mathbb{D}^U \mathbf{K}_2 \mathbb{U}$ the matrix \mathbb{U} moves extant individuals among stages without changing their age or frailty. Then, the matrix \mathbf{K}_2 rearranges the age classes within weak and strong juvenile and within weak and strong adults groups. The matrix \mathbb{D}^U advances surviving individuals to the next age class without changing their stage or

frailty. The matrix \mathbf{K}_3 rearranges frailty classes within juveniles of each age class and within adults of each age. The matrix \mathbb{H}^U moves individuals among frailty class without changing their stage or age. Finally, the permutation matrix $(\mathbf{K}_3\mathbf{K}_2)^\top$ rearranges the vector back to the stage-within-age-frailty groups arrangement of $\tilde{\mathbf{n}}$. The description of all these matrices is summarised in Table 1.

The families of matrices $\{\mathbf{U}_{jk}\}$, $\{\mathbf{D}_{jk}^U\}$ and $\{\mathbf{H}_{ij}^U\}$ that appear in the block-diagonal matrices $\mathbb{U}, \mathbb{D}^U, \mathbb{H}^U$ are the basis of the model. In the following we describe those matrices according to assumptions H1–H4. The stage transition matrix

$$\mathbf{U}_{jk} = \begin{pmatrix} u_{jk} & 0 \\ 1 - u_{jk} & 1 \end{pmatrix} \quad \text{eqn 38}$$

contains the transition probabilities between stages of extant individuals for stage j and frailty class k . Since death happens only in the second phase of the dynamics (assumption (H2)), each column of \mathbf{U}_{jk} sums to 1.

To construct the matrices describing age transitions and mortality, define the $\omega \times \omega$ matrix \mathbf{Y} with ones on the subdiagonal and in the lower right corner, for example for $\omega = 3$:

$$\mathbf{Y} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{eqn 39}$$

The 1 in the (ω, ω) entry makes the last age class open-ended.

The age transition matrices for juveniles and adults in frailty class k are

$$\mathbf{D}_{1k}^U = e^{-\nu}\mathbf{Y} \quad \text{eqn 40}$$

$$\mathbf{D}_{2k}^U = e^{-\mu z_k}\mathbf{Y}. \quad \text{eqn 41}$$

The parameter ν is the mortality rate of juveniles (stage 1), which advance to the next age class with probability $e^{-\nu}$

regardless of age or frailty. The survival of adults (stage 2) depends on frailty. The parameter μ is the baseline mortality rate of adult individual, and z_1 and z_2 are multiplicative factors associated with frailty classes 1 and 2. Since individuals are supposed to be *strong* in frailty class 1 and *weak* in frailty class 2, we assume $z_1 < z_2$.

The frailty transition matrix

$$\mathbf{H}_{ij}^U = \begin{pmatrix} h_1^i & 1 - h_1^i \\ 1 - h_1^i & h_2^i \end{pmatrix} \quad \text{eqn 42}$$

contains the transition probabilities between frailty classes for individuals for stage i and age class j . If frailty is a fixed property, then \mathbf{H}^U is an identity matrix.

To arrange the matrices $\mathbf{U}_{jk}, \mathbf{D}_{jk}^U$ and \mathbf{H}_{ij}^U into the block-diagonal matrices $\mathbb{U}, \mathbb{D}^U, \mathbb{H}^U$, we follow eqn (20)

$$\mathbb{U} = \text{diag}(\mathbf{U}_{11}, \dots, \mathbf{U}_{\omega 1}, \mathbf{U}_{12}, \dots, \mathbf{U}_{\omega 2}) \quad \text{eqn 43}$$

$$\mathbb{D}^U = \text{diag}(\mathbf{D}_{11}^U, \mathbf{D}_{21}^U, \mathbf{D}_{12}^U, \mathbf{D}_{22}^U) \quad \text{eqn 44}$$

$$\mathbb{H}^U = \text{diag}(\mathbf{H}_{11}^U, \mathbf{H}_{21}^U, \dots, \mathbf{H}_{1\omega}^U, \mathbf{H}_{2\omega}^U) \quad \text{eqn 45}$$

Construction of the fertility matrix $\tilde{\mathbf{F}}$

Following eqn (27) with $m = 3$, the hyperstate matrix $\tilde{\mathbf{F}}$ is constructed from the block-diagonal matrices \mathbb{F}, \mathbb{D}^F and \mathbb{H}^F

$$\tilde{\mathbf{F}} = (\mathbf{K}_3\mathbf{K}_2)^\top \mathbb{H}^F \mathbf{K}_3 \mathbb{D}^F \mathbf{K}_2 \mathbb{F}. \quad \text{eqn 46}$$

Reading from right to left within the matrix product $(\mathbf{K}_3\mathbf{K}_2)^\top \mathbb{H}^F \mathbf{K}_3 \mathbb{D}^F \mathbf{K}_2 \mathbb{F}$, the block-diagonal matrix \mathbb{F} first creates new individuals and places them in the juvenile stage, within the age and frailty class of their parents. Then, the matrix \mathbf{K}_2 rearranges the age classes within weak and strong juvenile and within weak and strong adults groups. The matrix \mathbb{D}^F places newborn individuals in the first age class without changing their stage or frailty. The matrix \mathbf{K}_3

Table 1. Matrices used in Section ‘An example: stage, age, and frailty’. ω denotes the number of age classes

Notation	Expression	Size	Description
Matrix describing the full population			
$\tilde{\mathbf{A}}$	$\tilde{\mathbf{U}} + \tilde{\mathbf{F}}$	$4\omega \times 4\omega$	Hyperstate population projection matrix
Matrices describing transitions and survival of existing individuals			
$\tilde{\mathbf{U}}$	$(\mathbf{K}_3\mathbf{K}_2)^\top \mathbb{H}^U \mathbf{K}_3 \mathbb{D}^U \mathbf{K}_2 \mathbb{U}$	$4\omega \times 4\omega$	Hyperstate transitions and survival matrix
\mathbb{U}	$\text{diag}(\mathbf{U}_{11}, \dots, \mathbf{U}_{\omega 1}, \mathbf{U}_{12}, \dots, \mathbf{U}_{\omega 2})$	$4\omega \times 4\omega$	Block-diagonal stage transition matrix
\mathbb{D}^U	$\text{diag}(\mathbf{D}_{11}^U, \mathbf{D}_{21}^U, \mathbf{D}_{12}^U, \mathbf{D}_{22}^U)$	$4\omega \times 4\omega$	Block-diagonal age transition matrix
\mathbb{H}^U	$\text{diag}(\mathbf{H}_{11}^U, \mathbf{H}_{21}^U, \dots, \mathbf{H}_{1\omega}^U, \mathbf{H}_{2\omega}^U)$	$4\omega \times 4\omega$	Block-diagonal frailty transition matrix
\mathbf{U}_{jk}	\mathbf{U}_{jk}	2×2	Stage transition matrix for age class j and frailty class k
\mathbf{D}_{jk}^U	\mathbf{D}_{jk}^U	$\omega \times \omega$	Age transition matrix for stage i and frailty class k
\mathbf{H}_{ij}^U	\mathbf{H}_{ij}^U	2×2	Frailty transition matrix for stage i and age class j
Matrices describing reproduction			
$\tilde{\mathbf{F}}$	$(\mathbf{K}_3\mathbf{K}_2)^\top \mathbb{H}^F \mathbf{K}_3 \mathbb{D}^F \mathbf{K}_2 \mathbb{F}$	$4\omega \times 4\omega$	Hyperstate reproduction matrix
\mathbb{F}	$\text{diag}(\mathbf{F}_{11}, \dots, \mathbf{F}_{\omega 1}, \mathbf{F}_{12}, \dots, \mathbf{F}_{\omega 2})$	$4\omega \times 4\omega$	Block-diagonal reproduction matrix
\mathbb{D}^F	$\text{diag}(\mathbf{D}_{11}^F, \mathbf{D}_{21}^F, \mathbf{D}_{12}^F, \mathbf{D}_{22}^F)$	$4\omega \times 4\omega$	Block-diagonal age assignment matrix
\mathbb{H}^F	$\text{diag}(\mathbf{H}_{11}^F, \mathbf{H}_{21}^F, \dots, \mathbf{H}_{1\omega}^F, \mathbf{H}_{2\omega}^F)$	$4\omega \times 4\omega$	Block-diagonal frailty heritability matrix
\mathbf{F}_{jk}	\mathbf{F}_{jk}	2×2	Reproduction matrix for age class j and frailty class k
\mathbf{D}_{jk}^F	\mathbf{D}_{jk}^F	$\omega \times \omega$	Age assignment matrix of for stage i and frailty class k
\mathbf{H}_{ij}^F	\mathbf{H}_{ij}^F	2×2	Frailty heritability matrix of for stage i and age class j

rearranges frailty classes within juveniles of each age class and within adults of each age. The matrix \mathbb{H}^F assigns newborn individuals to a frailty class, dependent on the frailty of their parents, without changing their stage or age. Finally, the permutation matrix $(\mathbf{K}_3\mathbf{K}_2)^\top$ rearranges the vector back to the stage-within-age-frailty groups arrangement of $\tilde{\mathbf{n}}$. The description of all these matrices is summarised in Table 1.

The families of matrices $\{\mathbf{F}_{jk}\}$, $\{\mathbf{D}_{ik}^F\}$ and $\{\mathbf{H}_{ij}^F\}$ which appear in the block-diagonal matrices \mathbb{F} , \mathbb{D}^F , \mathbb{H}^F are constructed according to assumptions H1–H4. The reproduction matrix

$$\mathbf{F}_{jk} = \begin{pmatrix} 0 & f_{jk} \\ 0 & 0 \end{pmatrix}, \quad \text{eqn 47}$$

describes the production of births by juveniles and adults of age class j and for frailty class, k . By assumption on the reproduction schedule (assumption H4),

$$f_{jk} = \begin{cases} r & \text{if } j \in [a_{\min}, a_{\max}] \\ 0 & \text{elsewhere.} \end{cases} \quad \text{eqn 48}$$

Newborn individuals are placed in the first age class, regardless of the stage, age or frailty class of their parents by the age assignment matrix with ones in the first row and zeros elsewhere, for example when $\omega = 3$

$$\mathbf{D}_{ik}^F = \mathbf{D}^F = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{eqn 49}$$

The transmission of frailty from parent to offspring is defined by the frailty heritability matrix

$$\mathbf{H}_{ij}^F = \mathbf{H}^F = \begin{pmatrix} p & 1-q \\ 1-p & q \end{pmatrix}, \quad \text{eqn 50}$$

where p is the probability of inheriting parental frailty for the offspring of a strong individual and q is the probability of inheriting parental frailty class for offspring of a weak individual. Assumption H3 implies that the relation between the frailty of parents and offspring is independent of parental age and stage.

As in eqns (43–45), we arrange the matrices \mathbf{F}_{jk} , \mathbf{D}_{ik}^F , \mathbf{H}_{ij}^F according to (20) into the block-diagonal matrices \mathbb{F} , \mathbb{D}^F , \mathbb{H}^F , respectively,

$$\mathbb{F} = \text{diag}(\mathbf{F}_{11}, \dots, \mathbf{F}_{\omega 1}, \mathbf{F}_{12}, \dots, \mathbf{F}_{\omega 2}) \quad \text{eqn 51}$$

$$\mathbb{D}^F = \text{diag}(\mathbf{D}^F, \mathbf{D}^F, \mathbf{D}^F, \mathbf{D}^F) \quad \text{eqn 52}$$

$$\mathbb{H}^F = \text{diag}(\mathbf{H}^F, \dots, \mathbf{H}^F) \quad \text{eqn 53}$$

Sequential dynamics of $\tilde{\mathbf{n}}(t)$

The hypermatrix \mathcal{N} describing the initial state of the population has the structure in Fig. 1 with rows (left to right) corresponding to developmental stages, columns (top to bottom) to age classes and skewers (front to back) to frailty classes. The resulting population vector is

$$\tilde{\mathbf{n}} = \text{vec}(\mathcal{N}) = \begin{pmatrix} n_{111} \\ n_{211} \\ \vdots \\ \frac{n_{1\omega 1}}{n_{112}} \\ n_{2\omega 1} \\ n_{112} \\ n_{212} \\ \vdots \\ \frac{n_{1\omega 2}}{n_{2\omega 2}} \end{pmatrix} \quad \text{eqn 54}$$

The vector $\tilde{\mathbf{n}}$ is obtained by stacking the columns of the first page of the hypermatrix, then the columns of the second page (cf. Fig. 1). In the first half of the vector, developmental stages are arranged within age classes for frailty class 1, and in the second half, the developmental stages are arranged with age classes for the frailty class 2.

The hyperstate population projection matrix is

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}} + \tilde{\mathbf{F}} \quad \text{eqn 55}$$

and dynamics of $\tilde{\mathbf{n}}(t)$ is

$$\tilde{\mathbf{n}}(t+1) = \tilde{\mathbf{A}}\tilde{\mathbf{n}}(t) \quad \text{eqn 56}$$

where $\tilde{\mathbf{n}}(0) = \tilde{\mathbf{n}}$.

The matrix $\tilde{\mathbf{A}}$ can be subjected to all the usual demographic analyses, including sensitivity analysis. In Section ‘Effect of heterogeneous frailty on population growth rate’, we present an analysis of the sensitivity of the population growth rate to changes in parameters.

Implementation 4.2 In the file `hyper_state_example.m`, available in the Supplementary Information material, we provide the MATLAB code for computing the hyperstate projection matrices $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{F}}$ for this example, given the transition rates $r, u_{jk}, h_1^1, h_2^1, p$ and q , and the parameters $a_{\min}, a_{\max}, \mu, \nu, z_1$ and z_2 .

EFFECT OF HETEROGENEOUS FRAILTY ON POPULATION GROWTH RATE

In this example, frailty is an organismal trait, not accounted for by age or stage, that affects mortality. Such a trait, if not accounted for, can distort the results of analyses based on other state variables. In this section, we investigate how the dynamics of frailty affect the long-term growth rate of a population. First, we investigate a scenario where frailty is an inherited trait that is constant over the lifetime of an individual. Secondly, we investigate a scenario where frailty is a changing trait along the lifetime of an individual, but it is not inherited. While in the former scenario the frailty is seen as a fixed trait of the individual, in the latter scenario, it is seen as a health status that varies during the lifetime of an individual. Frailty may or may not be inherited from parent to offspring.

Heritable frailty

Suppose that the frailty class of an individual is unchanging, and that new individuals inherit the frailty of their

parents with probability p . This scenario is incorporated in our model by defining the frailty heritability matrices

$$\mathbf{H}_{1j}^U = \mathbf{H}_{2j}^U = \mathbf{H}^U \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{eqn 57}$$

$$\mathbf{H}_{ij}^F = \mathbf{H}^F \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}. \quad \text{eqn 58}$$

When frailty is perfectly heritable ($p = 1$), the population in frailty class 1 is independent of the population in frailty class 2. The long-term growth rate λ of the total population is the long-term growth rate of the strongest subpopulation (here the population in frailty class 1), and the frequency of the weaker subpopulation declines to 0.

The long-term growth rate λ is an increasing function of the probability p ; the full heritability scenario ($p = 1$) maximises the long-term growth rate (Fig. 3). We say that the model is *fair* when $p = 1/2$; that is, distribution of offspring among frailty classes is independent of the frailty of their parents. Deviation from the fair scenario can enhance or harm the population. Deviations in the direction of isolating strong and weak classes by increasing p increase λ . Forcing ‘alternate’ mixing (i.e. decreasing p) reduces λ .

Reduction phenomenon for heritability

In this model, the heritability parameter p is responsible for ‘mixing’ the low frailty and high frailty components of the population. Figure 3 shows that λ is an increasing function of p ; that is, that greater mixing reduces growth. Altenberg (2012) has proven that mixing leads to reduced growth rates for a large class of unstructured population models, where the mixing may occur because of dispersal in spatial models, mutation in genetic models, etc. Li & Schreiber (2006) have proven a weaker version of the reduction principle for multistate models. In particular, their result implies that when the mixing only occur in one stage and one age

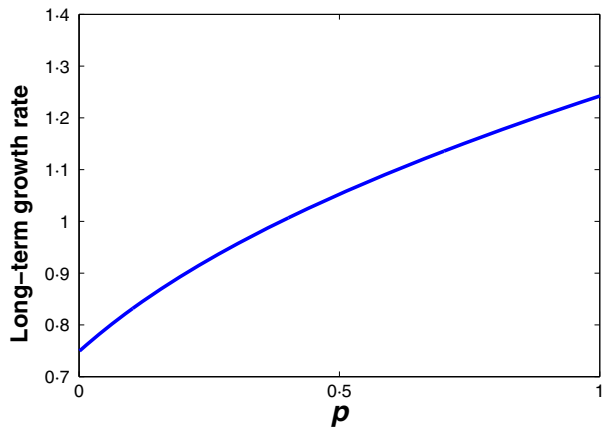


Fig. 3. The long-term growth rate λ as a function of p . The parameters are $\omega = 10, r = 1.3, u_{jk} = 0.5, a_{\min} = 0, a_{\max} = 10, \mu = 0.8, v = 0.2, z_1 = 0.2$ and $z_2 = 5$.

class – as it is the case here – it reduces growth. Because increases in p reduce the mixing between the more and less frail components of the population, the results in Fig. 3 is an example of this reduction phenomenon. Moreover, we can prove that λ is an increasing function of p for any reproductive rate $r > 0$ and any choice of frailty values z_1 and z_2 . (The proof, not shown here, uses the sensitivity analysis of λ to changes in p). This result does not hold universally; if other parameters (e.g. reproductive rate) depend on frailty, λ need not be an increasing function of p . Also, following Li & Schreiber (2006), if the mixing occurs in multiple age classes, mixing need not reduce growth.

Dynamic frailty

So far, we have assumed that frailty is fixed over the lifetime of an individual. However, individual frailty may increase or decrease dynamically due to stress, disease, etc (see Vaupel & Yashin 2006). In this section, we examine a dynamic frailty model. Frailty class 1 is healthy; frailty class 2 is associated with some event that affects individual health.

Following Vaupel & Yashin (2006), we set the frailty class of a newborn individual to 1 (healthy). The frailty of adults changes from class 1 to class 2 with probability α and from class 2 to class 1 with probability β . Vaupel & Yashin (2006) considered irrevocable changes, with $\beta = 0$, but we will consider reversible changes (e.g. disease that can be fully healed) with $\beta > 0$.

To incorporate dynamic frailty, we define frailty heritability matrices

$$\mathbf{H}_{1j}^U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{eqn 59}$$

$$\mathbf{H}_{2j}^U = \begin{pmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{pmatrix} \quad \text{eqn 60}$$

$$\mathbf{H}_{ij}^F = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}. \quad \text{eqn 61}$$

The matrix \mathbf{H}_{1j}^U describes the unchanging frailty state of juvenile individuals of any age j . The stochastic matrix \mathbf{H}_{2j}^U describes a two stages (healthy and ill) Markov chain that models the health status of adults of any age j . This process can be characterised by its autocorrelation ρ and the frequency of the healthy state during adult life. A positive autocorrelation leads to long runs of healthy or ill status; a negative autocorrelation leads to frequent reversals of health status. The matrix \mathbf{H}_{ij}^F assigns all newborn individuals, regardless of the health state or the age of their parents, to the healthy state.

Figure 4 shows that the long-term growth rate increases with the frailty autocorrelation for any frequency. The population increases more rapidly when sick individuals take a long time to recover and healthy individual stay healthy for long period of time. The same pattern is observed for any frailty inheritance matrix \mathbf{H}^F .

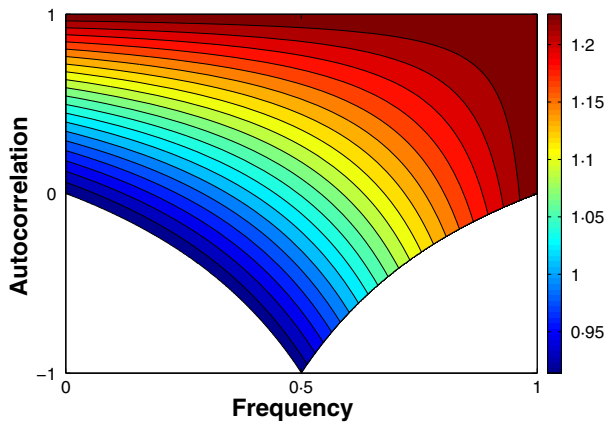


Fig. 4. The long-term growth rate λ as a function of the autocorrelation ρ and the frequency f of the low frailty state. The parameters are $\omega = 10$, $r = 1.3$, $u_{jk} = 0.5$, $a_{\min} = 0$, $a_{\max} = 10$, $\mu = 0.8$, $v = 0.2$, $z_1 = 0.2$ and $z_2 = 5$.

Discussion

Matrix population models project the density of a structured population to the next time step by matrix multiplication. It is often useful to classify individuals on two or more dimensions (e.g. Rogers 1967; Land & Rogers 1982; Andersen *et al.* 1993; Pascarella & Horvitz 1998; Hunter & Caswell 2005; Willekens 2014). Models with two dimensions are called *multistate* models; we have generalised this to an arbitrary number of dimensions in what we call *hyperstate* models.

In order to take advantage of the power of matrix analysis, when individuals are classified on more than one dimension, those dimensions must be rearranged into a vector, and an appropriate matrix must be constructed to project that vector. The vec-permutation model provides a methodical approach to writing the projection matrix as a product of block-diagonal matrices. The great advantage of this is that the transitions between stages along each dimension of the i -state space appear in well-defined locations, so that analyses can operate on them directly. In particular, it permits straightforward calculation of the sensitivity of any demographic output of the model to changes in any parameters.

The hyperstate model framework provides step-by-step construction and analysis for any number of dimensions. Naturally, as the number of dimensions increases, the amount of data required also increases. The number of transitions rates that must be specified increases exponentially with the number of dimensions. A model with s stages requires a matrix with s^2 entries. A two-dimensional model with s stages and g groups requires gs^2 entries in matrices describing stage transitions for each group, and sg^2 entries for matrices describing group transitions within each stage. A three-dimensional model with s , g and h stages along the three dimensions requires $ghs^2 + shg^2 + sgh^2$ entries and so on.

This is to be expected. The claim that the demography of a species depends on say, age, stage and frailty carries

the consequence that stage transitions may depend on age and frailty, while age transitions depend on stage and frailty, and frailty transitions depend on age and stage. If these three dimensions are truly important, so that their interactions must be included in the model, it should come as no surprise that the model requires data on those interactions.

In some special cases, the data requirements for hyperstate models can be reduced. The structure of the matrices describing transitions along certain dimensions may be constrained (e.g. age-specific matrices have positive entries only on the subdiagonal and the first row). In some cases, demographic information is available as a function of an environmental state (e.g. weather conditions or seasons). That state can be included as an additional dimension in a hyperstate model; if the environment affects the individuals but not the reverse, then only a single environment transition matrix must be estimated. Such extensions have been used to study stage-classified models in varying environments (Caswell 2009).

A key step in writing hyperstate models as a product of block-diagonal matrices is specifying the sequence of operation of the processes along each dimension of the i -state space. The order in which the block-diagonal matrices are multiplied in the formula (27) affects the resulting projection matrix $\tilde{\mathbf{A}}$ and thus any demographic output of the model. Sometimes an order is naturally imposed by seasonal variation: plants grow to maturity, then flower and then produce seeds, etc. Sometimes there is not an imposed order. For example, in a spatial size-structured model, the dispersal event can happen anytime in the time step. In this case, the order has to be chosen and the transition rates have to be estimated in accordance with this choice. The sequence can be particularly important in nonlinear models. For example, *seasonal compensation* (Boyce, Sinclair & White 1999), modifies the effects of mortality depending on whether it occurs before or after the operation of seasonal density dependence. Likewise, in a stage-structured host–parasite model, the sequence in which the mortality factors operate in the host’s life cycle modifies the stability properties of the system (Wang & Gutierrez 1980; May *et al.*, 1981).

We note parenthetically here that hyperstate models can also be considered in the context of tensor mathematics, which is designed to handle spaces of multiple dimensions. However, the conceptual framework and the analytical methods for the application of tensor methods to population models have yet to be developed.

The generalisation of the vec-permutation matrix (Magnus & Neudecker 1979) to higher dimensions involves significant technical difficulties. Those results have their own mathematical importance and are detailed in the Appendix S2. However, in spite of the technicalities underlying these models, implementation is straightforward. In Appendix S1, we provide the detailed description of the MATLAB functions needed to run the model. Also, the MATLAB code of the example described in Section ‘An

example: stage, age, and frailty' is given in the supplementary material; its adaptation to a different scenario would be straightforward.

Acknowledgements

This research was supported by Advanced Grant 322989 from the European Research Commission and NSF Grant DEB-1257545. The authors thank two anonymous referees for their useful comments on an earlier draft of this article.

Data accessibility

All of the MATLAB code for building the hyperstate matrix model and the example are provided in the Additional Supporting Information.

References

- Altenberg, L. (2012) Resolvent positive linear operators exhibit the reduction phenomenon. *Proceedings of the National Academy of Sciences of the United States of America*, **109**, 3705–3710.
- Andersen, P.K., Borgan, O., Gill, R.D. & Keiding, N. (1993) *Statistical Models Based on Counting Processes*. Springer-Verlag, New York, NY, USA.
- Boyce, M.S., Sinclair, A.R.E. & White, G.C. (1999) Seasonal compensation of predation and harvesting. *Oikos*, **87**, 419–426.
- Burns, C.E. & Grear, J.S. (2008) Effects of habitat loss on populations of white-footed mice: testing matrix model predictions with landscape-scale perturbation experiments. *Landscape Ecology*, **23**, 817–831.
- Caswell, H. (2001) *Matrix Population Models*, 2nd edn. Sinauer Associates, Sunderland, MA, USA.
- Caswell, H. (2006) *Applications of Markov chains in demography*. MAM2006: Markov Anniversary Meeting, pp. 319–334. Bosen Books, Raleigh, NC, USA.
- Caswell, H. (2007) Sensitivity analysis of transient population dynamics. *Ecology Letters*, **10**, 1–15.
- Caswell, H. (2008) Perturbation analysis of nonlinear matrix population models. *Demographic Research*, **18**, 59–116.
- Caswell, H. (2009) Stage, age and individual stochasticity in demography. *Oikos*, **118**, 1763–1782.
- Caswell, H. (2011) Beyond R_0 : demographic models for variability of lifetime reproductive output. *PLoS ONE*, **6**, e20809.
- Caswell, H. (2012) Matrix models and sensitivity analysis of populations classified by age and stage: a vec-permutation matrix approach. *Theoretical Ecology*, **5**, 403–417.
- Caswell, H. (2014a) Statistics of inter-individual variation in lifetime fertility: a Markov chain approach. Paper presented at Population Association of America Annual Meeting, 2014.
- Caswell, H. (2014b) A matrix approach to the statistics of longevity in heterogeneous frailty models. *Demographic Research*, **31**, 553–592.
- Caswell, H. & Salguero-Gómez, R. (2013) Age, stage, and senescence in plants. *Journal of Ecology*, **101**, 585–595.
- Caswell, H. & Shyu, E. (2012) Sensitivity analysis of periodic matrix population models. *Theoretical Population Biology*, **82**, 329–339.
- Dostal, P. (2007) Population dynamics of annuals in perennial grassland controlled by ants and environmental stochasticity. *Journal of Vegetation Science*, **18**, 91–102.
- Ezard, T.H.G., Bullock, J.M., Dalglish, H.J., Millon, A., Pelletier, F., Ozgul, A. & Koons, D.N. (2010) Matrix models for a changeable world: the importance of transient dynamics in population management. *Journal of Applied Ecology*, **47**, 515–523.
- Flockhart, D.T., Pichancourt, J.B., Norris, D.R. & Martin, T.G. (2015) Unravelling the annual cycle in a migratory animal: breeding-season habitat loss drives population declines of monarch butterflies. *Journal of Animal Ecology*, **84**, 155–165.
- Goldberg, E.E., Lynch, H.J., Neubert, M.G. & Fagan, W.F. (2010) Effects of branching spatial structure and life history on the asymptotic growth rate of a population. *Theoretical Ecology*, **3**, 137–152.
- Henderson, H.V. & Searle, S.R. (1981) The vec-permutation matrix, the vec operator and Kronecker products: a review. *Linear and Multilinear Algebra*, **9**, 271–288.
- Hernandez-Suarez, C., Rabinovich, J. & Hernandez, K. (2012) The long-run distribution of births across environments under environmental stochasticity and its use in the calculation of unconditional life-history parameters. *Theoretical Population Biology*, **82**, 264–274.
- Hunter, C.M. & Caswell, H. (2005) The use of the vec-permutation matrix in spatial matrix population models. *Ecological Modelling*, **188**, 15–21.
- Klepac, P. & Caswell, H. (2011) The stage-structured epidemic: linking disease and demography with a multi-state matrix approach model. *Theoretical Ecology*, **4**, 301–319.
- Land, K.C. & Rogers, A. (1982) *Multidimensional Mathematical Demography*. Academic Press, New York, NY, USA.
- Lebreton, J.D. (1996) Demographic models for subdivided populations: the renewal equation approach. *Theoretical Population Biology*, **49**, 291–313.
- Li, C.K. & Schreiber, S.J. (2006) On dispersal and population growth for multi-state matrix models. *Linear Algebra and its Applications*, **418**, 900–912.
- Magnus, J.R. & Neudecker, H. (1979) The commutation matrix: some properties and applications. *The Annals of Statistics*, **7**, 381–394.
- Magnus, J.R. & Neudecker, H. (1985) Matrix differential calculus with applications to simple, hadamard, and kronecker products. *Journal of Mathematical Psychology*, **29**, 474–492.
- Magnus, J.R. & Neudecker, H. (1999) *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., Chichester, UK.
- May, R.M., Hassell, M.P., Anderson, R.M. & Tonkyn, D.W. (1981) Density dependence in host-parasitoid models. *Journal of Animal Ecology*, **50**, 855–865.
- Metcalf, C.J.E., Lessler, J., Klepac, P., Morice, A., Grenfell, B.T. & Bjørnstad, O.N. (2012) Structured models of infectious disease: inference with discrete data. *Theoretical Population Biology*, **82**, 275–282.
- Metz, J.A.J. & Diekmann, O. (1986) *The Dynamics of Physiologically Structured Populations*. *Lecture Notes in Biomathematics*. Springer-Verlag, Berlin, Germany.
- Ozgul, A., Oli, M.K., Armitage, K.B., Blumstein, D.T. & Van Vuren, D.H. (2009) Influence of local demography on asymptotic and transient dynamics of a Yellow-Bellied Marmot metapopulation. *American Naturalist*, **173**, 517–530.
- Pascarella, J.B. & Horvitz, C.C. (1998) Hurricane disturbance and the population dynamics of a tropical understory shrub: megamatrix elasticity analysis. *Ecology*, **79**, 547–563.
- Perez-Heydrich, C., Oli, M.K. & Brown, M.B. (2012) Population-level influence of a recurring disease on a long-lived wildlife host. *Oikos*, **121**, 377–388.
- Rogers, A. (1967) Matrix analysis of interregional population growth and Distribution. *Papers of the Regional Science Association*, **18**, 177–196.
- Rogers, A. (1975) *Introduction to Multiregional Mathematical Demography*. Wiley, New York, NY, USA.
- Strasser, C.A., Neubert, M.G., Caswell, H. & Hunter, C.M. (2012) Contributions of high- and low-quality patches to a metapopulation with stochastic disturbance. *Theoretical Ecology*, **5**, 167–179.
- Tinker, M.T., Doak, D.F. & Estes, J.A. (2008) Using demography and movement behavior to predict range expansion of the southern sea otter. *Ecological Applications*, **18**, 1781–1794.
- Tuljapurkar, S., Horvitz, C.C. & Pascarella, J. (2003) The many growth rates and elasticities of populations in random environments. *American Naturalist*, **162**, 489–502.
- Vaupel, J.W. & Yashin, A.I. (2006) Unobserved population heterogeneity. *Demography: Analysis and Synthesis: A Treatise in Population Studies*, **1**, 271–278.
- Vaupel, J.W., Manton, K.G. & Stallard, E. (1979) The impact of heterogeneity in individual frailty on the dynamics of mortality. *Demography*, **16**, 439–454.
- Wang, Y.H. & Gutierrez, A.P. (1980) An assessment of the use of stability analyses in population ecology. *Journal of Animal Ecology*, **49**, 435–452.
- Warden, M.L., Haas, H.L., Rose, K.A. & Richards, P.M. (2015) A spatially explicit population model of simulated fisheries impact on loggerhead sea turtles (*Caretta caretta*) in the Northwest Atlantic Ocean. *Ecological Modelling*, **299**, 23–39.
- Wienke, A. (2011) *Frailty Models in Survival Analysis*. CRC Press, Boca Raton, FL, USA.
- Willekens, F. (2014) *Multistate Analysis of Life Histories with R*. Springer, New York, NY, USA.
- Yashin, A.I., Iachine, I.A. & Begun, A.S. (2000) Mortality modeling: a review. *Mathematical Population Studies*, **8**, 305–332.
- Yule, K.M., Miller, T.E.X. & Rudgers, J.A. (2013) Costs, benefits, and loss of vertically transmitted symbionts affect host population dynamics. *Oikos*, **122**, 1512–1520.

Received 21 April 2016; accepted 19 July 2016
Handling Editor: Mark Rees

Supporting Information

Additional Supporting Information may be found online in the supporting information tab for this article:

Appendix S1. Implementing hyperstate matrix models in MATLAB.

Appendix S2. Mathematical appendix.

Data s1 Matlab files. We provide the MATLAB files to carry out the functions described in Appendix S1 and the Example presented in the main text.