

University of St Andrews



Full metadata for this thesis is available in
St Andrews Research Repository
at:

<http://research-repository.st-andrews.ac.uk/>

This thesis is protected by original copyright

Unsigned - Contacted
Matters re omission 4/2/04
Supervisor left Uni
Co-Supervisor - Prof Buckland
25/2/04 Signed.

Multi-dimensional Modelling of
Physiologically and Temporally Structured
Populations

Camilla Elizabeth Dixon



Thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
in the School of Mathematical and Statistics
UNIVERSITY OF ST ANDREWS

August 2003



Th
E544

Declarations

1. I, Camilla Elizabeth Dixon, hereby certify that this thesis, which is approximately 43 500 words in length, has been written by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a higher degree.

date signature of candidate

2. I was admitted as a research student in October 1999 and as a candidate for the degree of PhD in October 1999; the higher study for which this is a record was carried out in the University of St Andrews between 1999 and 2003.

date signature of candidate

3. I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

date 25/2/04 signature of supervisor ...

4. In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker.

date signature of candidate

Abstract

Physiologically-structured population counts are sometimes the only available source of information about a population. Because such data are often sparse and noisy, they are difficult to model. Moreover the parameters of interest may be underlying demographic rates rather than population abundance. In this thesis, the possibility of using smoothing splines, demographic processes and environmental variables to improve estimates of birth and death rates from physiologically-structured population counts is explored.

A smooth physiologically-structured population model is proposed which makes appropriate use of demographic processes and allows explicit space-time dependence in estimated quantities (birth and death rates). A model fitting process is also defined. The model is tested using simulated data and is applied to Dover sole (*Solea solea*) data from the Bristol Channel. It can be applied when the life-history stages are defined by age. The method presented avoids mis-specification bias in birth and death rate estimates and improves their precision. It allows study of the relationship between vital rate estimates and environmental variables and, when coded, is straightforward to apply.

The more general inverse problem of obtaining birth, death and growth rates from stage abundances is also examined, when the physiological trait distinguishing stages need not be age. It is proven mathematically that unique solutions do not exist unless the trait is age, though limits for the growth and hence birth and death rates do exist when the growth rate is independent of the physiological trait. Simulations are used to demonstrate that plausible estimates of the 'true' birth, death and growth rates, cannot be identified in practice.

To overcome this difficulty, stage-age experiment data may be incorporated into physiologically-structured population models. A new method is discussed for doing this in a statistically justified manner.

Acknowledgements

I would like to thank my supervisor, Simon Wood, for his immeasurable help and patience throughout the preparation of this thesis. I have greatly appreciated his useful comments, technical aid, and assistance with clarification of ideas and language. I would also like to thank the research staff at the Centre for Research into Ecological and Environmental Modelling (CREEM) at the University of St Andrews, particularly David Borchers, Liz Clarke, Sharon Hedley, Mike Lonergan, Charles Paxton and Fiona Underwood for sharing their statistical, computational and linguistic skills. I am especially grateful to Sharon for her helpful criticisms and to both Fiona and Sharon for their support.

Thanks are also due to Joe Horwood at the Centre for Environment, Fisheries and Aquaculture, Lowestoft, for the Dover sole data used in chapter 3.

Financial support provided by the Environmental and Physical Sciences Research Council (EPSRC) is gratefully acknowledged.

Finally, I would like to thank Robin Chester, my brothers and parents for their invaluable encouragement and moral support.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research aim	4
1.3	Thesis organization	4
2	A Smooth Physiologically-Structured Population Model	5
2.1	Introduction	5
2.2	Smooth physiologically-structured population model	7
2.2.1	Model Summary	15
2.3	Fitting the SPSP model	17
2.3.1	The Objective Function	17
2.3.2	Quadratic approximation of the objective function	26

2.3.3	Model constraints	29
2.3.4	Obtaining model parameters for the true objective function	35
2.3.5	Starting and stopping the model fitting process	39
2.3.6	Summary of model fitting	41
2.3.7	Variance estimation	44
2.4	Discussion	46
3	Application of a Smooth Physiologically-Structured Population Model	49
3.1	Introduction	49
3.2	Simulation Testing	50
3.2.1	Simulated data	50
3.2.2	Models fitted to simulated data	61
3.3	Application to Dover Sole (<i>Solea solea</i>) in the Bristol Channel	76
3.3.1	The sole data	79
3.3.2	The fitted model	85
3.4	Discussion	90
4	The Inverse Problem for Size-Structured Data	95

4.1	Introduction	95
4.2	The inverse problem for physiologically-structured population data	97
4.2.1	A general model developed for physiologically-structured population counts	99
4.2.2	A method of solving the inverse problem for birth and death rates	105
4.3	Extension of the method to growth rates	106
4.4	Non-uniqueness of growth, birth and death rates	108
4.4.1	Mathematical demonstration	108
4.4.2	Limits on the growth rate	110
4.4.3	Avoiding the non-uniqueness problem using assumptions	112
4.4.4	Demonstration using simulated data	115
4.5	Discussion	121
5	A New Daily Egg Production Method	126
5.1	Introduction	126
5.2	The new DEPM model	128
5.2.1	Stage at age model for incubation experiment data	129
5.2.2	Ageing the physiologically-structured population	131

5.2.3	From stage abundances to cohort abundances	133
5.2.4	Modelling cohort abundances to obtain birth and death rates	136
5.2.5	Variance estimation	139
5.2.6	Model assumptions	141
5.2.7	Model summary	144
5.3	Results from application of the new DEPM model	148
5.4	Discussion	151
6	Discussion	154
	Bibliography	159
A	Notation	168
A.1	Notation for chapters 2 and 3	168
A.2	Notation for chapter 4	176
A.3	Notation for chapter 5	179
B	Fitting the SPSP Model for Three Covariates	183
B.1	Model components for $d = 3$	183
B.2	Jacobian matrix for $d = 3$	186

B.3	Wiggleness penalty matrix for $d = 3$	189
C	Programs used to Fit an SPSP Model to Stage Count Data	190
C.1	fitmodel.c	190
C.2	Programs used in the application of the SPSP model	220
D	Results of the application of the SPSP model	223
E	Simulating Data Using the McKendrick von Foerster Equation	227

List of Tables

3.1	Realized coverage probabilities for the models fitted to normal data with 10% noise.	63
3.2	Sample periods and times for Dover sole data.	82
A.1	Notation used in chapter 2.	175
A.2	Notation used in chapter 4.	179
A.3	Notation used in chapter 5.	182
D.1	Results of simulation study for the normal data with 10% CV.	226

List of Figures

2.1	An example of physiologically-structured population counts through time. . .	6
2.2	Development of a cohort of fish eggs spawned at x	8
2.3	Model structure for one set of environmental conditions.	11
2.4	Obtaining model parameters subject to inequality constraints using quadratic programming.	34
2.5	Illustration of the first steps of the golden section search for model parameters.	38
3.1	Sample points and knots for simulated data.	51
3.2	Stage age limits for simulated data.	53
3.3	Birth rate for simulated data.	54
3.4	Death rate for simulated data.	55
3.5	Stage populations with sampled counts for simulated data.	56

3.6	Stage counts for simulated normal data with 10% noise.	58
3.7	Stage counts for simulated normal data with 40% noise.	59
3.8	Stage counts for simulated Poisson data.	60
3.9	Model fit for simulation study data with no random errors.	61
3.10	Histograms and confidence intervals for the simulated normal data with 10% noise.	64
3.11	Model fit for the simulation study for normal data set 96 with 10% noise. . .	66
3.12	Estimated birth rate for the model fitted to normal data set 96 with 10% noise.	67
3.13	Estimated death rate for the model fitted to normal data set 96 with 10% noise	68
3.14	Model fit for the simulation study for normal data set 1 with 40% noise. . . .	69
3.15	Estimated birth rate for the model fitted to normal data set 1 with 40% noise.	70
3.16	Estimated death rate for the model fitted to Poisson data set 1.	71
3.17	Model fit for the simulation study for Poisson data set 1.	73
3.18	Estimated birth rate for the model fitted to Poisson data set 1.	74
3.19	Estimated death rate for the model fitted to Poisson data set 1.	75
3.20	Stage abundances per square metre for the sole egg data.	77
3.21	Original sample points and knots for the sole egg data.	80

3.22	Stage age limits for the sole egg data.	81
3.23	Sample points and knots for the sole egg data after scaling.	84
3.24	Model fit for the sole egg data.	86
3.25	Log of 1 plus the estimated stage populations for the model fitted to the sole egg data.	87
3.26	Log of 1 plus the estimated birth rate for the model fitted to the sole egg data.	88
3.27	Estimated death rate for the model fitted to the sole egg data.	89
4.1	An example of physiologically-structured population counts through time. . .	98
4.2	A schematic illustration of one stage of the physiologically-structured popula- tion model in equation 4.1.	100
4.3	Schematic illustration of the time-invariant growth and death case and the time- and size-invariant growth and death case.	114
4.4	Growth rate, death rate, birth rate and population density for simulated pop- ulation data	117
4.5	Alternative growth, death and birth rates for the population density in figure 4.4.	120
5.1	An example of the micro-birth rate $\beta(t)$	131
5.2	An example of the pdf $p(a \mathbf{x}, t)$ of age.	132

5.3	An example of a micro-birth rate, corresponding observed cohort abundances and macro-birth rate.	137
5.4	An example of a micro-birth rate, corresponding observed cohort abundances and time dependent macro-birth rate.	143
5.5	Sample locations around the Iberian Peninsula for the survey of sardine eggs in 1999.	149
E.1	Approximation of the population density function using the upwind difference scheme	229
E.2	The midpoint method	230

Chapter 1

Introduction

1.1 Motivation

Depletion of fish stocks and the implications for fishing communities and the marine environment continue to cause concern. Globally, marine fisheries employ approximately 27 million people and produce around 85 million tonnes each year. In 2000, international trade in fish products was valued at US\$55 billion. The Food and Agricultural Organization of the United Nations (FAO) estimates that globally 47% of fish stocks are fully exploited, 18% are fished beyond maximum sustainable limits, and 10% are so depleted that the populations have been reduced to very low levels. (The above statistics can be found in FAO (2002).) Clearly there is a need for effective management of fish stocks if the fish populations are to continue to sustain an important economy, and maintain their ecological role (Shepherd 1993).

The state of fisheries is just one problem for which ecologists, economists and politicians require reliable information about populations in order to make good management decisions.

In another example, the knowledge gained from a survey of locust egg beds in Afghanistan in 2002 enabled the FAO to co-ordinate a program which successfully saved the 2003 wheat crop from a severe locust threat (FAO News, 4th August 2003). To produce reliable information for decision makers, statistical models of available data must be of high calibre. A prerequisite of credible statistical modelling and analysis is good quality data. Lack of quality in either data or statistical inference may lead to a lack of confidence in the resulting (possibly poor) advice. This has been a political issue for the fisheries industry that increasingly sophisticated statistical techniques have as yet been unable to redress (EUROSTAT 1995).

It is often difficult and expensive to collect appropriate data, a situation which further argues for substantial effort in finding good models for statistical inference and prediction. Physiologically-structured population counts are a type of data that are relatively easy to observe. These counts can be collected for the wide range of populations in which individuals pass through a number of physiological stages before death or maturation. There are many statistical models for dealing with physiologically-structured count data (Tuljapurkar & Caswell 1997, Manly 1990), but such data are often sparse and noisy making the task of modelling a challenge.

Smoothing functions, and in particular smoothing splines, are useful tools for dealing with such data since they may be fitted even when data sampling is irregular. They do not impose a rigid structure on the function being modelled (which may lead to mis-specification bias), but instead simply assume that the function is smooth. Unlike non-parametric smoothing models which also waive strong structural assumptions, smoothing splines are easily understood as the solution of a clearly posed minimization problem. In addition, the complexity (wiggleness) of a smoothing spline can be chosen in a statistically justified manner (Craven & Wahba

1979, Wahba 1990) and computational tools exist to make this selection efficiently for a broad range of model forms (Wood 2000). Spline theory is well established in the literature; standard texts are Silverman (1985), Wahba (1990), Green & Silverman (1994). Clearly smoothing splines (and their multi-dimensional cousins, thin plate splines), have potential for application to sparse, noisy, physiologically-structured population data and in fact have been applied with success (see for example Wood & Horwood (1995)).

Often, birth and death rates are required from the data, rather than simply population density or size. Obtaining these underlying functions from the population density is called the ‘inverse problem’ (for example, Wood (1997)) because a population model is used to infer the birth and death rates rather than to calculate population dynamics from known rates. Solutions to inverse problems make use of the relationship of birth, death and possibly growth rates to population density. Incorporating these demographic processes into models of physiologically-structured population count data will allow birth and death rates to be estimated in a biologically defensible manner.

Using knowledge of demographic and possibly physiological processes should also increase the precision of predictions from population models. Precision can also be improved by including in population models environmental variables such as temperature or spatial location that are thought to account for heterogeneity in the data (see Borchers, Buckland, Priede & Ahmadi (1997) and Augustin, Borchers, Clarke, Buckland & Walsh (1998) for examples). In addition, these models can give insight into the relationship between environmental attributes and population density.

1.2 Research aim

This thesis ties together the potential sources of improvement for population models outlined above. The aim is to investigate whether smoothing splines, simple demographic processes and environmental variables can be used to obtain improved birth and death estimates from physiologically-structured population counts.

1.3 Thesis organization

To achieve this aim, a model for solving the inverse problem is defined in chapter 2, along with a process for fitting the model to physiologically-structured count data. The model incorporates thin plate splines of environmental variables using a basic demographic process to define its non-linear structure. It is applicable when the physiological stages are defined by age, the age limits of the stages are known for the data space, and the death rate is independent of age. The model is then tested with simulated data and applied to Dover sole (*Solea solea*) egg data from the Bristol Channel in chapter 3.

Chapter 4 addresses the more general inverse problem in which the physiological trait distinguishing the stages is not necessarily age, and growth and death rates are allowed to vary with the physiological trait.

Sometimes, individuals are not assigned to life-history stages according to age, but age is the trait used to define the stages when modelling. Chapter 5 presents a method of dealing more honestly with the relationship between stage and age.

Finally, chapter 6 draws together the work presented and summarizes the main results.

Chapter 2

A Smooth

Physiologically-Structured

Population Model

2.1 Introduction

Chapter 1 identified the need for good models for making effective policies when dealing with populations. Unfortunately, many models actually used as a basis for decisions perform poorly, partially due to the quality of the data. Stage-structured population counts are available in many situations, and should provide useful information if they can be modelled well.

In fisheries, one difficulty with estimating the population size directly from the adult fish

arises from their mobility. In contrast, fish eggs are stationary, and egg data could provide information about the adult spawning stock. Egg production methods have been developed to perform this task. The purpose of an egg survey is to obtain the birth rate, but only egg counts are available. These counts are difficult to turn into the birth rate because the mortality rate is unknown, but is expected to be high. As they develop, fish eggs pass through a series of life-history stages for which counts can be observed, and these physiologically-structured data offer the potential to estimate the birth and death rates simultaneously. This has been successfully done before for spatially homogenous closed populations, but not for spatially varying populations.

Sometimes stage counts are recorded for several sets of environmental conditions (these could include geographical locations and temperatures, for example), and the recordings repeated through time. This is illustrated for one set of environmental conditions in figure 2.1.

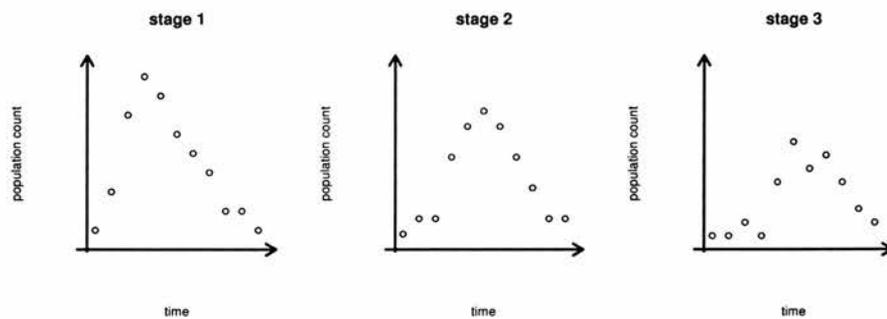


Figure 2.1: *An example of physiologically-structured population counts through time.*

Unfortunately, surveyed physiologically-structured population counts are often noisy, and contain many zeros. In particular this is the case for zooplankton and fish egg data. These problems make it difficult to obtain accurate and unbiased estimates of population density, birth and death rates, a challenge which is taken up in this chapter with the aim of defining

a good model which may be used to solve the inverse problem for physiologically-structured population count data. The definition will include a model fitting process.

But what makes a ‘good’ model? A good model should attempt to incorporate all of the data, that is, all developmental stages; should incorporate known demographic structure (tightly specified models leading to lower estimation variances); should explain heterogeneity in the data (e.g. in relation to environmental variables and time); should not impose convenient mathematical forms which have no biological basis; should not be intractable - the model should be fairly straightforward to implement and use.

A model is therefore proposed in section 2.2 in which the birth and death rates are represented using thin plate splines, which are functions of environmental variables and time. Demography then dictates a non-linear structure relating birth rates and death rates to populations in developmental classes. The resulting model is termed the smooth physiologically-structured population model. It is fitted to structured population counts using an approximating penalized linear regression problem as explained in section 2.3, which enables the model and its multiple smoothing parameters to be estimated fairly efficiently.

2.2 Smooth physiologically-structured population model¹

The smooth physiologically-structured population (SPSP) model is developed from a recognition of the demographic mechanisms at work:

Consider a cohort consisting of all individuals in a closed population born in environmen-

¹Notation for this section is given in table A.1 of appendix A. A summary of the SPSP model is also provided in section 2.2.1 for reference.

tal conditions $\{x_1, x_2, \dots, x_{d-1}\}$ at time x_d . The size of the cohort is initially equal to the birth rate $\beta(\mathbf{x})$ at that time in those conditions, $\mathbf{x} \equiv \{x_1, x_2, \dots, x_d\}$. Suppose that an individual's development can be divided into a series of identifiable stages, labelled $1, \dots, C$, with the transition between stages i and $i + 1$ occurring at age α_i , and that a proportion $\mu(x_1, x_2, \dots, x_{d-1}, x_d + a)$ per unit time die at time $x_d + a$ (where a is the cohort's current age). Note that the form of the death rate assumes that the environmental conditions x_1, x_2, \dots, x_{d-1} are time invariant; this can be overridden if the form of the time-dependence is known. For example, if x_2 is time dependent, replace $\mu(x_1, x_2, \dots, x_{d-1}, x_d + a)$ with $\mu(x_1, x_2(x_d + a), \dots, x_{d-1}, x_d + a)$. We do not consider what happens to the surviving individuals when they reach the maximum age α_C in the final (C th) stage at time $x_d + \alpha_C$ (it could be that in practice no individuals survive this long).

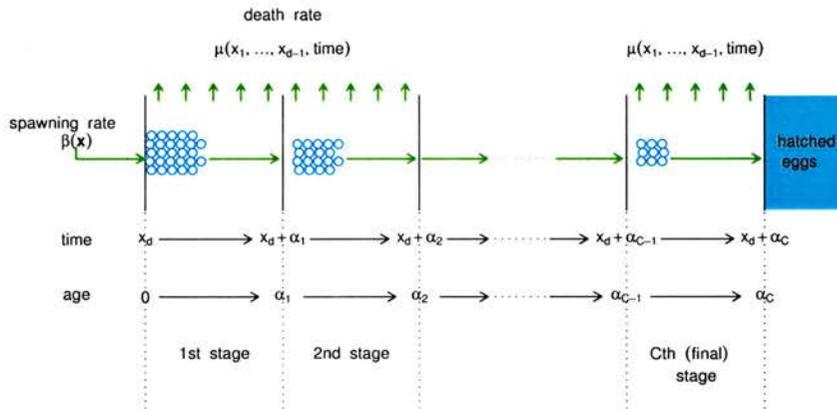


Figure 2.2: Development of a cohort of fish eggs spawned at \mathbf{x} .

Figure 2.2 illustrates the progress of such a cohort born in conditions \mathbf{x} . The example is of fish eggs which progress through C developmental (age) stages before hatching at age

α_C . All individuals born at the same time and in the same conditions are regarded as identical, that is, subject to the same death rate and progressing to the next developmental stage at the same age. Ignoring variation between individuals in this way will lead to an entirely deterministic model, which is reasonable for large populations over short timescales, especially when sampling error is large and can therefore be treated as the dominant source of stochasticity.

To produce a workable model, a number of assumptions are made, and these must be justified for the data set to which the SPSP model is applied. We assume that the population at any point can be treated as closed, i.e. that it is not subject to migration or advection, so that the population density at a sample point is only changed by the birth and death rates. In practice, it may rather be the case that migration and advection are small relative to the birth and death rates. Once born, the individuals in the population can only decrease in number at a rate equal to the death rate. So for $\eta(\mathbf{x}, a)$ individuals of age a at time x_d and in environmental conditions $(x_1, x_2, \dots, x_{d-1})$, the rate of decline of the population η with respect to time is equal to the total death rate at the time instant of interest:

$$\frac{\partial \eta(\mathbf{x}, a)}{\partial x_d} = -\mu(\mathbf{x})\eta(\mathbf{x}, a)$$

where $\mu(\mathbf{x})$ is the per capita death rate which is assumed to be the same for all ages at \mathbf{x} . This assumption must be reasonable for the species to which this SPSP model is applied, for example, the assumption is acceptable for fish eggs where the main causes of death are predation by non-visual predators or environmental effects. However the assumption may not hold when the main source of mortality is from visual predators since pelagic fish eggs

become less transparent as they develop (Fox, Bunn & Webb 2000). Solving for η we get

$$\eta(\mathbf{x}, a) = \eta(x_1, x_2, \dots, x_{d-1}, x_d - a, 0) \exp \left(- \int_{x_d-a}^{x_d} \mu(x_1, x_2, \dots, x_{d-1}, z) \partial z \right) .$$

Individuals aged a at time x_d were born (aged 0) at time $x_d - a$, so $\eta(x_1, x_2, \dots, x_{d-1}, x_d - a, 0) = \beta(x_1, x_2, \dots, x_{d-1}, x_d - a)$, and we can write

$$\eta(\mathbf{x}, a) = \beta(x_1, x_2, \dots, x_{d-1}, x_d - a) \exp \left(- \int_{x_d-a}^{x_d} \mu(x_1, x_2, \dots, x_{d-1}, z) \partial z \right) \quad (2.1)$$

that is, the population density at \mathbf{x} aged a is the number born in conditions $(x_1, x_2, \dots, x_{d-1})$ at time $x_d - a$, multiplied by their survival from birth till time x_d .

To obtain the stage abundance f_j for physiological stage j in \mathbf{x} , we simply integrate $\eta(\mathbf{x}, a)$ over the age range for that stage:

$$f_j(\mathbf{x}) = \int_{\alpha_{j-1}(\mathbf{x})}^{\alpha_j(\mathbf{x})} \eta(\mathbf{x}, a) \partial a . \quad (2.2)$$

Since age limits $(\alpha_{j-1}(\mathbf{x}), \alpha_j(\mathbf{x}))$ for the physiological stages ($j = 1, \dots, C$) are usually only available at sample times, the form of these functions is often unknown. This does not pose a problem for model fitting, since the age limits need only be known at the sample points. The form of the age limits as functions of explanatory variables is only required for predicting stage abundances between sample times. In this situation, the age limits can be considered to be smooth functions of the environmental conditions and time, constrained to ensure that $\frac{\partial a}{\partial x_d} = 1$.

Figure 2.3 illustrates the SPSP model structure for a particular set of environmental conditions, $(x_1, x_2, \dots, x_{d-1})$. The stage age limits are represented by curves (since they may change over time) approximately parallel to the time-axis. A diagonal line (since all individuals age at the rate of one day per day) follows the life history of a particular section

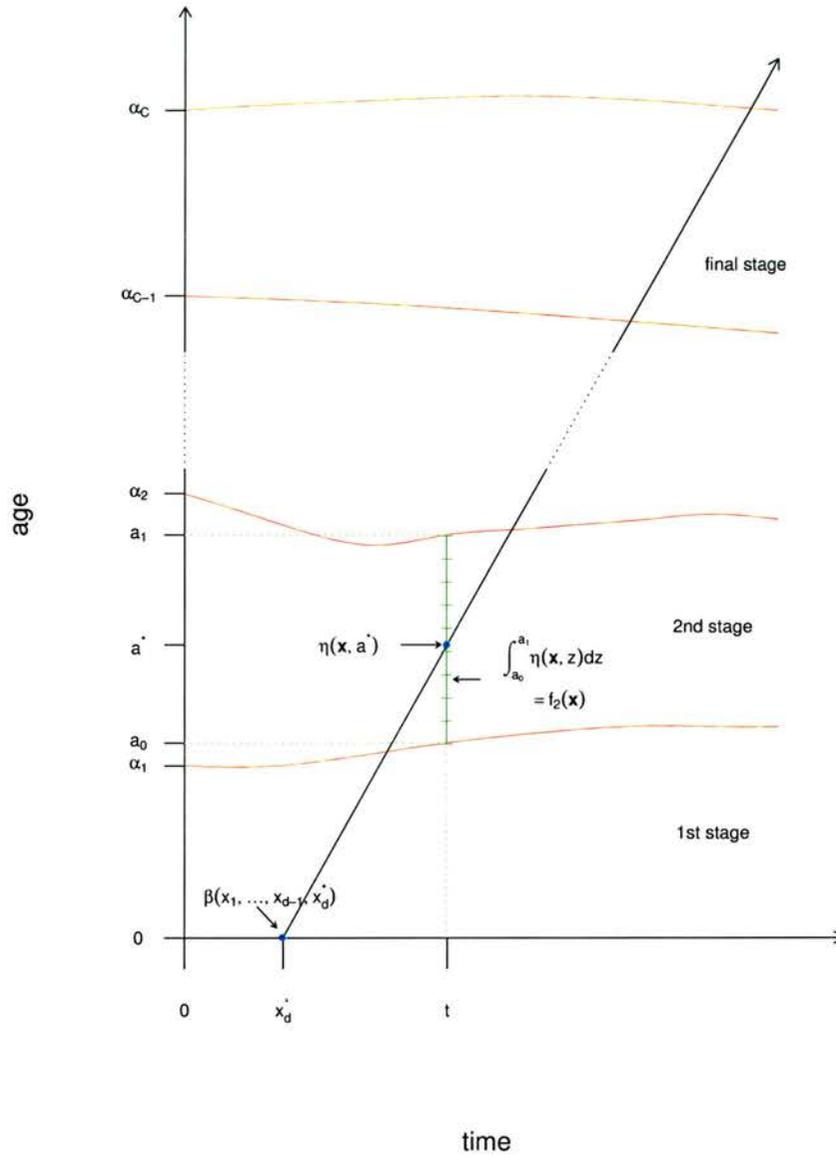


Figure 2.3: Model structure for the environmental conditions $(x_1, x_2, \dots, x_{d-1})$. The population density per unit age interval over time and age would be the ‘height’ of a surface in the direction perpendicular to the page. The wavy lines approximately parallel to the time axis are the age limits of the physiological stages. The progress of the cohort born at time x_d^* is shown by the diagonal line. The stage abundance for the second stage at time x_d is the integral of the population densities over the hatched line.

of the population, born at time x_d^* . The number of individuals per unit age interval in this cohort would be the ‘height’ of this diagonal line, extending perpendicularly upwards from the page. To obtain the population abundance $f_2(\mathbf{x})$ for the second developmental stage, the population density η at time x_d would be integrated over the hatched line from a_0 to a_1 .

Because the way in which the birth rate β depends on the environmental variables and time is usually unknown, it is sensible to assume that β is merely a smooth function of \mathbf{x} . This smooth function can be represented using a thin plate spline like basis. Suppose there are $d - 1$ environmental variables. Then

$$\begin{aligned}\beta(\mathbf{x}) &= \beta(x_1, \dots, x_{d-1}, x_d) \\ &= \theta \sum_{i=1}^K \gamma_i \xi(\|\mathbf{x} - \mathbf{k}_i\|) + \sum_{j=1}^M b_j \phi_j(\mathbf{x}).\end{aligned}\tag{2.3}$$

In practice, the elements of \mathbf{x} will usually be linearly scaled to have similar ranges. $\mathbf{k}_i = (k_{1i}, \dots, k_{di})$ for $i = 1, \dots, K$ are points called ‘knots’ chosen as part of the model specification to be spread ‘nicely’ over the survey domain. The first number in the subscript corresponds to the dimension (the environmental variable), with k_{di} corresponding to the i th knot in the time dimension. γ_i , $i = 1, \dots, K$ and b_j , $j = 1, \dots, M$ are model parameters to be estimated. The γ_i parameters correspond to a radial basis function (ξ) about each knot, and the b_j parameters correspond to linearly independent polynomial terms (ϕ_j). $\|\cdot\|$ is the usual Euclidean norm.

The radial basis function ξ has the form

$$\xi(\|\mathbf{x} - \mathbf{k}_i\|) = \begin{cases} \|\mathbf{x} - \mathbf{k}_i\|^{2m-d} \log(\|\mathbf{x} - \mathbf{k}_i\|) & \text{if } d \text{ is even} \\ \|\mathbf{x} - \mathbf{k}_i\|^{2m-d} & \text{if } d \text{ is odd} \end{cases}$$

where m is an integer satisfying $2m > d + 1$, to ensure a ‘visibly smooth function’ (Green & Silverman 1994, page 161). For $M = \binom{m+d-1}{d}$, $\{\phi_1 \dots \phi_M\}$ are linearly independent polynomials spanning the null space of the penalty terms given in section 2.3.1, that is, the M -dimensional space of polynomials in \mathbb{R}^d of total degree less than m . θ , the constant of proportionality, is given by

$$\theta = \begin{cases} \frac{(-1)^{m+1+\frac{d}{2}} 2^{1-2m}}{\pi^{\frac{d}{2}} (m-1)! (m-\frac{d}{2})!} & \text{if } d \text{ is even} \\ \frac{\Gamma(\frac{d}{2}-m) 2^{-2m}}{\pi^{\frac{d}{2}} (m-1)!} & \text{if } d \text{ is odd.} \end{cases}$$

This is adapted from Green & Silverman (1994, chapter 7) where more conventional thin plate splines are described in which the knots are the observed covariates \mathbf{x}_i , $i \in \{1, 2, \dots, n\}$ for n samples.

The benefit of using splines is that we don’t impose an overly restrictive parametric form on the form of the birth rate, but a smooth function is chosen according to the shape of the data. This smooth function should provide a good description of the birth rate implied by the data but should not reproduce excessive variation (wiggles) that can really be attributed to noise in the data. This is achieved by fitting the SPSP model using a penalized regression approach explained section 2.3.

The death rate is also represented using a thin plate spline like basis of the environmental variables and time:

$$\mu(\mathbf{x}) = \theta \sum_{i=1}^K \delta_i \xi(\|\mathbf{x} - \mathbf{k}_i\|) + \sum_{j=1}^M d_j \phi_j(\mathbf{x}) \quad (2.4)$$

with model parameters δ_i , $i = 1, \dots, K$ and d_j , $j = 1, \dots, M$, and all other terms defined as for the birth rate. Note that the birth and death rates are linear in the model parameters.

When using thin plate splines to model the birth and death rates, we assume that these rates are smooth functions of the covariates \mathbf{x} . The covariates do not include a development index, such as age, so all individuals at \mathbf{x} are subject to the same death rate, regardless of their level of physiological development. The SPSP model should not be applied to populations for which this assumption is unreasonable.

The same number of knots (K) are used for the death rate and the birth rate. This does not need to be the case, but has been adopted here for notational and computational convenience. Limiting the number of knots will limit how wiggly a thin plate spline can be. However, providing there are sufficient knots for a potentially very wiggly function, the smoothness of a thin plate spline can be selected entirely by a smoothing parameter. Thus the birth and death rates need not have the same degree of smoothness despite having the same number of knots because separate smoothing parameters will be selected for each of them (see section 2.3.1). The benefit of reducing the number of knots is a reduction in the computation time.

The observed population abundances y_{ji} can be modelled as observations of random variables Y_{ji} such that

$$E(Y_{ji}) = f_{ji} \quad Y_{ji} \sim \text{an exponential family distribution, } i = 1, \dots, n, j = 1, \dots, C \quad (2.5)$$

where f_{ji} is the predicted population density (fitted value) of the observed population count y_{ji} for developmental stage j at sample point \mathbf{x}_i . From this assumption, the form of the mean-variance relationship of the Y_{ji} 's is identified and can be incorporated into the model fitting process, in a manner similar to that for generalized linear models, or generalized additive models using iteratively re-weighted least squares (see section 2.3.1).

2.2.1 Model Summary

Equations 2.1, 2.2, 2.3, 2.4 and 2.5 define the population dynamics model described in this chapter, summarized below. The form of the SPSP model for $d = 3$ is described in appendix B.1. Table A.1 in appendix A lists the notation used in this chapter.

Model Summary

The main equations describing the non-linear smooth physiologically-structured population model are:

$$\beta(\mathbf{x}) = \theta \sum_{i=1}^K \gamma_i \xi(\|\mathbf{x} - \mathbf{k}_i\|) + \sum_{j=1}^M b_j \phi_j(\mathbf{x}) \quad \text{birth rate}$$

$$\mu(\mathbf{x}) = \theta \sum_{i=1}^K \delta_i \xi(\|\mathbf{x} - \mathbf{k}_i\|) + \sum_{j=1}^M d_j \phi_j(\mathbf{x}) \quad \text{death rate}$$

$$\eta(\mathbf{x}, a) = \beta(x_1, x_2, \dots, x_{d-1}, x_d - a) \exp\left(-\int_{x_d-a}^{x_d} \mu(x_1, x_2, \dots, x_{d-1}, z) \partial z\right) \quad \text{population density}$$

$$f_j(\mathbf{x}) = \int_{\alpha_{j-1}(\mathbf{x})}^{\alpha_j(\mathbf{x})} \eta(\mathbf{x}, a) \partial a \quad \text{jth stage population}$$

$$y_{ji} \quad \text{ith observed jth stage population}$$

$$E(Y_{ji}) = f_{ji} \quad \text{ith modelled jth stage population}$$

$$i = 1, \dots, n \quad j = 1, \dots, C \quad Y_{ji} \sim \text{an exponential family distribution}$$

Full details such as the form of θ , ξ and ϕ , are given in section 2.2.

continued on next page

continued from previous page

The assumptions for the SPSP model are

- the population is closed (no migration or advection)
- the birth and death rates are smooth functions of environmental conditions and time \mathbf{x}
- the death rate is independent of age
- the population counts Y_{ji} are observations from an exponential family distribution
- the age limits are known for each physiological stage at each sample point \mathbf{x}_i

2.3 Fitting the SPSP model

To provide an overview of the fitting procedure, section 2.3.6 contains the model fitting algorithm and data requirements.

2.3.1 The Objective Function

The objective function is most easily introduced by considering the simple least squares case before generalizing to allow for data from any exponential family distribution. Let $\epsilon_{ji} = y_{ji} - f_{ji}$ be the error between the model prediction f_{ji} and the observed value y_{ji} of stage j abundance at \mathbf{x}_i . To obtain a good fit to the data, the method of least squares is a sensible approach, since the aim is to minimize the total squared errors between the observed data and the values predicted by the model.

$$\min \left(\sum_{j,i} \epsilon_{ji}^2 \right) = \min \left(\sum_{j,i} (y_{ji} - f_{ji})^2 \right)$$

For notational convenience, the Euclidean norm of $(\mathbf{y} - \mathbf{g})$, $\sum_{j,i} (y_{ji} - f_{ji})^2$, can be written $\|\mathbf{y} - \mathbf{f}\|^2$. $\mathbf{y} = (y_{11} \dots y_{1n} y_{21} \dots y_{2n} \dots y_{C1} \dots y_{Cn})^T$ is a vector of the observed population counts in each stage and \mathbf{f} is the corresponding vector of fitted stage counts.

This goodness of fit measurement is appropriate for Gaussian data, but for data assumed to be observations of any other exponential family distribution, an approach used in generalized linear modelling can be followed, namely iteratively re-weighted least squares (IRLS). For this, a weight must be attached to each error, so we include the weights matrix W in the sum of squares term:

$$\min (\|W(\mathbf{y} - \mathbf{f})\|^2)$$

where

$$W = \text{diag}(w_{ji}) \quad w_{ji} = \frac{1}{\sqrt{\widehat{\text{Var}}(Y_{ji})}} \quad i = 1, \dots, n; j = 1, \dots, C$$

For example, population counts can be considered as observations from a Poisson process, in which case W will be $W = \text{diag}\left(\frac{1}{\sqrt{f_{ji}}}\right)$. Whatever the exponential family assumed for the population counts, the weights matrix is evaluated from the current update of the fitted model, \mathbf{f} , and so is estimated iteratively.

So far, the objective function to be minimized only contains a measure of how closely the model fits the data. However, when the data are noisy, we only want to capture the (smooth) trends inherent in the data and not the noise. To do this we add wiggleness penalties $P_d(\beta)$, $P_d(\mu)$ for the birth and death rates to the sum of squares term and control the trade-off between goodness of fit to the data and smoothness of the model by means of smoothing parameters λ_β , λ_μ :

$$\min (\|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta P_d(\beta) + \lambda_\mu P_d(\mu))$$

$\lambda_\beta = 0$ would imply a very wiggly birth rate function β capturing the noise in the data; at the other extreme, a very large value of λ_β , would imply a flat birth rate surface β ignoring much of the information contained in the data; similarly for the death rate μ . The following section sets out the form of the wiggleness penalties, then section 2.3.1 outlines how sensible smoothing parameters are obtained.

The wiggleness penalties

For a ‘visibly smooth’ function in d dimensions the penalty should contain the m th derivatives of the function (Green & Silverman 1994, page 159), where $2m > d+1$. For $d = 2$ and $m = 2$,

these derivatives are proportional to the strain energy, and natural splines are the functions that minimize the total strain (they were originally thin elastic rods used by design engineers). From Green & Silverman (1994, page 159) the penalty for a ‘visibly smooth’ birth rate thin plate spline β with various desired properties² is given by

$$P_d(\beta) = \int \cdots \int_{\mathbb{R}^d} \sum \frac{m!}{v_1!v_2! \cdots v_d!} \left(\frac{\partial^m \beta}{\partial x_1^{v_1} \partial x_2^{v_2} \cdots \partial x_d^{v_d}} \right)^2 dx_1 dx_2 \cdots dx_d \quad (2.6)$$

where the sum within the integral is over all positive integers v_1, v_2, \dots, v_d such that $v_1 + v_2 + \cdots + v_d = m$. Recall that β is linear in the model parameters, so we can write

$$\beta(\mathbf{x}) = \sum_{i=1}^K e_i \gamma_i + \sum_{j=1}^M T_j b_j = \begin{pmatrix} \mathbf{e} & \mathbf{T} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\gamma} \\ \mathbf{b} \end{pmatrix}$$

for vectors \mathbf{e} and \mathbf{T} of the form

$$\mathbf{e} = \begin{pmatrix} \theta\xi(\|\mathbf{x} - \mathbf{k}_1\|) & \cdots & \theta\xi(\|\mathbf{x} - \mathbf{k}_K\|) \end{pmatrix}^T \quad \text{and} \quad \mathbf{T} = \begin{pmatrix} \phi_1(\mathbf{x}) & \cdots & \phi_M(\mathbf{x}) \end{pmatrix}^T$$

and with $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_K)^T$ and $\mathbf{b} = (b_1, b_2, \dots, b_M)^T$.

The derivatives in the penalty are taken with respect to the covariates x_1, \dots, x_d so the derivatives are also linear in the model parameters:

$$\frac{\partial^m \beta}{\partial x_1^{v_1} \partial x_2^{v_2} \cdots \partial x_d^{v_d}} = \sum_{i=1}^K \frac{\partial^m e_i}{\partial x_1^{v_1} \partial x_2^{v_2} \cdots \partial x_d^{v_d}} \cdot \gamma_i + \sum_{j=1}^M \frac{\partial^m T_j}{\partial x_1^{v_1} \partial x_2^{v_2} \cdots \partial x_d^{v_d}} \cdot b_j,$$

and since the power of each polynomial term in \mathbf{T} is less than m ,

$$\begin{aligned} \frac{\partial^m \beta}{\partial x_1^{v_1} \partial x_2^{v_2} \cdots \partial x_d^{v_d}} &= \sum_{i=1}^K \frac{\partial^m e_i}{\partial x_1^{v_1} \partial x_2^{v_2} \cdots \partial x_d^{v_d}} \cdot \gamma_i + 0 \\ &= (\mathbf{e}'_i)^T \boldsymbol{\gamma}. \end{aligned}$$

²The penalty is invariant under translation and rotation of the coordinate system and has quadratic dependence on β leading to fitted curves and surfaces linear in the data.

In the last line l is used as an indicator for the derivative of interest (an element of the sum inside the integral in equation 2.6). In the penalty equation 2.6, the derivatives are squared and multiplied by a constant before being summed and integrated:

$$\begin{aligned} P_d(\beta) &= \int \cdots \int_{\mathbb{R}^d} \sum_l \frac{m!}{v_1! \cdots v_d!} ((e'_l)^T \gamma)^2 dx_1 \dots dx_d \\ &= \int \cdots \int_{\mathbb{R}^d} \sum_l \frac{m!}{v_1! \cdots v_d!} \gamma^T e'_l (e'_l)^T \gamma dx_1 \dots dx_d \end{aligned}$$

$$\text{setting } e_l^* = \frac{m!}{v_1! \cdots v_d!} e'_l (e'_l)^T \quad :$$

$$\begin{aligned} &= \int \cdots \int_{\mathbb{R}^d} \left\{ \gamma^T e_1^* \gamma + \cdots + \gamma^T e_{\max(l)}^* \gamma \right\} dx_1 \dots dx_d \\ &= \gamma^T \left(\int \cdots \int_{\mathbb{R}^d} e_1^* dx_1 \dots dx_d \right) \gamma + \cdots + \gamma^T \left(\int \cdots \int_{\mathbb{R}^d} e_{\max(l)}^* dx_1 \dots dx_d \right) \gamma \\ &= \gamma^T \left(\left(\int \cdots \int_{\mathbb{R}^d} e_1^* dx_1 \dots dx_d \right) + \cdots + \left(\int \cdots \int_{\mathbb{R}^d} e_{\max(l)}^* dx_1 \dots dx_d \right) \right) \gamma \end{aligned}$$

$$\text{setting } e^\dagger = \sum_l \left(\int \cdots \int_{\mathbb{R}^d} e_l^* dx_1 \dots dx_d \right) \quad :$$

$$= \gamma^T e^\dagger \gamma .$$

Fortunately, this penalty for β has a simpler form (Duchon 1977):

$$P_d(\beta) = \gamma^T E \gamma$$

where

$$E = \begin{pmatrix} 0 & \theta \xi (\|\mathbf{k}_1 - \mathbf{k}_2\|) & \dots & \theta \xi (\|\mathbf{k}_1 - \mathbf{k}_K\|) \\ \theta \xi (\|\mathbf{k}_2 - \mathbf{k}_1\|) & 0 & \dots & \theta \xi (\|\mathbf{k}_2 - \mathbf{k}_K\|) \\ \vdots & \vdots & & \vdots \\ \theta \xi (\|\mathbf{k}_K - \mathbf{k}_1\|) & \theta \xi (\|\mathbf{k}_K - \mathbf{k}_2\|) & \dots & 0 \end{pmatrix} .$$

As an illustration, the explicit form of this matrix for $d = 3$ is given in appendix B.3. For notational convenience, the penalty for the birth rate will be written:

$$P_d(\beta) = \mathbf{p}^T S_\beta \mathbf{p}$$

where S_β is the $(2K + 2M) \times (2K + 2M)$ matrix

$$S_\beta = \begin{bmatrix} E & 0_{K,M} & 0_{K,K} & 0_{K,M} \\ 0_{M,K} & 0_{M,M} & 0_{M,K} & 0_{M,M} \\ 0_{K,K} & 0_{K,M} & 0_{K,K} & 0_{K,M} \\ 0_{M,K} & 0_{M,M} & 0_{M,K} & 0_{M,M} \end{bmatrix}$$

($0_{i,j}$ is a matrix of zeros with i rows and j columns) and

$$\mathbf{p} = \left(\gamma_1 \quad \dots \quad \gamma_K \quad b_1 \quad \dots \quad b_M \quad \delta_1 \quad \dots \quad \delta_K \quad d_1 \quad \dots \quad d_M \right)^T .$$

Note that the penalty for the birth rate involves only model parameters of the birth rate function.

Since the population model contains two thin plate splines, the birth rate and the death rate, it will be fitted using two penalty functions, one for each. The penalty of the death rate is similar to that of the birth rate:

$$P_d(\mu) = \mathbf{p}^T S_\mu \mathbf{p}$$

and this time

$$S_\mu = \begin{bmatrix} 0_{K,K} & 0_{K,M} & 0_{K,K} & 0_{K,M} \\ 0_{M,K} & 0_{M,M} & 0_{M,K} & 0_{M,M} \\ 0_{K,K} & 0_{K,M} & E_{K,K} & 0_{K,M} \\ 0_{M,K} & 0_{M,M} & 0_{M,K} & 0_{M,M} \end{bmatrix}$$

with \mathbf{p} is defined as before. Note that this penalty only involves model parameters defining the death rate.

The objective function $F(\mathbf{p}, \lambda_\beta, \lambda_\mu)$ that we wish to minimize now consists of a sum of squares term and two wiggleness penalties, one for each thin plate spline in the SPSP model:

$$F(\mathbf{p}, \lambda_\beta, \lambda_\mu) = \|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta \mathbf{p}^T S_\beta \mathbf{p} + \lambda_\mu \mathbf{p}^T S_\mu \mathbf{p} . \quad (2.7)$$

Obtaining the smoothing parameters

The smoothing parameters, λ_β and λ_μ , are used to balance the trade-off between loyalty to the data and smoothness of the birth and death functions. The larger a smoothing parameter, the bigger the wiggleness penalty and the smoother the function; the smaller a smoothing parameter, the smaller the penalty and the closer the function matches the data. Increasing the number of knots in the thin plate splines increases the *potential* for wiggleness, but the level of wiggleness is selected by the smoothing parameters. Since the quantity of noise in the data is unknown, we don't know how smooth to make the model, so how are we to choose the smoothing parameters?

A systematic method used to obtain the smoothing parameters, which is sufficiently general for application to the SPSP model, is explained in Wood (2000). The paper describes a method for minimizing the generalized cross validation (GCV) score for the smoothing parameters, handily accompanied by a program (written in C called `Multismooth`, but also usable from R) which can perform the task. The method of generalized cross validation for fitting smooth functions has been available for some time (Craven & Wahba 1979), developed from ordinary cross validation. The SPSP model requires two smoothing parameters to be estimated and so multiple smoothing parameter GCV is required, as described in Gu & Wahba (1991) for kernel based spline models. Other methods for obtaining multiple smoothing pa-

rameters are also specific to the models considered (e.g. some generalized additive models in Hastie & Tibshirani (1990)). In Wood (2000) the method is generalized to be applicable for the wide range of models of the form $E(Y_{ji}) = f(\mathbf{p})$, fitted to the data points y_{ji} which are treated as observations of the random variate Y_{ji} from some exponential family distribution, where f can be a non-linear function of the model parameters \mathbf{p} . The SPSP model is of this form and so the smoothing parameters for the SPSP model can be selected using this method. Once the smoothing parameters are known, or estimates are available, the objective function 2.7 can be minimized to obtain the model parameters. Sections 2.3.2–2.3.4 contain the details.

A rough guide to GCV is given here to provide an insight into the methods by which the SPSP model is fitted.

Generalized cross validation (GCV) was developed from ordinary cross validation (OCV). The idea behind OCV was to find the smoothing parameters $\lambda_\beta, \lambda_\mu$ that give the smallest OCV score $V_o(\lambda_\beta, \lambda_\mu)$. The OCV score is a measure of how well a model predicts the observed data values for given values of $\lambda_\beta, \lambda_\mu$. It is calculated by fitting the model excluding one of the data points, finding the difference between the prediction of that point and the original observed value, then repeating the process for each data point and summing the errors:

$$V_o(\lambda_\beta, \lambda_\mu) = \frac{1}{nC} \sum_i \left(\hat{y}_i - \hat{f}^{(-i, \lambda_\beta, \lambda_\mu)}(\mathbf{x}_i) \right)^2$$

where $\hat{f}^{(-i, \lambda_\beta, \lambda_\mu)}$ is the minimizer of $\sum_{j \neq i} (y_j - f(\mathbf{x}_j))^2 + \lambda_\beta \times [\text{smoothing penalty}] + \lambda_\mu \times [\text{smoothing penalty}]$. (There are nC observations because there are n sample points and C developmental stages.) So $\hat{f}^{(-i, \lambda_\beta, \lambda_\mu)}$ is the model of a form already chosen, fitted to the data excluding one point only, which means that the models do not differ in structure for different

i ; only the model parameter estimates differ. $\hat{f}^{(-i, \lambda_\beta, \lambda_\mu)}(\mathbf{x}_i)$ is the prediction of the missing point y_i . Most simply, a grid search for $\lambda_\beta, \lambda_\mu$ could be performed in which the OCV score is calculated for a set of different $\lambda_\beta, \lambda_\mu$'s and the $\lambda_\beta, \lambda_\mu$ giving the smallest OCV score is selected, along with its associated f . Whatever method is used to search for the minimising smoothing parameter $\lambda_\beta, \lambda_\mu$, the OCV score will need to be calculated several times. The OCV score may not have a unique minimum.

The residuals in the OCV score can be written in terms of the residuals from the model $f^{(\lambda_\beta, \lambda_\mu)}(\mathbf{x})$ fitted to *all* of the data points ($\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$) weighted using elements of the 'influence' or 'hat' matrix $A(\lambda_\beta, \lambda_\mu)$. The influence matrix maps the observed values onto the fitted values:

$$\hat{f}^{(\lambda_\beta, \lambda_\mu)}(\mathbf{x}) = A(\lambda_\beta, \lambda_\mu)\mathbf{y} .$$

Thus the OCV score now becomes

$$V_o(\lambda_\beta, \lambda_\mu) = \frac{1}{n} \sum_i \left(\frac{y_i - \hat{f}^{\lambda_\beta, \lambda_\mu}(\mathbf{x}_i)}{1 - A_{ii}(\lambda_\beta, \lambda_\mu)} \right)^2 .$$

This OCV score is still quite expensive to compute, and has some poor properties (see argument in Wahba (1990)) so the GCV score can be used instead. The GCV score is simply the OCV score with each of the weights set equal to the average weight $\left(1 - \frac{\text{tr}(A)}{n}\right)^2$, where $\text{tr}(A)$ denotes the trace of the matrix A . Data assumed to be observations from non-normal exponential family distributions are accommodated using the weights matrix W . The GCV score, $V(\lambda_\beta, \lambda_\mu)$ is

$$V(\lambda_\beta, \lambda_\mu) = \frac{\|W(\mathbf{y} - \hat{f}^{(\lambda_\beta, \lambda_\mu)}(\mathbf{x}))\|^2/n}{\left(1 - \frac{\text{tr}(A)}{n}\right)^2} .$$

The GCV score may have several local minima, so care must be taken to find the smoothing parameters minimising $V(\lambda_\beta, \lambda_\mu)$. Although straightforward, and a sure method of finding the

global minimum, a grid search would be very expensive - taking order $nC(K+M)^2k^2$ floating point operations for the SPSP model when there are k grid points (reduced to nCK^2k^2 order operations when the equality constraints are included - see section 2.3.3). Wood's method (Wood 2000) uses only order $(K+M)^3$ operations, reduced to order K^3 operations when equality constraints are incorporated.

When more than one smoothing parameter is to be estimated, as in this case, we can write the smoothing parameters in terms of an overall smoothing parameter, ρ , and relative smoothing parameters for the birth and death functions, $\lambda_\beta^* = \frac{\lambda_\beta}{\rho}$ and $\lambda_\mu^* = \frac{\lambda_\mu}{\rho}$. So the objective function can be written

$$F(\mathbf{p}, \lambda_\beta, \lambda_\mu) = \rho \|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta^* \mathbf{p}^T S_\beta \mathbf{p} + \lambda_\mu^* \mathbf{p}^T S_\mu \mathbf{p} . \quad (2.8)$$

The procedure followed in Wood (2000) is to first find the level of overall smoothing (the value of ρ) that minimizes the GCV score by a direct search, and then update the relative weight given to each smooth (calculate λ_β^* and λ_μ^*). These two steps are iterated until an acceptable level of convergence of the smoothing parameters is achieved. The first step of this approach makes use of the efficient algorithms that exist for problems involving only one smoothing parameter; some fairly expensive transformations are performed initially so that an efficient direct search for ρ can be carried out. The second step involves applying Newton's method to update $\begin{pmatrix} \log(\lambda_\beta) & \log(\lambda_\mu) \end{pmatrix}^T$. (An explanation of Newton's method can be found in Gill, Murray & Wright (1981))

In the direct search for ρ , the GCV score and hence the trace of the influence matrix A must be calculated. If the model is linear in the model parameters, $\mathbf{f} = X\mathbf{p}$, the influence matrix can be written in terms of the design matrix X . However the SPSP model is non-linear,

so instead the influence matrix is written in terms of the SPSP model's Jacobian matrix. Essentially this means the objective function of a non-linear model must be approximated by one which is quadratic in the model parameters. The resulting iterative least squares scheme resembles that used for fitting generalized linear models.

2.3.2 Quadratic approximation of the objective function

The structure of the SPSP model f is the flexible non-linear form described in section 2.2. The particular model fitted to a set of data is determined by the model parameters, \mathbf{p} , and the smoothing parameters, λ_β and λ_μ . To make use of Wood's process for estimating these parameters, the sum of squares term in the objective function must be quadratic in the model parameters.

The smoothing part of the objective function 2.7 is already quadratic in the model parameters, so only the sum of squares term

$$\|\mathbf{f} - \mathbf{y}\|^2 = (\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y})$$

is of concern. A quadratic approximation can be found by using a suitable Taylor expansion.

The model parameters can be estimated by minimizing this sum of squares term, which can be performed iteratively. Let $\mathbf{p}^{[k]}$ denote the k th iterate of the model parameter estimates, and $\mathbf{f}^{[k]}$ denote the k th iterate of the SPSP model $f = E(Y)$ (the one with parameters $\mathbf{p}^{[k]}$) estimated at each data point. The model parameters can be written in terms of the k th estimate of them as follows:

$$\mathbf{p} = \mathbf{p}^{[k]} + \Delta$$

for small Δ . The SPSP model \mathbf{f} is a non-linear function of \mathbf{p} . Because of the relationship between \mathbf{p} and $\mathbf{p}^{[k]}$ given above, \mathbf{f} can be approximated by using a Taylor's series expansion about $\mathbf{p}^{[k]}$ up to the first derivative, resulting in an approximate model function which is linear in the model parameters:

$$f_{ji} \approx f_{ji}^{[k]} + \sum_{l=1}^{2K+2M} \frac{\partial f_{ji}^{[k]}}{\partial p_l^{[k]}} \Delta_l$$

$$\mathbf{f} \approx \mathbf{f}^{[k]} + J\Delta$$

where J is the Jacobian matrix with the element on the $((j-1)n+i)$ th row and l th column given by $J_{((j-1)n+i),l} = \frac{\partial f_{ji}^{[k]}}{\partial p_l^{[k]}}$. J has nC rows (the number of data points), and $2K+2M$ columns (the number of model parameters). The Jacobian matrix for the case $d=3$ can be found in appendix B.2. Substituting for \mathbf{f} in the Euclidean norm allows us to write the sum of squares as

$$(\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y}) \approx (\mathbf{f}^{[k]} + J\Delta - \mathbf{y})^T (\mathbf{f}^{[k]} + J\Delta - \mathbf{y})$$

$$(\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y}) \approx (\mathbf{f}^{[k]} - \mathbf{y})^T (\mathbf{f}^{[k]} - \mathbf{y}) + \Delta^T J^T J \Delta + 2\Delta^T J^T (\mathbf{f}^{[k]} - \mathbf{y}) .$$

This may be differentiated with respect to Δ and set equal to zero to obtain a value of Δ which minimizes the sum of squares:

$$2J^T J \Delta + 2J^T (\mathbf{f}^{[k]} - \mathbf{y}) = 0 \Rightarrow \Delta = (J^T J)^{-1} J^T (\mathbf{y} - \mathbf{f}^{[k]})$$

Now $\mathbf{p} = \Delta + \mathbf{p}^{[k]}$, so

$$\begin{aligned}
 \mathbf{p} &= (J^T J)^{-1} J^T (\mathbf{y} - \mathbf{f}^{[k]}) + \mathbf{p}^{[k]} \\
 &= (J^T J)^{-1} J^T (\mathbf{y} - \mathbf{f}^{[k]}) + (J^T J)^{-1} J^T J \mathbf{p}^{[k]} \\
 &= (J^T J)^{-1} J^T (\mathbf{y} - \mathbf{f}^{[k]} + J \mathbf{p}^{[k]}) \\
 &= (J^T J)^{-1} J^T \mathbf{z}^{[k]}
 \end{aligned}$$

when the ‘pseudo-data’ \mathbf{z} are set equal to $\mathbf{z}^{[k]} = \mathbf{y} - \mathbf{f}^{[k]} + J \mathbf{p}^{[k]}$. The sum of squares can now be approximated by a quadratic function $\|J \mathbf{p} - \mathbf{z}^{[k]}\|^2$ of the model parameters \mathbf{p} , since

$$\begin{aligned}
 \|J \mathbf{p} - \mathbf{z}^{[k]}\|^2 &= \|J(J^T J)^{-1} J^T (\mathbf{y} - \mathbf{f}^{[k]} + J \mathbf{p}^{[k]}) - (\mathbf{y} - \mathbf{f}^{[k]} + J \mathbf{p}^{[k]})\|^2 \\
 &= \|J(J^T J)^{-1} J^T (\mathbf{y} - \mathbf{f}^{[k]}) - (\mathbf{y} - \mathbf{f}^{[k]}) + J(J^T J)^{-1} J^T J \mathbf{p}^{[k]} - J \mathbf{p}^{[k]}\|^2 \\
 &= \|J \Delta - (\mathbf{y} - \mathbf{f}^{[k]})\|^2 \\
 &\approx \|\mathbf{f} - \mathbf{y}\|^2
 \end{aligned}$$

This approximation of the sum of squares leads to the approximation of the objective function being:

$$F(\mathbf{p}, \lambda_\beta, \lambda_\mu) \approx F^*(\mathbf{p}, \lambda_\beta, \lambda_\mu) = \|W(J \mathbf{p} - \mathbf{z}^{[k]})\|^2 + \lambda_\beta \mathbf{p}^T S_\beta \mathbf{p} + \lambda_\mu \mathbf{p}^T S_\mu \mathbf{p} \quad (2.9)$$

which is quadratic in the model parameters. The elements of the Jacobian matrix can be calculated either directly (for example, see appendix B.2), or by using finite differencing.

Wood’s program `Multismooth` updates the smoothing parameters and minimizes the quadratic approximation of the objective function to obtain the model parameters for this update. However, the space of mathematically and demographically plausible model parameters is

smaller than the entire space of possible model parameters for these smoothing parameters. Obtaining plausible model parameters is the subject of the next section.

2.3.3 Model constraints

The model parameters must be constrained in order to ensure sensible birth and death rate functions.

There are two different types of constraints that restrict the SPSP model - equality constraints and inequality constraints. Equality constraints are required for mathematical reasons - to ensure the wiggleness penalties of the birth and death rates are finite. Inequality constraints are required for demographic reasons - to ensure the birth and death rates are non-negative. Incorporating the equality constraints is the subject of the next section while the inequality constraints are dealt with in the succeeding section.

Incorporating equality constraints

For the penalty terms corresponding to the birth and death rates to be finite, the birth and death rates must both be *natural* thin plate splines. This means that the model parameters must satisfy

$$\mathbf{T}_{MK}\boldsymbol{\gamma} = 0_{M,1} \quad \text{and} \quad \mathbf{T}_{MK}\boldsymbol{\delta} = 0_{M,1} \quad (2.10)$$

where \mathbf{T}_{MK} is the $M \times K$ matrix

$$\mathbf{T}_{MK} = \begin{pmatrix} \phi_1(\mathbf{k}_1) & \phi_1(\mathbf{k}_2) & \dots & \phi_1(\mathbf{k}_K) \\ \phi_2(\mathbf{k}_1) & \phi_2(\mathbf{k}_2) & \dots & \phi_2(\mathbf{k}_K) \\ \vdots & \dots & \vdots & \\ \phi_M(\mathbf{k}_1) & \phi_M(\mathbf{k}_2) & \dots & \phi_M(\mathbf{k}_K) \end{pmatrix} .$$

Note that the equality constraints involve only the parameters corresponding to the radial basis functions, and these functions make up the elements of the penalty matrices.

These constraints must be incorporated into the model fitting process so that only viable models, that is, those whose parameters satisfy these constraints, may be selected. This may be achieved by using QT factorisation (Gill et al. 1981).

A $K \times K$ orthogonal matrix Q may be found such that

$$\mathbf{T}_{MK}Q = \begin{bmatrix} 0_{M,(K-M)} & \Delta \end{bmatrix}$$

where Δ is an $M \times M$ reverse lower triangular matrix such that $\Delta_{ij} = 0$ for $i+j \leq M$, $i, j = 1, \dots, M$. The first $(K-M)$ columns of Q form a matrix Z_{MK} such that $Z_{MK}^T Z_{MK} = I_{(K-M)}$ ($I_{(K-M)}$ is the $(K-M)$ square identity matrix) and $\mathbf{T}_{MK}Z_{MK} = 0_{M,(K-M)}$.

Setting $\gamma = Z_{MK}\gamma_z$ for γ_z a vector of length $(K-M)$,

$$\mathbf{T}_{MK}\gamma = \mathbf{T}_{MK}Z_{MK}\gamma_z = 0_{M,(K-M)}\gamma_z = 0_{M,1} .$$

Similarly, setting $\delta = Z_{MK}\delta_z$ for δ_z a vector of length $(K-M)$ implies $\mathbf{T}_{MK}\delta = 0_{M,1}$. Therefore a model satisfying the equality constraints 2.10 can be fitted by simply choosing any parameters γ_z and δ_z and replacing γ and δ with $Z_{MK}\gamma_z$ and $Z_{MK}\delta_z$ throughout the model fitting process.

For convenience the $(2K + 2M) \times 2K$ matrix Z and vector \mathbf{p}_z of length $2K$ are defined:

$$\mathbf{p}_z = \begin{bmatrix} \gamma_z \\ \mathbf{b} \\ \delta_z \\ \mathbf{d} \end{bmatrix} \quad Z = \begin{bmatrix} Z_{MK} & 0_{K,M} & 0_{K,(K-M)} & 0_{K,M} \\ 0_{M,(K-M)} & I_M & 0_{M,(K-M)} & 0_{M,M} \\ 0_{K,(K-M)} & 0_{K,M} & Z_{MK} & 0_{K,M} \\ 0_{M,(K-M)} & 0_{M,M} & 0_{M,(K-M)} & I_M \end{bmatrix} .$$

So $\mathbf{p} = Z\mathbf{p}_z$. To distinguish \mathbf{p} from \mathbf{p}_z , \mathbf{p}_z will be termed ‘constrained’ model parameters.

The objective function to be minimized can be written

$$\begin{aligned} F(\mathbf{p}, \lambda_\beta, \lambda_\mu) &= \|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta \mathbf{p}^T S_\beta \mathbf{p} + \lambda_\mu \mathbf{p}^T S_\mu \mathbf{p} \\ &= \|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta (Z\mathbf{p}_z)^T S_\beta Z\mathbf{p}_z + \lambda_\mu (Z\mathbf{p}_z)^T S_\mu Z\mathbf{p}_z \\ &= \|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta \mathbf{p}_z^T Z^T S_\beta Z\mathbf{p}_z + \lambda_\mu \mathbf{p}_z^T Z^T S_\mu Z\mathbf{p}_z \end{aligned}$$

setting $S_{\beta_z} = Z^T S_\beta Z$ and $S_{\mu_z} = Z^T S_\mu Z$,

$$F(\mathbf{p}_z, \lambda_\beta, \lambda_\mu) = \|W(\mathbf{y} - \mathbf{f})\|^2 + \lambda_\beta \mathbf{p}_z^T S_{\beta_z} \mathbf{p}_z + \lambda_\mu \mathbf{p}_z^T S_{\mu_z} \mathbf{p}_z$$

and its quadratic approximation becomes

$$\begin{aligned} F^*(\mathbf{p}, \lambda_\beta, \lambda_\mu) &= \|W(J\mathbf{p} - \mathbf{z}^{[k]})\|^2 + \lambda_\beta \mathbf{p}^T S_\beta \mathbf{p} + \lambda_\mu \mathbf{p}^T S_\mu \mathbf{p} \\ &= \|W(JZ\mathbf{p}_z - (\mathbf{y} - \mathbf{f} + JZ\mathbf{p}_z))\|^2 + \lambda_\beta \mathbf{p}_z^T S_{\beta_z} \mathbf{p}_z + \lambda_\mu \mathbf{p}_z^T S_{\mu_z} \mathbf{p}_z \end{aligned}$$

setting $J_z = JZ$ (notice $\mathbf{z} = \mathbf{y} - \mathbf{f} + JZ\mathbf{p}_z$),

$$F^*(\mathbf{p}_z, \lambda_\beta, \lambda_\mu) = \|W(J_z \mathbf{p}_z - \mathbf{z})\|^2 + \lambda_\beta \mathbf{p}_z^T S_{\beta_z} \mathbf{p}_z + \lambda_\mu \mathbf{p}_z^T S_{\mu_z} \mathbf{p}_z .$$

The model fitting process now involves searching for the smoothing parameters λ_β and λ_μ and the constrained model parameters \mathbf{p}_z . It becomes clear why the number of steps taken

to fit the SPSP model is reduced by incorporating equality constraints — there are $2M$ fewer parameters to estimate. The matrices involved in the objective function and all calculations are smaller and so once the matrix multiplications have taken place prior to performing the iterative procedure, fewer computations will be required to fit the model. Incorporating the equality constraints in this way also means that there must be more than $2M$ knots or there will be no constrained model parameters to estimate at all.

Incorporating inequality constraints

Villalobos & Wahba (1987) describe how to incorporate inequality constraints into a spline model using quadratic programming, so that only appropriate parameters can be chosen for the entire data space ($\forall \mathbf{x} \in \mathbb{X}$). However, their method would increase programming complexity and computation time substantially. Instead, constraints are imposed on only a finite set of points covering the data space, $\{(\mathbf{x}_i), i = 1, \dots, n^*\} \in \mathbb{X}$. The inequality constraints are

$$\beta(\mathbf{x}_i) \geq 0 \quad \text{and} \quad \mu(\mathbf{x}_i) \geq 0 \quad \text{for } i = 1, \dots, n^*. \quad (2.11)$$

Quadratic programming is the method employed within each iterate of the model fitting process to achieve this using functions available as part of the `mgcv` package for *R* written by Simon Wood (available from www.cran.r-project.org). The rest of this section outlines the procedure followed.

Since the birth and death rates can be written as linear functions of the model parameters,

the inequality constraints can be re-phrased

$$\begin{bmatrix} A_{n^*} & 0_{n^*,(K+M)} \\ 0_{n^*,(K+M)} & A_{n^*} \end{bmatrix} \mathbf{p} = \begin{bmatrix} A_{n^*} & 0_{n^*,(K+M)} \\ 0_{n^*,(K+M)} & A_{n^*} \end{bmatrix} Z \mathbf{p}_z \geq 0_{2n^*}$$

where the $n^* \times (K + M)$ matrix A_{n^*} is defined as

$$\begin{pmatrix} \xi(\|\mathbf{x}_1 - \mathbf{k}_1\|) & \xi(\|\mathbf{x}_1 - \mathbf{k}_2\|) & \dots & \xi(\|\mathbf{x}_1 - \mathbf{k}_K\|) & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \dots & \phi_M(\mathbf{x}_1) \\ \xi(\|\mathbf{x}_2 - \mathbf{k}_1\|) & \xi(\|\mathbf{x}_2 - \mathbf{k}_2\|) & \dots & \xi(\|\mathbf{x}_2 - \mathbf{k}_K\|) & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \dots & \phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \xi(\|\mathbf{x}_{n^*} - \mathbf{k}_1\|) & \xi(\|\mathbf{x}_{n^*} - \mathbf{k}_2\|) & \dots & \xi(\|\mathbf{x}_{n^*} - \mathbf{k}_K\|) & \phi_1(\mathbf{x}_{n^*}) & \phi_2(\mathbf{x}_{n^*}) & \dots & \phi_M(\mathbf{x}_{n^*}) \end{pmatrix}$$

It is advantageous to include more constraint points in the matrix than there are in the data set used to fit the SPSP model, that is, to have more inequality constraints than there are data points, $n^* > n$. For example, this would give more security when predicting birth or death rates on a fine grid over the data space.

As indicated by the name, quadratic programming requires the objective function to be quadratic in the model parameters. Thus the quadratic approximation of the objective function (equation 2.9) is used to obtain model parameters subject to the inequality constraints.

The idea of quadratic programming is illustrated in figure 2.4. Because visualizing a function in more than three dimensions is difficult, the figure uses an example with an objective function of a two-dimensional model parameter vector (the smoothing parameters are fixed at the current estimate). In the figure, the contours of the approximate objective function are drawn as dotted curves, shaded regions indicate parameter subspaces that are not permitted by the inequality constraints and red lines define the boundaries of the constraints.

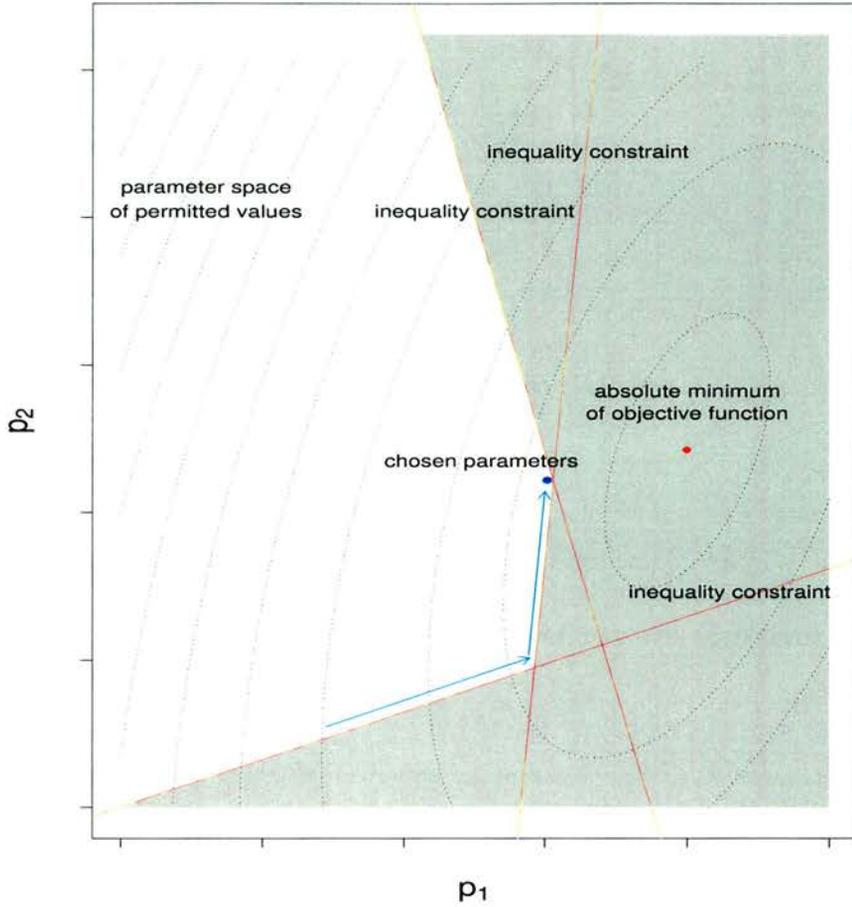


Figure 2.4: Contour plot showing an example approximate objective function when there are only two elements of the model parameter vector, (p_1, p_2) . Inequality constraints on the parameters are shown as shaded regions outlined by red lines. The subspace of permitted parameter values is unshaded. The arrows indicate a method of locating the parameters which minimize the objective function subject to the inequality constraints.

The location of the model parameters that minimize the entire approximate objective function is plotted as a red dot, which, as in the illustration, may lie in the prohibited parameter subspace. The location of the model parameters that minimize the approximate objective function subject to the inequality constraints is plotted as a blue dot. These parameters

are found using the quadratic programming algorithm: firstly, the quadratic approximation of the objective function $F^*(\mathbf{p}_z + \boldsymbol{\nu}_z, \lambda_\beta, \lambda_\mu)$ is minimized for $\boldsymbol{\nu}_z$. $\boldsymbol{\nu}_z$ would take \mathbf{p}_z to the minimizer of $F^*(\mathbf{p}_z + \boldsymbol{\nu}_z, \lambda_\beta, \lambda_\mu)$ (the red dot) ignoring the inequality constraints. Instead, the step $\alpha\boldsymbol{\nu}_z$ (a blue arrow) is taken, which is the longest possible step towards the minimizer without violating any inequality constraints (crossing any red lines). Then the new set of constrained parameters is set to $\mathbf{p}_z \leftarrow \mathbf{p}_z + \alpha\boldsymbol{\nu}_z$, and the process is repeated until α is zero because it is no longer possible to reduce further the value of the approximate objective function subject to the inequality constraints by changing the model parameters. The current update of the constrained model parameters, $\mathbf{p}_z^{[\text{temp}]}$, is set to this final vector.

Figure 2.4 illustrates this process by indicating the direction of the search for the model parameters with blue arrows. The search can be thought of as walking downhill until a wall - an inequality constraint boundary - is reached. The search continues along the wall until another wall is encountered, and the process repeats until it is no longer possible to move further downhill. The final location (the blue dot) indicates the model parameters used in the current update.

2.3.4 Obtaining model parameters for the true objective function

For each update of the smoothing parameters, the model parameters are updated subject to inequality constraints (and holding the smoothing parameters fixed), since the smoothing restrictions impact on the particular form of the splines. The updated model parameters will then be used to provide the next estimate of the Jacobian matrix and pseudo data. So far, the model parameters have been updated by minimizing the quadratic approximation of the objective function subject to constraints. However, if the model parameters \mathbf{p} are poor

estimates of the minimizing model parameters, then the quadratic approximation $\|J\mathbf{p} - \mathbf{z}^{[k+1]}\|$ may be a poor approximation of $\|\mathbf{f}^{[k+1]} - \mathbf{y}\|$ and the iterative model fitting process may converge slowly, fail to converge, or even diverge.

Therefore the next estimate of the model parameters $\mathbf{p}_z^{[k+1]}$ will be found using a search based on evaluation of the true objective function rather than the quadratic approximation. The parameter space searched is the line between the previous estimate of the model parameters $\mathbf{p}_z^{[k]}$ and the (current) estimate $\mathbf{p}_z^{[\text{temp}]}$ suggested by minimizing the quadratic approximation of the objective function. Ideally an efficient search should require relatively few evaluations of the objective function and each new parameter estimate should get closer to the solution. The golden section search is one such efficient method and is described in this section.

The method begins with two estimates of the model parameters - those from the previous iterate of the model fitting process, $\mathbf{p}^{[k]}$, and those of the current iterate, $\mathbf{p}^{[\text{temp}]}$. Both parameter vectors satisfy the equality and inequality constraints but I have dropped the subscript z in this section for clarity. The search is performed within the parameter subspace containing these two parameter vectors. To follow the search method, it helps to visualize the objective function and parameter space, so this section will be illustrated imagining that the model parameter vector has only two elements, (p_1, p_2) .

Consider the objective function for the smoothing parameters fixed to their current estimates over the space of all possible combinations of model parameters, as illustrated by the contour plot in figure 2.5(a). Although $\mathbf{p}^{[\text{temp}]}$ may not be heading directly from $\mathbf{p}^{[k]}$ in the direction that includes the global minimizer of the objective function, good progress can be made by finding the minimizer \mathbf{p}^* in the direction from $\mathbf{p}^{[k]}$ to $\mathbf{p}^{[\text{temp}]}$. In this way each iteration of

the model fitting process will proceed towards the global best parameters. The purpose of the golden section search is to find \mathbf{p}^* in the subspace between $\mathbf{p}^{[k]}$ and $\mathbf{p}^{[\text{temp}]}$, illustrated in figure 2.5(a) by the line connecting them. Only this subspace is searched to ensure that the parameter constraints are always met.

The golden section search proceeds as follows. Firstly, the intermediary parameter vectors \mathbf{p}_a and \mathbf{p}_b are calculated:

$$\begin{aligned}\mathbf{p}_a &= \mathbf{p}^{[k]} + \tau(\mathbf{p}^{[\text{temp}]} - \mathbf{p}^{[k]}) \\ \mathbf{p}_b &= \mathbf{p}^{[\text{temp}]} - \tau(\mathbf{p}^{[\text{temp}]} - \mathbf{p}^{[k]})\end{aligned}$$

for $\tau = \frac{2}{1+\sqrt{5}}$ which is the reciprocal of the golden ratio. Figure 2.5 (b) shows \mathbf{p}_a and \mathbf{p}_b for the example.

The following procedure is then repeated until $|F(\mathbf{p}_b) - F(\mathbf{p}_a)|$ is deemed sufficiently small:

$$\begin{aligned}\text{If } F(\mathbf{p}_b) < F(\mathbf{p}_a), \text{ set } & \mathbf{p}^{[\text{temp}]} \leftarrow \mathbf{p}_a \\ & \mathbf{p}_a \leftarrow \mathbf{p}_b \\ & \mathbf{p}_b \leftarrow \mathbf{p}^{[\text{temp}]} - \tau(\mathbf{p}^{[\text{temp}]} - \mathbf{p}^{[k]}) \\ \text{else if } F(\mathbf{p}_b) > F(\mathbf{p}_a), \text{ set } & \mathbf{p}^{[k]} \leftarrow \mathbf{p}_b \\ & \mathbf{p}_b \leftarrow \mathbf{p}_a \\ & \mathbf{p}_a \leftarrow \mathbf{p}^{[k]} + \tau(\mathbf{p}^{[\text{temp}]} - \mathbf{p}^{[k]}) .\end{aligned}$$

The operations must be performed in the order they are written; in each case, the third operation involves the latest forms of $\mathbf{p}^{[k]}$ and $\mathbf{p}^{[\text{temp}]}$. Figure 2.5(c) illustrates one such step for the case $F(\mathbf{p}_b) > F(\mathbf{p}_a)$.

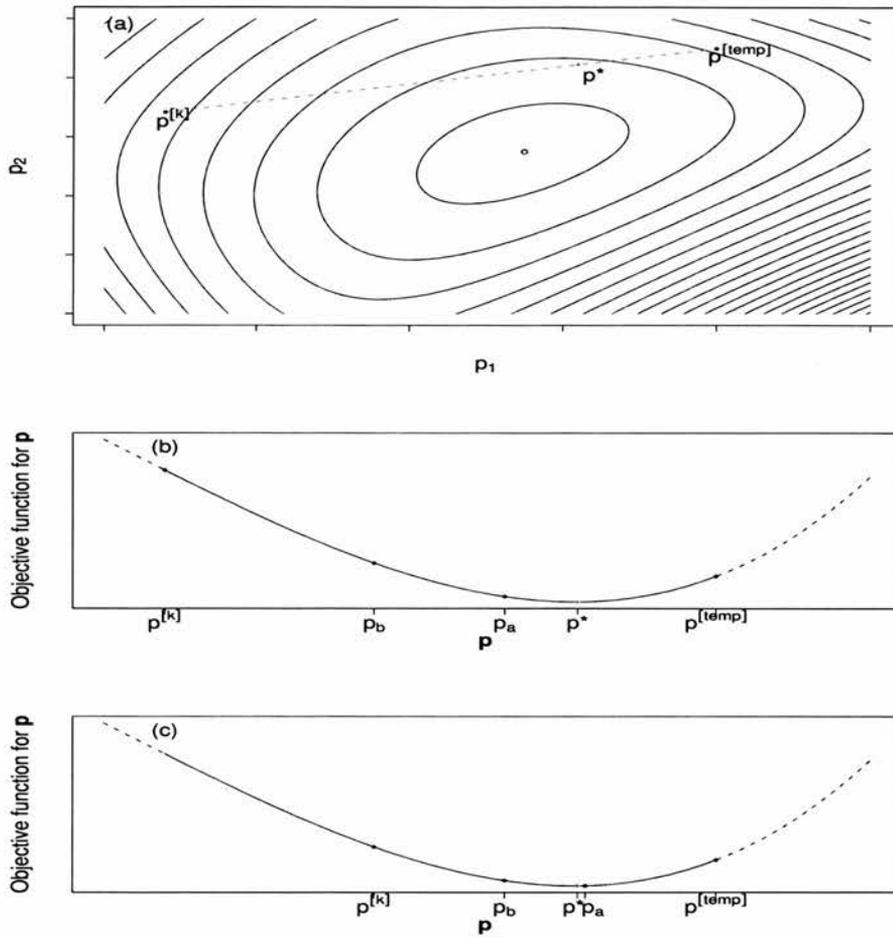


Figure 2.5: (a) Contour plot showing an example objective function when there are only two elements of the model parameter vector, (p_1, p_2) . The dashed line illustrates the path of the golden section search, given the previous and current estimates, $\mathbf{p}^{[k]}$ and $\mathbf{p}^{[temp]}$. (b) The objective function for the subspace of parameters considered in the golden section search, and parameter estimates involved in the first step of the search. (c) The objective function for the subspace of parameters considered in the golden section search, and parameter estimates involved in the second step of the search.

When $|F(\mathbf{p}_b) - F(\mathbf{p}_a)| < \text{conv_tol}$, (where conv_tol is the convergence tolerance level), the golden search is stopped and the set of parameters with the smallest objective function value is assigned to the new ‘current estimate’ of the model parameters, $\mathbf{p}^{[k+1]}$.

If $F(\mathbf{p}_b) < F(\mathbf{p}_a)$, set $\mathbf{p}^{[k+1]} \leftarrow \mathbf{p}_b$ otherwise set $\mathbf{p}^{[k+1]} \leftarrow \mathbf{p}_a$.

2.3.5 Starting and stopping the model fitting process

The previous sections (2.3.1-2.3.4) describe an iterative procedure for fitting the SPSP model. They give the method by which the smoothing parameters and model parameters are updated, and this section suggests a suitable starting procedure and gives the criteria for deciding when to stop updating the parameters.

The smoothing parameters do not need to be initialized since they are estimated within the model fitting algorithm using GCV. The model parameters are initialized by setting the birth function to a positive constant and the death functions to a positive constant. For example

$$\boldsymbol{\beta} = \begin{bmatrix} A_n & 0_{n,K+M} \\ 0_{n,K+M} & 0_{n,K+M} \end{bmatrix} Z\mathbf{p}_z = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\mu} = \begin{bmatrix} 0_{n,K+M} & 0_{n,K+M} \\ 0_{n,K+M} & A_n \end{bmatrix} Z\mathbf{p}_z = \begin{pmatrix} \mu_0 \\ \vdots \\ \mu_0 \end{pmatrix}$$

for $\beta_0 = 2$ and $\mu = 0.1$. To achieve this, the parameter b_1 corresponding to the polynomial term $\phi_1 = 1$ in the birth rate spline is set equal to the birth constant, and the parameter d_1 corresponding to the same polynomial term $\phi_1 = 1$ in the death rate spline is set equal to the death constant. (There will always be a $\phi_i = 1$ in the set of linearly independent polynomials spanning the M -dimensional space of polynomials in $\mathbb{R}^{d \geq 1}$ of total degree less than m . For convenience, this term has been chosen to be the first of the set $\{\phi_1 \dots \phi_M\}$.) All other

constrained parameters $(\gamma_z, b_2, \dots, b_M, \delta_z, d_2, \dots, d_M)$ are set to zero. These parameters satisfy the inequality constraints since the splines they define are both positive over \mathbb{R}^d and therefore the whole data space.

Suitable stopping criteria can be found in Gill et al. (1981). A fairly simple set of criteria have been adopted here.

- If the fitted stage populations are diverging, stop iterating. Divergence is deemed to occur when at least one element in the current fitted stage population vector $\mathbf{f}^{[k+1]}$ differs greatly from the corresponding element of the previous fitted stage population vector $\mathbf{f}^{[k]}$:

$$\text{if } D_{ji} = \frac{|f_{ji}^{[k+1]} - f_{ji}^{[k]}|}{1 + \left(\frac{f_{ji}^{[k+1]} + f_{ji}^{[k]}}{2}\right)} > \text{div_tol} \quad \text{for any } j \in \{1, \dots, C\} \text{ and } i \in \{1, \dots, n\}$$

→ stop iterating, divergence has occurred.

- If the fitted stage populations have converged, stop iterating. Convergence is said to have occurred when every element in the current fitted stage population vector $\mathbf{f}^{[k+1]}$ is sufficiently close to the corresponding element of the previous fitted stage population vector $\mathbf{f}^{[k]}$:

$$\text{if } D_{ji} = \frac{|f_{ji}^{[k+1]} - f_{ji}^{[k]}|}{1 + \left(\frac{f_{ji}^{[k+1]} + f_{ji}^{[k]}}{2}\right)} < \text{conv_tol} \quad \forall j \in \{1, \dots, C\} \text{ and } i \in \{1, \dots, n\}$$

→ stop iterating, convergence has occurred.

These convergence criteria involve the difference between consecutive estimates of the stage populations, scaled by the mean size of estimates. ‘1’ is added to the denominator to ensure

that divisions by numbers close to zero do not occur. `div_tol` and `conv_tol` are the divergence and convergence tolerance values.

2.3.6 Summary of model fitting

The algorithm for fitting the SPSP model given in the box below is made up of the steps explained in sections 2.3.1–2.3.5.

Model Fitting Algorithm

1. Initialize the constrained model parameters $\mathbf{p}_z^{[k]}$ to be zero except for those relating to the constant term, $b_1 = \beta_0 > 0$ and $d_1 = \mu_0 > 0$.
2. Use the constrained model parameters $\mathbf{p}_z^{[k]}$ to calculate the model parameters $\mathbf{p}^{[k]}$ using $\mathbf{p}^{[k]} = Z\mathbf{p}_z^{[k]}$. (Check that the constraints $T\mathbf{p} = \mathbf{0}$ are satisfied.)
3. Use $\mathbf{p}^{[k]}$ to obtain the stage abundances $\mathbf{f}^{[k]}$, the Jacobian matrix $J^{[k]}$, and the weight matrix $W^{[k]}$.
4. Transform $J^{[k]}$ to $J_z^{[k]} = J^{[k]}Z$ and calculate the pseudo-data vector $\mathbf{z}^{[k]}$
5. Use the method of Wood (2000) to perform the next iterate in the process of minimizing the penalized sum of squares

$$\|W^{[k]}(J^{[k]}\mathbf{p}_z - \mathbf{z}^{[k]})\|^2 + \lambda_\beta \mathbf{p}_z^T S_{\beta z} \mathbf{p}_z + \lambda_\mu \mathbf{p}_z^T S_{\mu z} \mathbf{p}_z \quad (2.12)$$

with respect to \mathbf{p}_z while estimating $\lambda_\beta^{[k+1]}$ and $\lambda_\mu^{[k+1]}$ by GCV. Discard the \mathbf{p}_z since these parameters may not satisfy the inequality constraints.

continued on next page

continued from previous page

6. With the smoothing parameters fixed at the current estimates of $\lambda_\beta^{[k+1]}$ and $\lambda_\mu^{[k+1]}$, minimize equation 2.12 with respect to \mathbf{p}_z subject to the constraints ensuring positive birth and death rates by quadratic programming. The resulting model parameters are labelled $\mathbf{p}_z^{[temp]}$.

7. Use the golden section search to find the constrained model parameters (assigning them to $\mathbf{p}_z^{[k+1]}$) which minimize the true objective function

$$\|W^{[k]}(\mathbf{y} - \mathbf{f}^{[k]})\|^2 + \lambda_\beta^{[k+1]} \mathbf{p}_z^T S_{\beta_z} \mathbf{p}_z + \lambda_\mu^{[k+1]} \mathbf{p}_z^T S_{\mu_z} \mathbf{p}_z$$

along the line from the previous constrained model parameter estimates $\mathbf{p}_z^{[k]}$ to the current estimates $\mathbf{p}_z^{[temp]}$.

8. Calculate $\mathbf{p}^{[k+1]}$ and hence $\mathbf{f}^{[k+1]}$ and $\mathbf{D} = (D_{11}, \dots, D_{Cn})^T$ from $\mathbf{p}_z^{[k+1]}$.

- If any element of \mathbf{D} is greater than *div_tol*, divergence has occurred and iteration must stop.
- If all elements of \mathbf{D} are less than *conv_tol*, convergence has occurred, in which case go to step 9.
- If neither of the above is true, set $\lambda_\beta^{[k]} \leftarrow \lambda_\beta^{[k+1]}$, $\lambda_\mu^{[k]} \leftarrow \lambda_\mu^{[k+1]}$ and $\mathbf{p}^{[k]} \leftarrow \mathbf{p}^{[k+1]}$, then repeat from step 2.

9. Use the latest model parameters $\mathbf{p}^{[k+1]}$ to calculate any parameters or functions of interest, for example, the birth rate and death rate at the sample points (β and μ). This completes the process.

Data Requirements

In order to fit the SPSP model f subject to the constraints $T\mathbf{p} = 0$, there are some requirements of the data and knots:

- The data space must be defined in d dimensions, that is, the data points and the knots should not be able to be described in $d - 1$ dimensions (e.g. if $d = 3$, the data points and knots should not all lie in one plane).
- $K \geq M$ so that the parameters can be constrained to satisfy the equality constraints $T\mathbf{p} = 0$ and still have degrees of freedom left over for fitting. (K is the number of knots, $M = \binom{m+d-1}{d}$, where m satisfies $2m > d + 1$.)
- $nC \geq 2K$, to ensure the Jacobian matrix has at least as many rows as columns so that the constrained model parameters can be estimated. (n is the number of sample points and C is the number of physiological stages.)

It is also desirable for the data to have a range of time much greater than the maximum age of the eggs. This is advantageous because in calculating the population density $\eta(x_1, \dots, x_{d-1}, t_1, \alpha_C)$ for the oldest population members at the first sample time t_1 , the birth rate is evaluated at time $t_1 - \alpha_C$, that is, beyond the time boundary of the data. In general, models do not extrapolate well, so it is a good idea to minimize the amount of extrapolation if possible.

2.3.7 Variance estimation

Point estimates and function estimates are obtained using the final model from the process detailed in previous sections. To measure the uncertainty of these estimates, variance or confidence intervals could be obtained from the covariance matrix for the model parameters. This analytic method is adopted by Wood (2000) who suggests a posterior covariance matrix for the model parameters

$$\text{Var}(\mathbf{p}_z) \approx \hat{\sigma}^2 (J_z^T W^2 J_z + \lambda_\beta S_{\beta_z} + \lambda_\mu S_{\mu_z})^{-1}$$

where $\hat{\sigma}^2 = \frac{\|W(\mathbf{y}-\mathbf{f})\|^2}{\text{tr}(I-A)}$ and A is the influence matrix such that $\mathbf{f} = A\mathbf{y}$. This follows from Silverman (1985) who derived his Bayesian confidence intervals from the method of Wahba (1983), using a simpler prior.

The analytic approach above does not cater for inequality constraints to which the SPSP model is subject. The constraints depend on the data so an analytic form of a variance estimate would differ for each application of the SPSP model. It therefore makes sense with the SPSP model to employ bootstrapping for variance estimation.

Wang & Wahba (1995) compared Bayesian confidence intervals with various intervals obtained from parametric bootstrapping and found that the approaches resulted in very similar intervals, lending credence to both. They concluded that, relative to the Bayesian method, bootstrap confidence intervals are straightforward to understand and to use with any distribution, and they provide better coverage when the sample size is small. The drawback of bootstrapping is that it is very computer intensive, but computer time is relatively cheap.

In the SPSP model application of chapter 3, non-parametric bootstrapping is adopted. The

advantage of non-parametric bootstrapping is that it will retain the features of the original data such as over-dispersion which may have been missed in the distributional assumption for Y_{ji} . A disadvantage is that it requires the samples to be independent, which may not be the case for spatial data, in which case the variance could be underestimated as correlation is not accounted for. If there is a suspicion that the data are correlated, a coarser unit of sampling could be adopted or a parametric bootstrap employed.

A bootstrap re-sample of the data is obtained by sampling n times with replacement from the sample points. For each of the re-sample points, the observed population counts for each stage, age limits for each stage, environmental conditions, and time are retained. The SPSP model is then fitted to the bootstrap re-sample and the required statistic calculated. Once this has been repeated b times, the bootstrap estimates and the original estimate (from the SPSP model fitted to the original data) are ordered, resulting in the empirical distribution of the statistic of interest. The $\frac{\alpha}{2}$ lower and upper percentiles of this distribution make up the $(1 - \alpha)100\%$ confidence interval (this is the percentile method of Efron (1982)). The standard error of the estimate is the standard deviation of the empirical distribution divided by $\sqrt{b + 1}$.

The percentile method can also be used to obtain confidence intervals for the birth rate, death rate, or stage abundance functions. For each point x_i , the empirical distribution of the function value at that point (for example, f_{1i}) is required. $(1 - \alpha)100\%$ confidence limits are found using the percentile method for each of these distributions separately. The $(1 - \alpha)100\%$ confidence interval for the whole function is made up of the confidence limits at each of the points and so will consist of predictions from functions fitted to different re-samples. The confidence intervals should contain the true function for $(1 - \alpha)100\%$ of the range of the data

space, and should be interpreted ‘across the function’ rather than point-wise.

If the parametric bootstrap were preferred, bootstrap re-samples can be obtained by generating new observations from the assumed (exponential family) distribution of Y_{ji} using the expected values from the fitted model. For example, if Y_{ji} is assumed to be Poisson distributed, a new bootstrap data point is generated from the $P(f_{ji})$ distribution. As with the non-parametric bootstrap method, a model is fitted to each bootstrap re-sample and the parameter of interest is calculated in each case.

2.4 Discussion

In this chapter, a population model and associated fitting method have been defined for solving the inverse problem from physiologically-structured data. A novel feature of the method is the fact that estimated vital rates can vary in space and time, but this is facilitated by the assumption that the death rates are independent of age. The model has the attributes of a ‘good’ model listed in the introduction to this chapter since

- the model makes use of known demographic mechanisms to incorporate all of the data into a single coherent model,
- heterogeneity in the data is accounted for by using thin plate splines to model the birth and death rates in terms of environmental variables,
- thin plate splines are used so that no restrictive parametric forms need be assumed for the model unknowns,
- the model should be straightforward to implement as the fitting procedure described in

the previous section is self-regulating (parameter selection is automated).

The SPSP model provides a general approach to modelling spatio-temporal population data that are physiologically-structured. In particular, it should help to provide improved birth and death rate estimates, and should give insight into how birth and death rates and population density relate to environmental conditions.

Because the model unknowns (the birth and death rates) are represented in a very flexible manner and the model assumptions are mild, the SPSP model is widely applicable. Using thin plate splines to represent the birth and death rate functions also avoids mis-specification bias which may occur when an incorrect parametric form is used. This can be more important than is often assumed: Wood & Thomas (1999) found that predictions of population dynamics can occasionally be drastically sensitive to model selection, despite this sensitivity being missed by usual sensitivity analysis methods. They suggested that the inclusion of phenomenological elements in models and assumptions about unknown causes of population dynamics should be discouraged.

Precision is gained through the inclusion of demographic structure in the form of the non-linear model, and by accounting for heterogeneity in the data with respect to environmental conditions. Involving demographic processes in the model form means that data from the whole population can be included in one comprehensive model, and separate models are not required for each developmental stage. As well as being convenient, this is of particular importance for egg production methods in fish stock assessment, which have traditionally used counts from only the first physiological stage in the calculation of the egg production estimate (the estimated total number of births). This traditional practice is very wasteful

especially considering information about mortality is contained the observed abundances of all stages (Wood & Nisbet 1991), as demonstrated by the SPSP model.

The SPSP model could be further developed to use generalized additive models (GAMs) rather than thin plate splines to represent the model unknowns. GAMs are sums of smoothing functions, such as thin plate splines. GAMs should add greater flexibility since the modeller would be able to choose to have additive as well as interactive environmental effects. The more environmental conditions included in the model, the more difficult the integration of the death rate, the integration of the population density and the differentiation involved in the Jacobian matrix may be. Appendix B shows the analytic integration of the death rate, the analytic derivatives in the Jacobian matrix, and the numerical integration of the population density for three dimensional case ($d = 3$). As d increases, it may not be feasible to perform the integrations and differentiation analytically and so numerical schemes may have to be used. If additive models consisting of several smoothing functions of low dimensions replaced the multi-dimensional thin plate splines, the integration and differentiation may be made easier. The inclusion of GAMs would require an adaption of the model fitting process.

Alternative strategies for variance estimation are also worth investigating. The non-parametric bootstrap assumes that the samples are independent, which may not be the case for some spatially distributed data for example. A parametric alternative, or different sampling units may useful.

Once coded, the SPSP model should be straightforward to implement. In the next chapter, the SPSP model is shown in operation, and its practicability is investigated.

Chapter 3

Application of a Smooth Physiologically-Structured Population Model

3.1 Introduction

Chapter 2 describes a smooth physiologically-structured population (SPSP) model and explains a method of fitting the model to stage-structured population counts. The purpose of this chapter is to test the model's performance as a means of obtaining reliable information about a population, and as a practical method for straightforward application to physiologically-structured count data. This is accomplished using a simulation study (section 3.2), and an application to Dover sole (*Solea solea*) egg data from the Bristol Channel (section 3.3).

Simulation testing is the only way to check that the model can indeed produce reliable birth and death rate estimates, since the true birth and death rates are known and properties of estimators are analytically intractable. The quantity of random error included in the data can be controlled so the performance of the model can be tested for different situations.

There is usually a set of associated problems unique to any given real data set, so application of the model to the Bristol Channel sole data is useful for seeing how practicable the model is.

In both situations, the model was fitted using the process described in the previous chapter written as a C program, `fitmodel.c`, which can be found in appendix C. `fitmodel.c` makes use of `matrix.c`, `gcv.c` and `qp.c` which perform matrix algebra, generalized cross validation for multiple smoothing parameters and quadratic programming respectively. These latter three libraries are freely available as part of the R package `mgcv` (www.cran.r-project.org, Wood 2001).

3.2 Simulation Testing

3.2.1 Simulated data

The simulated data were created using two environmental variables (x_1, x_2) and time (x_3). This corresponds to the variables available for the sole data described in section 3.3. Appendix B describes the form of the model components, Jacobian matrix and wiggleness penalty matrix for this $d = 3$ situation. For the simulation study, the environmental variables consist of 36 co-ordinates of a spatial grid, repeated for three sample times, $\{-40, 0, 40\}$. Sixteen

knots were selected over the spatial grid for each knot time. The ‘knot times’ (k_3) are the same as the sample times. The sample points and the knots are both shown in figure 3.1. The total number of sampled points is $n = 108$ producing 432 stage counts since there are four developmental stages, and there are a total of $K = 48$ knots.

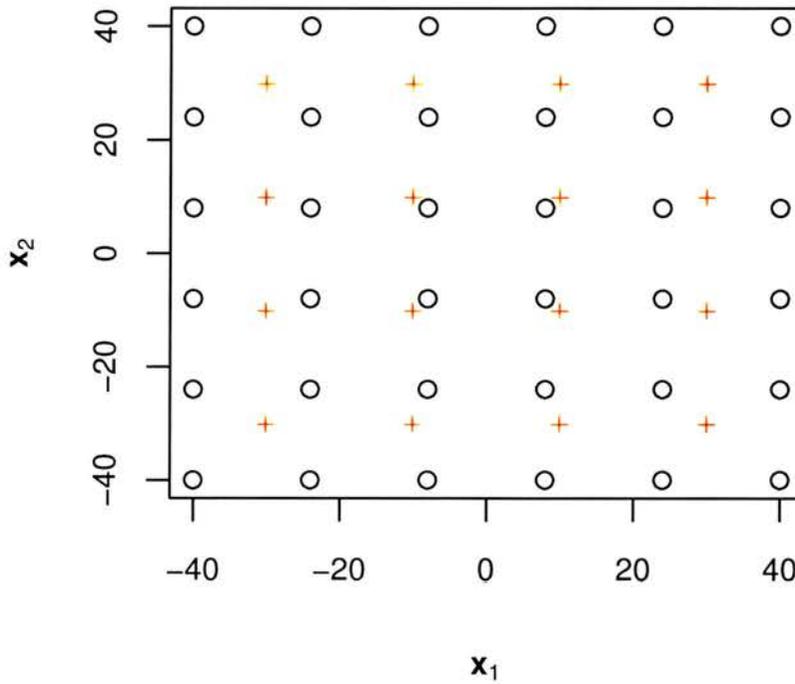


Figure 3.1: *Spatial sample points (black circles) and knots (red crosses) for sole egg data. These were the same for each sample time $\{-40, 0, 40\}$.*

Note that the sample points have the same range in each dimension and are centred about zero. If the values of x_1, x_2 and x_3 have small range but are of large magnitude then the polynomial basis terms can be close to co-linear causing numerical problems. Such problems were avoided by ensuring the sample points were symmetric in range about zero for each

dimension. The sample points have the same range in each dimension because this prevents anisotropic smoothing problems. Such problems can occur because the smoothness penalties and the spline basis penalties are isotropic in x_1, x_2 and x_3 . The penalties measure the wiggleness of the splines, so for example, if the range of x_1 is much larger than the range of x_2 and number of wiggles ‘allowed’ is the same per unit distance in each direction, the fitted surface will be smoother over x_2 than over x_1 . This may only be a problem for ‘related’ covariates such as longitude and latitude, but not for ‘unrelated’ covariates such as longitude and time. The choice of knots follows from the choice of sample points, so the knots are also reasonably spread and symmetric about zero.

Counts in four developmental stages were created using stage age limits which change with time but not space. The ages at entry to and exit from each of the stages are shown for each sample time in figure 3.2. (The R function `agelims.f` used to generate these limits can be found in appendix C.) The SPSP model is more suited to more frequent sampling so that the time lapse between samples is less than the maximum age individuals can reach. However, this is not the case for the sole egg data to which the model is applied in section 3.3, so the model was tested out with infrequent sampling in the simulation study.

The model structure of section 2.2 was used to construct birth and death rates, and hence stage populations, having selected some constrained model parameters. The data \mathbf{y} consist of evaluations of the stage populations at the sample points. Figures 3.3(a)–(c) and 3.4(a)–(c) show the birth and death rate functions for each sample time with the locations of the sample points marked. Figures 3.5I(a)–IV(c) show the stage populations with the sampled stage counts.

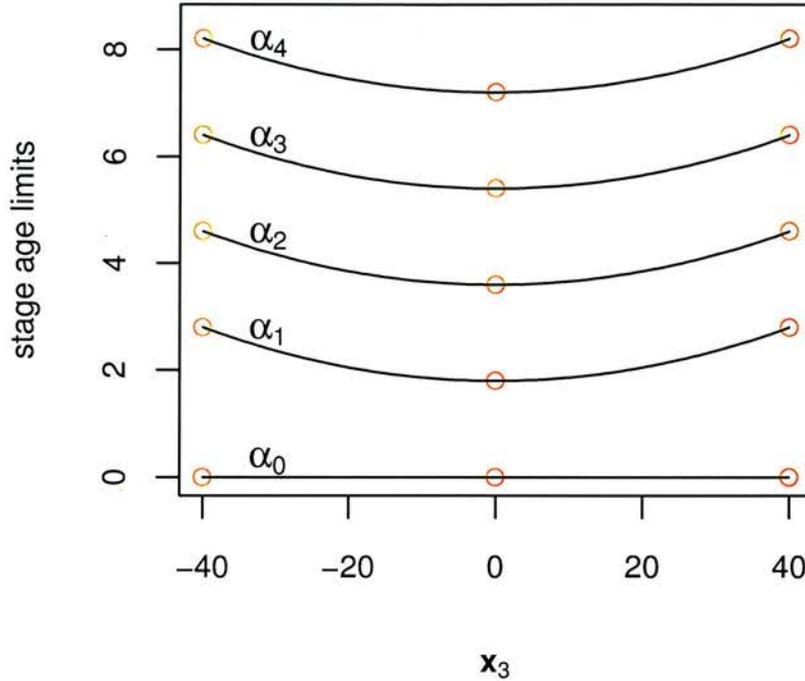


Figure 3.2: Age limits $\alpha_0, \dots, \alpha_4$ against time x_3 for the four developmental stages. The black lines show the age limits as functions of time while the red dots show the age limits for each sample time used to create the simulated data. The age limits do not differ over space.

Random errors (noise) were added to the stage counts to create data sets for modelling. This was achieved by generating the (noisy) stage counts from normal distributions with means equal to the ‘perfect’ stage counts shown in figure 3.5, and standard errors equal to 10% and 40% of those means. This is not very realistic considering the population data the model is designed for, so random errors were also introduced to the stage counts by generating the

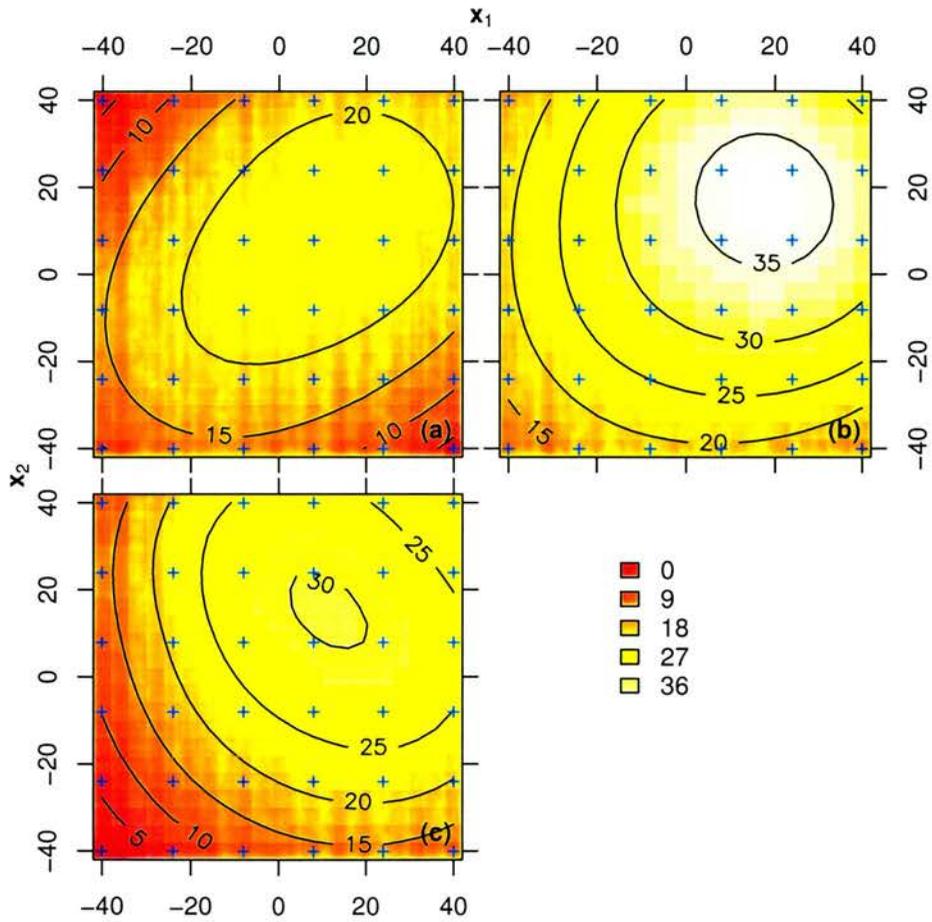


Figure 3.3: Simulated birth rate functions for the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.

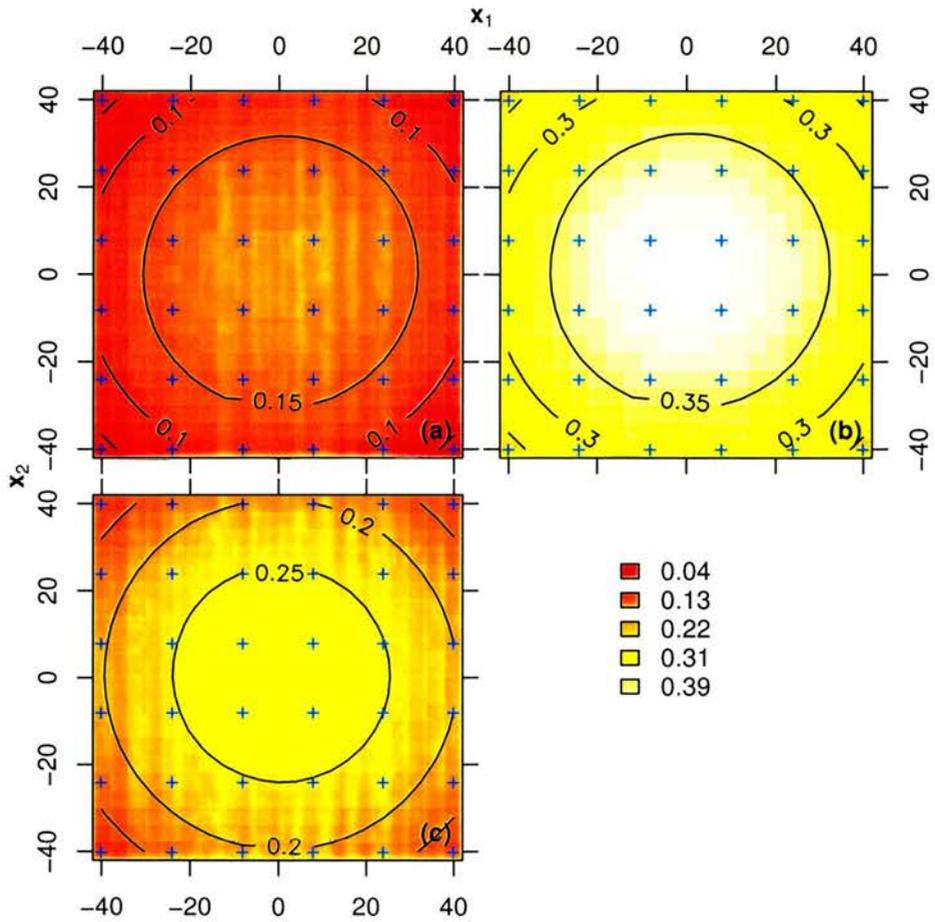


Figure 3.4: Simulated death rate functions for the three sample times (a) -40, (b) 0, (c)

40. The sample point locations are shown as blue crosses.

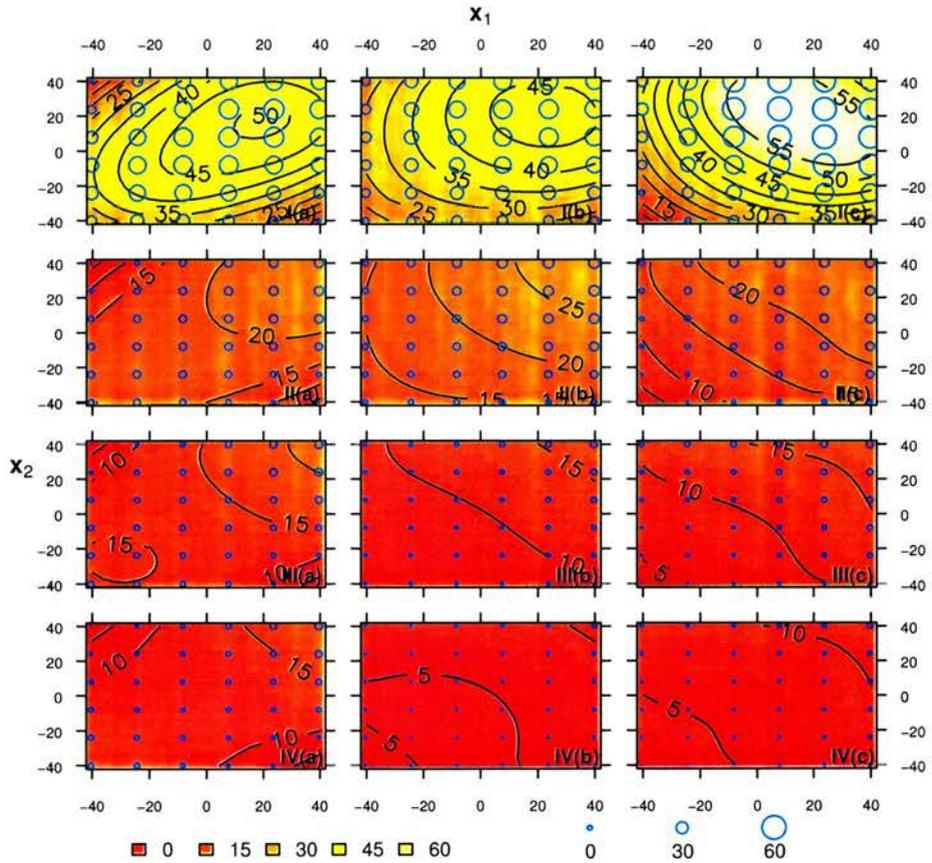


Figure 3.5: Stage populations in each of the four developmental stages (I–IV) and three sample times (a) -40, (b) 0, (c) 40, together with the sample stage counts shown as circles. Circle size is proportional to population size. (The figure layout is such that developmental stage increases from top to bottom and time increases from left to right.)

counts from Poisson distributions with means equal to the ‘true’ stage counts, t_{ji} :

$$\text{‘normal data with 10% noise’}: Y_{ji} \sim N(t_{ji}, (0.1t_{ji})^2),$$

$$\text{‘normal data with 40% noise’}: Y_{ji} \sim N(t_{ji}, (0.4t_{ji})^2),$$

$$\text{‘Poisson data’}: Y_{ji} \sim P(t_{ji}).$$

The R function `addnoise.f` in appendix C performed the task of creating the data sets in this manner. Examples of the resulting noisy stage counts are shown in figures 3.6I(a)–IV(c), 3.7I(a)–IV(c) and 3.8I(a)–IV(c). A couple of examples of Poisson data and a couple of examples of normal data with 40% noise were created. Most of the simulation study focusses on the normal data with 10% noise for which 100 data sets were simulated.

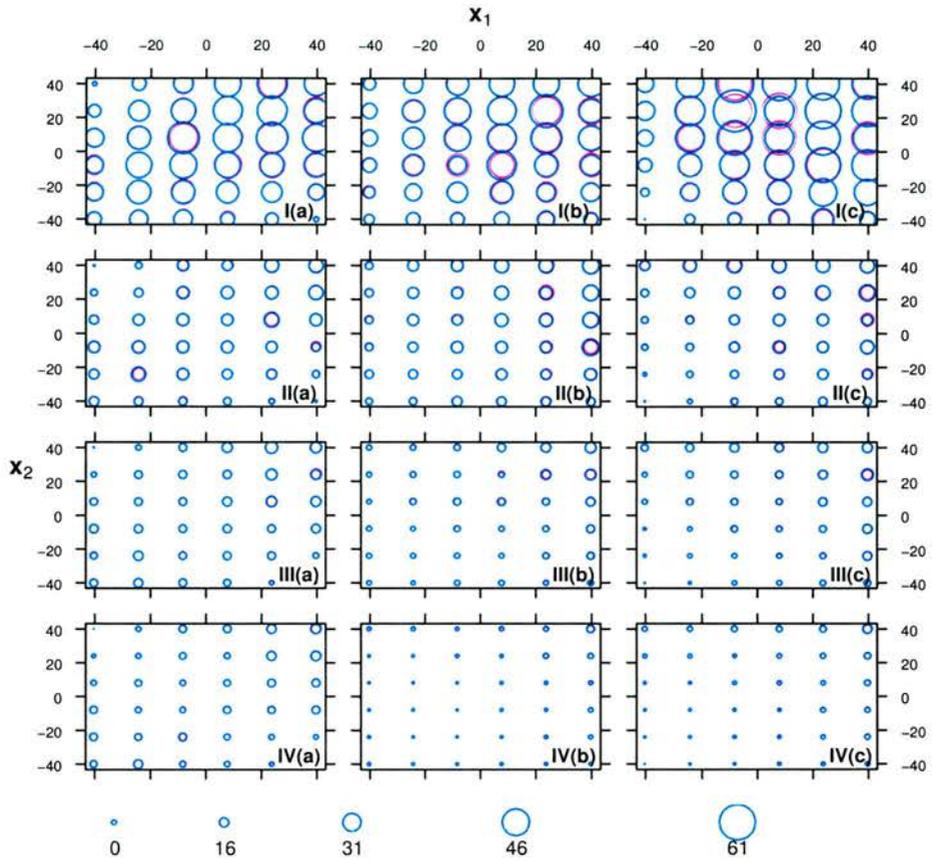


Figure 3.6: Stage counts for simulated normal data with 10% noise in each of the four developmental stages (I–IV) and three sample times (a) -40, (b) 0, (c) 40, for generated data set 96, shown as royal blue circles. The original stage counts are shown as pink. The pink circles do not show up when the noisy counts differ little from the original counts. Circle size is proportional to population size.

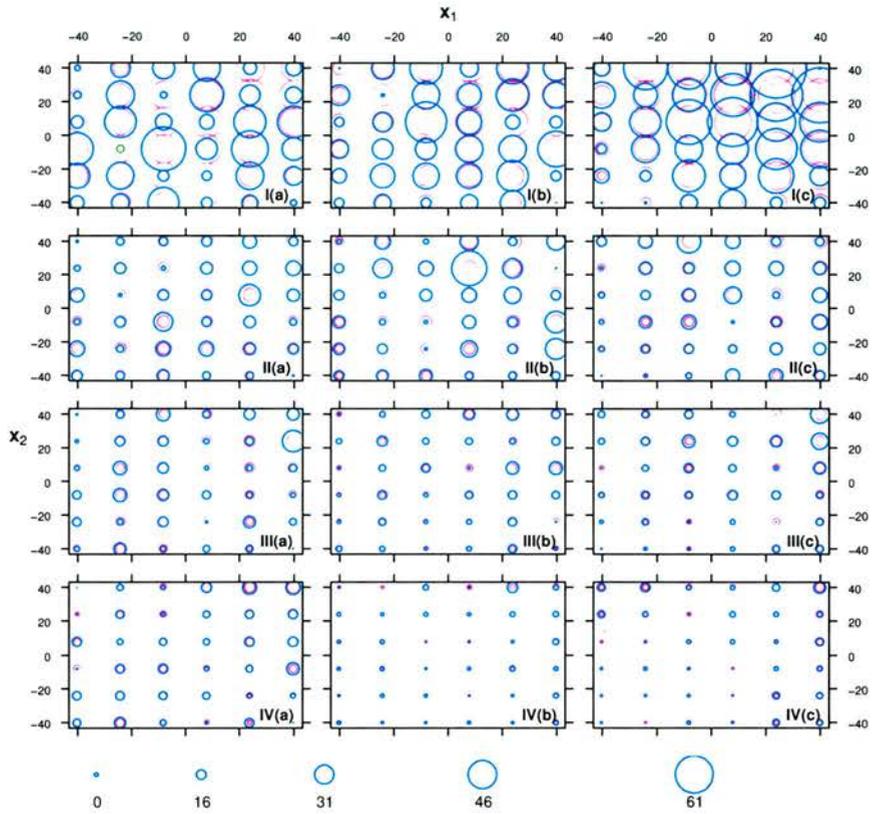


Figure 3.7: Stage counts for simulated normal data with 40% noise in each of the four developmental stages (I-IV) and three sample times (a) -40, (b) 0, (c) 40, for generated data set 1, shown as royal blue circles. The original stage counts are shown as pink. The pink circles do not show up when the noisy counts differ little from the original counts. A green circle indicates a negative value (there are two for this set). Circle size is proportional to population size.

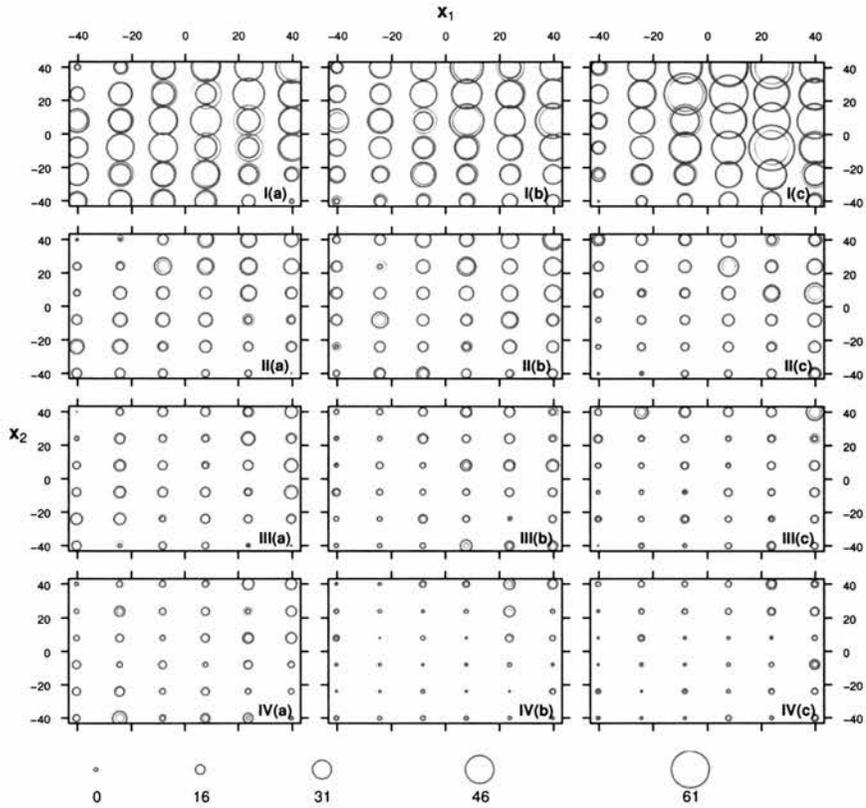


Figure 3.8: Stage counts for simulated Poisson data in each of the four developmental stages (I-IV) and three sample times (a) -40, (b) 0, (c) 40, for generated data set 1, shown as royal blue circles. The original stage counts are shown as pink. The pink circles do not show up when the noisy counts differ little from the original counts. Circle size is proportional to population size.

3.2.2 Models fitted to simulated data

The model was first fitted to the data without any noise, to make sure that the original birth rate, death rate and stage population functions were reproduced. Indeed they were, as can be seen in figures 3.9(a)–(d) which illustrate the model fit for the functions calculated on a finer grid (1323 points) than the original grid of points to which the model was fitted (108 points). As would be expected, the amount of variation in the data explained by the model, R-squared, was 1.

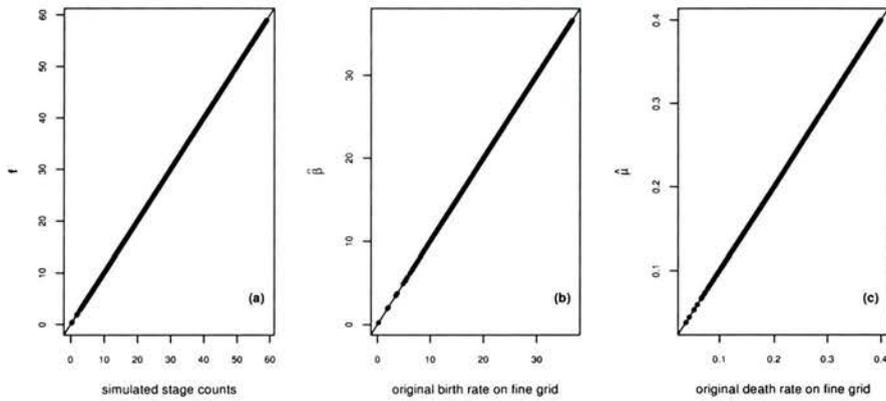


Figure 3.9: Model fit for simulation study data with no random errors, evaluated on a fine grid: (a) fitted stage populations versus original sampled stage populations, (b) estimated birth rates versus original birth rates, (c) estimated death rates versus fitted death rates.

In the context of egg production methods for fish stock assessments, the parameter of interest to be obtained by modelling the egg population is the total number of eggs spawned, P_0 . In a more general context this is the total number of individuals born which can be obtained by integrating the birth rate over the explanatory variables and time for the survey region and survey period, or approximated more simply by summing the birth rate evaluated at a grid of points over the survey, and multiplying by the (space-time) volume represented by a

single grid point:

$$P_0 = \Delta x_1 \Delta x_2 \Delta x_3 \sum_{x_1} \sum_{x_2} \sum_{x_3} (\beta(\mathbf{x})) .$$

(The C routine for calculating P_0 is contained within the function `fitmodel.c`.) \widehat{P}_0 will be termed the ‘total birth estimate’. There are some eggs in the later stages of the simulated population at the first sample time (see figure 3.5(a)), implying the birth rate must have been non-zero prior to sampling. Since the summation over x_3 in the calculation of \widehat{P}_0 begins at the first sample time, not all eggs produced will be included in the total birth estimate for the simulated data. When a real survey is to be performed, it must be designed to ensure sampling occurs as close to the beginning and end of the spawning period as possible.

For the simulated data, $\widehat{P}_0 = 13\ 162\ 343$ and for the model fitted to these data, $\widehat{P}_0 = 13\ 162\ 300$, out by just 0.0003%. Results for the fitted model are given in the first row of table D.1 which can be found in appendix D to compare with the results for the models fitted to the normal data with 10% noise.

In order to obtain the realized coverage probability of the confidence intervals, 100 sets of normal data with 10% noise were created, and each was bootstrapped non-parametrically 99 times using the R routine `bootdata.f` (appendix C). The model was then fitted to each bootstrap re-sample. This resulted in 100 estimates of P_0 and associated 80%, 90% and 95% confidence intervals calculated from the bootstrap quantiles. The confidence intervals and their histograms are shown in figure 3.10. There can be reasonable trust in the confidence intervals since the realized coverage probabilities were fairly close to the confidence levels (see table 3.1).

Figures 3.10(d)–(f) give a good indication of coverage of the confidence limits for each fitted

Confidence level	Realized coverage probability	% of CIs too low	% of CIs too high
80%	0.81	17%	2%
90%	0.89	11%	0%
95%	0.97	3%	0%

Table 3.1: *Realized coverage probabilities of the bootstrap confidence intervals for the models fitted to normal data with 10% noise.*

model. There is a slight negative bias (the bias of the total birth estimates ranges from -6.7% to 3.5% with mean -1.5%) which explains why all of the simulation confidence intervals that fail to include the true P_0 (except for two 80% confidence intervals) are too low rather than too high. This issue requires further investigation to discover if this is a systematic problem with the method.

All of the models converged, with 75% of them converging within 5 steps. The amount of variation in the data sets explained by the fitted models ranged from 93% to 98%, showing that in each case the model fits well. Note that because random errors were added to the data, we would not want R-squared to be 100%, or the smoothing parameters would not be doing their job. Table D.1 in appendix D contains the results for the 100 fitted models.

The model fitted to data set 96 is used to illustrate the performance of a typical model selected at random. Figures 3.11(b) and (c) indicate that the model fits the data well. The red dots have a narrower spread than the black dots showing that the model resembles the original true data more closely than the noisy data do, and therefore the model has eliminated some noise. The model has also managed to reproduce the original birth and death rates

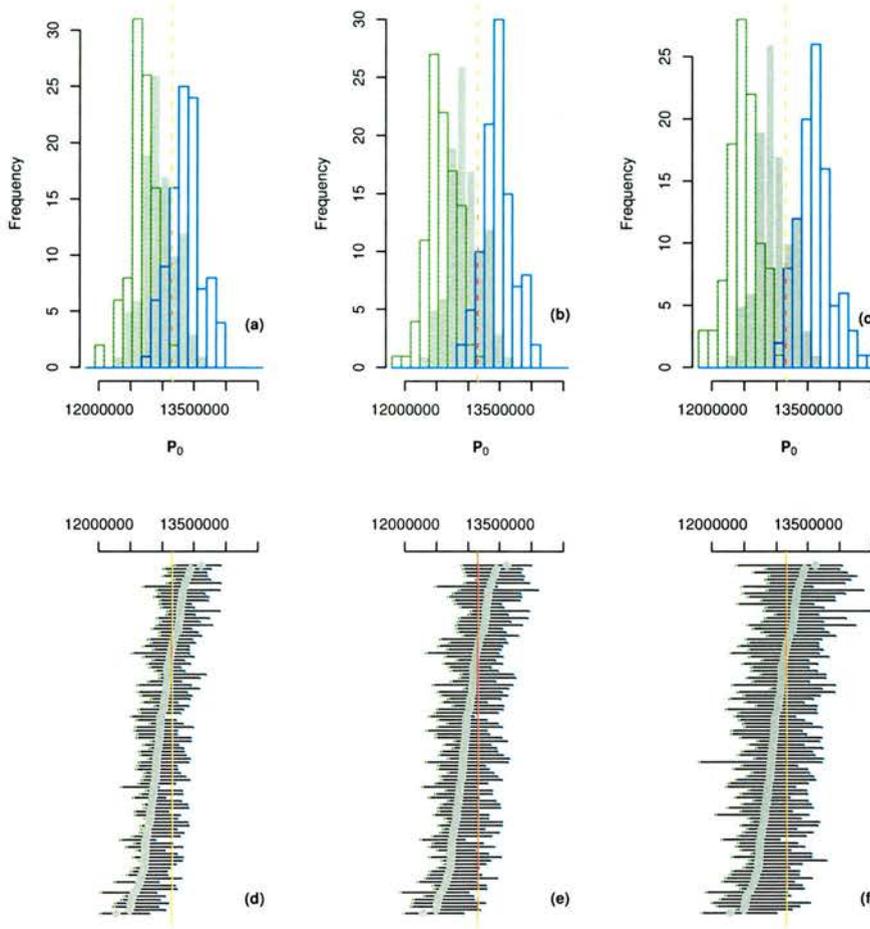


Figure 3.10: For the models with 99 non-parametric bootstrap re-samples fitted to 100 sets of simulated normal data with 10% noise: histograms of estimated total number of individuals born \widehat{P}_0 (grey), with lower (green) and upper (blue) confidence limits, with the true value of P_0 shown as a red dashed line, for confidence levels of (a) 80%, (b) 90%, and (c) 95%; the estimated total number of individuals born \widehat{P}_0 (grey) in ascending order, with associated lower (green) and upper (blue) confidence limits, with the true value of P_0 shown as a red line, for confidence levels of (d) 80%, (e) 90%, and (f) 95%.

fairly closely (see figures 3.11(d) and (e) and compare figures 3.12 and 3.13 to figures 3.3 and 3.4). The distribution of the bootstrap estimates of the total number of individuals born (\widehat{P}_0) is normal except in the extremes, so the bootstrap confidence intervals could have been replaced with normal ones providing the confidence level was sufficiently small. The outliers are excluded by the 90% (bootstrap) confidence interval (figures 3.11(f) and (g)).

The results for the model fitted to the normal data set 1 with 40% noise are shown in figure 3.14. The model now accounts for less of the variation in the data (R squared is 59%) but still is much closer to the original true data than the noisy data to which it was fitted (figures 3.14(b) and (c)). The estimated birth rate corresponds well to the true birth rate (figure 3.14(d)), though it is not so close a fit as for the models fitted to the data with 10% noise. The fitted death rate also performs well (figure 3.14(e)), with a slight underestimation for the larger death rate values. Despite two of the stage counts in the data being negative (which wouldn't happen in reality), the estimated birth and death rates, and hence stage counts, are all positive suggesting that the inequality constraints are functioning correctly. Figures 3.15 and 3.16 show that the fitted birth and death rates are good estimates of the originals with only slight differences in shape. The peak of the estimated birth rate at the final sample time is situated slightly too close to the upper range of x_1 and x_2 , and the estimated death rate is slightly less symmetrical in x_1 and x_2 than the original.

Looking at figure 3.14(f), the estimate of the total number of births (12 295 700) is negatively biased by 6.6% but the 90% confidence interval (10 306 500, 13 679 800) includes the true value of 13 162 343. The Q Q plot suggests that the bootstrap quantiles could not be replaced by a normal confidence interval since they are not normally distributed for a large part of the tails.

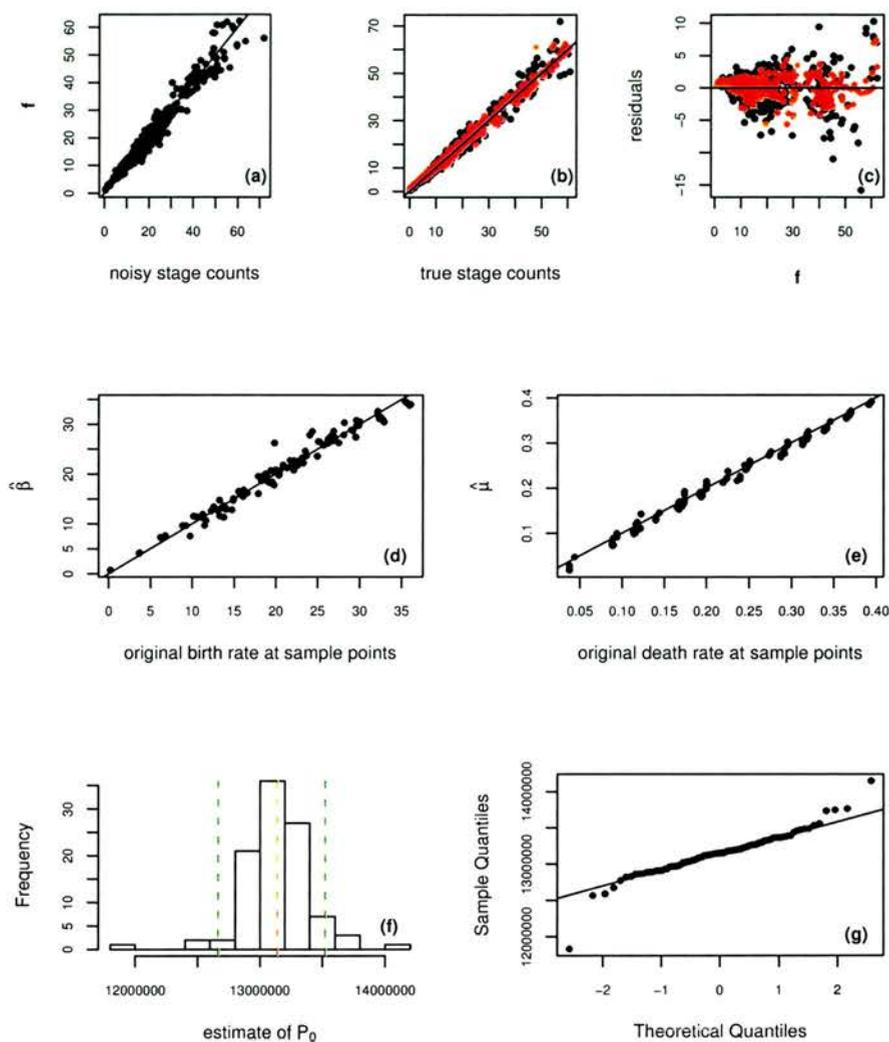


Figure 3.11: The model fitted to simulated normal data with 10% noise sample 96: (a) fitted stage populations versus noisy sampled stage populations, (b) noisy stage populations versus true sampled stage populations (in black) and fitted stage populations versus true sampled stage populations (in red), (c) residuals versus fitted values relative to the noisy counts (in black) and the true stage counts (in red), (d) estimated birth rates versus original birth rates, (e) estimated death rates versus fitted death rates, (f) histogram of bootstrap estimates of the total number of individuals born with the model estimate indicated by the red dashed line and the 90% confidence limits by the green dashed lines, (g) normal Q-Q plot of bootstrap results versus theoretical quantiles.

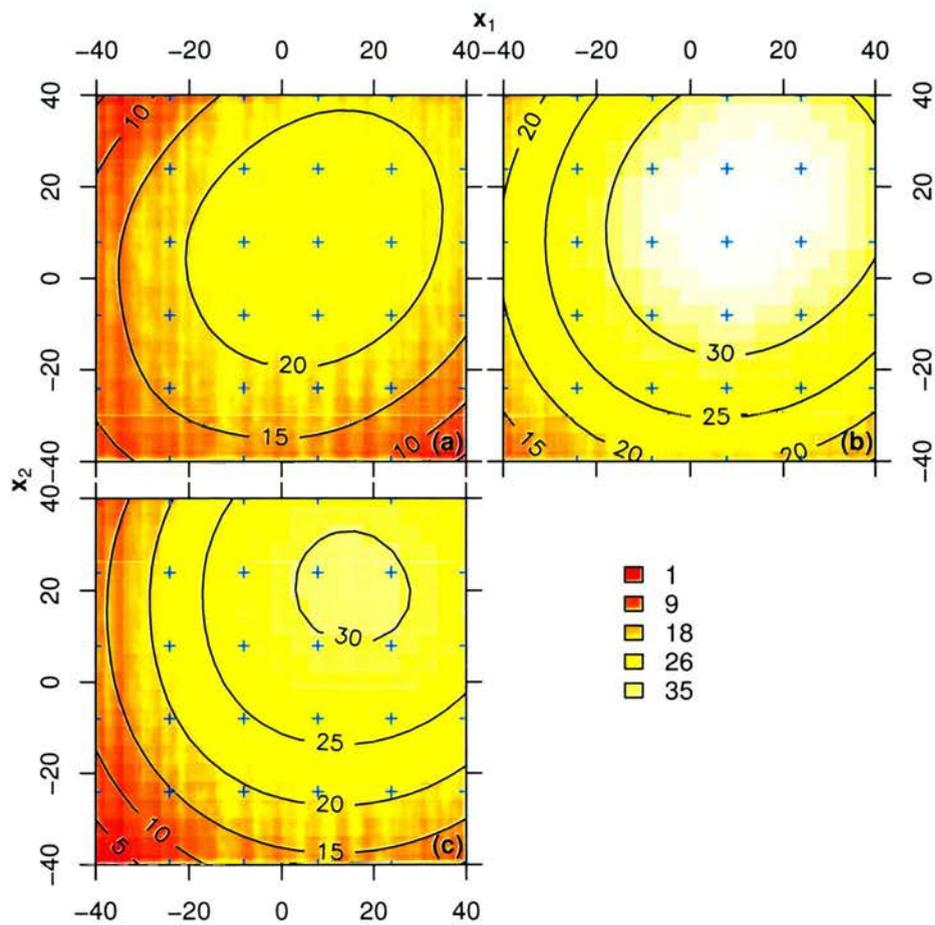


Figure 3.12: Estimated birth rate for the model fitted to normal data set 96 with 10% noise at the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.

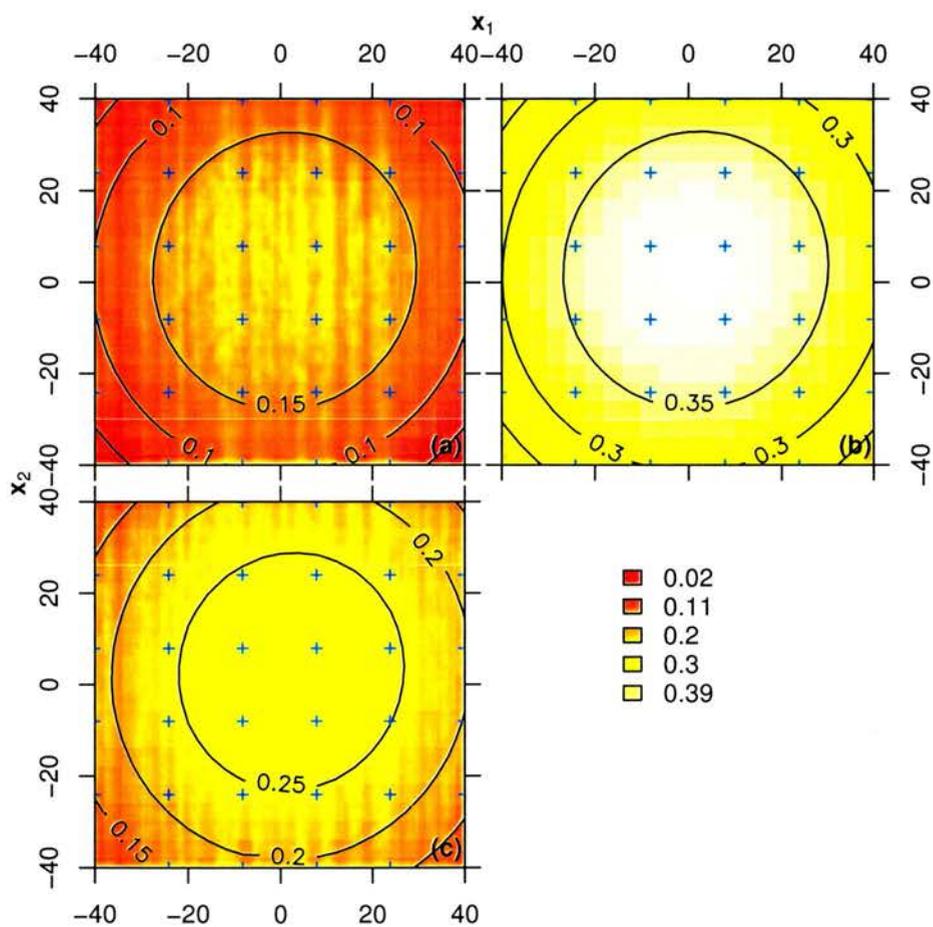


Figure 3.13: Estimated death rate for the model fitted to normal data set 96 with 10% noise at the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.

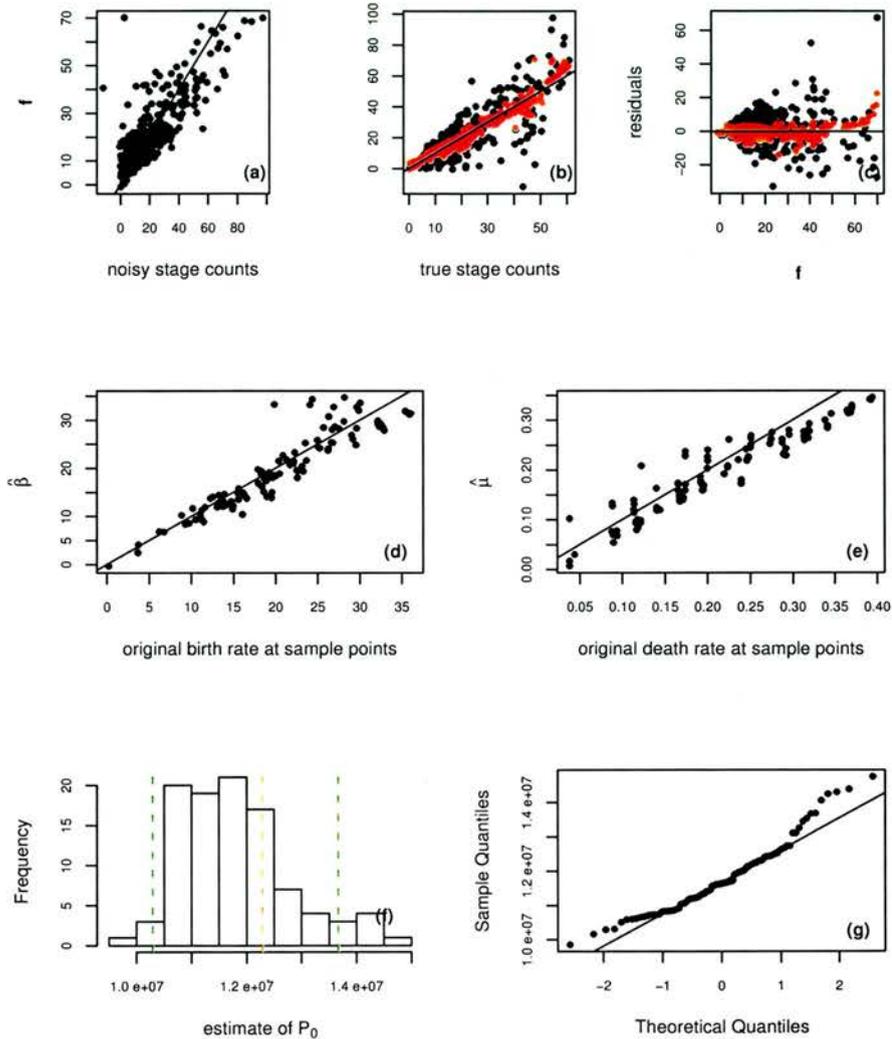


Figure 3.14: Model fitted to simulated normal data set 1 with 40% noise: (a) fitted stage populations versus noisy sampled stage populations, (b) noisy stage populations versus true sampled stage populations (in black) and fitted stage populations versus true sampled stage populations (in red), (c) residuals versus fitted values relative to the noisy counts (in black) and the true stage counts (in red), (d) estimated birth rates versus original birth rates, (e) estimated death rates versus fitted death rates, (f) histogram of bootstrap estimates of the total number of individuals born with the model estimate indicated by the red dashed line and the 90% confidence limits by the green dashed lines, (g) normal Q-Q plot of bootstrap results versus theoretical quantiles.

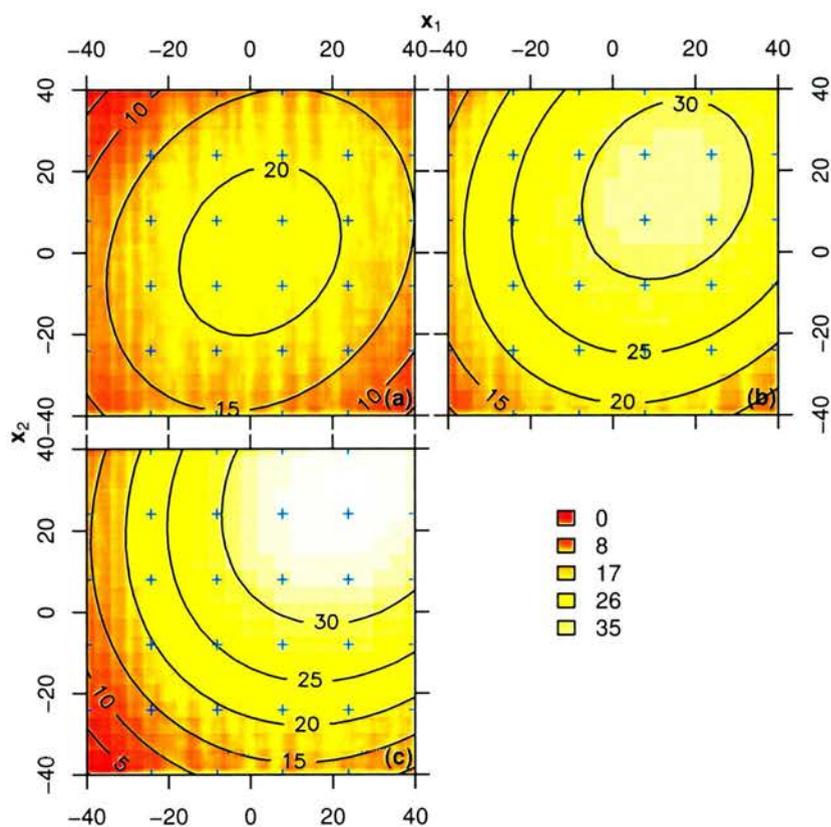


Figure 3.15: *Estimated birth rate for the model fitted to normal data set 1 with 40% noise at the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.*

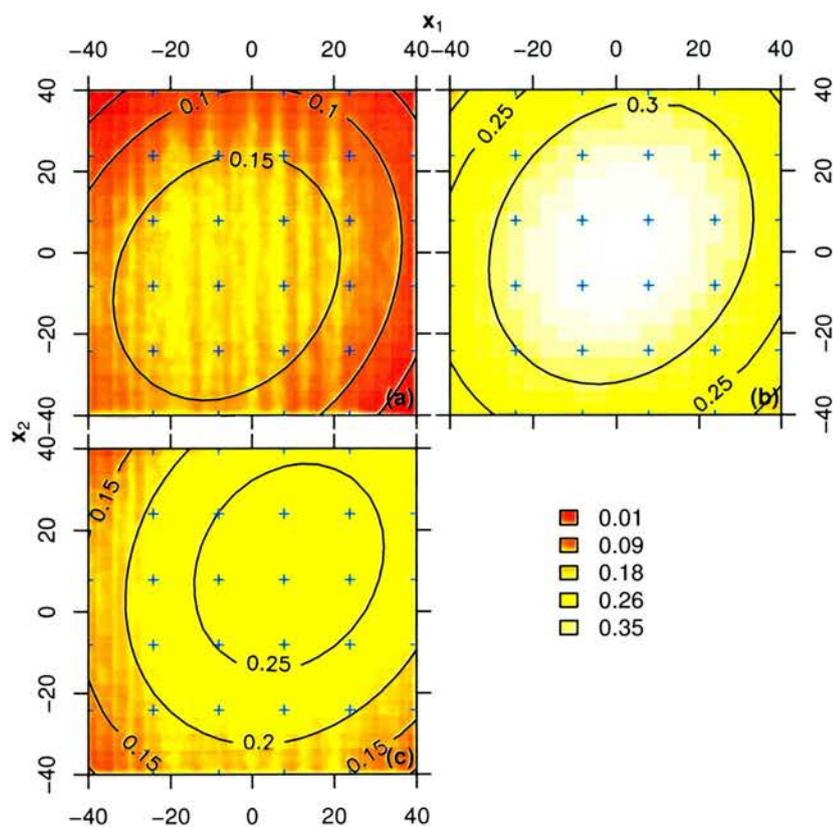


Figure 3.16: Estimated death rate for the model fitted to normal data set 1 with 40% noise at the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.

The results for the model fitted to the Poisson data set 1 is shown in figure 3.17. Again, the model accounts for most of the variation in the data (R-squared is 85%) as can be seen in figures 3.17(b) and (c). Figures 3.17(d) and (e) show that the fitted model underestimates the birth rate and the death rate for the large values of these functions. This leads to a 8.7% negatively biased total egg production estimate (12 012 700) and a 90% confidence interval (11 403 400, 12 570 600) which does not include the true value, since the estimate is very precise (the CV of the total egg production estimate is just 0.3%). The bootstrap estimates are symmetrical in distribution (figure 3.17(f)) and the confidence intervals could have been replaced by normal confidence intervals other than in the upper tail (figure 3.17(g)). Other than having smaller peaks, the birth rate estimate (figure 3.18) has a fairly close resemblance to the original, as does the death rate (figure 3.19). Both differ from the original in the same way as the model fitted to normal data set 1 with 40% noise.

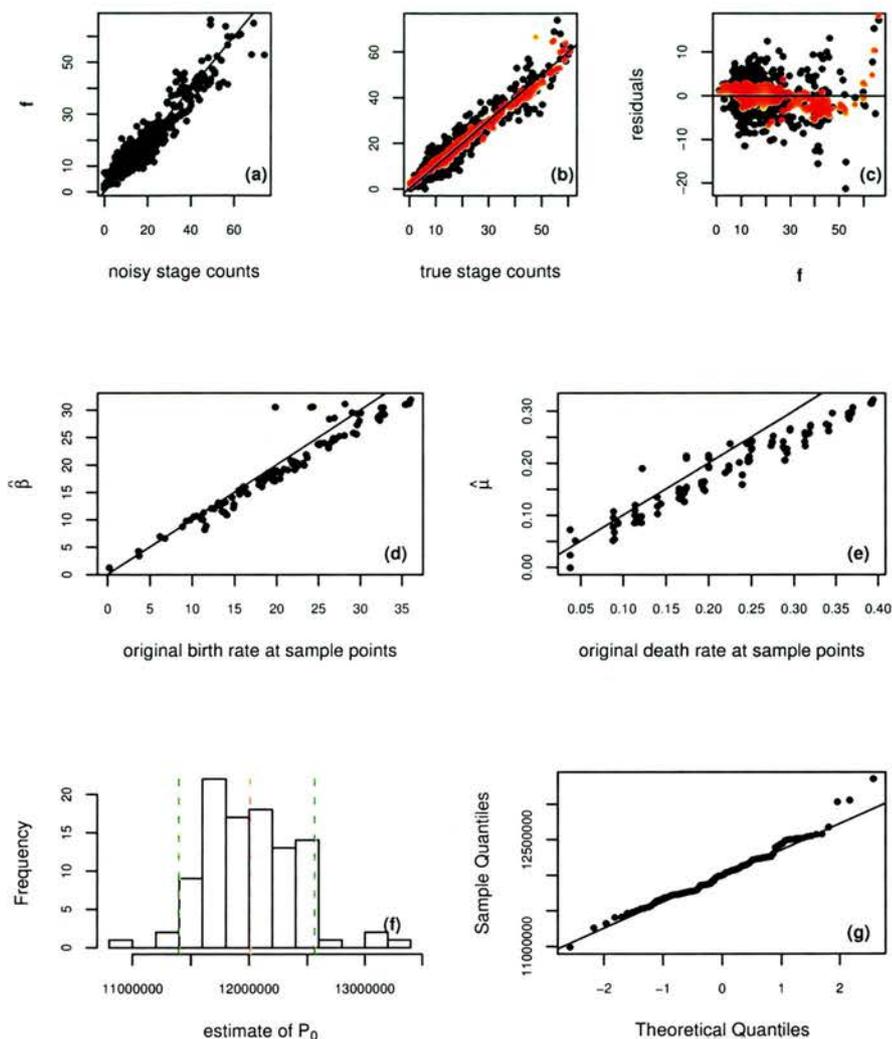


Figure 3.17: Model fitted to simulated Poisson data sample 1: (a) fitted stage populations versus noisy sampled stage populations, (b) noisy stage populations versus true sampled stage populations (in black) and fitted stage populations versus true sampled stage populations (in red), (c) residuals versus fitted values relative to the noisy counts (in black) and the true stage counts (in red), (d) estimated birth rates versus original birth rates, (e) estimated death rates versus fitted death rates, (f) histogram of bootstrap estimates of the total number of individuals born with the model estimate indicated by the red dashed line and the 90% confidence limits by the green dashed lines, (g) normal Q-Q plot of bootstrap results versus theoretical quantiles.

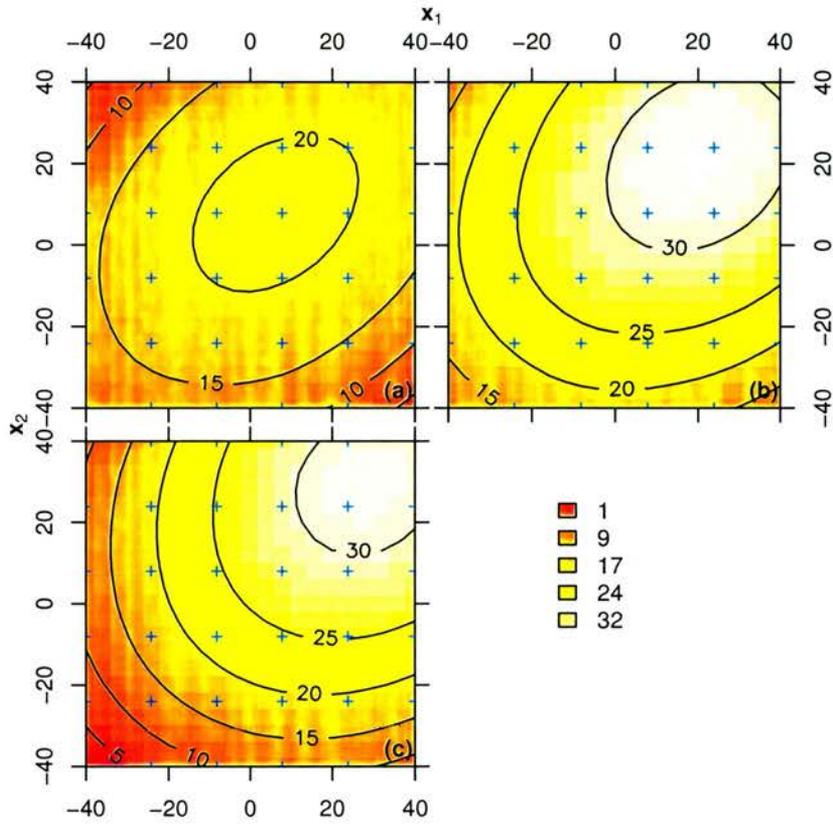


Figure 3.18: Estimated birth rate for the model fitted to Poisson data set 1 at the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.

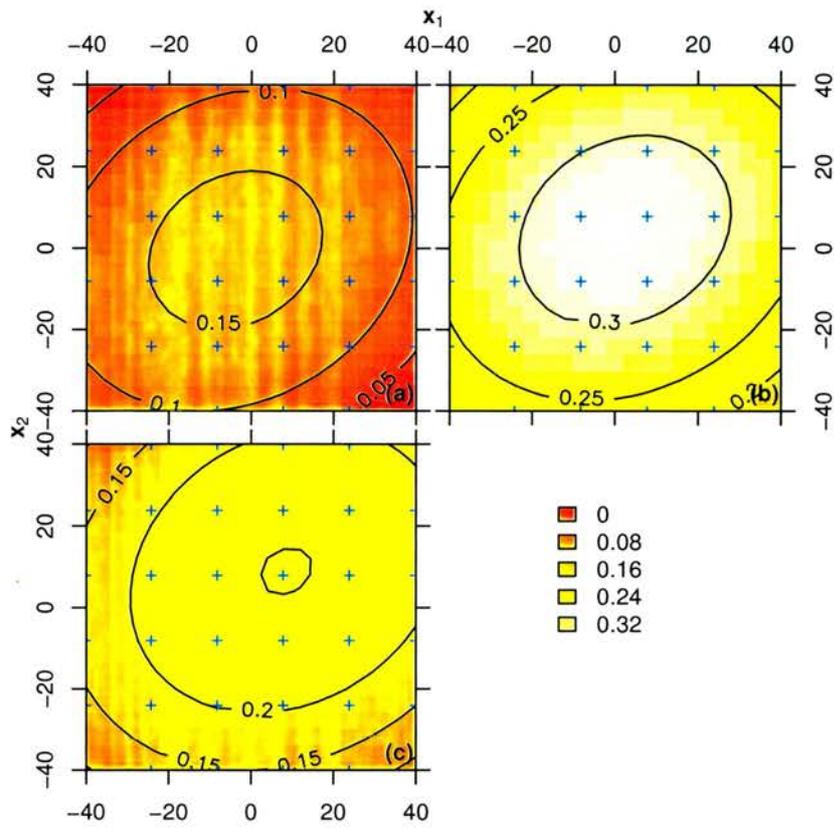


Figure 3.19: Estimated death rate for the model fitted to Poisson data set 1 at the three sample times (a) -40, (b) 0, (c) 40. The sample point locations are shown as blue crosses.

3.3 Application to Dover Sole (*Solea solea*) in the Bristol Channel

Fish egg surveys are undertaken in order to determine spawning rates, from which the adult fish biomass may be inferred using fecundity estimates. The approach divides into two main categories, the annual egg production method (AEPM) used for determinate spawners, and the daily egg production method (DEPM) for determinate or indeterminate spawners¹. Both of these methods are described in Gunderson (1993, chapter 4). The Dover sole from the Bristol Channel are determinate spawners (Horwood & Greer Walker 1990) and AEPM is used to find their spawning stock biomass. The greatest source of uncertainty in egg production methods is in obtaining the spawning rates from the egg counts, rather than in converting these rates into fish stock estimates, so improvement of this aspect of the method will have greatest impact on the precision of the adult biomass estimates (Borchers et al. 1997).

Egg data consist of observations at several locations throughout the spawning region and, for the AEPM, several times throughout the spawning season. Usually egg counts are recorded for a number of different developmental classes, and other environmental information may be taken, such as bottom depth and water temperature for the location of the eggs.

An example of such egg data are some Dover sole (*Solea solea*) egg data from the Bristol Channel collected in 1990, shown in figure 3.20.

The adult population that spawned the eggs observed in this data set occupied (and some

¹A species consists of determinate spawners if the total number of eggs produced by an female over an entire breeding season can be determined by counting the oocytes before the season begins.

3.3 Application to Dover Sole (*Solea solea*) in the Bristol Channel

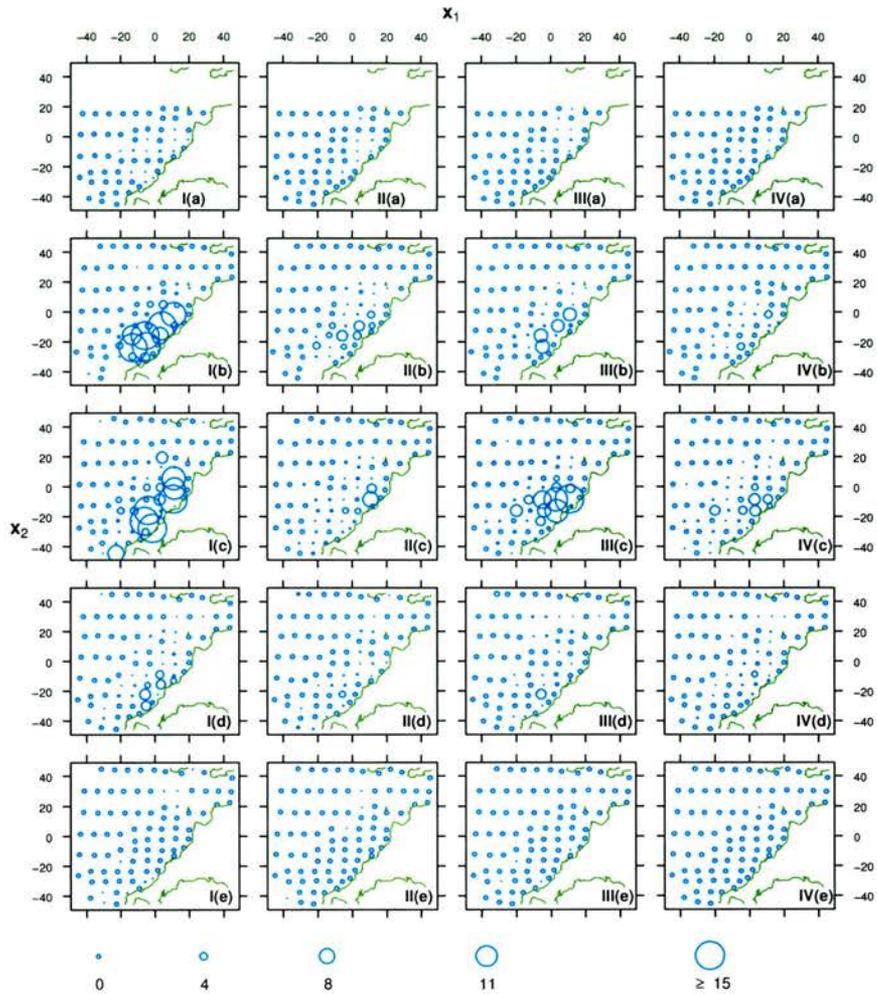


Figure 3.20: Stage abundances per square metre in each of the four developmental stages (I–IV) and five sample times (a) 49.5, (b) 70.0, (c) 93.5, (d) 110.5 and (e) 140.5 Julian days, for the sole data from the Bristol channel. Circle size is proportional to population size, as shown in the legend. (Note that the figure layout is the reverse of that for the simulation study. Here, developmental stage increases from left to right and time increases from top to bottom.)

veterans may still occupy) ICES² divisions VIIIf–g, approximately. ICES considers the sole of

²International Council for the Exploration of the Sea

these divisions to be fished outside safe biological limits and so good estimates of spawning stock biomass (SSB) are essential for deciding safe fishing mortality levels. Fishermen from the UK, Belgium, France and Ireland elicit a good price for this popular restaurant fish, so they too have an interest in the long-term survival of the population. Though sole SSB is currently estimated using catch-per-unit-effort (adult) data from commercial fleets and surveys, egg production methods are also appropriate (spawning and nursery grounds are well defined), and indeed were one of the methods tried in 1990. Although an egg production estimate for 1990 is no longer of vital importance, it is a useful exercise to apply the SPSP model to this sole egg data set. If the SPSP model produces reliable estimates of egg production fairly simply, then egg production methods might become a more attractive option for stock assessment scientists.

The sole data suffer from the problems typical for physiologically-structured egg abundances — they are sparse and noisy, and were expensive to gather. It is therefore difficult, but imperative, to model the data carefully in order to obtain good spawning rate estimates.

For the SPSP model to be relevant for the sole data, the model assumptions (see section 2.2.1) must be applicable. According to Horwood (1993), the Dover sole that spawn in the Bristol channel are a closed population; it seems reasonable to assume that the birth and death rates of the sole are smooth functions of the covariates, space and time; the death rate is likely to be independent of age since the most common cause of death is being eaten; age limit estimates are available for each stage and so in the absence of any further information, they are assumed to be the true values; the data consist of abundances and so it is common practice to assume that the observations are from a Poisson distribution. There may be a problem with the last assumption since the data are not raw abundances but have been scaled to population

per square metre of sea surface area. It is uncertain in the absence of further information, whether the scaled data is under- or over-dispersed relative to the Poisson distribution. Still, the Poisson assumption might reasonably be expected to provide satisfactory point estimates even if the assumption is invalid, and non-parametric bootstrapping ought to cope with most of the change in dispersion.

However, the fitted model *might* be adversely affected. Suppose the Poisson assumption has been made. If the data are under-dispersed, then the fitted model may be too smooth, because smoothing parameter selection is made on the premise of Poisson variation, so the smoothing parameters will be too large for under-dispersed data. The opposite effect is obtained if the data are over-dispersed: the fitted model may be too wiggly.

3.3.1 The sole data

At a grid of longitude and latitude locations in the Bristol Channel, samples of sole eggs were taken in 1990. A description of how the samples were taken is found in Horwood (1993, p. 231). For each sample, the number of eggs per square metre was recorded for each developmental stage. This procedure was repeated for five periods throughout the spawning season, from February till May, and the staged abundances per square metre are shown in figure 3.20. It is clear that a large proportion (61%) of the samples have no eggs recorded for any stage. The figure also shows that the population peaked around early April (the middle of the spawning season), when eggs were found throughout the region. At other times, most of the eggs were found in the south-east of the sample region.

The sample locations ranging from 50.00 to 51.75°N and -7.0 to -3.5°E, differ for each sample

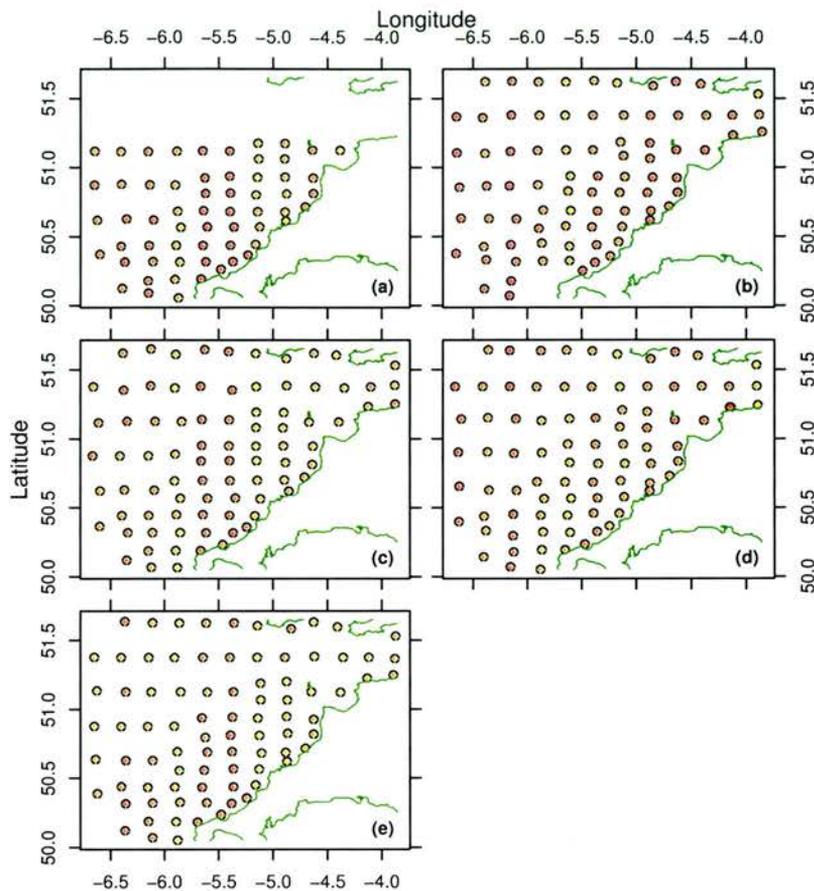


Figure 3.21: Original spatial sample points (black circles) and knots (red crosses) for sole egg data for the sample times in Julian days (a) 49.5 (18th February), (b) 70.0 (11th March), (c) 93.5 (3rd April), (d) 110.5 (20th April) and (e) 140.5 (20th May).

period, as shown in figure 3.21. Knots were placed at each of the $n = 394$ sample points (so $K = 394$), to ensure sufficient potential for very flexible splines, given the rather large size of some of the stage abundances (up to 35.923) relative to the large proportion of abundances less than one (90%). (Stage ‘abundance’ is abundance per square metre).

Age limits for the four stages of egg development are thought to vary with temperature, which changes over time, so age limits were estimated for each sample period using a modification

of Riley (1974)'s stage durations obtained from egg incubation experiments (Horwood 1993). The recorded age limits are shown in figure 3.22. The relatively long duration of stage three explains the higher population abundances for that stage, despite the eggs of all the stages being subject to the same death rate for any particular sample point.

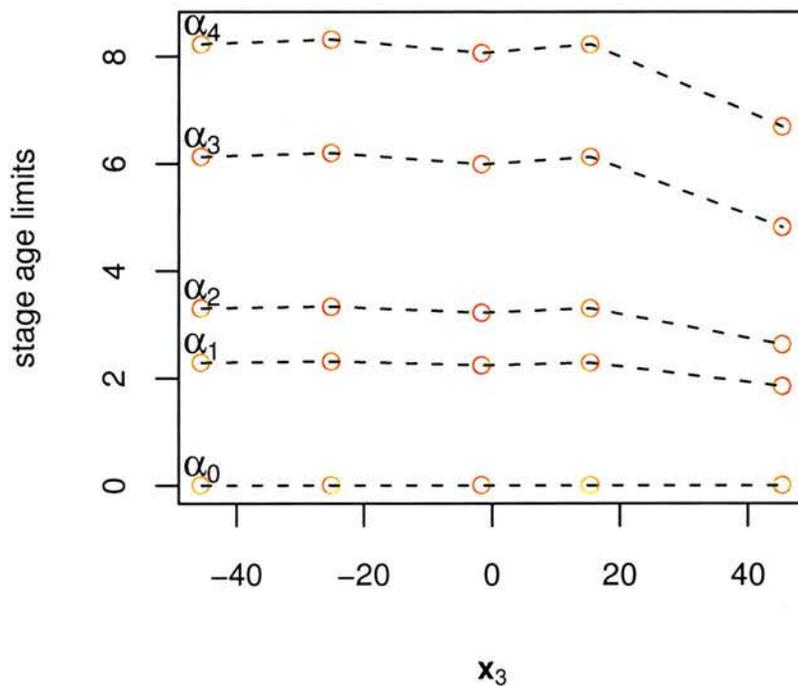


Figure 3.22: Age limits in days $\alpha_0, \dots, \alpha_4$ against time x_3 for the four developmental stages. The red dots show the age limits recorded for each sample time, the black dashed lines are just there to aid the eye in linking age limits for the same development stage. The age limits do not differ over space.

Because the data were originally used to provide snapshots of the egg population separately for each sample period, only a midpoint sample time (in Julian days) has been recorded for

each sample period. The effect of this is to impose the assumption that the entire survey region was sampled within one day, although in fact it took several days to survey the whole region. The sample times used for modelling are listed for the sample periods of the sole egg survey in table 3.2. Note that the SPSP model is capable of modelling using the true sample times, but these are not available for the sole data.

Sample period	Mid point	Sample time (x_3) in Julian Days
15–22 February	18th February	49.5
9–13 March	11th March	70
1–6 April	3rd April	93.5
17–24 April	20th April	110.5
18–23 May	20th May	140.5

Table 3.2: Sample periods and times for Dover sole data.

The Bristol Channel is at approximately 50°latitude, which means that the survey region does not represent a flat (grid) surface. To account for this the spatial variables (x_1, x_2) in degrees East and North can be transformed to kilometres from Greenwich Mean Time (GMT) and from the equator. This is be done by

- creating a set of GMT points \mathbf{v} which have the same latitude as \mathbf{x} but with longitude 0, and a set of equator points \mathbf{w} which have the same longitude as \mathbf{x} but with latitude 0,

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & x_2 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} x_1 & \mathbf{0} \end{bmatrix} ;$$

- calculating the great circle distances between \mathbf{x} and the GMT points to obtain x_1 in

nautical miles, and between \mathbf{x} and the equator points to obtain x_2 in nautical miles,

$$x_1 \leftarrow 60^\circ \times \arccos(\sin(x_2) \sin(\mathbf{v}_2) + \cos(x_2) \cos(\mathbf{v}_2) \cos(|x_1 - \mathbf{v}_1|))$$

$$x_2 \leftarrow 60^\circ \times \arccos(\sin(x_2) \sin(\mathbf{w}_2) + \cos(x_2) \cos(\mathbf{w}_2) \cos(|x_1 - \mathbf{w}_1|)) ;$$

- converting these distances from nautical miles to kilometres by multiplying by 1.852.

This is coded in the R routine `converttokm.f` which can be found in appendix C.2.

Unlike the simulated data, the variables $\mathbf{x} = (x_1, x_2, x_3)$ evaluated at the sample points are not symmetric about zero, nor do they have the same range in all dimensions. To avoid potential problems (outlined in section 3.2), these sample points were linearly re-scaled, by first making them symmetric in range about zero,

$$x_1 \leftarrow x_1 - 0.5(\min(x_1) + \max(x_1))$$

$$x_2 \leftarrow x_2 - 0.5(\min(x_2) + \max(x_2))$$

$$x_3 \leftarrow x_3 - 0.5(\min(x_3) + \max(x_3)) ;$$

and then ensuring they all have the same range as the time variable (since this means the age limits do not need adjusting),

$$x_1 \leftarrow x_1 \times \frac{\max(x_3) - \min(x_3)}{\max(x_1) - \min(x_1)}$$

$$x_2 \leftarrow x_2 \times \frac{\max(x_3) - \min(x_3)}{\max(x_2) - \min(x_2)} .$$

The knots were similarly transformed (using the R routine `scale.f` in appendix C.2), and

the resulting covariates and knots are illustrated in figure 3.23. Because the above operations are linear, they do not affect the form of the model.

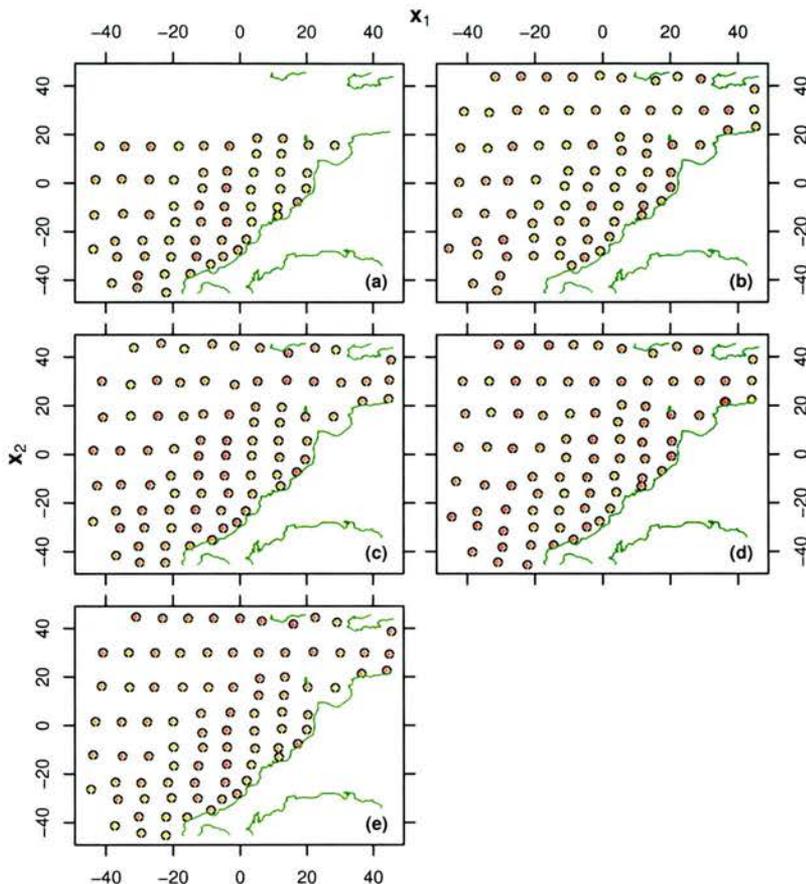


Figure 3.23: Spatial sample points (black circles) and knots (red crosses) for simulated data for the sample times (a) -45.5, (b) -25.0, (c) -1.5, (d) 15.5 and (e) 45.5. The points and knots have been scaled as indicated in the text, in preparation for model fitting.

The SPSP model can be fitted since the data requirements (see section 2.3.6) are satisfied:

- the sample points define the data space in three dimensions (longitude, latitude and time) and do not lie in one plane;
- for $d = 3$ dimensions, $m = 3$ to satisfy $2m > d$, so $M = \binom{m+d+1}{d} = \binom{3+3+1}{3} = 10$.

$K = 394$ knots were chosen and so the requirement $K > M$ was satisfied;

- the number of staged abundances $nC = 394 \times 4 = 1576$ is more than twice the number of knots $2K = 788$.

In addition, the time range is 91 days, the maximum age for eggs in the final development stage is less than a tenth of this, 8.32 days, so extrapolation of the birth and death rate evaluations in calculating the final stage population at the first sample time is not too much of a problem. The inequality constraints are defined on a fine grid of points covering the sample region, from $\alpha_4(\leq 8.32)$ days before the first sample time to the final sample time.

3.3.2 The fitted model

The SPSP model converged in 4 iterations. It has an R-squared of 66% and thus explains much the variation in the data. The fitted values are shown against the sampled stage abundances in figure 3.24(a), both variables were square rooted to produce figure 3.24(b), and the residuals are shown in figure 3.24(c). From these it can be seen that the SPSP fits underestimates the higher abundances, most notably the extreme observation of 35.923 eggs per square metre. The data are extremely wiggly and it could be that the model is too smooth leading to underestimation of higher abundances. This could be because sampling is infrequent through time, or there are too few knots, or the data are under-dispersed. The first reason is highly likely to be the case as much happens to the egg population between the first and second sample time. Since most of the total observed eggs come from the larger observations, underestimation of these implies negative bias overall.

Figure 3.24(b) shows more clearly that the model fails to reproduce a sufficiently large number

of zeros. However the proportion of fitted abundances less than one (89.6%) almost matches the proportion for the observed abundances (89.9%).

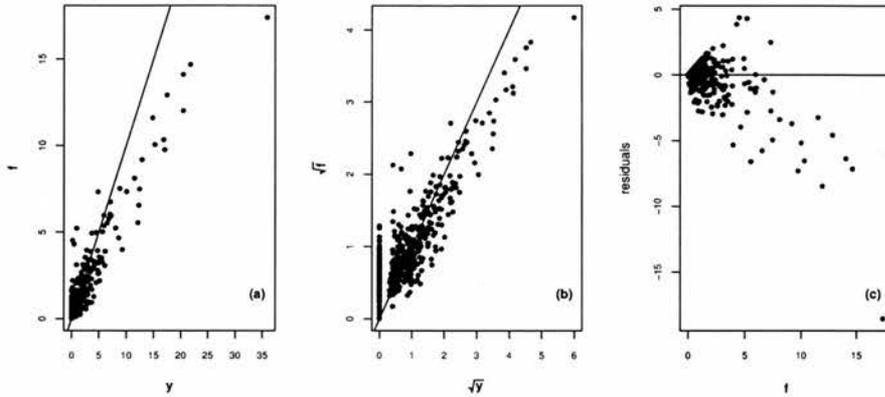


Figure 3.24: Model fit for the sole egg data: (a) fitted stage populations versus original sampled stage populations, (b) square root of fitted stage populations versus square root of original sampled stage populations, (c) residuals versus fitted values.

When the (raw) residual plot (figure 3.24(c)) is compared with randomly generated Poisson residuals, under or over dispersion are not obvious. A possible source of random variation, other than measurement error, is advection of the eggs, which the SPSP model does not allow for. As the maximum age of the eggs is 8.32 days, the eggs are unlikely to drift far. Also, models of residual water flow including the effect of prevailing winds suggest advection is negligible (Pingree & Griffiths 1980), but these results differ from those obtained using drifter experiments reported in Horwood (1993), and so thoughts on advection are inconclusive.

Because of the large proportion of low abundances and the few extremely high abundances, the logarithm of 1 plus the estimated stage abundance ($\log(1 + f_j(\mathbf{x}))$, $j = 1, \dots, 4$) is shown in figure 3.25 rather than the abundance itself. Estimated abundance is presented for each of the four developmental stages and five sample times. Other than underestimating

the peaks, the stage abundance functions capture the features of the sampled abundances of figure 3.20 including negligible abundance for most of the survey, with the majority of population appearing off the north Cornwall coast in the south-east of the survey region. The smooth surfaces are much easier to interpret than the separate noisy samples.

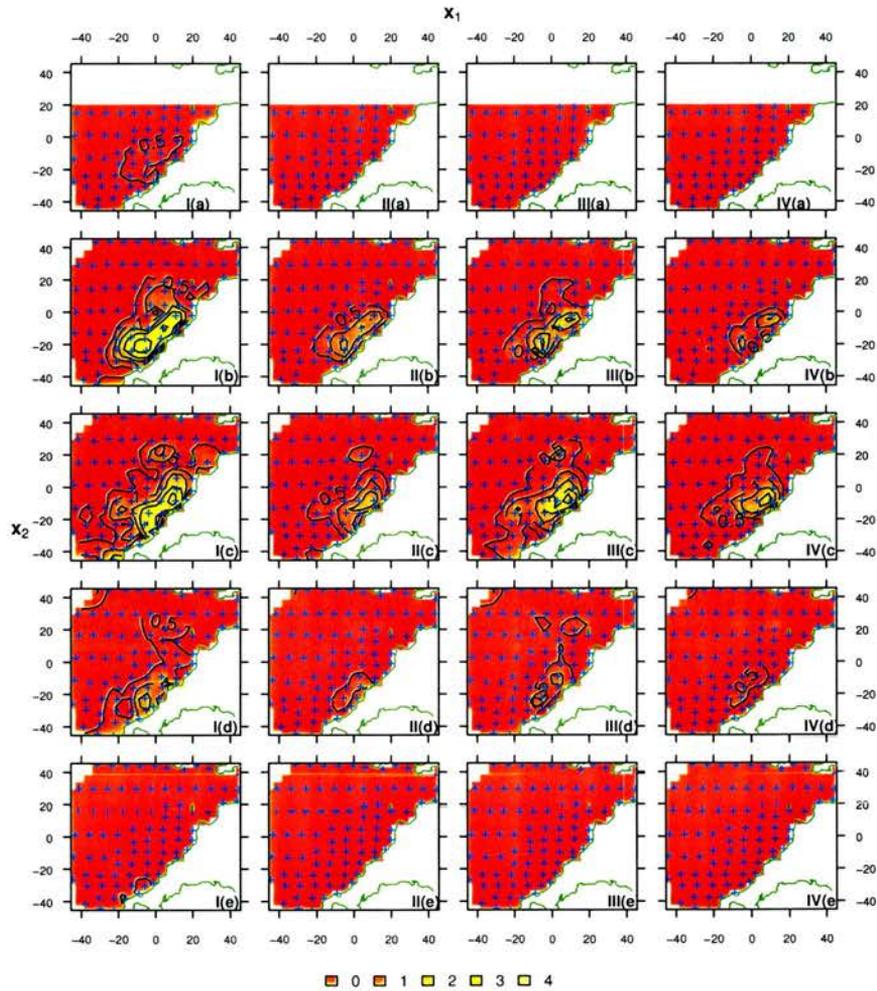


Figure 3.25: *Log of 1 plus the estimated stage populations ($\log(1 + f_j(\mathbf{x}))$) in each of the four developmental stages (I–IV) and five sample times (a) 18th February, (b) 11th March, (c) 3rd April, (d) 20th April, (e) 20th May, for the model fitted to the sole egg data. The sample locations are shown as blue crosses.*

3.3 Application to Dover Sole (*Solea solea*) in the Bristol Channel

As with the stage abundance surfaces, the logarithm of 1 plus the estimated spawning rate ($\log(1 + \beta(\mathbf{x}))$) is shown in figure 3.26. The plots suggest that the sampling did cover the whole spawning season and that most of the spawning occurred in the southeastern corner of the survey region, off the north Cornwall coast. Spawning is negligible in February, increases dramatically within a fairly tight region by mid March, continues at a high rate in the south-east and less intensely throughout the region till mid April, and then dwindles to almost nothing by mid May.

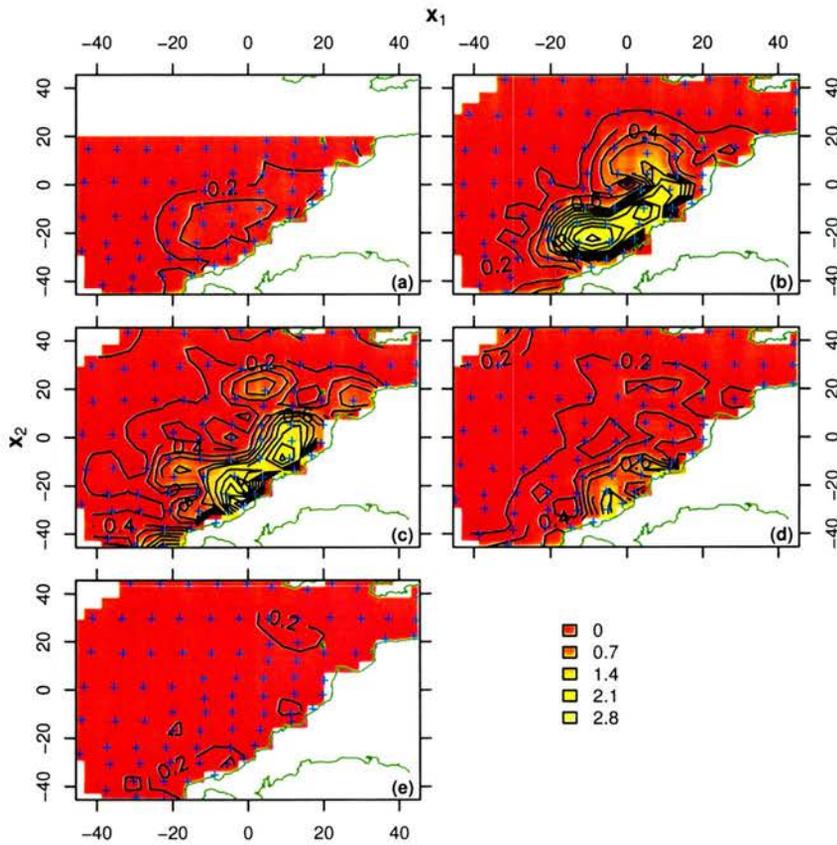


Figure 3.26: Log of 1 plus the estimated birth rate ($\log(1 + \beta(\mathbf{x}))$) for the model fitted to the Dover sole egg data at the five sample times (a) 18th February, (b) 11th March, (c) 3rd April, (d) 20th April, (e) 20th May. The sample locations are shown as blue crosses.

The death rate function in figure 3.27 shows that mortality is highest in areas where most of the spawning occurs, presumably because predators are drawn to regions rich in food.

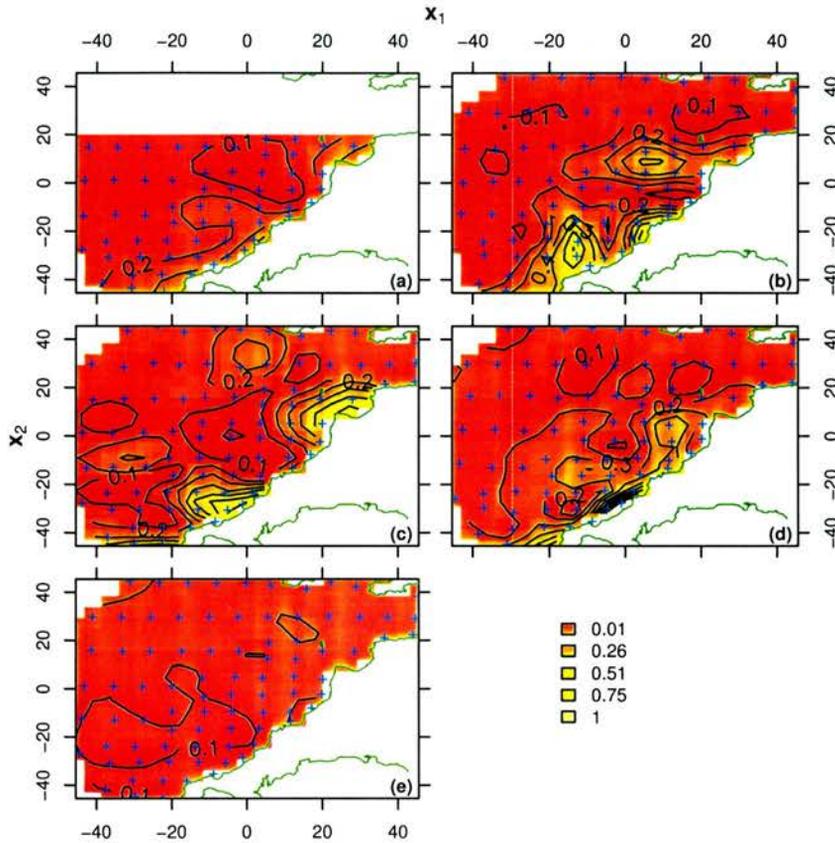


Figure 3.27: Estimated death rate for the model fitted to Poisson data set 1 at the five sample times (a) 18th February, (b) 11th March, (c) 3rd April, (d) 20th April, (e) 20th May.

The sample locations are shown as blue crosses.

A total egg production estimate for 1990 was calculated by summing the predicted birth rates over a fine grid of points throughout the survey region and period. This yielded an estimate of $\widehat{P}_0 = 8.71 \times 10^7$ for the SPSP model with a CV of 1%. This seems unrealistic — a much higher CV would be expected since the model must rely on the smoothness assumptions due to the lack of information between the infrequent sample times during which much activity

takes place (e.g. between the first and second sample times, the population increases from almost zero to its apparent maximum value). The small CV may be a consequence of the non-parametric bootstrap algorithm, which may not reflect the variance at each sample location adequately. Alternatively, a parametric bootstrap might perform better (see section 2.3.7).

The total egg production estimate cannot be compared directly with Horwood (1993) because the net size and other conversion parameters are unknown.

3.4 Discussion

The purpose of this chapter was to test the SPSP model to see if it produces reliable information about a population and if it is straightforward to apply to physiologically-structured count data.

The testing was carried out by applying the SPSP model to several simulated data sets and to sole egg abundances from the Bristol Channel. Having ensured the model assumptions were valid and the data requirements met, fitting the model involved

1. providing the data, that is, the stage abundances, age limits and any environmental data at a set of sample points. This data was also required for each bootstrap re-sample;
2. using a simple linear manipulation of the environmental measurements and time to ensure they had the same range as the sample times and were symmetric in range about zero;
3. choosing knots and inequality constraints and providing the environmental conditions for these;

4. running the model fitting program.

Thus, once the fitting process has been coded, the SPSP model is straightforward to implement. In addition, the fitting algorithm has proven to be fairly efficient, since convergence of the model occurred within five iterations for the majority of simulated data examples and for the model applied to the Dover sole data.

It can be seen from the simulation study that the SPSP model does produce reliable information about a population. Both qualitative and quantitative trends in the true birth and death rates were reproduced fairly closely by the models, even for fairly noisy data. The amount of variation in the data explained by the model was appropriate for the amount of random error in the data (which shouldn't be explained).

There is negative bias of the total birth estimates as a consequence of the SPSP model including smoothing functions, but the bias is small when the residuals are fairly symmetric. The size of the bias is unknown for the sole egg data but is likely to be larger as the larger values making up the greatest proportion of the total birth estimate were underestimated (see figure 3.24).

The SPSP model should perform best when sampling is more frequent than the age range of the population. This was not the case for the simulated data sets or the sole egg abundances, and this may be the cause of the negative bias in the total egg production estimate. In the large time lapse between sample times, there is no information about the population, so the model relies upon the smoothness assumption. However, for the sole egg data it is clear from figure 3.20 that much of the population's activity occurs between the first and second sample times and between the third and fourth sample times, so the underlying population

functions are unlikely to be smooth. Bias may arise when integration of the too smooth birth function is performed in the calculation of the total birth estimation. However the source of bias requires further investigation.

The implications of negative bias are a little assuaged since the realized coverage probabilities of the simulation study imply reliable confidence interval estimation using non-parametric bootstrapping.

In each simulated data set, the variance of the distribution from which the observations were generated differed for each sample point. In this situation for Gaussian data, the objective function of section 2.3.1 becomes sub-optimal because the variance is assumed to be constant. The model performed well in spite of the infrequency of sample times and only partial validity of distributional assumptions. However, the source of negative bias in the application to simulated data should be investigated. The investigation could begin by fitting the SPSP model to simulated data with frequent sampling through time to see if the negative bias disappears. The conclusion would then be that the bias occurs when sampling is infrequent because the fitted smooth function is too smooth an approximation to the unobserved population activity.

As explained in the previous section, the SPSP model provides an improvement on methods previously applied to the sole data. Inference is placed on a more rigorous footing since ad hoc methods are not required and confidence intervals are available through bootstrapping. The SPSP unifies the data set providing a complete picture of the population so that all stage populations can be modelled together, and the birth and death rates can be modelled explicitly. This, and allowing the death rate to vary over the available environmental condi-

tions and time has improved the precision of the total spawning estimate, as anticipated in chapter 2. The gain in precision will have a substantial impact on the precision of spawning stock biomass estimates, as Horwood found that the egg production estimate accounted for 41% of the variation in the estimate of female Bristol Channel sole spawning biomass.

It is possible that inclusion of additional environmental conditions would further improve the precision of the total sole egg production estimate, and would provide useful insight into how spawning and egg death rates relate to variables such as temperature, salinity and depth. Indeed, maps of these conditions for the Bristol Channel in 1990 suggest that temperature is correlated to egg density (Horwood 1993), and Ben-Tuvia (1990) found that geographical position and temperature have a large effect on sole presence. Unfortunately measurements of conditions other than space and time were not available for the sole egg samples but the SPSP model should have captured some of the temperature effect since temperature is correlated with time and less strongly with space.

Application of the SPSP model in a real situation has demonstrated the model's usefulness as a method of solving the inverse problem and obtaining information from sparse and noisy data. The small CV obtained for the sole egg data needs to be examined, and an alternative strategy for variance estimation should be found. The non-parametric bootstrap method used assumes the independence of the samples, but the egg abundances appear to be spatially correlated, so perhaps a coarser sampling unit could assist. Inclusion of environmental conditions in the model means that a wide range of inference is possible; for example, geographical locations are included in the sole data which, in conjunction with the SPSP model, can be used to identify regions of highest egg mortality. Such inference can further inform any decision making for population management.

Although the SPSP model has coped reasonably well with the sole egg data, the large quantity of zeros in the sole egg data has proven difficult to model and may be part of the reason for underestimation of the larger observations. A possible solution might be to model the presence/absence of eggs and to combine this with a model of egg abundance given presence (Borchers et al. 1997), but this is a non-trivial exercise for this demographically based model. An alternative might be to make use of the zero inflated Poisson (ZIP) distribution of Lambert (1992).

The SPSP model is restricted to the situation in which the physiological trait defining the stages is age. The following chapter considers a generalization of the inverse problem when the physiological trait need not be age.

Chapter 4

The Inverse Problem for Size-Structured Data

4.1 Introduction

In previous chapters I considered estimation of birth and death rates from structured population data and assumed that something about the growth rates is known (in the sole egg case this meant that the age boundaries of the stages were known). In this chapter I examine the feasibility of estimating growth rates along with birth and death rates from structured population data. By growth rate I mean the speed at which an individual develops, $\frac{dm}{dt}$, where m is the physiological trait used to measure development.

Growth is an important physiological rate, for example it can play a part in survival strategy, (Hare & Cowen 1997). Inferring the causes of population dynamics from population counts or densities is known as the ‘inverse problem’ because we use a population model to help

us to infer the growth, birth and death rates, rather than to find out what happens to the population dynamics given certain rates. The SPSP model of chapter 2 was a special case of a method for solving the inverse problem in which the physiological trait was age, and the death rate was independent of age.

For convenience, the growth rates, birth rates and death rates will sometimes be referred to collectively as the 'vital rates'. These rates could depend on some environmental variables, but the inverse problem can be explored more simply by ignoring this dependence, and so environmental variables are not considered in this chapter. Allowing the vital rates to depend on these variables would not change the findings of this chapter.

Obtaining demographic rates or information about development directly may be difficult, as Hirst & Shearer (1997) point out for copepods, because there are practical challenges when observing in the field, yet creating realistic laboratory conditions also poses problems. When the amount of information available is limited, a model used to obtain the vital rates should be general; the temptation to include biologically insupportable assumptions which are convenient only for the mathematics in an attempt to make the problem more tractable should be resisted. This investigation shows that it is not possible to obtain unique growth and mortality rates using stage population counts without very strong assumptions, or sufficient data so as to render the model superfluous.

An outline of the chapter is now given. Firstly, a general model relating population abundances to the vital rates is defined in section 4.2.1. It is shown that, without some assumptions, no vital rates can be obtained, even if the growth rates are known. A successful method for solving the inverse problem for only birth and death rates is then considered

in section 4.2.2 which requires only mild assumptions. In section 4.3, I then consider the extension of this method to include estimation of growth rates, but the model produced is mathematically demonstrated in section 4.4.1 to support an infinite number of alternative sets of growth, birth and death rates for the same population counts. In section 4.4.2 limits for the growth rate are shown to exist, deriving from the fact that death rates must be non-negative, but it is also made evident that these limits cannot be obtained in practice. The existence of plausible alternative sets of birth, death and growth rates is then illustrated in section 4.4.4 using a simulated example with perfect data, and known original vital rates.

4.2 The inverse problem for physiologically-structured population data

Biologists and ecologists often collect physiologically-structured population data, that is, population counts for each life-history stage of a particular species. (When censuses rather than samples are taken, such data are also called repeat census data.) In many cases these are counts through time, $y_j(t_i)$ for each stage j , $j = 1, \dots, C$, and sample time t_i , $i = 1, \dots, n$. This is illustrated in figure 4.1.

It is always good economy to make full use of data, which is often expensive to collect, and intuitively we may expect to obtain a lot of information from physiologically-structured population data:

- by comparing population abundances in neighbouring stages we should be able to learn about the mortality of a population,

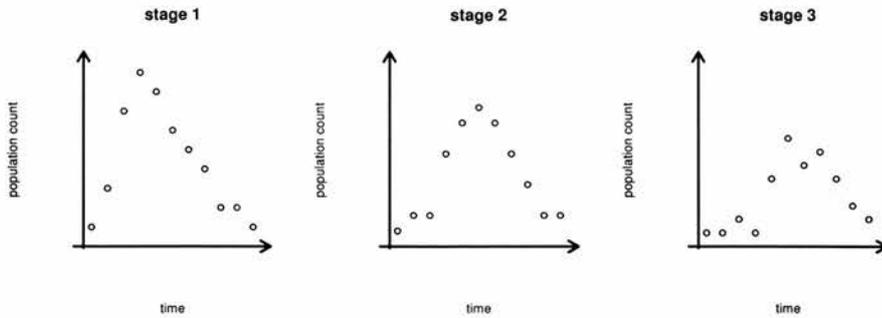


Figure 4.1: *An example of physiologically-structured population counts through time.*

- by comparing the timings of peak populations we should gain knowledge about development time from one stage to the next.

This would be particularly useful for populations where reliable vital rate estimates cannot be obtained by direct observation. Wood (1997) discusses this inverse problem for physiologically-structured populations, including model fitting techniques. Aksnes, Miller, Ohman & Wood (1997) provide a review of the wide variety of estimation techniques used for obtaining the vital rates from copepod population data. Since copepod data are often segregated by physiological stage, this paper is a useful study of physiologically-structured population models with their assumptions, common features and distinctive attributes.

In the following section, I attempt to find a very general model for physiologically-structured data, applicable when there is little biological information available, which can be used to determine the vital rates. Such a model must relate stage abundances to vital rates. Because models are simplified descriptions of the real world, assumptions must be made about what happens in reality. To produce a generally applicable model which does not dictate the form of the vital rates but allows the data to speak for itself, as few assumptions as possible should be made and they should be based on biological knowledge (not conjecture).

4.2.1 A general model developed for physiologically-structured population counts

To make any progress in developing a physiologically-structured population model, we need some information about the developmental stages, in addition to the stage abundances. The information required consists of the lower and upper limits of the physiological trait (m) used to define development, for each of the C developmental classes: $(m_{1L}, m_{1U}), (m_{2L}, m_{2U}), \dots, (m_{CL}, m_{CU})$. A physiological trait could be mass, age, length or some other indicator of development.

Many physiologically-structured population models (including this one) assume that individual members of a population pass through a series of life-history stages and they can never revert back to a previous stage. As the model developed here is deterministic, it is assumed that individuals born at the same time will essentially be identical and are referred to here as a cohort. At any point in time, they will be subject to the same development rate and per capita death rate and thus remain identical for the rest of their lives. The development, or growth rate, of an individual, is the rate at which their physiological trait increases, $\frac{dm}{dt}$. The per capita death rate is the probability that an individual dies in any particular instant.

With this information and these assumptions, we can attempt to solve the inverse problem by formulating a model. In order to write down a model, it is helpful to think about how individuals progress through the life history stages, through time. We can think of the stages as conveyor belts, as illustrated in figure 4.2. Individuals enter a stage by dropping onto the left end of a conveyor belt, then they travel along it until they drop off the right end, unless they evaporate (die) before reaching the end. The rate at which individuals drop onto the

first (top left) conveyor belt is the birth rate, with no individuals surviving to drop off the right end of the last (bottom right) conveyor belt.

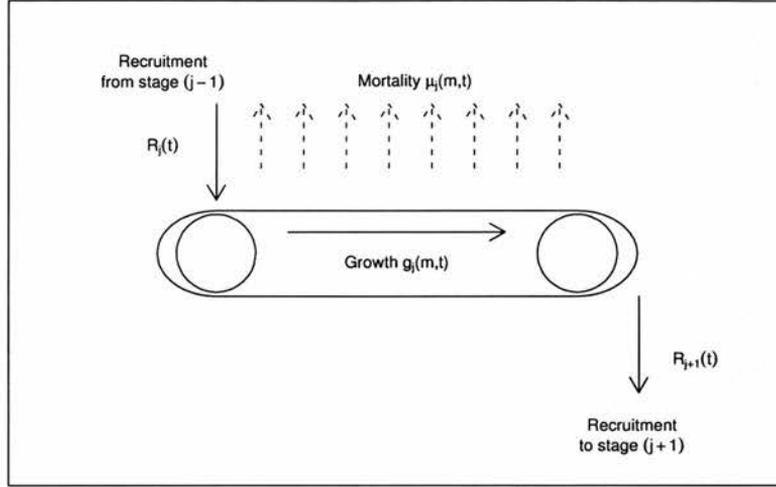


Figure 4.2: A schematic illustration of one stage of the physiologically-structured population model described by equation 4.1. This diagram is derived from a diagram in Gurney & Nisbet (1998, page 252). Individuals drop onto the conveyor belt (stage) at a rate of $R_j(t)$, then travel at a speed of $g_j(m,t)$ until they drop off the other end at a rate of $R_{j+1}(t)$, unless they evaporate from the conveyor belt (die at a rate of $\mu_j(m,t)$) beforehand.

For a closed population, the stage abundance (total number of individuals on one conveyor belt) can be increased by individuals being recruited from the previous stage, and decreased by individuals maturing to the next stage or dying:

$$\begin{aligned} \frac{dN_j}{dt} &= R_j(t) - R_{j+1}(t) - \int_{m_{jL}}^{m_{jU}} \mu_j(x,t)\eta_j(x,t)dx \quad i = 1, \dots, C-1 \\ \frac{dN_C}{dt} &= R_C(t) - \int_{m_{CL}}^{m_{CU}} \mu_C(x,t)\eta_C(x,t)dx \end{aligned} \quad (4.1)$$

where for $j = 1, \dots, C$, $N_j(t)$ is the j th stage abundance at time t , $\eta_j(m, t)$ is the within stage population density — the number of individuals in stage j at time t with physiological trait m , $R_j(t)$ is the recruitment rate into stage j at time t , and $\mu_j(m, t)$ is the death rate at time t for individuals with physiological trait m . No individual can mature from the final stage, so $R_{C+1}(t)$ does not exist. The recruitment to the first stage, $R_1(t)$ is the population recruitment, or birth rate. For a summary of the notation used throughout this chapter, see table A.2 in appendix A.

In order to make use of equation 4.1, we need to know $\frac{dN_j}{dt}$. We can figure this out by fitting C smooth functions of time $N_j(t)$ to the stage abundances $y_j(t_i)$, $i = 1, \dots, n$, separately for each stage $j = 1, \dots, C$. This is like fitting a smooth curve separately to each graph in figure 4.1, so that

$$y_j(t_i) = N_j(t_i) + \epsilon_{ji} \quad j = 1, \dots, C \quad \text{and} \quad i = 1, \dots, n$$

where ϵ_{ji} is a random error term. These abundance functions $N_j(t)$ are then be used to obtain the change in the population counts with respect to time, $\frac{dN_j}{dt}$ analytically or numerically.

The rate at which individuals drop off the right end of one conveyor belt is the same as the rate at which they drop onto the left end of the next conveyor belt. This is equivalent to saying that the rate of maturation from one stage is the same as the rate of recruitment to the next, which is reflected in the system of ordinary differential equations (4.1) by $R_{j+1}(t)$ appearing in the equations for the j th and the $(i + 1)$ th stage. The individuals that mature to stage $j + 1$ will be those that grew and survived through stage j :

$$R_{j+1}(t) = R_j(t - \tau_j(t)) \exp^{-\int_{t-\tau_j(t)}^t \mu_j(m(x), x) dx} \quad (4.2)$$

where $\tau_j(t)$ is the stage duration for the j th stage at time t . The exponent term is the through

stage survival. Note that the physiological trait m is a function of time, since different cohorts may grow at different rates. This cohort have physiological trait $m(t - \tau_j(t)) = m_{jL}$ when they enter stage j at time $t - \tau_j(t)$, and $m(t) = m_{jU}$ when they mature from stage j at time t . The recurrence relation given by equation 4.2 means that recruitment to stage $j + 1$, R_{j+1} , can be written in terms of the birth rate, R_1 , and the death rates μ_k for each of the previous stages $k = 1, \dots, j$, where all of these rates relate to the same cohort of individuals.

The growth rate $g_j(m, t)$ is the speed at which individuals travel towards the right end of the conveyor belt. If all individuals within one stage grow at the same rate ($g_j(m, t) = g_j(t)$), the growth rate is the speed at which the conveyor belt moves. In the more general case where the growth rate may differ for individuals with contrasting physiological traits within a stage (g_j is a function of m as well as t), the growth rate is a combination of the speed at which the conveyor belt moves, and the stretch or contraction of the belt.

The growth rate $g_j(m, t)$ is included in the model in the recruitment terms and the stage durations $\tau_j(t)$ of the recurrence relation 4.2. It can be incorporated explicitly in birth rate as

$$R_1(t) = \eta_1(m_{1L}, t)g_1(m_{1L}, t) \tag{4.3}$$

that is, the population density of newborns multiplied by the growth rate of newborns. The j th stage duration $\tau_j(t)$ can be calculated (non-trivially) from the growth rate for that stage, $g_j(m, t)$, using the result

$$\int_{t-\tau_j(t)}^t g_j(m(x), x)dx = m_{jU} - m_{jL} . \tag{4.4}$$

The model described by equations 4.1, 4.2, 4.3 and 4.4 requires knowledge of the within stage density functions, $\eta_j(m, t)$, whereas only stage abundance functions, $N_j(t)$, can be obtained

from stage abundance observations $y_j(t_i)$. The relationship between the stage abundance and within stage density can be written as

$$N_j(t) = \int_{m_{jL}}^{m_{jU}} \eta_j(x, t) dx.$$

$N_j(t)$ is like a summary of $\eta_j(m, t)$, and so contains less detailed information. Since it is not possible to obtain $\eta_j(m, t)$ from $N_j(t)$ unless we assume that there are smooth transitions between the stages, we cannot use equations 4.1, 4.2, 4.3 and 4.4 to solve the inverse problem in practice. A smoothness assumption would imply there are no physiological jumps between stages so that $\eta_j(m_{jU}, t) = \eta_j(m_{(i+1)L}, t)$, $\mu_j(m_{jU}, t) = \mu_j(m_{(i+1)L}, t)$ and $g_j(m_{jU}, t) = g_j(m_{(i+1)L}, t)$. For now, this particular assumption will not be made, so the model will instead be simplified to eliminate the need for $\eta_j(m, t)$. We choose to do this by replacing the stage death rate function $\mu_j(m, t)$ by its average for the stage, $\mu_j(t) = \frac{1}{m_{jU} - m_{jL}} \int_{m_{jL}}^{m_{jU}} \mu_j(x, t) dx$ in the model equations:

$$\frac{dN_j}{dt} = R_j(t) - R_{j+1}(t) - \mu_j(t)N_j(t) \quad i = 1, \dots, C - 1 \quad (4.5)$$

$$\frac{dN_C}{dt} = R_C(t) - \mu_C(t)N_C(t)$$

$$R_{j+1}(t) = R_j(t - \tau_j(t)) \exp^{-\int_{t-\tau_j(t)}^t \mu_j(x) dx} \quad i = 1, \dots, C - 1 \quad (4.6)$$

$$\int_{t-\tau_j(t)}^t g_j(m(x), x) dx = m_{jU} - m_{jL} \quad i = 1, \dots, C. \quad (4.7)$$

This means we now assume that all individuals within a stage are subject to the same death rate at any given time instant. $R_1(t)$ can no longer be written in terms of the death rate (because we'd need to know $\eta_1(m_{1L}, t)$) so it is now a function to be estimated.

To summarize, for a closed population:

- Available data consist of stage abundances and the stage limits of some physiological trait for C life-history classes.

- It is assumed that individuals born at the same time are and remain identical throughout their lives, and that individuals pass through the life-history stages in ascending order. At any one instant, all individuals within a stage are subject to the same death rate.
- C ordinary differential equations relate the stage abundances to the recruitment functions and death rates (equation system 4.5). The recruitment functions can be written in terms of the birth rate, the death rates and stage durations (equation 4.6), and the stage durations can be obtained from the growth rates and stage limits (equation 4.7).
- We want to estimate the death rate $\mu_j(t)$ and growth rate $g_j(m(t), t)$ for each stage, $j = 1, \dots, C$, (the stage durations $\tau_j(t)$ follow from the growth rates), and the birth rate $R_1(t)$.

Suppose we knew the growth rates, and hence could calculate the stage durations. For example, the growth rates could have been obtained from an independent study. Alternatively, we could take the physiological trait to be age, implying $g_j(m(t), t) = 1$ and $\tau_j(t) = m_{jU} - m_{jL}$, $j = 1, \dots, C$ (the difference between the age limits for each stage). We would then only need to figure out $C + 1$ unknowns, the death rates plus the birth rate. However, we only have C equations for doing this (the equation system 4.5) so it is impossible to obtain both the birth and death rates from this model. (Even if we knew $R_1(t)$, death rate estimates from this model are unstable. See Wood & Nisbet (1991, section 2.4).)

We have met with the trade-off problem between the generality of a model and ability to fit it to the available data. The more general the model, the fewer the assumptions involved, so the harder it is to fit, because more parameters need to be estimated from the same amount of

data. Wood (1997) discusses this trade-off for physiologically-structured population models.

4.2.2 A method of solving the inverse problem for birth and death rates

As mentioned at the end of the previous section, we can simplify the inverse problem by setting the physiological trait to be age, $m = a$. We no longer need to consider growth since the growth rate is simply 1 (individuals age one day per day), and the stage durations are given by the difference between the age limits for each stage. It is now only necessary to obtain the birth and death rates.

Because we have insufficient information to estimate birth and death, a further assumption must be made in order to progress: that the underlying population density is a smooth function of age, $\eta(a, t)$. It follows that the death rate must be a continuous function of age and time, $\mu(a, t)$, so an individual making the transition from one stage to the next will not be subject to sudden changes in its chance of survival. In terms of biology, this is equivalent to saying that there are no major physiological or environmental changes from one stage to the next — so this section's model could not be used for tadpoles transforming to frogs, for example.

This assumption of smooth $\eta(a, t)$ is quite mild, allowing mortality to be a function of the physiological trait (age), so different cohorts within the same stage are not restricted to being subject to the same mortality rate. Instead of reducing the model given by equations 4.1, 4.2, 4.3 and 4.4 to the one given by equations 4.5, 4.6 and 4.7, we can replace it with a partial differential equation model:

$$\frac{\partial \eta}{\partial t} + \frac{\partial \eta}{\partial a} + \mu(a, t)\eta(a, t) = 0 . \tag{4.8}$$

This description of the relationship between the population density and the vital rates is the McKendrick von Foerster equation for age (Sharpe & Lotka 1911, Lotka 1925, McKendrick 1926, von Foerster 1959). The birth rate is simply $\eta(0, t)$, the population density at age 0, and the death rate can be obtained by rearranging equation 4.8 for $\mu(a, t)$. This method of solving the inverse problem for birth and mortality, with the incorporation of the demographic restrictions $\eta(a, t) \geq 0$ and $\mu(a, t) \geq 0$, is fully detailed in Wood (1994). Briefly, a smooth population density surface $\eta(a, t)$ (e.g. a smoothing spline) is estimated from abundance data $y_j(t_i) = \int \eta(a, t_i) da + \epsilon_j$, using, for example, a smoothing spline. Equation 4.8 is then used to obtain the death rate function $\mu(a, t)$. This method has been more recently applied to copepods in Norway by Eiane, Aksnes, Ohman, Wood & Marinussen (2002).

4.3 Extension of the method to growth rates

An obvious extension to this method is to try to obtain growth rates in addition to birth and death rates. This is desirable for biologists who may want to know how fast individuals develop, whether this changes for different stages of development, and whether it changes over time — seasonally, or from year to year. In addition, it is often difficult to age a species, though it is usually much easier to measure its size, for example, its length or mass. Species for which it is easier to measure size include zooplankters (Aksnes et al. 1997) and fish (Crowder, Rice, Miller & Marschall 1992). Crowder et al. (1992) also recommend the use of size for fish because there is evidence to suggest that the mechanisms governing their survival depend on size, and Easterling, Ellner & Dixon (2000) found size to be a better predictor of plant growth for northern monkshood *Aconitum noveboracense* than age. Growth must

be included in any model relating population dynamics to birth and death rates using size, rather than age, as a developmental measurement. In all that follows, size can refer to any continuous developmental measurement.

Because it is desirable to use methods with as few arbitrary assumptions as possible, it is logical to approach the problem of obtaining growth rates by extending the method used for obtaining birth and death rates in section 4.2.2. To do this, we can assume that growth is a smooth function of size (the physiological trait) and time. This means that as a population member develops from one stage to the next, its growth rate changes gradually. With this assumption, we can describe the population dynamics using the McKendrick von Foerster equation for size:

$$\frac{\partial \eta}{\partial t} + \frac{\partial (g\eta)}{\partial m} + \mu\eta = \frac{\partial \eta}{\partial t} + g \frac{\partial \eta}{\partial m} + \eta \frac{\partial g}{\partial m} + \mu\eta = 0 \quad (4.9)$$

The birth rate is given by $R(t) = \eta(m_{min}, t)g(m_{min}, t)$. de Roos (1997) provides a derivation of this scheme. This model requires very few assumptions, and therefore is very general. Equation 4.8 is a special case of equation 4.9 with $g(m, t) = 1$.

Unfortunately, given only population counts and stage size limits, it is impossible to determine *unique* growth and hence birth and death rates without strong assumptions, as the following section demonstrates.

4.4 Non-uniqueness of growth, birth and death rates

4.4.1 Mathematical demonstration

To prove uniqueness in mathematics, it is common to try alternative solutions, and show that to satisfy known equations, the alternatives must be equal. For a given population density function, η , it is possible to have alternative sets of growth and death rates that still satisfy equation 4.9, and so the vital rates are not uniquely defined. Because birth is a function of growth, it is not unique either.

Suppose a population $\eta(m, t)$ has ‘true’ death rate $\tilde{\mu}(m, t)$, growth rate $\tilde{g}(m, t)$ and birth rate $\tilde{R}(t) = \eta(m_{min}, t)\tilde{g}(m_{min}, t)$. For the ideal case in which $\eta(m, t)$ is known perfectly from the population counts $N_j(t)$, the derivatives $\frac{\partial \eta}{\partial t}$ and $\frac{\partial \eta}{\partial m}$ are also known. If a working method cannot be obtained for this situation, then no method will be available when knowledge of $\eta(m, t)$ is incomplete. η and \tilde{g} are assumed to be smooth functions and

$$\frac{\partial \eta}{\partial t} + \tilde{g} \frac{\partial \eta}{\partial m} + \eta \frac{\partial \tilde{g}}{\partial m} + \tilde{\mu} \eta = 0 \quad (4.10)$$

is obtained from equation 4.9.

Consider a constant perturbation Δg to \tilde{g} , producing a new growth rate $g^* = \tilde{g} + \Delta g$. Since Δg is constant, g^* is as smooth as \tilde{g} . η is fixed, so its smoothness doesn’t change. There is now a new death rate μ^* , which must also satisfy equation 4.9, and a new birth rate $R^*(t) = \eta(m_{min}, t)g^*(m_{min}, t)$. If g , μ and R were unique, equation 4.9 could only be

satisfied if $g^* = \tilde{g}$, $\mu^* = \tilde{\mu}$ and $R^* = \tilde{R}$. However,

$$\begin{aligned} \frac{\partial \eta}{\partial t} + g^* \frac{\partial \eta}{\partial m} + \eta \frac{\partial g^*}{\partial m} + \mu^* \eta &= \frac{\partial \eta}{\partial t} + (\tilde{g} + \Delta g) \frac{\partial \eta}{\partial m} + \eta \frac{\partial (\tilde{g} + \Delta g)}{\partial m} + \mu^* \eta \\ &= \frac{\partial \eta}{\partial t} + \tilde{g} \frac{\partial \eta}{\partial m} + \eta \frac{\partial \tilde{g}}{\partial m} + \Delta g \frac{\partial \eta}{\partial m} + \eta \frac{\partial \Delta g}{\partial m} + \mu^* \eta . \end{aligned}$$

From equation 4.10 and the fact that $\frac{\partial \Delta g}{\partial m} = 0$ since Δg is a constant,

$$\begin{aligned} \frac{\partial \eta}{\partial t} + g^* \frac{\partial \eta}{\partial m} + \eta \frac{\partial g^*}{\partial m} + \mu^* \eta &= -\tilde{\mu} \eta + \Delta g \frac{\partial \eta}{\partial m} + \mu^* \eta \\ &= 0 \end{aligned}$$

so

$$\mu^* = \tilde{\mu} - \frac{\Delta g}{\eta} \frac{\partial \eta}{\partial m} \tag{4.11}$$

which shows that if $\tilde{\mu}$ and η are smooth, then so is μ^* . Since μ^* is non-negative,

$$\begin{aligned} \Delta g < \frac{\tilde{\mu} \eta}{\frac{\partial \eta}{\partial m}} &\quad \text{if} \quad \frac{\partial \eta}{\partial m} > 0 \\ \Delta g > \frac{\tilde{\mu} \eta}{\frac{\partial \eta}{\partial m}} &\quad \text{if} \quad \frac{\partial \eta}{\partial m} < 0 \\ \Delta g \text{ can be anything} &\quad \text{if} \quad \frac{\partial \eta}{\partial m} = 0 . \end{aligned}$$

We also require that R^* is non-negative:

$$R^*(t) = \eta(m_{min}, t) \tilde{g}(m_{min}, t) + \eta(m_{min}, t) \Delta g \geq 0$$

so $\Delta g \geq -\tilde{g}(m_{min}, t)$, which is relevant when $\frac{\partial \eta}{\partial m} \leq 0$.

These inequalities for Δg show that a Δg can always be found so that the death rate μ^* and the birth rate R^* are non-negative. That is, if \tilde{g} and $\tilde{\mu}$ are the true growth and death rates,

alternative growth and death rates, g^* and μ^* can be found which also satisfy the McKendrick von Foerster equation for size. Therefore there is no unique set of growth, birth and death rates (g, μ, R) for any given population density surface.

4.4.2 Limits on the growth rate

Unless discussing a particular species, there is no physiological restriction on growth — for example, growth can be negative for perennial plants. However, the demographic restriction that the death rate must be non-negative imposes limits on growth via equation 4.9. Frustratingly, we can't find these limits because they involve an unknown integration constant! An exception is the particular case in which growth is independent of size.

To find the limits on growth, equation 4.9 is rearranged so that all the growth terms are on one side:

$$\begin{aligned} \frac{\partial \eta}{\partial t} + g \frac{\partial \eta}{\partial m} + \eta \frac{\partial g}{\partial m} + \mu \eta &= 0 \\ \frac{\partial g}{\partial m} + \frac{1}{\eta} \frac{\partial \eta}{\partial m} g &= -\frac{1}{\eta} \frac{\partial \eta}{\partial t} - \mu \end{aligned}$$

which is a linear first order partial differential equation in g . From Kreyszig (1993, pages 30 – 31), the solution (for y) of

$$\frac{dy}{dx} + p(x)y = r(x)$$

is

$$y = \exp^{-h} \left[\int \exp^h r(x) dx + c \right] \quad \text{where} \quad h = \int p(x) dx$$

and c is an integration constant. For equation 4.9, $x = m$, $y = g$, $p(m) = \frac{1}{\eta} \frac{\partial \eta}{\partial m}$, and

$r(m) = -\frac{1}{\eta} \frac{\partial \eta}{\partial t} - \mu$. Then

$$h = \int \frac{1}{\eta} \frac{\partial \eta}{\partial m} \partial m = \int \frac{1}{\eta} \partial \eta = \ln \eta$$

and

$$\begin{aligned} g &= \exp^{-\ln \eta} \left[\int_{m_{\min}}^{m_{\max}} \exp^{\ln \eta} \left(-\frac{1}{\eta} \frac{\partial \eta}{\partial t} - \mu \right) \partial m + c \right] \\ &= -\frac{1}{\eta} \left[\int_{m_{\min}}^{m_{\max}} \left(\frac{\partial \eta}{\partial t} + \mu \eta \right) \partial m - c \right]. \end{aligned}$$

Now $\mu \eta \geq 0$, so

$$\begin{aligned} g &\leq -\frac{1}{\eta} \left[\int_{m_{\min}}^{m_{\max}} \frac{\partial \eta}{\partial t} \partial m - c \right] \\ &\quad \left(= -\frac{1}{\eta} \left[\frac{\partial N}{\partial t} + c^* \right], \quad N = \text{total population abundance per unit time} \right) \end{aligned}$$

We could use numerical methods to estimate all the quantities involved except for the integration constant c . c is independent of m but could depend on t . In order to calculate c , we'd need to know the upper limit of g , but if we knew this, we wouldn't have to calculate it!

For the special case when growth is non-negative, $0 \leq -\eta^{-1} \left[\int_{m_{\min}}^{m_{\max}} \frac{\partial \eta}{\partial t} \partial m - c \right]$, so $c \leq -\int_{m_{\min}}^{m_{\max}} \frac{\partial \eta}{\partial t} \partial m$. c may depend on time but we can say that $c \leq \max_t \left[\int_{m_{\min}}^{m_{\max}} \frac{\partial \eta}{\partial t} \partial m \right]$. Numerical calculation of this upper limit on c will allow calculation of a generous upper limit on g .

Limits on growth, when growth is independent of size

For the specific case in which g is a function of t only, the limits of g can be obtained. If $g = f(t)$, $\frac{\partial g}{\partial m} = 0$, and equation 4.9 for growth becomes

$$\frac{\partial \eta}{\partial t} + g \frac{\partial \eta}{\partial m} + \mu \eta = 0$$

we know that $\mu\eta \geq 0$, so

$$\frac{\partial\eta}{\partial t} + g\frac{\partial\eta}{\partial m} \leq 0$$

When $\frac{\partial\eta}{\partial m} > 0$, $0 \leq g \leq -\frac{\partial\eta}{\partial t} \frac{\partial\eta}{\partial m}$. When $\frac{\partial\eta}{\partial m} < 0$, $g \geq -\frac{\partial\eta}{\partial t} \frac{\partial\eta}{\partial m}$.

From this we find that the lower limits of the growth rate are given by

$$gL = \begin{cases} 0 & \frac{\partial\eta}{\partial m} > 0 \\ -\frac{\frac{\partial\eta}{\partial t}}{\frac{\partial\eta}{\partial m}} & \frac{\partial\eta}{\partial m} < 0 \text{ and } \frac{\partial\eta}{\partial t} > 0 \\ 0 & \frac{\partial\eta}{\partial m} < 0 \text{ and } \frac{\partial\eta}{\partial t} \leq 0 \end{cases}$$

and the upper limits are

$$gU = \begin{cases} -\frac{\frac{\partial\eta}{\partial t}}{\frac{\partial\eta}{\partial m}} & \frac{\partial\eta}{\partial m} > 0 \\ \text{no limit} & \frac{\partial\eta}{\partial m} < 0 \end{cases}$$

When $\frac{\partial\eta}{\partial m} = 0$, there are no limits for g .

Limits on the growth rate imply limits on the death rate, in addition to non-negativity. No attempt has been made to find the range of possible death rates, although this might be useful. So long as a growth rate is chosen within the growth rate limits, the death rate will also lie within its limits.

4.4.3 Avoiding the non-uniqueness problem using assumptions

One possible tactic for obtaining unique vital rates is to assume a particular functional form for the growth rates and the death rates. For example, Rothschild, Sharov, Kearsley & Bondarenko (1997), Edvardsen, Zhou, Tande & Zhu (2002) and Meise, Collie, Widman &

Howell (1999) assume time-invariant growth and death rates. Time-invariance is a strong assumption but it may be reasonable when growth is deterministically linked to a measurable covariate such as temperature. In many cases, temperature may be sufficiently constant that time invariance could be assumed.

Equation 4.11 allows alternative time-invariant growth, birth and death rates when the population density is also time invariant (this would imply constant recruitment). However, a unique solution does exist when the birth rate is allowed to vary with time but the growth and death rates *must* be time-invariant. This is because the population density η will vary with time so equation 4.11 implies any alternative death rate would also vary with time, even though the alternative growth rate is time-invariant.

When considered intuitively, it may be expected that a unique solution is obtainable. Cohorts will traverse the size-time plane in parallel curves and the height of the population density surface will reduce at the same level in the size direction, maintaining the relative proportions of the cohorts in the time direction. This is illustrated in figure 4.3(a) where population density is indicated by colour. The shape of $\frac{R(t)}{g(m_{min})}$ should be clearly discernable traversing the size-time plane, reducing in height, but otherwise unchanging. To obtain the growth rate, a cohort could be traced (e.g. the peak of the population density surface) and the growth rate $\frac{\partial m}{\partial t}$ estimated. From this the death and birth rates could be found.

Without further investigation, it is difficult to see how the growth and death and birth rates may be calculated in a mathematically rigorous fashion from the population density, since the solution and upper limit for g given at the start of section 4.4.2 involve an unknown constant of integration.

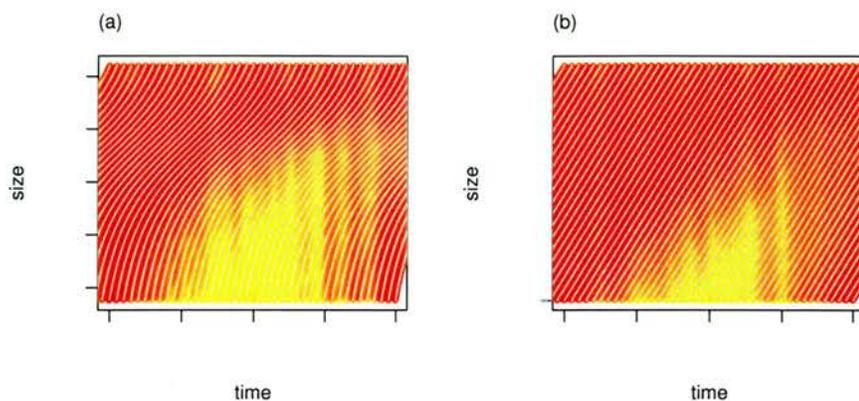


Figure 4.3: Schematic illustration of showing cohorts traversing the size-time plane with population density indicated by colour (red is low, yellow is high density) for the special cases (a) growth and death are independent of time, (b) growth and death are independent of size and time. In both cases, recruitment depends on time.

If the even stronger assumption of constant growth and mortality is made, then unique solutions to the McKendrick von Foerster equation do exist because the population density will not be constant unless the death rate is zero, which is not demographically reasonable. An alternative constant growth rate would imply, through equation 4.11, an alternative non-constant mortality rate, which is not permitted by the assumption. The constant growth and mortality situation is illustrated in figure 4.3(b).

There is another special case in which it may be possible to determine unique vital rates. When the birth rate consists of very sharp pulses, it may be possible to ‘observe’ the growth rate of these pulse cohorts directly to obtain $g = \frac{\partial m}{\partial t}$ along the pulses. Hence the birth and death rates may be obtained for the pulses but it is unclear how the vital rates may be obtained between the pulses.

A potential inhibitor to a solution in this case is the smoothness assumption which facilitated

the approach to a solution in the first instance! If the pulse in the population density is sufficiently sharp for observation of the growth rate, it may be too sharp for the smoothness assumption to hold.

The SPSP model assumes that mortality is independent of size. In this case, there are no unique solutions and the upper limits on the growth rate cannot be found because the number of unknowns in equation 4.9 cannot be reduced as it can when growth is independent of size. (The inverse problem could be solved using the SPSP model because the physiological trait was age so the growth rate was known.)

Strong assumptions should not be made without substantial biological evidence, so that misspecification bias can be avoided, and invalid solutions are not interpreted as the truth. If sufficient biological evidence of the vital rates is available to select appropriate vital rates, solving the inverse problem does not greatly enhance current knowledge.

4.4.4 Demonstration of the non-uniqueness of growth, birth and death rates using simulated data

Section 4.4.1 demonstrated mathematically that given a population density function $\eta(m, t)$, we cannot obtain unique set of growth, birth and death rates, g , μ and R . However, there may still be a lingering hope that in practical terms, we can pin down approximately true vital rates. This might be possible if the range of alternative vital rates is narrow, or if sets of vital rates that differ from the true set contain insufficiently smooth death rates.

This section removes this possibility by using some simulated data to show that the range of alternative sets of vital rates is wide, and that they include sufficiently smooth death rates.

Hence there is no way of deciding between them.

Simulated data set

A data set consisting of 37471 ‘perfect’ observations of a population density function of size and time was created (i.e. the observations do not include random errors). The density function was simulated using a chosen set of vital rates by approximating equation 4.9 using the upwind difference scheme, then integrating the resulting ordinary differential equations using the midpoint method. Further details of the method of simulation can be found in appendix E. The original vital rates and population density surface are shown in figure 4.4.

The birth rate function used to create the data set was

$$R(t) = 10^6 f_1(0.5 - f_2) + 500$$

where $f_1 = \frac{1}{200\pi} \exp -0.5 \left(\frac{t-120}{100} \right)^2$ and $f_2 = \frac{1}{40\pi} \exp -0.5 \left(\frac{t-120}{20} \right)^2$. We assume that all individuals are born to the same (minimum) size.

The death rate used was

$$\begin{aligned} \mu(m, t) = & 0.25 [-0.0255 - (5.911 \times 10^{-6})t + 0.00063m] \\ & - 0.25 [(9.487 \times 10^{-7})m^2 + (5.029 \times 10^{-10})m^3 - (8.645 \times 10^{-14})m^4] . \end{aligned}$$

The growth rate used was

$$g(m, t) = 1 + 0.2 \frac{f_3(t)(m - 120)(f_4(t) - m)}{(f_4(t) - 120)}$$

where

$$f_3(t) = 0.083 \sin \left(\frac{365t}{2\pi} \right) + 0.185$$

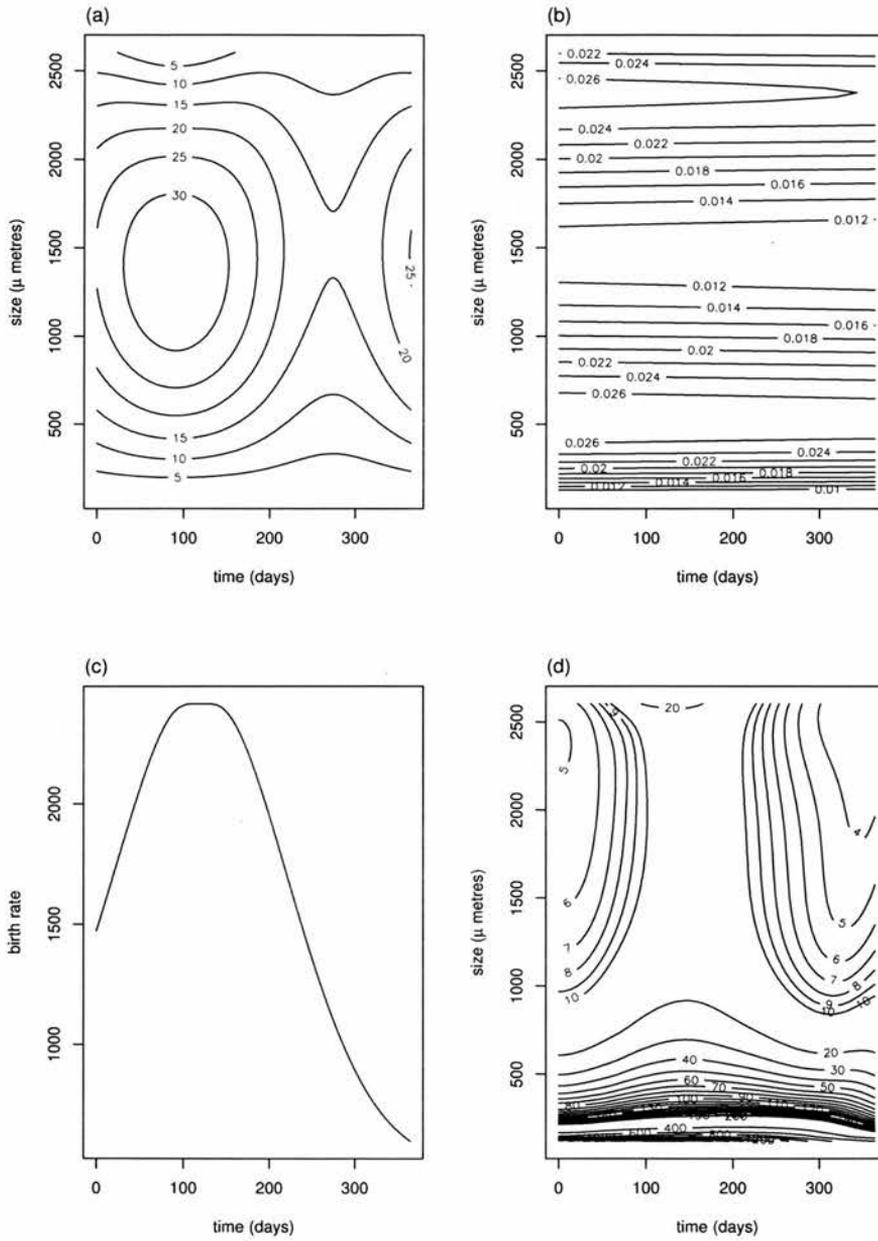


Figure 4.4: The (a) growth rate, (b) death rate, (c) birth rate and the (d) population density produced by these vital rates. Size is measured in micrometres notated ' μ metres' on the y-axes.

and

$$f_4(t) = -156.5 \sin\left(\frac{365t}{2\pi}\right) + 2759.5$$

so the derivative of g with respect to size m is

$$\frac{\partial g}{\partial m} = 0.2 \frac{f_3(t)(120 + f_4(t) - 2m)}{(f_4(t) - 120)}.$$

These vital rates were chosen to be approximately consistent with copepod populations, where size is taken to be length. Size information was found in Campbell, Wagner, Teegarden, Boudreau & Durbin (2001), and from this growth rates were obtained. Ohman & Wood (1996) give egg abundances through time which suggested a suitable birth rate function, although the function was adapted to have a flatter peak so that the progression of the peak population density can be seen more clearly in the population density function (figure 4.4(d)). Ohman & Wood (1996) also contained mortality estimates which could be used in conjunction with Campbell et al.'s (2001) size data to obtain a death rate function of size and time.

When simulating a population, it is necessary to choose an initial size distribution. This was chosen by assuming that the vital rates had been in equilibrium prior to the start time (t_0),

$$\eta(m, t_0) = R(0) \exp\left(-\int_{m_{min}}^m \frac{\mu(x, t_0)}{g(x, t_0)} dx\right).$$

This is equivalent to choosing any initial population and iterating the population density estimates through time for a very long time. This should make plots of the population density function less visually confusing, so that we can see to some extent how the population dynamics are generated.

Alternative vital rates

As an illustration of the impossibility of solving the inverse problem for growth, birth and death, three alternative sets of vital rates were found for the simulated population density. The alternative vital rates are shown in figure 4.5. Because the original growth rate was dependent on size, limits cannot be found for the growth rate although they do exist.

The first alternative growth rate is set to $g_a(m, t) = 0.5g(m, t)$, i.e. the same functional form but on a smaller scale — the growth is half as slow as the original. The first alternative death rate is found using a rearrangement of equation 4.9,

$$\mu_a(m, t) = \frac{-1}{\eta(m, t)} \left(\frac{\partial \eta}{\partial t} + g_a(m, t) \frac{\partial \eta}{\partial m} + \eta(m, t) \frac{\partial g_a}{\partial m} \right)$$

and the first alternative birth rate using $R_a(t) = \eta(m_{min}, t)g_a(m_{min}, t)$.

The second alternative growth rate is set to $g_b(m, t) = 10g(m, t)$ — ten times faster than the original — and the corresponding second alternative birth and death rates are found using formulae equivalent to those above.

These alternative growth, birth and death rates are shown in graphs (a) – (f) of figure 4.5. The constants 0.5 and 10 were found by trial and improvement, since limits on the growth rate could not be calculated. Firstly, a multiplying constant was chosen, then $g_j(m, t)$ and the corresponding μ_j ($j = a, b$) were calculated, then μ_j was checked to ensure it was non-negative. This process was repeated until an approximately minimum constant was selected. Before an approximately maximum constant was reached, unreasonably large values of g_b were found (for example $3000\mu m$ per day implying that individuals could be born and reach maximum size in one day), so the round constant 10 was decided upon.

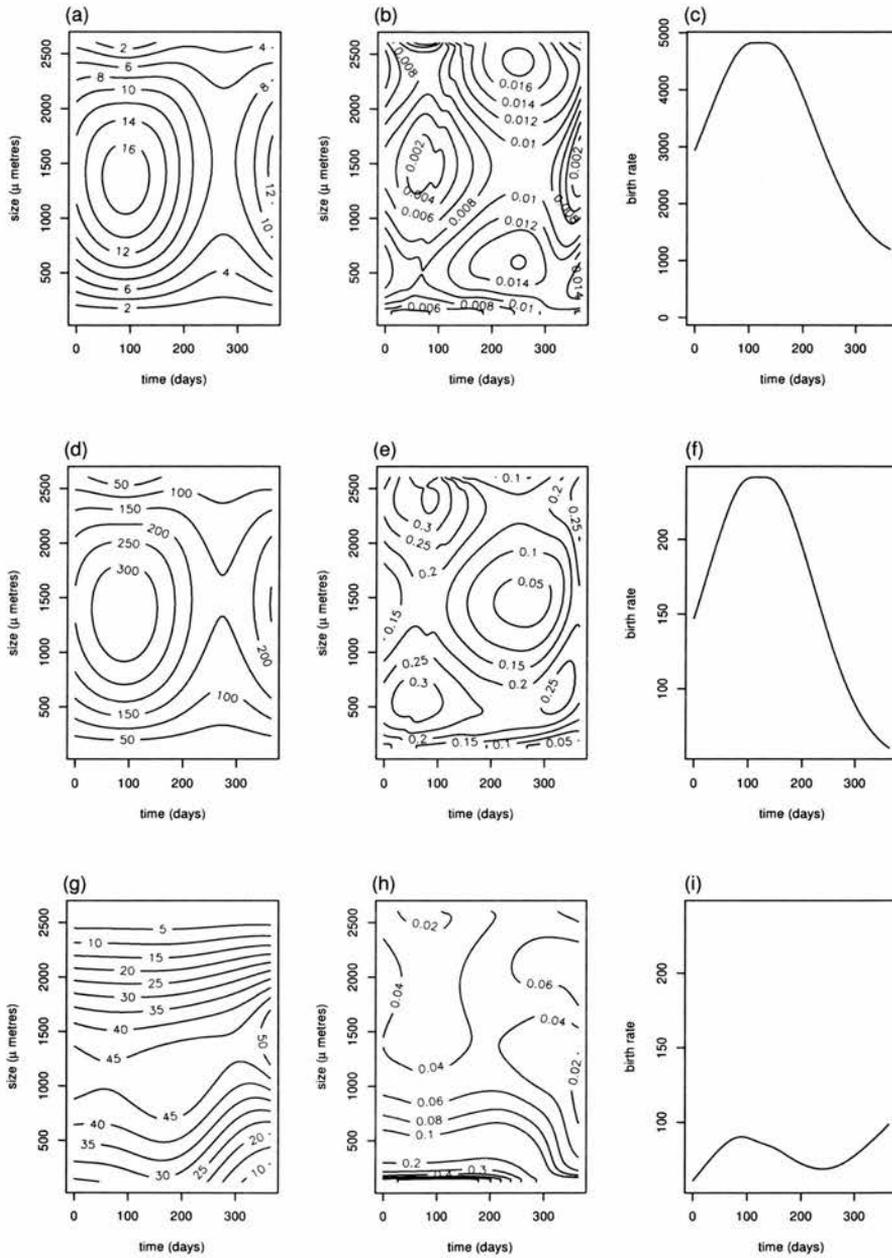


Figure 4.5: Alternative growth, death and birth rates for the population density in figure 4.4. (a) $g_a(m, t)$, (b) $\mu_a(m, t)$, (c) $R_a(m, t)$, (d) $g_b(m, t)$, (e) $\mu_b(m, t)$, (f) $R_b(m, t)$, (g) $g_c(m, t)$, (h) $\mu_c(m, t)$, (i) $R_c(m, t)$. Note that the y-axis scale of (c) differs from that of (f) and (i). Size is measured in micrometres notated ‘ μ metres’ on the y-axes.

These show that even if the functional form of g is known (or assumed), there is still a range of scales that g can take. This might be restricted by the desired smoothness of μ — for this example, μ is not very smooth for the very small g and very large g , and so something closer to the original g may be chosen. Another restriction might be the expected range of the birth rate function. However, if the functional form of g has merely been assumed, how do we know it's the correct form?

To show that different functional forms for growth are possible, a third alternative growth rate was chosen. Again, trial and error was the method used to find a growth rate with a corresponding non-negative death rate. The growth rate chosen was

$$g_c(m, t) = (m_{max} - m) \times \frac{100}{f_5(t)\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{m-f_6(t)}{f_5(t)}\right)^2}$$

where

$$f_5(t) = \frac{-(t-100)^2}{200} + 1050 \quad \text{and} \quad f_6(t) = \frac{t}{2} \cos\left(\frac{\pi(t-10)}{150}\right) + 1800$$

i.e. a scaled Gaussian function of size, where the mean and standard deviation vary with time, multiplied by a linear function of size. The corresponding birth and death rates can be obtained in the same way as for μ_a and R_a using g_c instead of g_a . This growth rate and the corresponding birth and death rates are shown in graphs (g), (h) and (i) of figure 4.5.

4.5 Discussion

When studying a population and its role in the wider ecological community, it is important to obtain correct growth, birth and death rates, since these rates drive the population dynamics. For example, Ohman & Hirche (2001) report that marine ecosystem models are sensitive to

the form of the zooplankton mortality term. In the case of copepods, the correct growth term is required for appreciating their role in energy and nutrient flow and transformation (Hirst & Sheader 1997). Biologists also use the growth and death rates to figure out the survival strategies of a species (Hare & Cowen 1997). Incorrect estimates of the vital rates will have consequences for conclusions made by these scientists. In turn, the conclusions can have serious repercussions since they may shape political or managerial decisions, and for the example of the fishing industry, will effect communities, the economy and, not least, fish populations.

The purpose of this chapter was to investigate the inverse problem of obtaining growth, birth and death rates from limited demographic and physiological data. The investigation has concluded that a unique solution to the inverse problem cannot be found for limited information, without making strong arbitrary assumptions — we can't get something for nothing!

A mathematical demonstration of this fact might not be convincing if, in practice, only a narrow range of vital rates is feasible. However, the simulation example shows that it is perfectly possible to have different combinations of the vital rates between which it is impossible to choose without more biological knowledge. Furthermore, the simulation example used data without random errors and therefore knowledge of the population density function η was perfect. This is an unlikely scenario in practice, which can only mean that a real example will involve even more uncertainty.

This is a rather gloomy situation, but it has not deterred some from obtaining solutions to the inverse problem. The results of this chapter imply that previous claims of unique

solutions to the inverse problem for growth, birth and death rates are false and therefore there must be limitations to the claims or flaws in the arguments supporting them. For example, Rothschild et al. (1997) can obtain growth and death rates, which are assumed to be time invariant, because in their analytic solution to the inverse problem, they reduce the number of parameters that must be estimated. There is no explanation of how this was done. Their numerical solutions provide good estimates of the stage populations, but this does not indicate that unique growth and death rates have been found. In fact, the errors for the fitted parameters which involve growth and death increase with the stage number, leading to large death rates compensating for small growth rates by the eighth developmental stage. The growth rate is sufficiently large to imply negative death rates for two of the stages, so even their own inequality constraints are not satisfied. In many cases, the 95% confidence intervals for the estimated parameters include negative rates. In Ackleh (1999), a least squares technique is developed for identifying the parameters (essentially the vital rates) of a non-linear size-structured population model. However, the method requires that some of the model parameters are already known. This can be seen in the numerical experiments where only the birth and growth rates need to be estimated because the (true) death rate is known. Huntley, Zhou & Lopez (1994) claim to solve the inverse problem for stage specific mortality and development of *Calanoides acutus*, but there is a flaw in the derivation of their analytic solution (they claim that their equation 11 implies that one of the development rates and one of the mortality rates must be zero, which reduces the number of parameters to be estimated and affects the values of these remaining estimates. This is not necessarily the case as two functions of time can sum to a non-zero constant.) Past methods may provide a solution to the inverse problem, but not a unique solution, and there is no way of telling whether the particular solution presented is the correct one.

To determine the true vital rates when the physiological trait is not age, we need more information than just stage abundances and stage limits. Easterling et al. (2000) suggest obtaining vital rates from records of marked individuals studied throughout their lives and using these in a population model. Somerton & Kobayashi (1992) use size at age data to obtain growth and mortality estimates when using (size-biased) size-frequency data. As a minimum, the growth rate must be obtained independently using size at age data. Once the growth rate is known, the birth and death rates can be obtained from the McKendrick von Foerster equation 4.9. Size at age data may be expensive to collect and there can be problems in collecting such data from the field or the laboratory, as Hirst & Sheader (1997) point out for copepods.

There are a number of special cases identified in section 4.4.3 (and others may exist) in which it seems intuitively possible to obtain unique vital rates and these require scrutiny. They include the situation where the population density and birth rate vary with time, but the growth and death rates must be time invariant. Time invariance can sometimes be assumed when the sample period is sufficiently short, so unique solutions would be valuable. Another special case is the situation when recruitment to the population occurs in sharp pulses. For both of these cases, the growth rate can be observed in a graph of population density, but the mathematical problem of obtaining the vital rates requires further investigation. The smoothness assumption may mean that the pulsating birth rate case cannot be solved. In fact, it appears that the smoother the population density, the wider the range of plausible vital rates; the less smooth the population density, the narrower the range of plausible vital rates but also the less valid the smoothness assumption which facilitates the search for a solution!

Section 4.2.2 explained that the inverse problem can be solved when the physiological trait is age, so the SPSP model of chapter 2 was able to obtain correct birth and death rate estimates, as demonstrated in chapter 3. Unfortunately, age is often used artificially, since individuals are ascribed to stages according to other physiological criteria. This issue has been addressed by a new model developed as part of European project 99/080 (Borchers, Wood, Stratoudakis, Lopes, Bernal, Lago de Lanzós & Uriarte 2003) and will be introduced in the next chapter.

Chapter 5

A New Daily Egg Production

Method

5.1 Introduction

Individuals in populations for which stage abundances are observed are often allocated to stages according to physiological criteria that are not easily linked to age (see Gunderson (1993, pages 136–142) for an example of anchovy egg classification). Chapter 4 made clear that more than just population counts at developmental measurements are needed to obtain unique growth, birth and death rates. It was also found that when the developmental measurement is age, growth is defined as one day per day and the vital rates can then be obtained. Age is often used artificially to distinguish stages so that the inverse problem can be solved for the physiologically-structured population data.

Indeed this was the case in chapter 3 when the SPSP model was applied to Dover sole

egg data from the Bristol Channel. In the SPSP model, individuals are treated as if they automatically progress into the next developmental stage as soon as they reach the upper age limit of the current stage. The stage age limits were estimated for the sole egg samples from a model for stage duration (Horwood & Greer Walker 1990). However, the sole eggs were categorized into stages according to physiological development, not age, using the scheme of Riley (1974). This means that sole eggs of the same age could be in different stages of development, a situation which is not catered for in the SPSP model.

A new daily egg production method (DEPM) which addresses the ageing issue has been developed as part of a project for the European Commission (Borchers et al. 2003). The method has particular relevance to this thesis because it has been created to obtain birth and death rates from egg data consisting of physiologically-structured population counts. In addition, it provides a ray of hope after the rather cheerless conclusions of the last chapter for populations characterized by traits other than age, since it gives an example of how careful subsidiary experiments can help address the problem of not knowing the growth rate.

The purpose of this chapter is to consider the new DEPM method as a way forward for solving the inverse problem for physiologically-structured populations by carefully incorporating stage at age experiment data. This will be achieved by first describing the new method in section 5.2. Results achieved through application of the method are considered in section 5.3. Finally, in section 5.4 some implications and possible future developments are discussed.

5.2 The new DEPM model

The purpose of the new DEPM model is to obtain an estimate of the spawning stock biomass (SSB) of a fish population. This is achieved by estimating the total daily egg production then converting the estimate to SSB using adult fish parameters (see Gunderson (1993, page 130)). The scope of this thesis is restricted to the consideration of physiologically-structured populations so only the egg part of the new model shall be discussed.

The new DEPM model makes appropriate use of experimental data to obtain a link between developmental stage and age. This relationship can then be combined with the physiologically-structured population counts in a statistically justified manner to obtain the death rate and birth rate of the population. The stage-age relationship is modelled using a GLM and cohort abundance is modelled with a GAM.

The problem of artificial ageing has been overlooked in the past because we have been able to obtain 'age limits' for each stage from experimental stage-age data by dubious means:

The stage-age data consist of a time series of stage counts recorded for incubation experiments using individuals sampled from the population of interest. The age a of the sampled individuals is given by the observation time and so is a known quantity. Incubations may be set up for a discrete set of environmental conditions $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{e_n}\}$, (where e_n is the number of different sets used).

Past methods have estimated stage durations by modelling this type of data with age as a function of stage (for example, Lo (1985)). However, it is clear that in the experiment, age is a known quantity and stage is a random variate, so stage ought to be modelled as a function

of age. This causes a difficulty as the main goal is to model the physiologically-structured population survey data, and for that we need to turn stages into ages. In the new DEPM model, the problem is solved using a simple Bayesian approach and some clever stage at age modelling pioneered by Ibaibarriaga and Loneragan (Borchers et al. 2003, appendix A3) independently. (The scheme is also known as polytomous logistic regression, for which Hosmer & Lemeshow (1989) provide details.)

The various model components are explained in sections 5.2.1–5.2.6 and summarized in section 5.2.7. The notation used throughout this chapter is listed in table A.3 of appendix A.

5.2.1 Stage at age model for incubation experiment data

The stage-age experiment data consist of the number of individuals in each stage for a discrete set of age and environmental condition combinations, (a_i, \mathbf{x}_i) , $i = 1, \dots, e_n$. We require a model that allows us to estimate proportions in each stage for *any* combination of age and environmental conditions, $p(s|a, \mathbf{x})$. This is the probability mass function (pmf) of stage given age and environmental conditions. In theory, stage also depends on prior conditions but in practice this is ignored in egg production methods, as the conditions in the volume of water that the eggs inhabit (of which temperature is the most influential) should change little over a few days.

Let the number of individuals in each stage for a single age-environment combination be n_1, n_2, \dots, n_C where C is the maximum number of life-history stages recorded. Define a new variable

$$N_s = \sum_{j=s}^C n_j ,$$

the number of individuals of stage s and above. Let $P[s^+|(s-1)^+]$ denote the probability of an individual being in class s or above, given that it is in class $s-1$ or above. Then the expected value of N_s is

$$E(N_s) = P[s^+|(s-1)^+]N_{s-1} .$$

Using N_s as the response variable, and N_{s-1} as the corresponding binomial denominator, we can model $P[s^+|(s-1)^+]$ with a binomial GLM. In this way $P[s^+|(s-1)^+]$ is allowed to depend on age, the recorded environmental conditions, and perhaps stage s :

$$h(P[s^+|(s-1)^+]) = \eta(a, \mathbf{x}, s)$$

where h is a probit or logit link function and $\eta(a, \mathbf{x}, s)$ is the linear predictor defining the dependence on age, environmental conditions, and stage. Note that the environmental conditions used here must be a subset of those recorded for the physiologically-structured population survey data.

Given the fitted model, the required probabilities are easily obtained. Since the event ‘in stage s or above’ is a subset of the event ‘in stage $s-1$ or above’, we have that

$$P[s^+] = \prod_{j=2}^s P[j^+|(j-1)^+]$$

and from this we can get the probability $P[s]$ of being in stage s :

$$P[s] = P[s^+] - P[(s+1)^+] .$$

Through the CLM model, $P[s]$ is a function of age and the environmental conditions, and therefore gives the required pmf $p(s|a, \mathbf{x})$. This model is straightforward to fit, never implies negative stage durations, but also allows a good deal of flexibility in the shape of $p(s|a, \mathbf{x})$.

5.2.2 Ageing the physiologically-structured population

The final aim of the egg part of the new DEPM model is to obtain the number of individuals born to a population in one day P_0 , which is the ‘macro-birth rate’ $B(\mathbf{x})$ (for example, eggs spawned per day at \mathbf{x}) integrated over the survey region. The ‘micro-birth rate’ $\beta(t)$ discussed in this section is on a finer time-scale (for example, eggs spawned per hour) and is time dependent. The method is intended for populations in which distinct cohorts can be identified, so this micro-birth rate is expected to be intermittent, as seen in figure 5.1. The period of the micro-birth rate is the macro time unit, which conceivably might not be day. The macro-birth rate $B(\mathbf{x})$ is the integral of the micro-birth rate $\beta(t)$ over a macro time unit, ($B(\mathbf{x}) = \int_{\text{one macro time unit}} \beta(t) dt$) that is, the number of individuals born in a cohort. In figure 5.1 the macro-birth rate would be the area under one hump of the micro-birth rate function. The form of the micro-birth rate as a function of time is assumed to be known. It could also vary with the environment, but again the form of the relationship would have to be known.

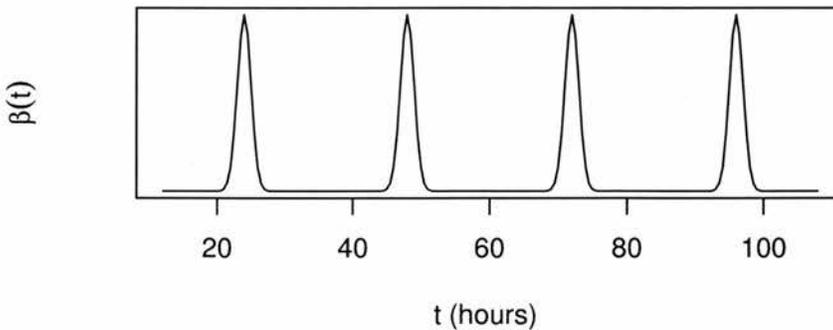


Figure 5.1: An example of a micro-birth rate, $\beta(t)$, against time. Only the shape of the function is assumed to be known, therefore the y-axis is arbitrary and no units are given.

In the absence of mortality, the micro-birth rate $\beta(t)$ gives the shape of the (improper) distribution of population ages at any time t :

$$p^*(a|t) \propto \beta(t - a) .$$

Given a death rate $\mu(\mathbf{x})$ that is independent of time, which is a reasonable assumption if sampling takes place over a short period, this in turn can be transformed into the shape of the (proper) probability density function (pdf) for age at time t :

$$p(a|\mathbf{x}, t) \propto \beta(t - a)e^{-\mu(\mathbf{x})a} .$$

Figure 5.2 illustrates this pdf for the micro-birth rate of figure 5.1.

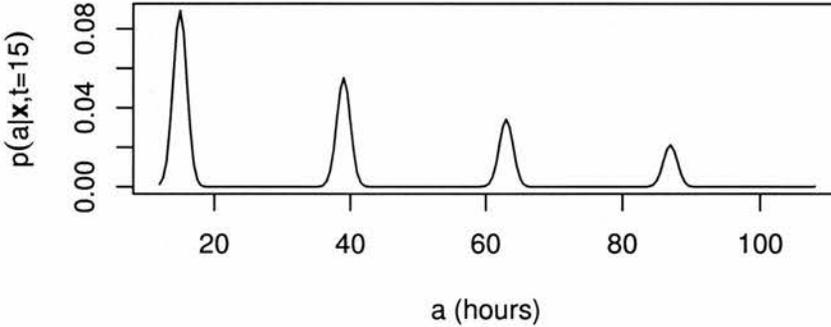


Figure 5.2: An example of the probability density function $p(a|\mathbf{x}, t)$ of age for environmental \mathbf{x} at time $t = 15$ hours, against age. This pdf corresponds to the micro-birth rate of figure 5.1.

The previous section showed how experimental stage at age data can be used to provide an estimate of the pmf of stage given age, $p(s|a, \mathbf{x})$. To be able to make use of the physiologically-structured population survey data, we require the pdf $p(a|s, \mathbf{x}, t)$ of age given stage for envi-

ronment \mathbf{x} at time t . Bayes' theorem lets us turn the pmf $p(s|a, \mathbf{x})$ into the pdf $p(a|s, \mathbf{x}, t)$:

$$p(a|s, \mathbf{x}, t) \propto p(s|a, \mathbf{x})p(a|\mathbf{x}, t) \propto p(s|a, \mathbf{x})\beta(t-a)e^{-\mu(\mathbf{x})a}.$$

The normalising constant for this pdf can be found by the numerical evaluation of:

$$\int_0^\infty p(s|a, \mathbf{x})\beta(t-a)e^{-\mu(\mathbf{x})a} da.$$

So

$$p(a|s, \mathbf{x}, t) = \frac{p(s|a, \mathbf{x})\beta(t-a)e^{-\mu(\mathbf{x})a}}{\int_0^\infty p(s|a, \mathbf{x})\beta(t-a)e^{-\mu(\mathbf{x})a} da}.$$

Evaluation of $p(a|s, \mathbf{x}, t)$ requires three components: the micro-birth rate $\beta(t)$, which is currently given by assumption; the pdf of stage given age $p(s|a, \mathbf{x})$, which is estimated from experimental data; and the death rate $\mu(\mathbf{x})$. Because the death rate is one of the functions to be obtained by solving the inverse problem, it must be estimated iteratively. Section 5.2.7 describes this procedure.

5.2.3 From stage abundances to cohort abundances

A cohort from a population consists of individuals in a particular age group. We know from the previous chapters of this thesis that it is possible to find unique solutions to the inverse problem for a population characterized by age. So estimation of the birth and death rates of the surveyed physiologically-structured population is possible if we can use the pdf $p(a|s, \mathbf{x}, t)$ of age given stage to allocate members of the population to cohorts.

Given a sample time t , cohort allocation can be performed by working out from the micro-birth rate assumption, the age range of eggs within each of the cohorts and calculating the proportion of individuals of each stage within this age range. For example, suppose the

macro-time unit is day. If, given a sampling time t , last night's cohort includes all individuals between ages $\alpha_0(t)$ and $\alpha_1(t)$ hours, then the proportion of the stage s population in last night's cohort would be

$$\int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}, t) da .$$

If n_{si} stage s individuals were observed in environment \mathbf{x}_i ,

$$n_{si} \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da$$

individuals from stage s would be allocated to last night's cohort. For C life-history stages, the total number of individuals y_{1i} at \mathbf{x}_i from all stages allocated to last night's cohort would be

$$y_{1i} = \sum_{s=1}^C n_{si} \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da .$$

The mean age of the cohort is also easily obtained. Using the same example, the mean age \bar{a}_{1i} for last night's cohort would be

$$\bar{a}_{1i} = \frac{\sum_{s=1}^C \left(n_{si} \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) a da / \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da \right)}{\sum_{s=1}^C n_{si} \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da}$$

The term

$$\frac{\int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) a da}{\int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da}$$

is just the mean age of eggs from stage s in last night's cohort. The overall mean age \bar{a}_{1i} has been obtained by taking a weighted average of such terms over all stages, the weights being the number of individuals in the cohort from the corresponding stage. \bar{a}_{1i} simplifies a little to

$$\bar{a}_{1i} = \frac{\sum_{s=1}^C n_{si} \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) a da}{\sum_{s=1}^C n_{si} \int_{\alpha_0(t)}^{\alpha_1(t)} p(a|s, \mathbf{x}_i, t) da} .$$

The mean cohort age is required because the cohort abundance model of the next section finds the macro-birth rate in terms of macro time units. The mean cohort age takes this form so that it implicitly contains the death rate to adjust for the relatively few old individuals in each cohort — for each cohort, the mean age of survivors is not equal to the mean age of the individuals who were born to the cohort.

The abundances y_{ki} and mean ages \bar{a}_{ki} are found for any other cohort $k \in \mathbb{N}$ (for example, cohorts of previous evenings) in the same way using the age range $(\alpha_{k-1}(t), \alpha_k(t))$. In the software provided to fit the new DEPM model (available from www.ruwpa.st-and.ac.uk/depm/), the integrals required are all evaluated using crude summations of gridded values of the integrands.

In summary, a new ‘data set’ has been created from the observed stage abundances and the estimated pdf $p(a|s, \mathbf{x}, t)$, consisting of the expected abundance and the expected average age for each cohort at each sample point. Ideally, the model would predict numbers in each physiological stage directly, but this is impossible and we require information about age. The procedure used to obtain the new data set is sensible and so using pseudo-data provides the best way forward. A side effect of this approach is that cohorts can contain fractions of eggs, but in practice this does not matter for fitting and inference.

A common problem is that cohorts can be substantially incomplete. This happens when sampling has occurred part way through a macro-time unit so only a small proportion of the cohort have been born. If the population under study matures at the end of the C th stage, rather than die of old age, old cohorts may be incomplete because some members have matured. There is no way of distinguishing between death and maturation in the model,

and indeed chapter 4 indicates it would be foolish to try. These incomplete cohorts must be discarded for the model of the next section to be a valid description of cohort abundance. Figure 5.3 shows for a given environment \mathbf{x} sampled at time t , (a) the observed cohort abundances y_k derived from (b) the assumed micro-birth rate $\beta(t)$, having been reduced by mortality and for the fourth cohort, maturation. The shaded region indicates which cohorts are to be discarded. The parameter of interest, the macro-birth rate $B(\mathbf{x})$, is shown in figure 5.3(c). Section 5.2.4 explains how $B(\mathbf{x})$ is to be estimated from y_k .

5.2.4 Modelling cohort abundances to obtain birth and death rates

We are now in a position to solve the inverse problem. The ‘data’ to be used are the set of ‘observations’ y_{ki} of cohort abundances Y_{ki} and cohort mean ages \bar{a}_{ki} for a number of complete cohorts k at every set of environmental conditions \mathbf{x}_i recorded at sample time t_i . The model structure used is

$$f_k(\mathbf{x}_i) = E(Y_{ki}) = R_i B(\mathbf{x}_i) e^{-\mu(\mathbf{x}_i) \bar{a}_{ki}} .$$

The term R_i is the effective area sampled, and $\log(R_i)$ is an offset. If R_i is the same for all $i = 1, \dots, n$, it can be discarded, but is often required for surveys of pelagic fish. The dependence of the macro-birth rate $B(\mathbf{x})$ and the death rate $\mu(\mathbf{x})$ on environmental conditions \mathbf{x} is represented using a GAM with a log link function. Hastie & Tibshirani (1990) is a standard text on GAM theory; Borchers et al. (2003, appendix 1) describes the statistical theory underlying the methods used here.

For illustration, consider a survey of the eggs of a pelagic fish species. The environmental conditions measured might be longitude x_1 , latitude x_2 , depth x_3 and temperature x_4 , $\mathbf{x} =$

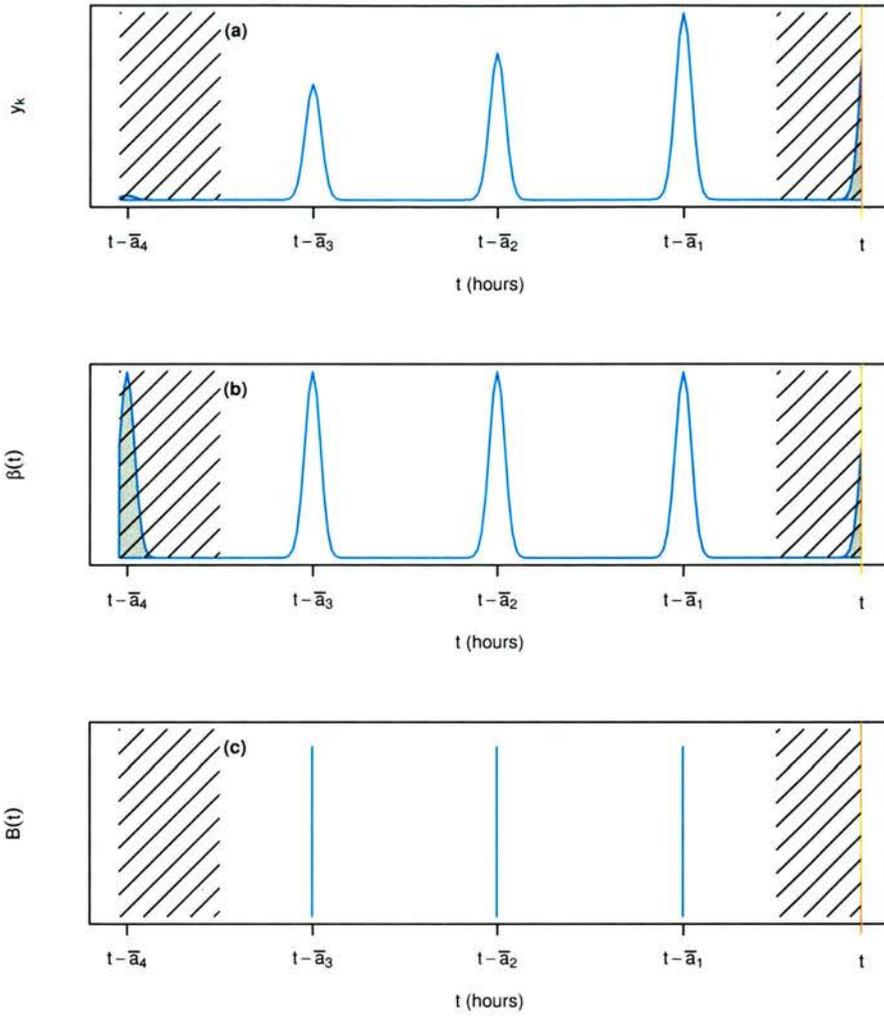


Figure 5.3: Suppose sampling occurs for environment x at time t , indicated by the red line in the above graphs. The observed cohort abundances y_k $k = 1, \dots, 4$ shown in (a), are all that have survived and not yet matured from the (b) individuals born according to the micro-birth rate before sampling took place. Ultimately we are trying to estimate the height of the macro-birth rate $B(x)$ shown in (c). For this example, the macro time unit could be day.

$(x_1, x_2, x_4, x_4)^T$. Suppose that the macro-birth rate (daily spawning rate) is thought to depend on longitude, latitude and depth, and the death rate is thought to depend on latitude

and temperature. Then the model can be written

$$\log(f_{ki}) = \log(R_i) + S_1(x_{1i}, x_{2i}) + S_2(x_{3i}) - S_3(x_{2i})\bar{a}_{ki} - S_4(x_{4i})\bar{a}_{ki}.$$

The macro birth rate is represented by the exponent of the sum of the first two smoothing functions, $B(\mathbf{x}) = \exp(S_1(x_1, x_2) + S_2(x_3))$, and the death rate is represented by the sum of the third and fourth smoothing functions, $\mu(\mathbf{x}) = S_3(x_2) + S_4(x_4)$. As with SPSP model, the explanatory variables in the multi dimensional smoothing functions must have similar ranges. The method of fitting the cohort abundance model to obtain the smoothing functions is explained in Borchers et al. (2003, appendix 1).

Briefly, the model can be found by minimizing an objective function within an iteratively re-weighted least squares loop. Like the SPSP model, the objective function consists of a sum of squares and a sum of wiggleness penalties, with an extra fixed penalty term without associated smoothing parameters. Wood (2000) provided a method developed from Gu & Wahba (1991) for minimizing a wide range of penalized sum of squares objective functions including those of GAMS, but excluding fixed penalty terms. (The fixed penalty is used to help counter lack of identifiability so that separate smoothing functions for birth and death can be dependent on the same environmental variables). Wood's new method in appendix 1 of the EC project report caters for a fixed penalty term in the objective function, and improves the numerical stability and efficiency of the fitting process.

The cohort abundance model is very similar to the SPSP model of chapter 2. In the SPSP model stages were treated as cohorts and the 'cohort' age limits were used to define the function; in the cohort abundance model of the new DEPM method, the mean ages of the cohorts are used. The cohort abundance model structure is the same as that of the SPSP model

when the birth and death rates are independent of time, as it utilizes the same demographic processes, and in both models, smoothing functions are used to relate the birth and death rates to environmental conditions.

5.2.5 Variance estimation

Variance estimates for any parameters of interest can be obtained using non-parametric bootstrapping. Variation due to the ageing procedure is included using a two-step process. For each of the b bootstraps,

1. randomly sample c_n times with replacement from the stage at age experiment data. Each sampling unit contains a unique combination of environmental conditions and age (\mathbf{x}_i, a_i) with their associated observed 'stage and above' abundances N_1, \dots, N_C . The stage at age multinomial GLM is fitted to this set.
2. randomly sample n times with replacement from the physiologically structured population abundances. Each sampling unit contains a unique combination of environmental conditions and age (\mathbf{x}_i, t_i) with their associated observed stage abundances n_1, \dots, n_C . (Note that other sampling units, such as 'transect's, could be used.) The stage at age model from the previous step is used to convert these to cohort abundances y_{kt} and mean ages \bar{a}_{ki} , and then a GAM is fitted. The GAM is used to predict any parameters of interest which are then stored.

The list of b bootstrap parameter estimates make up the empirical distribution of the parameter and can be used to calculate variance and confidence intervals. For example, in DEPM,

the total daily egg production, P_0 , is estimated. Confidence intervals for \widehat{P}_0 can be found using the standard error from the bootstrap estimates and a log-normal assumption.

When applying the new DEPM model to real data in the course of the EC project, a problem with this non-parametric bootstrapping method emerged. If one-dimensional smooths are included in the GAM, and the most extreme value(s) of the explanatory variable are not included in a bootstrap re-sample, severe overestimation of the parameter of interest might result for that re-sample. This occurs when a tail of the smoothing function increases toward the extreme, at either extreme. The parameter of interest usually requires prediction of a fitted function (for example, the macro-birth rate) for the whole of the original data space, and prediction at the extremes of this space therefore requires extrapolation. The result of severe overestimation for a few bootstrap re samples is overestimation of the variance. When the tail of the smoothing function decreases towards the extremes and extrapolation is required, the problem is not so acute, because underestimation is restricted by the log link function. The problem did not arise for the multi dimensional smooths because gradients at the extremes of the fitted surfaces are influenced by many neighbouring points.

The tactic used to deal with the overestimation problem was to force each bootstrap re sample to include those samples with extreme values of the environmental variables used in the one dimensional smooths. Bayesian confidence regions for the GAM fitted to the cohort abundances are described in Borchers et al. (2003, appendix 1).

The methods used above are not the only or best methods. The parametric bootstrap is an alternative which can be made to model the dependence within the data but does require that the distributional assumptions are reasonable. The fish egg survey data for which the

DEPM model was created are expected to be over-dispersed relative to a Poisson distribution (so a negative binomial distribution could be used which may create computational complexities), and further difficulties arise from a possibility that the eggs disperse as they age. Should the parametric bootstrap be selected, a parametric bootstrapping scheme for combining the experiment and survey data would have to be devised. A better scheme for (non-parametrically) bootstrapping the stage at age experiment data than currently used would be to condition on age to 'replicate' the experiment. The Bayesian bootstrap (Rubin 1981) could be investigated as a method for dealing with the overestimation problem for the one-dimensional smooths in the GAM. Different schemes for confidence interval estimation are possible, such as the BC_a (bias-corrected and accelerated) percentile method (Efron & Tibshirani 1993).

5.2.6 Model assumptions

The assumptions for the new DEPM model are listed in the third box of section 5.2.7. The model assumptions are similar to those of the SPSP model, but the SPSP model assumes that stage age limits are known, whereas the new DEPM model makes the more reasonable assumption that the cohort age limits are known. Knowledge of the cohort age limits follows from the micro-birth rate, the macro time unit, and the sample time. The assumption that the shape of the micro-birth rate is known may cause difficulties as it will have to be derived from separate studies. For example, problems arose when the new DEPM model was applied to sardine (*Sardina pilchardus*) data because traditional opinion about the timing of the peak of the micro-birth rate (the spawning rate) differed from the findings of Ré, Farinha & Meneses (1988) and Ganias, Somarakis, Machias & Theodorou (2003). Furthermore, the

standard deviation of the micro-birth rate suggested by the hourly distribution of stage 1 abundances for all of the sardine surveys differs from that obtained by Ré et al. (1988).

The time-invariance assumption for the macro-birth rate is fundamental to the DEPM method for estimating spawning stock biomass. However, the model could be adapted to relax the assumption for other populations and purposes. In this case, a time series of observed stage abundances must be available.

When the time-invariance assumption for the macro-birth rate is relaxed, the cohort abundance model becomes

$$f_k(\mathbf{x}_i) = R_i B(\mathbf{x}_i, t_i - \bar{a}_{ki}) e^{-\mu(\mathbf{x}_i) \bar{a}_{ki}}$$

and in the sum of the smoothing functions representing the logarithm of the macro-birth rate, one explanatory variable will be time. Figure 5.4(c) illustrates the time-dependent macro-birth rate as a dashed blue line. For results to make sense, prediction of the macro-birth rate should only occur at macro time units.

The attraction of assuming that the death rate is independent of time is clear. The survival term in the pdf of age given stage and the GAM is uncomplicated, and so the design matrix of the GAM can be written $\mathbf{X} = [\mathbf{X}_B, -\bar{\mathbf{a}}\mathbf{X}_\mu]$, where \mathbf{X}_B corresponds to the smooth functions making up the logarithm of the macro-birth rate, \mathbf{X}_μ corresponds to the smooth functions of the death rate, and $\bar{\mathbf{a}} = (\dots \bar{a}_{k1} \dots \bar{a}_{kn} \dots)^T$. Using the new model fitting techniques, a GAM with this design matrix is identifiable even if the same environmental variables are used for the macro-birth rate and death.

Relaxing the time-invariance assumption for the death rate is not straightforward. The pdf

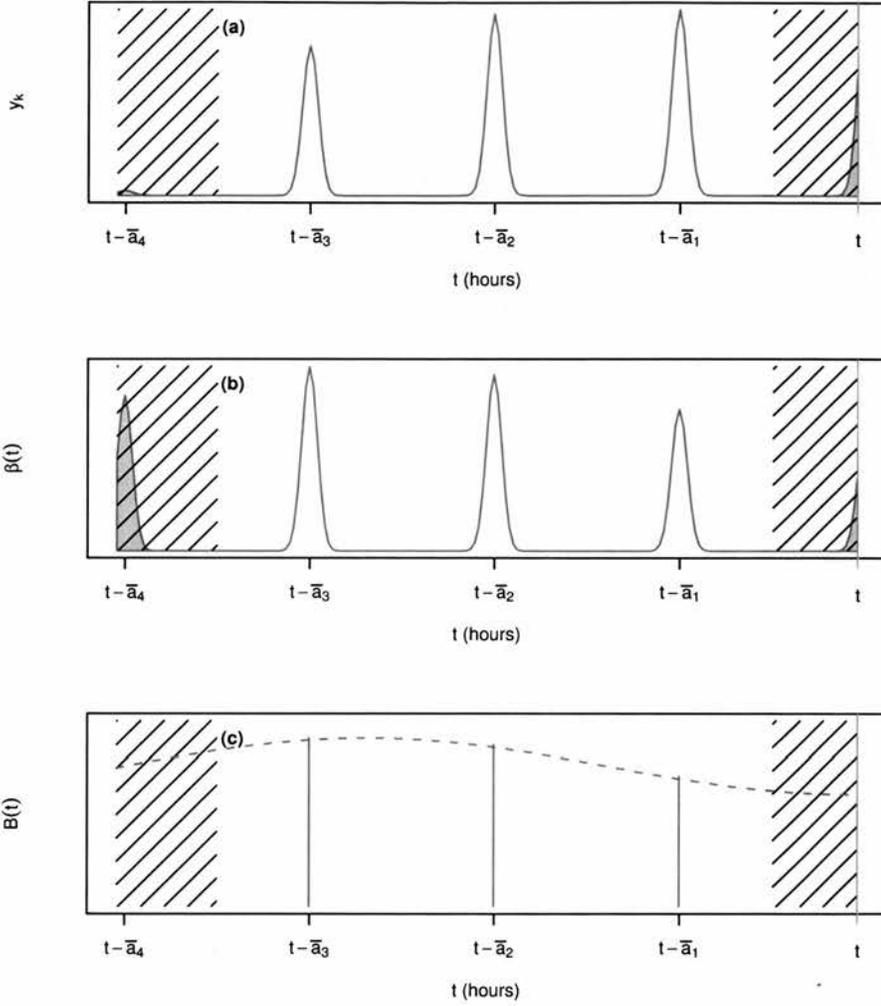


Figure 5.4: Suppose sampling occurs for environment \mathbf{x} at time t , indicated by the red line in the above graphs. The observed cohort abundances y_k $k = 1, \dots, 4$ shown in (a), are all that have survived and not yet matured from the (b) individuals born according to the micro-birth rate before sampling took place. Ultimately we are trying to estimate the macro-birth rate curve $B(\mathbf{x})$ shown by the dashed line in (c).

of age given stage and the cohort abundance model change to

$$\begin{aligned}
 p(a|s, \mathbf{x}, t) &= \frac{p(s|a, \mathbf{x})\beta(t - a) \exp\left(-\int_{t-a}^t \mu(\mathbf{x}, t) dt\right)}{\int p(s|a, \mathbf{x})\beta(t) \exp\left(-\int_{t-a}^t \mu(\mathbf{x}, t) dt\right) da} \\
 f_k(\mathbf{x}_i) &= R_i B(\mathbf{x}_i, t_i - \bar{a}_{ki}) \exp^{-\int_{t_i - \bar{a}_{ki}}^{t_i} \mu(\mathbf{x}_i, t) dt} .
 \end{aligned}$$

It is not obvious how this new version of the cohort abundance model is to be fitted since the design matrix for a single GAM is no longer easy to define. Furthermore, there could be problems for populations with narrow micro-birth rate peaks, such as sardine and anchovy. Generalizing the new DEPM model to allow time dependence of the death rate therefore requires further study. It might be possible to model the logarithm of cohort abundance using two separate GAMs — one for the logarithm of the macro-birth rate and one for the death rate in a process similar to that of the SPSP model.

5.2.7 Model summary

The new DEPM model is made up from a number of components encapsulated below. Notation for this chapter is listed in table A.3 of appendix A.

Model Components

Equations for utilizing the stage at age experiment data:

$$N_s = \sum_{j=s}^C n_j$$

$$h\left(\frac{E(N_s)}{N_{s-1}}\right) = h(P[s^+|(s-1)^+]) = \eta(\mathbf{x})$$

$$P[s^+] = \prod_{j=2}^s P[j^+|(j-1)^+]$$

$$p(s|a, \mathbf{x}) = P[s] = P[s^+] - P[(s+1)^+]$$

Equation for pdf of age given stage at time t :

$$p(a|s, \mathbf{x}, t) = \frac{p(s|a, \mathbf{x})\beta(t-a)e^{-\mu(\mathbf{x})a}}{\int p(s|a, \mathbf{x})\beta(t-a)e^{-\mu(\mathbf{x})a} da}$$

Equations for converting stage abundances to cohort abundances:

$$y_{ki} = \sum_{s=1}^C n_{si} \int_{\alpha_{k-1}(t)}^{\alpha_k(t)} p(a|s, \mathbf{x}_i, t) da$$

$$\bar{a}_{ki} = \frac{\sum_{s=1}^C n_{si} \int_{\alpha_{k-1}(t)}^{\alpha_k(t)} p(a|s, \mathbf{x}_i, t) a da}{\sum_{s=1}^C n_{si} \int_{\alpha_{k-1}(t)}^{\alpha_k(t)} p(a|s, \mathbf{x}_i, t) da}$$

Equations for modelling cohort abundances

$$E(Y_{ki}) = f_{ki} \quad Y_{ki} \sim \text{an exponential family distribution}$$

$$f_k(\mathbf{x}_i) = R_i B(\mathbf{x}_i) e^{-\mu(\mathbf{x}_i) \bar{a}_{ki}}$$

$$\log(f_k(\mathbf{x}_i)) = \log(R_i) + \log(B(\mathbf{x}_i)) - \mu(\mathbf{x}_i) \bar{a}_{ki}$$

For environmental conditions \mathbf{x}_i sampled at time t_i , $i = 1, \dots, n$, stages $s = 1, \dots, C$, cohorts k , ($k \in \mathbb{N}$ restricted to those cohorts that are complete).

Model Assumptions

The model assumptions are:

- the population is closed (no migration or advection)
- the macro-birth rate $B(\mathbf{x})$ is independent of time (this is a fundamental assumption of DEPM)
- the death rate $\mu(\mathbf{x})$ is independent of age and time
- $\mu(\mathbf{x})$ and $\exp(B(\mathbf{x}))$ are smooth functions of the environmental conditions \mathbf{x}
- the cohort abundances Y_{ki} are ‘observations’ from an exponential family distribution
- the count N_s of individuals in stage s or above from the incubation experiment are observations from a binomial distribution with binomial denominator N_{s-1}
- the form of micro-birth rate $\beta(t)$ is known and it has a constant period, hence the age limits α_k of the cohorts are also known

The process by which the model is fitted is summarized below. Model fitting software is available as an R package called `egg` which uses the `mgcv` library. The software can be found by following links from <http://www.ruwpa.st-andrews.ac.uk/depm/>.

Model Fitting Algorithm

1. Fit a binomial GLM to the stage at age experiment data using N_s as the response, with N_{s-1} as the corresponding binomial denominator. Convert the fitted response to the probability of stage given age $p(s|a, \mathbf{x})$.
2. Initialize the death rate estimate $\mu(\mathbf{x})$.
3. Obtain the pdf $p(a|s, \mathbf{x}, t)$ of age given stage from $p(s|a, \mathbf{x})$, the current estimate of $\mu(\mathbf{x})$, and the assumed micro-birth rate form $\beta(t)$, using Bayes' rule.
4. For each sample point \mathbf{x}_i , use $p(a|s, \mathbf{x}, t)$ to convert stage abundances n_{si} , $s = 1, \dots, C$, to cohort abundances y_{ki} , $k = 1, \dots$ with associated mean ages \bar{a}_{ki} . Delete incomplete cohorts from this new 'data set'.
5. Fit a GAM with log link function to the cohort abundances y_{ki} with effective area sampled R_i as an offset to obtain estimates of the macro-birth rate $B(\mathbf{x})$ and the death rate $\mu(\mathbf{x})$ in terms of the environmental conditions \mathbf{x} .
6. Repeat from step 3 until $\mu(\mathbf{x})$ converges. Once convergence has occurred, calculate any parameters of interest from the final model and use bootstrapping to get variance estimates.

The DEPM model was originally developed with certain data already in mind. The characteristics of the data led to choices about the form of the model, so important features of the data are listed here.

Features of the Data

- Stage durations vary depending on stage and temperature (from approximately 8 to 45 hours for sardine, approximately 5 to more than 100 hours for anchovy), and several stages may last longer than one macro-time unit (38% for sardine, 58% for anchovy). If stages had lasted less than one macro time unit, the need for the cohort ageing process would be less apparent.
- Although sampling was not repeated at any one sample point, sampling occurred frequently (never with lapses of more than one micro-time unit) within and through macro time units so it was possible to observe the timing of the micro birth rate. (The micro birth rate was expected to be the same throughout the survey region).
- The micro-birth rate is cyclic — it repeats once per macro-time unit. For the data used, the micro birth rate consisted of a sharp peak at night dropping to zero elsewhere. The pulse had a width of around 2 hours for a macro time unit of day. Such sharp pulses would make it difficult to model the micro-birth rate directly using a GAM.
- Sampling lasted over many (approximately 30, 20 and 70) macro-time units so the assumption that the death rate is independent of time is rather suspect.
- Sampling in space was fairly frequent and regular, as can be seen in figure 5.5.

5.3 Results from application of the new DEPM model

The new DEPM model fitting process was coded and then tested using simulated data (Borchers et al. 2003, appendix 3), before being applied to anchovy (*Engraulis encrasicolus*)

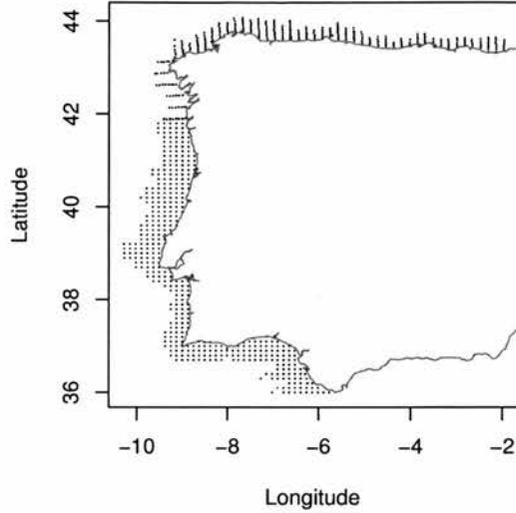


Figure 5.5: Sample locations around the Iberian Peninsula for the survey of sardine eggs in 1999.

egg data from the Bay of Biscay by AZTI¹ and sardine (*Sardina pilchardus*) egg data from the Iberian Peninsula by IPIMAR² and IEO³.

Egg incubation experiments had been performed for the anchovy population in 1992 (Motos 1994), and for the sardine population in 2002 using the same incubators. Multinomial models as described in section 5.2.1 were fitted to the data from these experiments. The models were found to be more satisfactory than traditional models:

- The theoretical basis for the multinomial model is more sound because stage is treated as a random variate and age is fixed. In addition, stage is considered (correctly) to be a discrete random variable, whereas in the past it has been handled as a continuous

¹Instituto Tecnológico Pesquero y Alimentario

²Instituto de Investigação das Pescas e do Mar

³Instituto Español de Oceanografía

random variable (Lo 1985).

- The model is more flexible than those previously available as past methods allow only one age value for each stage, whereas the GLM method allows a range of ages for each stage. This improves modelling of stage duration. For example, in Lo (1985) and Motos (1994), stage duration is constant with respect to environmental conditions but using the multinomial GLM, dependence on environmental variables can easily be modelled. In Miranda, Cal & Iglesias (1990), stage duration can vary continuously with stage and temperature, but the relationship is monotonic. Stages 3 and 4 of the sardine eggs have much shorter durations than the other stages and so cannot be dealt with by a monotonic relationship. The flexible GLM was able to capture the short stage durations by fitting small probabilities of being in that stage. Unfortunately, the functional form of the multinomial model means that each stage has a long-tailed age distribution, in that the probability of being in stage s is non zero despite age being far beyond the range for that stage.

Physiologically-structured egg abundance surveys have been carried out annually for the Bay of Biscay anchovy population, and rather less regularly for the sardine population of the Iberian Peninsula. The new DEPM model was applied to the anchovy survey data of 1996–1999 and to the sardine survey data of 1988, 1990, 1997, 1999 and 2002, to obtain total daily egg production estimates.

AZTI, IPIMAR and IEO report reasonable parity between the new and traditional point estimates of the total daily egg production, when the traditional prediction grid was used for both. Larger discrepancies for the sardine eggs could be explained by artifacts of differing

data sets.

Improvements in precision (coefficient of variation reductions of 5–12%) were recorded for the sardine estimates, which should impact on the precision of the adult spawning stock biomass (SSB) estimate since the total daily egg production estimate typically accounts for more than 35% of the SSB variance estimate. The improvements occurred despite the variance estimates for the new method incorporating uncertainty due to the ageing procedure, which the traditional estimates do not include. Precision of the total daily egg production estimate for the anchovy was not improved, but it is expected that the traditional variance estimates are negatively biased. The coefficients of variation for the new method were still very respectable at 13–21%, and are more realistic than the traditional version.

The GAM model confirmed relationships between anchovy egg population density and salinity and sea surface temperature, and mapped the migration of sardine spawning from 1988 to 1997. No relationship was found between sardine egg population density and environmental variables.

5.4 Discussion

The new DEPM method provides a new approach for dealing more honestly with physiologically-structured population data. With such a model available, the strategy of using age to artificially distinguish physiological stages when attempting to solve the inverse problem can be abandoned. As with the SPSP model, the new DEPM model is based on smoothing functions allowing flexibility in the form of the macro-birth and death rates. Both models use the same demographic framework to relate abundance to the birth and death rates. Because it deals

with structured populations that are assigned to stages using criteria other than age, the DEPM model does require stage at age experimental data and knowledge of the shape of the micro-birth rate, besides the physiologically-structured population abundance data.

Software incorporating elegant and robust fitting methods for the new DEPM model is freely available. The model is straightforward to apply using this software and useful statistics and surface plots can be produced. GLM and GAM model selection techniques can be employed to make informed decisions about the choice of environmental variables.

Variance estimates have been made more honest than those obtained for previous studies by including uncertainty due to the ageing procedure, and, for the case of sardines, precision was improved despite this.

The flexibility afforded by using a GAM to model cohort abundance, macro-birth and death rates can often reveal interesting points for further investigation. For example, higher estimates of the total daily egg production method were obtained for the anchovy data when the spatial integration grid was finer than that used traditionally. This could occur because fine detail is smoothed out when integration is performed on a coarse grid. Another issue was raised when an anchovy cohort abundance model with constant death was compared with a model in which death was allowed to vary over space. Total daily egg production estimates from the variable death GAM were higher than estimates from the constant death GAM by up to 6.7% (though for one year it was lower by 1%). The map of the death rate over space revealed that high death rates coincided with high egg production. This finding prompted recognition of the need to investigate by simulation and field study whether the death rate really is higher in these regions, or whether mortality is compounded with dispersion due to

development (younger eggs are more clumped than older eggs). In more traditional methods, the death rate was assumed to be constant and so such issues were not revealed.

Further generalization of the model to allow for time dependence of the macro-birth rate is straightforward, so long as time series physiologically-structured population counts are available. It may be possible to relax the time-invariance of the death rate but this would require further investigation. Further work could also be done to improve variance estimation. For the stage at age experiment data, bootstrapping could be conditioned on age. The Bayesian bootstrap (Rubin 1981) could be used to tackle the problem of extreme overestimation when using one dimensional smoothing functions. Another issue requiring study is the effect of dispersion. Dispersion is not allowed for in the DEPM model but fish eggs are thought to disperse as they age, and the effects of dispersion and the implications for modelling are yet to be understood.

In chapter 4 the impossibility of solving the inverse problem for physiologically-structured population data when age does not define stage was expounded. The new DEPM model makes solution of the inverse problem possible for just this situation by incorporating additional stage at age data in a statistically justified manner.

Chapter 6

Discussion

The primary aim of this thesis was to investigate whether smoothing splines, simple demographic processes and environmental variables can be used to obtain improved birth and death estimates from physiologically-structured population counts. This aim has been addressed in chapters 2–5 which consider the inverse problem for structured population counts. In each case, a simple demographic process is used to relate population density per unit age to birth and death rates, and in chapters 2, 3 and 5, the population density function is also linked to stage abundances. Environmental variables are incorporated by including multi-dimensional smoothing splines in the SPSP model of chapter 2, and by including GAMs consisting of smoothing splines in the new DEPM model presented in chapter 5.

The investigation carried out in this thesis has found that, for certain conditions, smoothing splines, simple demographic processes and environmental variables can be used to improve estimates of birth and death rates from physiologically-structured population counts. The conditions are that the physiological stages can be defined by age and age limits are known

for each stage throughout the data space, or that additional stage at age data and knowledge of the shape of the micro-birth rate are available. Improvements of the birth and death rate estimates are made in precision, reliability and avoidance of mis-specification bias, the ease with which the estimates can be obtained, and the amount of detail available (the rates are specified for the environmental variables). These points are argued for in the discussion sections of chapters 2, 3 and 5.

Chapter 4 discovered that a unique solution to the inverse problem cannot be found when the physiological trait defining the stages is not age, though a unique solution may be possible in a few special cases, most notably the time-invariance case. Limits can be imposed upon the growth, and hence death and birth rates when growth is independent of the measured physiological trait, but these limits may not restrict the vital rates sufficiently to be of practical use. This means that when physiologically-structured population counts are collected, additional data is required, relating the physiological trait defining the stages to age. If this additional data is not available, the staged population counts cannot be used.

When the physiological trait itself is age, I have provided in this thesis an SPSP model for obtaining estimates of population density, birth and death rates from the stage abundances. When the physiological trait is not age, the new DEPM model provides a method of combining stage at age experimental data with stage abundances to obtain estimates of population density, birth and death rates. For this, the form of the micro-birth rate, that is, the shape of the birth rate within the time unit of the (macro-)birth rate to be estimated, must be known.

When model fitting software is available, both of these models are straightforward to apply. Software already exists and is freely available at www.ruwpa.st-andrews.ac.uk/depm for the

new DEPM model, and some computer code for fitting the 3-dimensional version of the SPSP model is contained in Appendix C. This democratizes the information to be gleaned from physiologically-structured population data but care is still needed in applying the models to the data and making inferences. In addition, ad hoc or inappropriately simple modelling tools should be selected less often for stage abundance data, because users should find the relatively sophisticated new DEPM and SPSP models fairly straightforward to implement.

In relation to stock assessment in the fisheries industry, the OECD found that it had

... to acknowledge that the confidence limits attached to the estimates are wide and contribute significantly to a lack of confidence in the resulting advice. (EU-ROSTAT 1995)

Both models can be used within egg production methods to improve the precision and narrow the confidence intervals of total spawning stock biomass estimates and these improvements are important since they can redress the lack of confidence in scientific advice to the fisheries industry.

Improved precision follows from incorporation of simple demographic processes and environmental conditions into the models. These model attributes also enhance the information from stage count data that has been made accessible, as the models can be used to study the correspondence between population density, birth or death rates and environmental conditions.

Furthermore, assumptions previously used to help simplify the inverse problem have been reduced, and the smoothing splines from which the SPSP and new DEPM model are built only require fairly reasonable assumptions for the data. It may be difficult to decide whether

strong assumptions about a population are correct in the absence of comprehensive data. Reasonable assumptions mean that results should not suffer from mis-specification bias. By not pretending to be certain about the shape of the unknown vital rates, variance estimates will be more honest. However in the new DEPM model, the knowledge of the shape of the micro-birth rate is assumed to be complete, but this knowledge can be contentious and hard to obtain so mis-specification problems may arise.

Areas for continued work arising from this thesis are:

- Extension of the SPSP model to use GAMs to model the birth and death rates, and possible extension of the DEPM model to allow time dependence of the macro-birth rate and the death rate.
- Improved variance estimation procedures — the non-parametric bootstrap used for the SPSP and DEPM models involves assumptions which may not be valid, including independence, which appears not to hold for some spatially distributed fish egg abundances. Alternative methods for obtaining the CV and confidence intervals could also be explored.
- There are various specific issues with the application of the SPSP model to the sole egg data requiring examination, and it may turn out that the model should not be applied to data sampled infrequently through time.
- The implications of possible fish egg dispersion for the SPSP and DEPM models needs to be explored, since neither model allows for dispersion.
- Although the general inverse problem for size and time dependent growth and death rates cannot be uniquely solved, there are some special cases (including time invariance

and pulsing birth rates) which are worthy of further study, since intuitively it appears that unique solutions may exist in these cases.

In summary, the two models described in this thesis allow useful and accurate information to be obtained for any population for which physiologically-structured count data and stage-age data are available. In the introduction to this thesis, it was pointed out that useful and accurate information are prerequisites for sound population management decisions. The impact of good population management is illustrated by the results of the FAO's handling of the locust outbreak in Afghanistan (FAO News, 4th August 2003) reported in chapter 1. By providing credible modelling tools, the results of this thesis increase the possibility of better population management decisions for the case when physiologically-structured count data and stage-age data are being used. As a consequence of this, the potential for trust between scientists, decision makers, and those who are affected by the decisions, is enhanced.

Bibliography

- Ackleh, A. S. (1999), "Parameter identification in size structured populations models with nonlinear individual rates", *Mathematical and Computer Modelling* **30**, 81–92.
- Aksnes, D. L., Miller, C. B., Ohman, M. D. & Wood, S. N. (1997), "Estimation techniques used in studies of copepod population dynamics a review of underlying assumptions", *Sarsia* **82**, 279–296.
- Augustin, N. H., Borchers, D. L., Clarke, E. D., Buckland, S. T. & Walsh, M. (1998), "Spatiotemporal modelling for the Annual Egg Production Method of stock assessment using Generalized Additive Models", *Canadian Journal of Fisheries and Aquatic Science* **55**, 2608–2621.
- Ben-Tuvia, A. (1990), "A taxonomic reappraisal of the Atlanto-Mediterranean soles *Solea solea*, *S. senegalensis* and *S. lascaris*", *Journal of Fish Biology* **36**, 947–960.
- Borchers, D. L., Buckland, S. T., Priede, I. G. & Ahmadi, S. (1997), "Improving the precision of the Daily Egg Production Method using Generalized Additive Models", *Canadian Journal of Fisheries and Aquatic Sciences* **54**, 2727–2742.

BIBLIOGRAPHY

- Borchers, D. L., Wood, S. N., Stratoudakis, Y., Lopes, P., Bernal, M., Lago de Lanzós, A. & Uriarte, A. (2003), "Using environmental variables with improved DEPM methods to consolidate the series of sardine and anchovy estimates", Final report of project 99/080, European Commission.
- Campbell, R. G., Wagner, M. M., Teegarden, G. J., Boudreau, C. A. & Durbin, E. G. (2001), "Growth and development rates of the copepod *Calanus finmarchicus* reared in the laboratory", *Marine Ecological Progress Series* **221**, 161–183.
- Craven, P. & Wahba, G. (1979), "Smoothing noisy data with spline functions", *Numerical Mathematics* **31**, 377–403.
- Crowder, L. B., Rice, J. A., Miller, T. J. & Marschall, E. A. (1992), "Empirical and theoretical approaches to sine-based interactions and recruitment variability in fishes", in D. L. DeAngelis & L. J. Gross, eds, "Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems", Chapman and Hall, New York, chapter 11, pp. 237–255.
- de Roos, A. M. (1997), "A gentle introduction to physiologically-structured population models", in S. Tuljapurkar & H. Caswell, eds, "Structured-Population Models in Marine, Terrestrial, and Freshwater Systems", number 18 in "Population and Community Biology", Chapman and Hall, pp. 119–204.
- Duchon, J. (1977), "Splines minimizing rotation-invariant semi-norms in sobolev spaces", in W. Schempp & K. Zeller, eds, "Constructive Theory of Functions of Several Variables", number 571 in "Lecture Notes in Mathematics", Springer-Verlag, Berlin, pp. 85–100.

- Easterling, M. R., Ellner, S. P. & Dixon, P. M. (2000), "Size-specific sensitivity: applying a new structured population model", *Ecology* **81**, 694–708.
- Edvardsen, A., Zhou, M., Tande, K. S. & Zhu, Y. (2002), "Zooplankton population dynamics: measuring *in situ* growth and mortality rates using an Optical Plankton Counter", *Marine Ecology Progress Series* **227**, 205–219.
- Efron, B. (1982), *The Jackknife, the Bootstrap, and Other Resampling Plans*, number 38 in "Conference Board of the Mathematical Sciences, Regional Conference Series in Applied Mathematics", Society for Industrial and Applied Mathematics - National Science Foundation.
- Efron, B. & Tibshirani, R. J. (1993), *An Introduction to the Bootstrap*, number 57 in "Monographs on Statistics and Applied Probability", Chapman & Hall.
- Eiane, K., Aksnes, D. L., Ohman, M. D., Wood, S. & Marinussen, M. B. (2002), "Stage-specific mortality of *Calanus spp.* under different predation regimes", *Limnology and Oceanography* **47**, 636–645.
- EUROSTAT (1995), "A review of the quality and reliability of fishery statistics", in "Report of the Workshop on the Significance of Reliable Statistics to Conduct Effective Management", Organisation for Economic Co-operation and Development, pp. 185–187.
- FAO (2002), "The state of world fisheries and aquaculture", Technical report, Fisheries Department, Food and Agriculture Organization of the United Nations.
- Fox, C. J., Bunn, N. A. & Webb, T. (2000), "A literature review of studies on fish egg mortality: Implications for the estimation of spawning stock biomass by the Annual Egg

- Production Method.”, Science Series Technical Report 111, The Centre for Environment, Fisheries and Aquaculture Science.
- Ganias, K., Somarakis, S., Machias, A. & Theodorou, A. J. (2003), “Evaluation of spawning frequency in a Mediterranean sardine population (*Sardina pilchardus sardina*)”, *Marine Biology* **142**(6), 1169–1179.
- Gill, P. E., Murray, W. & Wright, M. H. (1981), *Practical Optimization*, Academic Press.
- Green, P. G. & Silverman, B. W. (1994), *Nonparametric Regression and Generalised Linear Models; A Roughness Penalty Approach*, number 58 in “Monographs on Statistics and Applied Probability”, Chapman and Hall.
- Gu, C. & Wahba, G. (1991), “Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method”, *SIAM Journal on Scientific and Statistical Computing* **12**(2), 383–398.
- Gunderson, D. R. (1993), *Surveys of Fisheries Resources*, John Wiley & Sons, Inc.
- Gurney, W. S. C. & Nisbet, R. M. (1998), *Ecological Dynamics*, Oxford University Press.
- Hare, J. A. & Cowen, R. K. (1997), “Size, growth, development and survival of the planktonic larvae of *Pomatomus saltatrix* (Pisces: Pomatomidae)”, *Ecology* **78**, 2415–2431.
- Hastie, T. J. & Tibshirani, R. J. (1990), *Generalised Additive Models*, Chapman and Hall, London.
- Hirst, A. G. & Sheader, M. (1997), “Are *in situ* weight-specific growth rates body-size independent in marine planktonic copepods? a re-analysis of the global syntheses and a new empirical model”, *Marine Ecology Progress Series* **154**, 155–165.

BIBLIOGRAPHY

- Horwood, J. (1993), "The Bristol Channel Sole (*Solea solea*(L.)): A fisheries case study", *Advances in Marine Biology* **29**, 215–367.
- Horwood, J. W. & Greer Walker, M. (1990), "Determinacy of fecundity in sole (*Solea solea*) from the Bristol Channel", *Journal of the Marine Biological Association of the United Kingdom* **70**, 803–813.
- Hosmer, D. W. & Lemeshow, S. (1989), *Applied Logistic Regression*, Wiley, New York.
- Huntley, M. E., Zhou, M. & Lopez, M. D. G. (1994), "*Calanoides acutus* in Gerlache Strait, Antarctica II. solving an inverse problem in population dynamics", *Deep Sea Research Part II - Topical Studies in Oceanography* **41**(1), 209–227.
- Kreyszig, E. (1993), *Advanced Engineering Mathematics*, 7 edn, John Wiley & Sons.
- Lambert, D. (1992), "Zero-inflated Poisson regression, with an application to defects in manufacturing", *Technometrics* **34**, 1–14.
- Lo, N. C. H. (1985), "A model for temperature-dependent northern anchovy egg development and an automated procedure for the assignment of age to staged eggs", in R. Lasker, ed., "An Egg Production Method for Estimating Spawning Biomass of Pelagic Fish: Application to the Northern Anchovy, *Engraulis mordax*", number 36 in "US Department of Commerce, NOAA Technical Report", NMFS, pp. 43–50.
- Lotka, A. J. (1925), *Elements of Physical Biology*, Williams and Wilkins, Baltimore.
- Manly, B. F. J. (1990), *Stage-Structured Populations, Sampling, Analysis and Simulation*, number 7 in "Population and Community Biology", Chapman and Hall.

BIBLIOGRAPHY

- McKendrick, A. G. (1926), "Applications of mathematics to medical problems", *Proceedings of the Edinburgh Mathematical Society* **54**, 98–130.
- Meise, C., Collie, J. S., Widman, J. & Howell, P. (1999), "Growth and mortality of juvenile winter flounder in two New England estuaries", *Estuaries* **22**(2A), 297–303.
- Miranda, A., Cal, R. M. & Iglesias, J. (1990), "Effect of temperature on the development of eggs and larvae of sardine *Sardina pilchardus* (Walbaum) in captivity", *Journal of Experimental Marine Biology and Ecology* **140**(1–2), 69–77.
- Motos, L. (1994), Estomación de la biomasa desovante de la población de anchoa del golfo de Vizcaya, engraulis encrasicolus, a partir de su producción de huevos. Bases metodológicas y aplicación. Memoria presentada para defensa de la Tesis Doctoral, PhD thesis, Universidad del País Vasco.
- Ohman, M. D. & Hirche, H. J. (2001), "Density-dependent mortality in an oceanic copepod population", *Nature* **412**, 638–641.
- Ohman, M. D. & Wood, S. N. (1996), "Mortality estimation for planktonic copepods: *Pseudocalanus newmani* in a temperate fjord", *Limnology and Oceanography* **41**, 126–135.
- Pingree, R. D. & Griffiths, D. K. (1980), "Currents driven by a steady uniform wind stress on the shelf seas around the British Isles", *Oceanologica Acta* **3**, 227–236.
- Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1988), *Numerical Recipes in C*, Cambridge University Press, Cambridge, United Kingdom.
- Ré, P., Farinha, A. & Meneses, I. (1988), "Diel spawning time of sardine, *Sardina pilchardus* (Walbaum, 1792) (Teleostei, Clupeidae), off Portugal", *Investigación Pesquera* **52**(2), 207–213.

BIBLIOGRAPHY

- Riley, J. D. (1974), "The distribution and mortality of sole eggs (*Solea solea* (L.)) in inshore areas", in J. H. S. Blaxter, ed., "The Early Life History of Fish", Springer-Verlag, Berlin, pp. 39–52.
- Rothschild, B. J., Sharov, A. F., Kearsley, A. J. & Bondarenko, A. S. (1997), "Estimating growth and mortality in stage-structured populations", *Journal of Plankton Research* **19**, 1913–1928.
- Rubin, D. B. (1981), "The bayesian bootstrap", *The Annals of Statistics* **9**(1), 130–134.
- Sharpe, F. R. & Lotka, A. J. (1911), "A problem in age-distribution", *Philosophy Magazine* **21**, 435–438.
- Shepherd, J. G. (1993), "Why fisheries need to be managed. Why technical conservation measures on their own are not enough.", Laboratory Leaflet 71, Ministry of Agriculture, Fisheries and Food (MAFF), Directorate of Fisheries Research, Lowestoft.
- Silverman, B. W. (1985), "Some aspects of the spline smoothing approach to nonparametric regression curve fitting", *Journal of the Royal Statistical Society B* **47**, 1–52.
- Somerton, D. A. & Kobayashi, D. R. (1992), "Inverse method for mortality and growth estimation: A new method for larval fishes", *Fishery Bulletin* **90**, 368–375.
- Tuljapurkar, S. & Caswell, H. (1997), *Structured-Population Models in Marine, Terrestrial, and Freshwater Systems*, number 18 in "Population and Community Biology", Chapman and Hall.
- Villalobos, M. & Wahba, G. (1987), "Inequality-constrained multivariate smoothing splines with application to the estimation of posterior probabilities", *Journal of the American Statistical Association* **82**, 239–248.

BIBLIOGRAPHY

- von Foerster, H. (1959), "Some remarks on changing populations", in F. Stohlmán, ed., "The Kinetics of Cellular Proliferation.", Grune & Stratton, New York, pp. 382–407.
- Wahba, G. (1983), "Bayesian confidence intervals for the cross validated smoothing spline", *Journal of the Royal Statistical Society* **45**, 133–150.
- Wahba, G. (1990), *Spline Models for Observational Data*, Vol. 59 of *Reg. Conference Series Appl. Math.*, CBMS-NSF.
- Wang, Y. D. & Wahba, G. (1995), "Bootstrap confidence intervals for smoothing splines and their comparison to bayesian 'confidence intervals'", *Journal of Statistical Computation and Simulation* **51**(2–4), 263–297.
- Wood, S. N. (1994), "Obtaining birth and mortality patterns from structured population trajectories", *Ecological Monographs* **64**(1), 23–44.
- Wood, S. N. (1997), "Inverse problems and structured-population dynamics", in S. Tuljapurkar & H. Caswell, eds, "Structured-Population Models in Marine, Terrestrial, and Freshwater Systems", number 18 in "Population and Community Biology", Chapman and Hall, chapter 19, pp. 555–586.
- Wood, S. N. (2000), "Modelling and smoothing parameter estimation with multiple quadratic penalties", *Journal of the Royal Statistical Society* **62**(2), 412–428.
- Wood, S. N. & Horwood, J. W. (1995), "Spatial distribution functions and abundances inferred from sparse noisy plankton data: an application of constrained thin-plate splines", *Journal of Plankton Research* **17**(6), 1189–1208.
- Wood, S. N. & Nisbet, R. M. (1991), "Estimation of mortality rates in stage-structured populations", *Lecture Notes in Biomathematics*.

BIBLIOGRAPHY

- Wood, S. N. & Thomas, M. B. (1999), "Super-sensitivity to structure in biological models",
Proceedings of the Royal Society of London series B **266**, 565–570.

Appendix A

Notation

A.1 Notation for chapters 2 and 3

Symbol	Meaning
<i>Constants</i>	
C	Number of life-history stages.
d	Number of dimensions (explanatory variables) including time.
K	Number of knots used in each of the thin plate splines.
n	Number of sample points.
n^*	Number of inequality constraints.

continued on next page

continued from previous page

Symbol	Meaning
m	An integer satisfying $2m > d+1$ to ensure the splines are ‘visibly smooth’.
$M = \begin{pmatrix} m + d - 1 \\ d \end{pmatrix}$	Number of linearly independent polynomial terms in the thin plate splines.
$\tau = \frac{2}{1+\sqrt{5}}$	Reciprocal of the golden ratio.
β_0, μ_0	Initial birth and death rate constants. These must be positive.
dev_tol, conv_tol	Divergence and convergence tolerance levels used for deciding when to stop the model fitting process.
b	Number of bootstrap re-samples.
Data and predicted values	
$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_d \end{pmatrix}$	The explanatory variables, i.e. environmental conditions and time.
x_d	Time.
\mathbf{x}_i	i th sample point for which stage counts are available.
$\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$	All sample points for which stage counts are available.
\mathbb{X}	The data space.

continued on next page

continued from previous page

Symbol	Meaning
a	Age.
α_{j-1}	Age at entry to the j th stage.
α_C	Maximum age attainable which is the age at departure from the final (C th) stage.
$\mathbf{k}_i = (k_{1i}, \dots, k_{di})$	i th knot used in the thin plate splines ($i = 1, \dots, K$).
Y_{ji}	Random variate from an exponential family distribution, representing the j th stage count at the i th sample point.
y_{ji}	Observation of Y_{ji} .
$\mathbf{y} = (y_{11} \dots y_{1n} y_{21} \dots y_{Cn})$	
t_{ji}	i th observation of the j th ‘true’ stage count for the simulated data (i.e. one without any noise).
P_0	Total number of births over the survey region and period.
$f_j(\mathbf{x})$	Predicted j th stage abundance at \mathbf{x} .
f_{ji}	Predicted j th stage abundance at i th sample point.
$\mathbf{f} = (f_{11} \dots f_{1n} f_{21} \dots f_{Cn})$	

continued on next page

continued from previous page

Symbol	Meaning
$\hat{f}^\rho(\mathbf{x})$	Predicted stage population at \mathbf{x} for a model fitted using all data points with overall smoothing parameter ρ .
$\epsilon_{ji} = y_{ji} - f_{ji}$	Error - the difference between the stage abundance predicted by the model and the observed stage abundance.
Functions and conventions	
$\beta(\mathbf{x})$	Birth rate at (\mathbf{x}) .
$\mu(\mathbf{x})$	Death rate at (\mathbf{x}) .
$\xi(\ \mathbf{x} - \mathbf{k}_i\)$	Radial basis function corresponding to the i th knot, forming part of the thin plate splines.
$\{\phi_1, \dots, \phi_M\}$	Set of linearly independent polynomials spanning the M -dimensional space of polynomials in \mathbb{R}^d of total degree less than m forming part of the thin plate splines.
$\eta(\mathbf{x}, a)$	Population density for individuals aged a at \mathbf{x} .
$F(\mathbf{p}, \lambda_\beta, \lambda_\mu)$	Objective function for model parameters \mathbf{p} and smoothing parameters $\lambda_\beta, \lambda_\mu$.
$F^*(\mathbf{p}, \lambda_\beta, \lambda_\mu)$	Quadratic approximation of the objective function for model parameters \mathbf{p} and smoothing parameters $\lambda_\beta, \lambda_\mu$.

continued on next page

continued from previous page

Symbol	Meaning
$F(\mathbf{p}_z, \lambda_\beta, \lambda_\mu)$	Objective function and its quadratic approximation for constrained model parameters \mathbf{p}_z and smoothing parameters $\lambda_\beta, \lambda_\mu$.
$F^*(\mathbf{p}_z, \lambda_\beta, \lambda_\mu)$	
$P_d(\beta), P_d(\mu)$	Wiggleness penalties corresponding to the birth and death rates for d dimensions.
$V_o(\rho)$	Ordinary cross validation (OCV) score for overall smoothing parameter ρ .
$V(\rho)$	Generalized cross validation (GCV) score for overall smoothing parameter ρ .
$\ \cdot\ $	Euclidean norm of \cdot .
$tr(\cdot)$	Trace of the matrix \cdot .
$\hat{\cdot}$	Estimate of \cdot .
$\cdot^{[k]}$	k th iterate estimate of \cdot .
\cdot^T	Transpose of the matrix \cdot .
Parameters	
θ	Constant of proportionality in the thin plate splines.
γ_i, δ_i	Model parameters relating to the radial basis functions of the i th knot in the birth and death rates.

continued on next page

continued from previous page

Symbol	Meaning
b_j, d_j	Model parameters relating to the j th polynomial terms of the birth and death rates.
$\gamma_{z_i}, \delta_{z_i}$	Constrained model parameters relating to the radial basis functions of the i th knot in the birth and death rates ($i = 1, \dots, K - 10$).
$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_1 & \dots & \gamma_K \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} b_1 & \dots & b_M \end{pmatrix}, \boldsymbol{\delta} = \begin{pmatrix} \delta_1 & \dots & \delta_K \end{pmatrix}, \boldsymbol{d} = \begin{pmatrix} d_1 & \dots & d_M \end{pmatrix}$ $\boldsymbol{\gamma}_z = \begin{pmatrix} \gamma_{z_1} & \dots & \gamma_{z_K} \end{pmatrix}, \boldsymbol{\delta}_z = \begin{pmatrix} \delta_{z_1} & \dots & \delta_{z_K} \end{pmatrix}$ $\boldsymbol{p} = \begin{pmatrix} \boldsymbol{\gamma} & \boldsymbol{b} & \boldsymbol{\delta} & \boldsymbol{d} \end{pmatrix}^T$ $\boldsymbol{p}_z = \begin{pmatrix} \boldsymbol{\gamma}_z & \boldsymbol{b} & \boldsymbol{\delta}_z & \boldsymbol{d} \end{pmatrix}^T$	Model parameters.
	Constrained model parameters.
$\lambda_\beta, \lambda_\mu$	Smoothing parameters corresponding to the birth and death rates.
ρ	Overall smoothing parameter.
$\lambda_\beta^* = \frac{\lambda_\beta}{\rho}, \lambda_\mu^* = \frac{\lambda_\mu}{\rho}$	Relative smoothing parameters corresponding to the birth and death rates.

continued on next page

continued from previous page

Symbol	Meaning
Matrices	
$W = \text{diag}(w_{ji}),$ $w_{ji} = \frac{1}{\widehat{\text{Var}}(Y_{ji})}$	Matrix of weights. w_{ji} is the weight for the j th stage population at the i th sample point.
E	Radial basis function matrix involved in the matrix forms of the wiggleness penalties, the ij th element of which is $\theta\xi(\ \mathbf{k}_i - \mathbf{k}_j\)$.
S_β, S_μ	Matrix forms of the wiggleness penalties for the birth and death rates.
$S_{\beta_z} = Z^T S_\beta Z,$	Matrix forms of the constrained wiggleness penalties for the birth and death rates.
$S_{\mu_z} = Z^T S_\mu Z$	
$\mathbf{e} = \left(\theta\xi(\ \mathbf{x} - \mathbf{k}_1\) \quad \dots \quad \theta\xi(\ \mathbf{x} - \mathbf{k}_K\) \right)^T, \mathbf{T} = \left(\phi_1(\mathbf{x}) \quad \dots \quad \phi_M(\mathbf{x}) \right)^T$ $\begin{pmatrix} \mathbf{e} & \mathbf{T} \end{pmatrix}$	Design matrix for the thin plate splines
$A(\rho)$	Influence matrix for the overall smoothing parameter ρ , such that $\hat{f}^\rho(\mathbf{x}) = A(\rho)\mathbf{y}$.
J, J_z	Jacobian matrix and constrained Jacobian matrix.
$\mathbf{z} = \mathbf{f} - \mathbf{y} + J\mathbf{p}$	Pseudo-data.

continued on next page

continued from previous page

Symbol	Meaning
T_{MK}	Equality constraints matrix.
Z	$(2K 2M) \times 2K$ matrix used for ensuring the equality constraints are satisfied.
$0_{A,B}$	A zero matrix with A rows and B columns.
I_A	The identity matrix with A rows and columns.
A_n, A_{n^*}	Design matrices of the thin plate splines for n and n^* sample points \mathbf{x}_i , $i = 1, \dots, n$ or n^* . The i th row is the same as $\begin{pmatrix} \mathbf{e} & \mathbf{T} \end{pmatrix}$ evaluated at the i th data point.
$D = \begin{pmatrix} D_{11} & \dots & D_{Cn} \end{pmatrix}^T$	Vector of differences between consecutive iterates of the predicted stage populations.

Table A.1: Notation used in chapter 2.

A.2 Notation for chapter 4

Symbol	Meaning
Constants	
C	Number of life-history classes/stages.
n	Number of sample times.
c, c^*	Unknown constants of integration.
Data and predicted values	
t	Time.
t_i	i th sample time.
t_0	Start time for population under consideration.
$\tau_j(t)$	Stage duration for stage j at time t .
a	Age.
$m, (m(t))$	Physiological trait or size, which can be a function of time.
m_{min}	Minimum physiological trait or size. The size at which individuals are born.
m_{max}	Maximum physiological trait or size attainable by an individual. This can be an asymptotic size that no individual will reach in practice.

continued on next page

continued from previous page

Symbol	Meaning
m_{jL}	Physiological trait or size at which an individual recruits to stage j .
m_{jU}	Physiological trait or size at which an individual matures from stage j .
$y_j(t_i)$	Observed population abundance for stage j at sample time t_i .
$N_j(t)$	Stage abundance — the number of individuals in stage j at time t .
$\epsilon_{ji} = y_j(t_i) - N_j(t_i)$	Random error for stage j at the i th sample time.
Functions and conventions	
$\eta_j(m, t)$	Population density function for stage j at time t and physiological trait m .
$\eta(m, t)$	(Smooth) population density function at time t and physiological trait m .
$\eta(a, t)$	(Smooth) age-dependent population density function at time t and age a .
$\mu_j(m, t)$	Death rate function for stage j at time t and physiological trait m .

continued on next page

continued from previous page

Symbol	Meaning
$\mu(m, t)$	(Smooth) death rate function at time t for physiological trait m .
$\mu(a, t)$	(Smooth) age-dependent death rate function at time t for physiological trait a .
$R_j(t)$	Recruitment function for stage j at time t . $R_1(t)$ is the birth rate.
$R(t)$	(Smooth) recruitment function at time t . This is the birth rate.
$g_j(m, t) = \frac{dm}{dt}$	Growth rate function (development rate function) for stage j — the rate of change with respect to time of the physiological trait of an individual in stage j of physiological trait m at time t .
τ	'True'.
$g(m, t) = \frac{dm}{dt}$	(Smooth) growth rate function at time t for physiological trait m .
g_L	Lower limit on the growth rate.
g_U	Upper limit on the growth rate.
g^*, g_a, g_b, g_c	Alternative growth rates.
$\mu^*, \mu_a, \mu_b, \mu_c$	Corresponding alternative death rates.
R^*, R_a, R_b, R_c	Corresponding alternative birth rates.

continued on next page

continued from previous page

Symbol	Meaning
$f_i(t), i = 1, \dots, 6$	Functions of time used in the calculation of $\tilde{R}(t)$, $\tilde{g}(m, t)$, and $g_c(m, t)$ in the simulation demonstration.

Table A.2: Notation used in chapter 4.

A.3 Notation for chapter 5

Symbol	Meaning
Constants	
C	Number of life-history stages.
e_n	Number of sample points of the stage-age experiment.
n	Number of sample points of the physiologically-structured population survey.
Data and predicted values	
s	Developmental stage.
k	Cohort.

continued on next page

continued from previous page

Symbol	Meaning
\mathbf{x}	The environmental conditions.
t	Time.
\mathbf{x}_i	Environmental conditions at the i th sample point.
a	Age.
a_i	Age at the i th sample point.
$\alpha_k(t)$	Upper age limit for cohort k , given sampling time t .
\bar{a}_{ki}	Mean age of cohort k at the i th sample point.
$\bar{\mathbf{a}} = (\dots \bar{a}_{k1} \dots \bar{a}_{kn} \dots)^T$	
n_s	Number of individuals in stage s .
n_{si}	Number of individuals in stage s at \mathbf{x}_i .
N_s	Number of individuals in stage s or above.
Y_{ki}	Random variate from an exponential family distribution, representing the k th cohort abundance at the i th sample point.
y_{ki}	'Observation' of Y_{ki} .
R_i	i th effective area sampled.

continued on next page

continued from previous page

Symbol	Meaning
Functions and conventions	
$p(s a, \mathbf{x}) = P[s]$	The probability mass function of stage given age for environmental conditions \mathbf{x} .
$p(a s, \mathbf{x}, t)$	The probability density function of age given stage, for the environmental conditions \mathbf{x} and time t .
$P[s^+ (s-1)^+]$	The probability of an individual being in class s or above, given that it is in class $s-1$ or above.
$P[s^+]$	The probability of an individual being in stage s or above.
$p(a t)$	The probability density function of age at time t .
$p(a s, \mathbf{x}, t)$	The probability density function of age given stage at \mathbf{x} and t .
$\eta(a, \mathbf{x})$	The linear predictor defining the dependence of $P[s^+ (s-1)^+]$ on age, environmental conditions, and perhaps stage.
$\beta(t)$	The micro-birth rate, i.e. the number of births per micro unit of time.
$B(\mathbf{x})$	The macro-birth rate at \mathbf{x} , i.e. the number of births per macro unit of time.

continued on next page

continued from previous page

Symbol	Meaning
$B(\mathbf{x}, t)$	Time dependent macro-birth rate at \mathbf{x} , i.e. the number of births per macro unit of time.
$\mu(\mathbf{x})$	The death rate at \mathbf{x} .
$\mu(\mathbf{x}, t)$	Time dependent death rate at \mathbf{x} .
f_{ki}	Fitted cohort abundance $E(Y_{ki})$.
P_0	Total daily egg production.
$h(\cdot)$	Probit or logit link function of \cdot .
$E(\cdot)$	The expected value of \cdot .
\cdot^T	The transpose of matrix \cdot .
Matrices	
\mathbf{X}_B	Columns of the design matrix of the cohort abundance GAM corresponding to the smooth functions making up the logarithm of the macro-birth rate.
\mathbf{X}_μ	Columns of the design matrix of the cohort abundance GAM corresponding to the smooth functions of the death rate.
$\mathbf{X} = [\mathbf{X}_B, -\bar{a}\mathbf{X}_\mu]$	Design matrix of the cohort abundance GAM.

Table A.3: Notation used in chapter 5.

Appendix B

Fitting the SPSP Model for Three Covariates

In chapter 3 the SPSP model was applied to data with three covariates ($d = 3$), having been defined for any d dimensions in chapter 2. This appendix presents the model components and the matrices involved in the objective function when $d = 3$. These items were used when encoding the fitting process in `fitmodel.c` (appendix C).

B.1 Model components for $d = 3$

When $d = 3$, $m = 3$ to satisfy $2m > d + 1$, and $M = \binom{m+d-1}{d} = \binom{5}{3} = 10$. In the birth and death rate thin plate splines the constant of proportionality, θ , is

$$\theta = \frac{\Gamma\left(\frac{d}{2} - m\right) 2^{-2m}}{\pi^{\frac{d}{2}}(m-1)!} = \frac{\Gamma\left(\frac{-3}{2}\right) 2^{-6}}{\pi^{\frac{3}{2}}(2)!} = \frac{1}{96\pi},$$

the radial basis function for the i th knot $\xi(\|\mathbf{x} - \mathbf{k}_i\|)$ is

$$\xi(\|\mathbf{x} - \mathbf{k}_i\|) = \|\mathbf{x} - \mathbf{k}_i\|^3,$$

and the linearly independent polynomial terms $\{\phi_1 \dots \phi_{10}\}$ spanning the $M = 10$ -dimensional space of polynomials in \mathbb{R}^d are $\left\{ 1 \ x_1 \ x_2 \ x_3 \ x_1^2 \ x_2^2 \ x_3^2 \ x_1x_2 \ x_1x_3 \ x_2x_3 \right\}$, where x_1, x_2 are two environmental variables and x_3 is time. Using this information, the birth rate can be written

$$\begin{aligned} \beta(\mathbf{x}) = & \frac{1}{96\pi} \sum_{i=1}^K \gamma_i \|\mathbf{x} - \mathbf{k}_i\|^3 \\ & + b_1 + b_2x_1 + b_3x_2 + b_4x_3 + b_5x_1^2 + b_6x_2^2 + b_7x_3^2 + b_8x_1x_2 + b_9x_1x_3 + b_{10}x_2x_3. \end{aligned}$$

Similarly the death rate is

$$\begin{aligned} \mu(\mathbf{x}) = & \frac{1}{96\pi} \sum_{i=1}^K \delta_i \|\mathbf{x} - \mathbf{k}_i\|^3 \\ & + d_1 + d_2x_1 + d_3x_2 + d_4x_3 + d_5x_1^2 + d_6x_2^2 + d_7x_3^2 + d_8x_1x_2 + d_9x_1x_3 + d_{10}x_2x_3. \end{aligned}$$

The integral of the death rate from time $x_3 - a$ to x_3 is required for calculating the population density $\eta(\mathbf{x}, a)$. The integral is given by

$$\begin{aligned}
 & \int_{x_3-a}^{x_3} \mu(x_1, x_2, z) \partial z = \\
 & \frac{1}{96\pi} \sum_{i=1}^K \delta_i \left\{ \frac{1}{4} \|\mathbf{x} - \mathbf{k}_i\|^3 (x_3 - k_{3i}) - \frac{1}{4} \|\mathbf{x}^* - \mathbf{k}_i\|^3 (x_3 - a - k_{3i}) \right\} + \\
 & \frac{1}{96\pi} \sum_{i=1}^K \delta_i \left\{ \frac{3}{8} \left((x_1 - k_{1i})^2 + (x_2 - k_{2i})^2 \right) \left(\|\mathbf{x} - \mathbf{k}_i\| (x_3 - k_{3i}) - \|\mathbf{x}^* - \mathbf{k}_i\| (x_3 - a - k_{3i}) \right) \right\} \\
 & + \frac{1}{96\pi} \sum_{i=1}^K \delta_i \left\{ \frac{3}{8} \left((x_1 - k_{1i})^2 + (x_2 - k_{2i})^2 \right)^2 \left(\ln(x_3 - k_{3i} + \|\mathbf{x} - \mathbf{k}_i\|) \right) \right\} \\
 & - \frac{1}{96\pi} \sum_{i=1}^K \delta_i \left\{ \frac{3}{8} \left((x_1 - k_{1i})^2 + (x_2 - k_{2i})^2 \right)^2 \left(\ln(x_3 - a - k_{3i} + \|\mathbf{x}^* - \mathbf{k}_i\|) \right) \right\} \\
 & + d_1 a + d_2 x_1 a + d_3 x_2 a + d_5 x_1^2 a + d_6 x_2^2 a + d_8 x_1 x_2 a \\
 & + (d_4 + d_9 x_1 + d_{10} x_2) \left(x_3 a - \frac{a^2}{2} \right) + d_7 \left(x_3^2 a - x_3 a^2 + \frac{a^3}{3} \right)
 \end{aligned}$$

where $\mathbf{x}^* = (x_1, x_2, x_3 - a)$. Finally, the predicted j th stage counts $f_j(\mathbf{x})$, are calculated by integrating the population density function over the age range for the j th stage. The integration was performed numerically using Simpson's rule:

$$f_j(\mathbf{x}) = \int_{\alpha_{j-1}(\mathbf{x})}^{\alpha_j(\mathbf{x})} \eta(\mathbf{x}, a) \partial a = \frac{h}{3} (g_0 + 4g_1 + 2g_2)$$

$$\text{for } g_0 = \eta(\mathbf{x}, a_0) + \eta(\mathbf{x}, a_{2K})$$

$$g_1 = \eta(\mathbf{x}, a_1) + \eta(\mathbf{x}, a_3) + \dots + \eta(\mathbf{x}, a_{2K-1})$$

$$g_2 = \eta(\mathbf{x}, a_2) + \eta(\mathbf{x}, a_4) + \dots + \eta(\mathbf{x}, a_{2K-2})$$

where $h = \frac{\alpha_j - \alpha_{j-1}}{2K}$ and $a_0 = \alpha_{j-1}$, $a_1 = a_0 + h$, \dots , $a_{2K-1} = a_0 + (2K-1)h$, $a_{2K} = a_0 + 2Kh = \alpha_j$, and K is the number of integration intervals.

B.2 Jacobian matrix for $d = 3$

The Jacobian matrix J has nC rows (the number of data points), and $2K + 2M$ columns (the number of model parameters) with the element on the $((j - 1)n + i)$ th row and l th column given by the derivative of the j th stage count function predicted at the i th sample point with respect to the l th model parameter,

$$J_{((j-1)n+i),l} = \frac{\partial f_{ji}^{[k]}}{\partial p_l^{[k]}}.$$

Because Simpson's rule was used to integrate $\eta(\mathbf{x}, a)$ to obtain $f_j(\mathbf{x})$, f_{ji} is a linear combination of $\eta(\mathbf{x}_i, a)$ for a set of $2K$ ages encompassing stage j . This means that if the derivatives of $\eta(\mathbf{x}_i, a)$ with respect to the model parameters can be found, so too can the derivatives of f_{ji} :

$$\frac{\partial f_{ji}}{\partial p_l} = \frac{h}{3} (g'_0 + 4g'_1 + 2g'_2)$$

$$\text{for } g'_0 = \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_0)} + \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_{2K})}$$

$$g'_1 = \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_1)} + \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_3)} + \dots + \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_{2K-1})}$$

$$g'_2 = \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_2)} + \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_4)} + \dots + \frac{\partial \eta}{\partial p_l} \Big|_{(\mathbf{x}_i, a_{2K-2})}$$

where $h = \frac{\alpha_j - \alpha_{j-1}}{2K}$ and $a_0 = \alpha_{j-1}$, $a_1 = a_0 + h$, \dots , $a_{2K-1} = a_0 + (2K - 1)h$, $a_{2K} = a_0 + 2Kh = \alpha_j$, and K is the number of integration intervals.

From equation 2.1,

$$\eta(\mathbf{x}_i, a) = \beta(\mathbf{x}_i^*) \exp^{-D(\mathbf{x}_i, a)},$$

where for clarity, the integral of the death rate from $x_{3i} - a$ to x_{3i} is denoted $D(\mathbf{x}_i, a)$, and

$\mathbf{x}_i^* = (x_{1i}, x_{2i}, x_{3i} - a)$. Then

$$\begin{aligned} \frac{\partial \eta}{\partial p_l} &= \exp(-D(\mathbf{x}_i, a)) \frac{\partial \beta}{\partial p_l} \Big|_{\mathbf{x}_i^*} + \beta(\mathbf{x}_i^*) \frac{\partial(\exp(-D))}{\partial p_l} \\ &= \left(\frac{\partial \beta}{\partial p_l} \Big|_{\mathbf{x}_i^*} - \beta(\mathbf{x}_i^*) \frac{\partial D}{\partial p_l} \Big|_{(\mathbf{x}_i, a)} \right) \exp(-D(\mathbf{x}_i, a)) \end{aligned}$$

for which the derivatives of β and D have yet to be defined. These derivatives have different forms for the model parameters of the birth and death rates pertaining to the radial basis functions ($\gamma_l, \delta_l, l = 1, \dots, K$) and those pertaining to the polynomial terms ($b_j, d_j, j = 1, \dots, M = 10$).

The partial derivatives of $\beta(\mathbf{x}_i^*)$ with respect to the model parameters are

$$\begin{aligned} \frac{\partial \beta}{\partial \gamma_l} \Big|_{\mathbf{x}_i^*} &= \frac{1}{96\pi} \|\mathbf{x}_i^* - \mathbf{k}_l\|^3 & \frac{\partial \beta}{\partial b_j} \Big|_{\mathbf{x}_i^*} &= \phi_j(\mathbf{x}_i^*) \\ \frac{\partial \beta}{\partial \delta_l} \Big|_{\mathbf{x}_i^*} &= 0 & \frac{\partial \beta}{\partial d_j} \Big|_{\mathbf{x}_i^*} &= 0 \end{aligned}$$

for $l = 1, \dots, K$ and $j = 1, \dots, 10$.

The partial derivatives of $D(\mathbf{x}_i, a)$ with respect to the model parameters are

$$\begin{aligned} \frac{\partial D}{\partial \gamma_l} \Big|_{(\mathbf{x}_i, a)} &= 0 & \frac{\partial D}{\partial b_j} \Big|_{(\mathbf{x}_i, a)} &= 0 \\ \frac{\partial D}{\partial \delta_l} \Big|_{(\mathbf{x}_i, a)} &= \frac{1}{96\pi} \left\{ \frac{1}{4} (\|\mathbf{x}_l - \mathbf{k}_l\|^3 (x_{3i} - k_{3l}) - \|\mathbf{x}_i^* - \mathbf{k}_l\|^3 (x_{3i} - a - k_{3l})) \right\} \\ &+ \frac{1}{96\pi} \left\{ \frac{3}{8} ((x_{1i} - k_{1l})^2 + (x_{2i} - k_{2l})^2) (\|\mathbf{x}_i - \mathbf{k}_l\| (x_{3i} - k_{3l})) \right\} \\ &+ \frac{1}{96\pi} \left\{ \frac{3}{8} ((x_{1i} - k_{1l})^2 + (x_{2i} - k_{2l})^2) (-\|\mathbf{x}_i^* - \mathbf{k}_l\| (x_{3i} - a - k_{3l})) \right\} \\ &+ \frac{1}{96\pi} \left\{ \frac{3}{8} ((x_{1i} - k_{1l})^2 + (x_{2i} - k_{2l})^2)^2 (\ln(x_{3i} - k_{3l} + \|\mathbf{x}_i - \mathbf{k}_l\|)) \right\} \\ &- \frac{1}{96\pi} \left\{ \frac{3}{8} ((x_{1i} - k_{1l})^2 + (x_{2i} - k_{2l})^2)^2 (\ln(x_{3i} - a - k_{3l} + \|\mathbf{x}_i^* - \mathbf{k}_l\|)) \right\} \end{aligned}$$

$$\begin{aligned} \frac{\partial D}{\partial d_1} \Big|_{(\mathbf{x}_i, a)} &= a & \frac{\partial D}{\partial d_2} \Big|_{(\mathbf{x}_i, a)} &= ax_{1i} & \frac{\partial D}{\partial d_3} \Big|_{(\mathbf{x}_i, a)} &= ax_{2i} \\ \frac{\partial D}{\partial d_4} \Big|_{(\mathbf{x}_i, a)} &= ax_{3i} - \frac{a^2}{2} & \frac{\partial D}{\partial d_5} \Big|_{(\mathbf{x}_i, a)} &= ax_{1i}^2 & \frac{\partial D}{\partial d_6} \Big|_{(\mathbf{x}_i, a)} &= ax_{2i}^2 \\ \frac{\partial D}{\partial d_7} \Big|_{(\mathbf{x}_i, a)} &= ax_{3i}^2 - a^2 x_{3i} + \frac{a^3}{3} & \frac{\partial D}{\partial d_8} \Big|_{(\mathbf{x}_i, a)} &= ax_{1i} x_{2i} \\ \frac{\partial D}{\partial d_9} \Big|_{(\mathbf{x}_i, a)} &= ax_{1i} x_{3i} - \frac{a^2}{2} x_{1i} & \frac{\partial D}{\partial d_{10}} \Big|_{(\mathbf{x}_i, a)} &= ax_{2i} x_{3i} - \frac{a^2}{2} x_{2i} \end{aligned}$$

for $l = 1, \dots, K$.

Using the above results, the derivatives of η can be written in a more transparent form,

$$\begin{aligned} \frac{\partial \eta}{\partial \gamma_l} &= \exp^{-D(\mathbf{x}_i, a)} \frac{\partial \beta}{\partial \gamma_l} \Big|_{\mathbf{x}_i^*} & \frac{\partial \eta}{\partial b_j} &= \exp^{-D(\mathbf{x}_i, a)} \frac{\partial \beta}{\partial b_j} \Big|_{\mathbf{x}_i^*} \\ \frac{\partial \eta}{\partial \delta_l} &= -\eta \frac{\partial D}{\partial \delta_l} \Big|_{(\mathbf{x}_i, a)} & \frac{\partial \eta}{\partial d_j} &= -\eta \frac{\partial D}{\partial d_j} \Big|_{(\mathbf{x}_i, a)} \end{aligned}$$

for $i = 1, \dots, n$ and $j = 1, \dots, 10$.

B.3 Wiggleness penalty matrix for $d = 3$

The non-zero component of the wiggleness penalty matrices S_β and S_μ for three dimensional thin plate splines is simply

$$E = \begin{pmatrix} 0 & \frac{1}{96\pi} \|\mathbf{k}_1 - \mathbf{k}_2\|^3 & \dots & \frac{1}{96\pi} \|\mathbf{k}_1 - \mathbf{k}_K\|^3 \\ \frac{1}{96\pi} \|\mathbf{k}_2 - \mathbf{k}_1\|^3 & 0 & \dots & \frac{1}{96\pi} \|\mathbf{k}_2 - \mathbf{k}_K\|^3 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{96\pi} \|\mathbf{k}_K - \mathbf{k}_1\|^3 & \frac{1}{96\pi} \|\mathbf{k}_K - \mathbf{k}_2\|^3 & \dots & 0 \end{pmatrix}$$

To obtain S_{β_z} and S_{μ_z} , the more convenient forms of the penalties, see section 2.3.1.

Appendix C

Programs used to Fit an SPSP Model to Stage Count Data

C.1 `fitmodel.c`

`fitmodel.c` is a C program that can be used to fit the SPSP model described in chapter 2 when there are two environmental variables ($d = 3$). The program was used in combination with the programs `matrix.c`, `gcv.c` and `qp.c` to produce the results given in chapter 3. The latter three libraries are available as part of the R package `mgcv` (www.cran.r-project.org, Wood 2001). The code for the program `fitmodel.c`, written by Simon Wood and myself, follows.

C.1 fitmodel.c

```
/* Camilla Dixon (with some contributions from Simon Wood)
   August 2003 */
/*
INPUT
=====
data    z1, z2, time, population counts
ages    age limits for each stage at each sample point
knots   z1, z2, time at knots
minmax  number of sample points, number of knots, number of age stages,
        min and max of z1, of z2, of time, number of unique sample times
constraintsgrid  z1, z2, time for inequality constraints (birth and
        death must be bigger than 0 at these locations). A finer grid
        than the sample grid is recommended
sumbXY  z1 and z2 for calculating total egg production estimate after
        final iterate of model fitting.

PROGRAMS
=====
fitmodel.c fits SPSP model using above list of inputs and the
        following C-programs...
matrix.c  matrix algebra routines from mgcv package
matrix.h  header file for matrix.c
gcv.c     GCV/multiple smoothing parameter selection routines
        from mgcv package
gcv.h     header file for gcv.c
qp.c     quadratic programming routines from mgcv package
qp.h     header file for qp.c
general.h header file needed for mgcv package routines
makeoutput.c  makes a file called output which fitmodel will add to

OUTPUT
=====
output  fitmodel adds a line to this file consisting of the
        final R-squared, objective function value, number of
        iterates done, egg production estimate, and total egg
        populations for stage 1 time 1, stage 2 time 1, ...,
        stage 4 time 5 (24 columns).
yparami for each iterate i, fitmodel produces yparami, which
        contains the fitted counts for each sample point and
        stage, the model parameters and the smoothing parameters
        (3 columns). NB by using comments "/*" to switch the
        choice of call to getJg() inside fitmodel.c, you can
        select to only produce yparam0 after the model has
        converged (useful for bootstrapping).

USE
===
First compile and run makeoutput.c (only once):
gcc -o makeoutput makeoutput.c
./makeoutput

To compile and run fitmodel.c:
gcc -o fitmodel fitmodel.c matrix.c gcv.c qp.c -lm -O3
./fitmodel
```

```
*/
/*
*****
*                               PRE-AMBLE                               *
*****
*/
#include <stdio.h>           // basic functions available
#include <math.h>            // maths functions available
#include <stdlib.h>          // enables use of "calloc" for arrays
#include "matrix.h"         // matrix algebra available
#include "gcv.h"            // smoothing parameter selection available
#include "qp.h"             // quadratic programming available
#define ANSI
// Define global constants (need changed for different inputs)...
#define NPENALTY 2          // number of penalty terms
#define NCONSTRAINTS 6229 // number of inequality constraints
#define MAXCOUNT 200      // max number of iterations
#define MAXSIMP 20         // number of terms for simpsons rule
                          // numerical integration
#define GRIDSIZE 161.0     // number of grid steps for total birth
#define SIZEXY 18701       // length of XY vector for total birth
#define DISTN 2            // distribution assumed: 1=normal
                          //                               2=poisson
#define CONVERGE 1e-3      // convergence tolerance
#define MAXDIFF 1e10       // divergence tolerance
#define MAXCHECK 1e-3      // used for checking derivatives etc
#define PI 3.1415927
/*
*****
*                               LIST OF FUNCTIONS                               *
*****
*/
void ErrorMessage(char *str,int fatal);
void myprintmat(matrix A, char *filename);
double meanmat(matrix X);
void ndata(int *n, int *k, int *s);
void stageage(matrix ages);
void getdata(matrix cov, matrix y, matrix knots);
double birth(matrix param, matrix s, matrix knots);
double death(matrix param, matrix s, matrix knots);
double intdfunction(matrix s, matrix knots, double x, matrix param);
double intdeath(matrix s, double birthday, matrix param, matrix knots);
void intpopn(matrix s, matrix knots, matrix param, matrix age,
             matrix beta, matrix f1, double *integral);
void deriv2(matrix s, matrix knots, double age, double *deriv);
void derivg2(matrix s, matrix knots, matrix param, double *deriv,
             matrix age, matrix beta, matrix f1);
void getJg(matrix param, double *smoothp, matrix cov, matrix knots,
           matrix g, matrix J, matrix w, int choice, int count);
void fillemat(matrix knots, matrix E);
void adjustJ(matrix J, matrix Jz, matrix Z);
void adjustp(matrix param, matrix paramz, matrix Z);
void constraints(matrix Z, matrix cov, matrix knots, matrix Ain, matrix b);
```

```
void checkcons(matrix param, matrix T);
void inequalcons(matrix paramz, double *smoothp, matrix Ain, matrix b,
  matrix Jz, matrix w, matrix z, matrix Pa0);
double object(matrix pz, matrix Z, matrix Pa0, double *smoothp);
void goldensmooth(matrix oldpz, matrix piz, matrix Z, matrix Pa0, double
  *smoothp, double *ob);
int criteria(matrix oldg, matrix g);
double rsq(matrix y, matrix g, int K);
void sumbirth(matrix param, matrix knots, double dimsize, matrix ans);
void fit(void);
/*
  *****
  *                THE ACTUAL FUNCTIONS!                *
  *****
*/
void ErrorMessage(char *str,int fatal)
/* {ErrorMessage} is necessary when matrix.c is included
*/
{
  printf("%s",str); if(fatal) {getc(stdin); exit(-1);}
}

void myprintmat(matrix A, char *filename)
/* {myprintmat} puts the matrix A into file filename
*/
{
  FILE *fp;
  int i, j;
  if((fp=fopen(filename, "w"))==NULL){
    printf("Error opening printmat file \n"); getc(stdin);
    exit(1);
  }
  for(i=0; i<A.r; i++){
    for(j=0; j<A.c; j++){
      fprintf(fp,"%lg ",A.M[i][j]);
    }
    fprintf(fp,"\n");
  }
  fclose(fp);
  return;
}

double meanmat(matrix X)
/* {meanmat} finds the "mean" value of the matrix (X) by summing all
  elements and dividing by the number of elements.
*/
{
  int i, j;
  double total=0.0, mean;
  for(i=0; i<X.r; i++){
    for(j=0; j<X.c; j++) total += X.M[i][j];
  }
  mean = total/(X.r*X.c);
}
```

```
    return(mean);
}

void ndata(int *n, int *k, int *s)
/* {ndata} reads number of data points (n), number of knots (k),
   number of stages (s) from the file "minmax"
*/
{
    FILE *fp;
    int i, j, l;
    if((fp=fopen("minmax", "r"))==NULL){
        printf("Error opening file 'minmax' \n"); getc(stdin);
        exit(1);
    }
    fscanf(fp,"%d%d%d",&i,&j,&l);
    *n = i;
    *k = j;
    *s = l;
    fclose(fp);
    return;
}

void stageage(matrix ages)
/* {stageage} fills the matrix (ages) with stage age limits from
   the file "ages"
*/
{
    FILE *fp;
    int i, j;
    double a;
    if((fp=fopen("ages", "r"))==NULL){
        printf("Error opening file \"ages\" \n");
        getc(stdin);
        exit(1);
    }
    for(i=0; i<ages.r; i++) {
        for(j=0; j<ages.c; j++){
            fscanf(fp,"%lg",&a);
            ages.M[i][j] = a;
        }
    }
    fclose(fp);
    return;
}

void getdata(matrix cov, matrix y, matrix knots)
/* {getdata} obtains covariates (sample points (cov)) and stage
   abundances (y) from the file "data", and the knots (knots) from
   the file "knots".
*/
{
    FILE *fp;
    int i, j, n, ns, K;
```

```
double z1, z2, t, y1;
K = knots.r;
n = cov.r;
ns = y.r/n;
if((fp=fopen("data", "r"))==NULL){
    printf("Error opening file \"data\" \n");
    getc(stdin);
    exit(1);
}
for(j=0; j<ns; j++){
    for(i=0; i<n; i++){
        fscanf(fp, "%lg%lg%lg%lg", &z1, &z2, &t, &y1);
        if(j==0){ // read covariates once only
            cov.M[i][0]=z1;
            cov.M[i][1]=z2;
            cov.M[i][2]=t;
        }
        y.V[j*n+i] = y1;
    }
}
fclose(fp);
if((fp=fopen("knots", "r"))==NULL){
    printf("Error opening file \"knots\" \n");
    getc(stdin);
    exit(1);
}
for(i=0; i<K; i++){
    fscanf(fp, "%lg%lg%lg", &z1, &z2, &t);
    knots.M[i][0]=z1;
    knots.M[i][1]=z2;
    knots.M[i][2]=t;
}
fclose(fp);
return;
}

double birth(matrix param, matrix s, matrix knots)
/*
    {birth} evaluates the birth rate at (s) (3 dimensions) for the knots
    (knots) and model parameters (param). {birth} is a thin plate spline.
*/
{
    int n, j;
    double sum=0.0,temp=0.0,partsum=0.0,theta,**knotsM,sv0,sv1,sv2,*knotp,
        x,x0,x1,x2,*paramV;
    n = knots.r;
    theta = 1.0/(96.0*PI);
    knotsM=knots.M;paramV=param.V;
    sv0=s.V[0];sv1=s.V[1];sv2=s.V[2];
    // radial basis part:
    for(j=0; j<n; j++)
    { knotp=knotsM[j];
        x = sv0 - *knotp;knotp++;
    }
}
```

```
    x1 = sv1 - *knotp;knotp++;
    x2 = sv2 - *knotp;
    temp=x*x+x1*x1+x2*x2;
    temp=temp*sqrt(temp);
    partsum += paramV[j]*temp;
}
// polynomial part:
temp = paramV[n]+paramV[n+1]*sv0+paramV[n+2]*sv1;
temp += paramV[n+3]*sv2;
temp += paramV[n+4]*sv0*sv0+paramV[n+5]*sv1*sv1;
temp += paramV[n+6]*sv2*sv2+paramV[n+7]*sv0*sv1;
temp += paramV[n+8]*sv0*sv2+paramV[n+9]*sv1*sv2;
sum = theta*partsum + temp;
return(sum);
}

double death(matrix param, matrix s, matrix knots)
/*
  {death} evaluates the death rate at (s) (3 dimensions) for the knots
  (knots) and model parameters (param). {death} is a thin plate spline.
*/
{
  int n, j, N;
  double sum=0.0,temp=0.0,partsum=0.0,theta,sv0,sv1,sv2,
    **knotsM,*knotp,*paramV,x,x1,x2;
  n = knots.r;
  N = 2*n+10;
  theta = 1.0/(96.0*PI);
  sv0=s.V[0];sv1=s.V[1];sv2=s.V[2];knotsM=knots.M;paramV=param.V;
  // radial basis part:
  for(j=0; j<n; j++)
  { knotp=knotsM[j];
    x = sv0 - *knotp;knotp++;
    x1 = sv1 - *knotp;knotp++;
    x2 = sv2 - *knotp;
    temp = x*x + x1*x1 + x2*x2;
    temp=temp*sqrt(temp);
    partsum += paramV[n+10+j]*temp;
  }
  // polynomial part...
  temp = paramV[N]+paramV[N+1]*sv0+paramV[N+2]*sv1;
  temp += paramV[N+3]*sv2;
  temp += paramV[N+4]*sv0*sv0+paramV[N+5]*sv1*sv1;
  temp += paramV[N+6]*sv2*sv2+paramV[N+7]*sv0*sv1;
  temp += paramV[N+8]*sv0*sv2+paramV[N+9]*sv1*sv2;
  sum = theta*partsum + temp;
  return(sum);
}

double intdfunction(matrix s, matrix knots, double x, matrix param)
/* {intdfunction} evaluates the integral of the death rate w.r.t. time
  at spatial co-ordinates (s) and time (x), for knots (knots) and model
  parameters (param). The integration has been tested by numerical
```

```

    comparison with the trapezoidal rule.
*/
{
    int n, N, j, i;
    double sum=0.0, D1, D2, D3, D4, U, parsum, theta,sv0,sv1,sv2,x0,x1,x2,
        **knotsM,*knotp,rU,*paramV;
    n = knots.r;
    N = 2*n+9;
    theta = 1.0/(96.0*PI);
    sv0=s.V[0];sv1=s.V[1];sv2=s.V[2];knotsM=knots.M;paramV=param.V;
// integration of the radial basis part:
for(j=0; j<n; j++)
{ knotp=knotsM[j];
  x0=sv0- *knotp;knotp++;
  x1=sv1- *knotp;knotp++;
  x2=x- *knotp;
  D1=x0*x0+x1*x1;
  U=D1+x2*x2;rU=sqrt(U);
  parsum=U*rU*0.25*x2;
  parsum += 0.375*D1*x2*rU;
  if ((x2+rU)>1e-6) parsum+=0.375*D1*D1*log(x2+rU);
  sum += parsum*paramV[n+10+j];
}
sum = theta*sum;
// integration of the polynomial part:
D2 = paramV[N+1] + paramV[N+2]*sv0 + paramV[N+3]*sv1;
D2 += paramV[N+5]*sv0*sv0 + paramV[N+6]*sv1*sv1;
D2 += paramV[N+8]*sv0*sv1;
D3 = (paramV[N+4] + paramV[N+9]*sv0+paramV[N+10]*sv1)*0.5;
D4 = paramV[N+7]/3.0;
sum += D2*x + D3*x*x + D4*x*x*x;
return(sum);
}

double intdeath(matrix s, double birthday, matrix param, matrix knots)
/* {intdeath} evaluates the integral of the death rate over time, from
   time of birth (birthday) till now (3rd element of (s)), for knots
   (knots) and model parameters (param). The integration has been
   tested by numerical comparison with the trapezoidal rule.
*/
{
    double answer;
    answer = intdfunction(s,knots,s.V[2],param);
    answer -= intdfunction(s,knots,birthday,param);
    return(answer);
}

void intpopn(matrix s, matrix knots, matrix param, matrix age,
    matrix beta, matrix f1, double *integral)
/* {intpopn} calculates the stage abundance (integral) at (s) for the
   the stage defined by the age limits (age.V[0]) and (age.V[age.r-1]).
   This is the integral of the population density per unit age
   (=beta*exp(-f1)) over these ages at (s) for knots (knots)

```

```

and model parameters (param). The Simpson rule is used for the
numerical integration and has been numerically compared with the
Trapezoidal rule.
The set of ages (age), birthrates (beta) and integrals of the death
rates (f1) are stored because they will be needed elsewhere.
*/
{
  int n, ns, i, k=MAXSIMP;
  double h, sum1=0.0, sum2=0.0, sum3=0.0, A, B;
  matrix s2;
  s2 = initmat(3L, 1L);
  ndata(&n, &i, &ns);
/* numerical integration using the Simpson rule is
  A = alpha_j = a0, B = alpha_(j+1) = a2k
  h = (B-A)/2k
  ai = a0+i*h
  Simp = (h/3)*{f(A)+4f(a1)+2f(a2)+...+2f(2ak-2)+4f(2ak-1)+f(B)}
*/
  A = age.V[0];           // age at entry to stage j
  B = age.V[age.r-1];    // age at departure from stage j
  h = (B-A)/(2.0*k);
  mcopy(&s, &s2);
  s2.V[2] = s.V[2] - A;
  beta.V[0] = birth(param, s2, knots);
  if(A<1.0e-7) f1.V[0] = 0.0; // if A=0 death integral from t to t!
  else f1.V[0] = intdeath(s, s2.V[2], param, knots);
  s2.V[2] = s.V[2] - B;
  beta.V[beta.r-1] = birth(param, s2, knots);
  f1.V[f1.r-1] = intdeath(s, s2.V[2], param, knots);
// sum1 = f(A) + f(B)
  sum1 = beta.V[0]*exp(-1.0*f1.V[0]);
  sum1 += beta.V[beta.r-1]*exp(-1.0*f1.V[f1.r-1]);
// sum2 = f(a1) + f(a3) + ... + f(a(2k-1))
  for(i=1; i<2*k; i+=2) {
    age.V[i] = A + i*h;
    s2.V[2] = s.V[2] - age.V[i];
    beta.V[i] = birth(param, s2, knots);
    f1.V[i] = intdeath(s, s2.V[2], param, knots);
    sum2 += beta.V[i]*exp(-1.0*f1.V[i]);
  }
// sum3 = f(a2) + f(a4) + ... + f(a(2k-2))
  for(i=2; i<(2*k-1); i+=2) {
    age.V[i] = A + i*h;
    s2.V[2] = s.V[2] - age.V[i];
    beta.V[i] = birth(param, s2, knots);
    f1.V[i] = intdeath(s, s2.V[2], param, knots);
    sum3 += beta.V[i]*exp(-1.0*f1.V[i]);
  }
  *integral = h*(sum1 + 4.0*sum2 + 2.0*sum3)/3.0;
  freemat(s2);
  return;
}

```

```

void deriv2(matrix s, matrix knots, double age, double *deriv)
/* {deriv2} calculates the first derivative (deriv) of beta={birth} or
f1={intdeath} at position (s) and age (age), for knots (knots),
w.r.t. the model parameters.
The derivatives have been checked numerically using finite
differencing: df/dx = (f(x+h) - f(x))/h for fixed h.
*/
{
  int i,j, K, n;
  double length, sum=0.0, D1, U, sum1=0.0, sum2=0.0, birthday, theta,
  sv0,sv1,sv2,x,x1,x2,x3,**knotsM,rU;
  sv0=s.V[0];sv1=s.V[1];sv2=s.V[2]; // save repeated address calculations.
  knotsM=knots.M;
  ndata(&n, &K, &i);
  birthday = sv2 - age;
  theta = 1.0/(96.0*PI);
  // dbeta/dgamma_i, i = 1,...,K === dbeta/dparam_i, i=1,...,K :
  for(i=0;i<K;i++)
  { x=sv0-knotsM[i][0];x1=sv1-knotsM[i][1];x2=birthday-knotsM[i][2];
    length=x*x+x1*x1+x2*x2;
    sum = length*sqrt(length);
    deriv[i] = theta*sum;
  }
  // dbeta/db_j, j=1,...,M === dbeta/dparam_i, i=K+1,...,K+M :
  deriv[K]=1.0;
  deriv[K+1]=sv0;
  deriv[K+2]=sv1;
  deriv[K+3]=birthday;
  deriv[K+4]=sv0*sv0;
  deriv[K+5]=sv1*sv1;
  deriv[K+6]=birthday*birthday;
  deriv[K+7]=sv0*sv1;
  deriv[K+8]=sv0*birthday;
  deriv[K+9]=sv1*birthday;
  for(i=0;i<K;i++)
  // df1/ddelta_i, i=1,...,K === df1/dparam_i, i=K+M+1,...,2K+M :
  { j=i+K+10; // index of derivative
    x=sv0-knotsM[i][0];x1=sv1-knotsM[i][1];
    x2=birthday-knotsM[i][2];x3=sv2-knotsM[i][2];
    D1 = x*x + x1*x1;
    U = D1 + x3*x3;rU=sqrt(U);
    sum1 = U*rU*0.25*x3;
    sum1 += 0.375*D1*x3*rU;
    // log(f) -> 0 as (f) -> 0, for f=x-k3+sqrt(U)
    // so only add next bit if f sufficiently large
    if((x3+rU)>1.0e-6) sum1 += 0.375*D1*D1*log(x3+rU);
    U = D1 + x2*x2;rU=sqrt(U);
    sum2 = U*rU*0.25*x2;
    sum2 += 0.375*D1*x2*rU;
    // log(f) -> 0 as (f) -> 0, for f=x-k3+sqrt(U)
    // so only add next bit if f sufficiently large
    if((x2+rU)>1.0e-6) sum2 += 0.375*D1*D1*log(x2+rU);
    deriv[j] = theta*(sum1-sum2);
  }
}

```

```

}
// df1/dd_j, j=1,...,M === df1/dparam_i, i=2K+M+1,...,2K+2M :
j++;deriv[j]=age;
j++;deriv[j]=age*sv0;
j++;deriv[j]=age*sv1;
j++;deriv[j]= age*sv2 - age*age*0.5;
j++;deriv[j]=age*sv0*sv0;
j++;deriv[j]=age*sv1*sv1;
j++;deriv[j]=age*age*age/3.0 + age*sv2*sv2 - age*age*sv2;
j++;deriv[j]=age*sv0*sv1;
j++;deriv[j] = age*sv0*sv2 - age*age*sv0*0.5;
j++;deriv[j]= age*sv1*sv2 - age*age*sv1*0.5;
}

void derivg2(matrix s, matrix knots, matrix param, double *deriv,
matrix age, matrix beta, matrix f1)
/* {derivg2} finds the first derivative (deriv) of fitted stage counts
w.r.t. the model parameters (param) at (s) for knots (knots). The
Simpson rule is used with the same grid as in {intpopn}, defined by
(age), using the birth rate values (beta) and survival values (f1).
*/
{
int n, ns, i, k=MAXSIMP, K, j;
double h,*sum1,*sum2,*sum3,answer,A,B,*temp,*f1V,*betaV,mult;
ndata(&n, &K, &ns);
temp=(double *)calloc((size_t)((2*K+20)*4),sizeof(double));
// sharing out memory to temporary arrays:
sum1=temp+2*K+20;sum2=sum1+2*K+20;sum3=sum2+2*K+20;
f1V=f1.V;betaV=beta.V; // save some address calculation
/* numerical integration of deta/dparam using the Simpson rule...
A = alpha_j = a0, B = alpha_(j+1) = a2k
h = (B-A)/2k
ai = a0+i*h
Simp = (h/3)*{f(A)+4f(a1)+2f(a2)+...+2f(2ak-2)+4f(2ak-1)+f(B)}
*/
A = age.V[0]; // age at entry to stage j
B = age.V[age.r-1]; // age at departure from stage j
h = (B-A)/(2.0*k); // integration step
// sum1 = f(a0)+f(a2k)
i=0;
deriv2(s,knots,age.V[i],sum1);
mult = exp(-f1V[i]);for(j=0;j<K+10;j++) sum1[j] *= mult;
mult = -exp(-f1V[i])*betaV[i];for(j=K+10;j<2*K+20;j++) sum1[j] *= mult;
i=f1.r-1;
deriv2(s,knots,age.V[i],temp);
mult = exp(-f1V[i]);for(j=0;j<K+10;j++) temp[j] *= mult;
mult = -exp(-f1V[i])*betaV[i];for(j=K+10;j<2*K+20;j++) temp[j] *= mult;
for(j=0;j<2*K+20;j++) sum1[j] += temp[j];
// sum2 = f(a1) + f(a3) + ... + f(a(2k-1))
for(i=1; i<2*k; i+=2)
{ deriv2(s,knots,age.V[i],temp);
mult = exp(-f1V[i]);for(j=0;j<K+10;j++) temp[j] *= mult;
mult = -exp(-f1V[i])*betaV[i];for(j=K+10;j<2*K+20;j++) temp[j] *= mult;

```

```

    for(j=0;j<2*K+20;j++) sum2[j] += temp[j];
}
// sum3 = f(a2) + f(a4) + ... + f(a(2k-2))
for(i=2; i<(2*k-1); i+=2)
{ deriv2(s,knots,age.V[i],temp);
  mult = exp(-f1V[i]);for(j=0;j<K+10;j++) temp[j] *= mult;
  mult = -exp(-f1V[i])*betaV[i];for(j=K+10;j<2*K+20;j++) temp[j] *= mult;
  for(j=0;j<2*K+20;j++) sum3[j] += temp[j];
}
for(j=0;j<2*K+20;j++) deriv[j] = h*(sum1[j]+4.0*sum2[j]+2.0*sum3[j])/3.0;
free(temp);
}

void getJg(matrix param, double *smoothp, matrix cov, matrix knots,
matrix g, matrix J, matrix w, int choice, int count)
/* {getJg} calculates the population density (g) for each data point
(cov) and age stage, using model parameters (param), and knots
(knots). It also fills the jacobian matrix (J) and weights matrix
(w) if (choice)>0. It writes the file "yparam[count]" containing
(g), (param) and smoothing parameters (smoothp) when count is not 99.
*/
{
FILE *fp3, *fp2, *fp1;
char f[10], filename[10]="yparam";
int i, j, k, K, n, ns, np, nr;
double integral,*deriv;
matrix ages, age, beta, f1, s;
K = knots.r;
n = cov.r;
ns = g.r/n;
np = param.r;
nr = 2*MAXSIMP+1;
deriv=(double *)calloc((size_t)np,sizeof(double));
ages = initmat((long) n, (long) ns+1);
s = initmat(3L, 1L);
age = initmat((long) nr, 1L);
beta = initmat((long) nr, 1L);
f1 = initmat((long) nr, 1L);
stageage(ages);
// work through the stages and data points:
for(j=0; j<ns; j++){
  for(i=0; i<n; i++){
    s.V[0] = cov.M[i][0];
    s.V[1] = cov.M[i][1];
    s.V[2] = cov.M[i][2];
    age.V[0] = ages.M[i][j];
    age.V[age.r-1] = ages.M[i][j+1];
// obtain the population estimates g = E(y) :
    intpopn(s,knots,param,age,beta,f1,&integral);
    g.V[n*j+i] = integral;
// obtain the weights :
    if(DISTN==1){w.V[n*j+i] = 1.0;}
    if(DISTN==2){

```

```
        w.V[n*j+i] = 1.0/fabs(integral);
        if(integral<=1e-6) {
            // printf("g=%lg ",integral);
            g.V[n*j+i] = 0.0;
            integral = 0.0;
        }
    }
}
// obtain the Jacobian if choice>0:
    if(choice>0){
        derivg2(s,knots,param,deriv,age,beta,f1);
        for(k=0; k<np; k++) { J.M[j*n+i][k]=deriv[k]; }
    }
}
}
// if count==99 finish now:
    if(count==99){
        freemat(ages);
        freemat(s);
        freemat(beta);
        freemat(f1);
        freemat(age);
        return;
    }
// write fitted values, model and smoothing parameters to "yparam[count]"
    sprintf(filename,"%s%d",filename,count);
    printf(" writing %s\n",filename);
    if((fp1=fopen("log", "a"))==NULL){
        printf("Error opening file \"log\" \n");
        exit(1);
    }
    fprintf(fp1," Writing %s\n",filename);
    fflush(fp1);
    fclose(fp1);
    if((fp3=fopen(filename, "w"))==NULL){
        printf("Error opening file \"%s\" \n",filename);
        exit(1);
    }
    for(j=0; j<ns; j++){
        for(i=0; i<n; i++){
            fprintf(fp3,"% .18lf",g.V[n*j+i]);
            if(n*j+i<2*K+20)fprintf(fp3,"\t%.18lf",param.V[n*j+i]);
            if(n*j+i<2)fprintf(fp3,"\t%.18lf",*(smoothp+n*j+i));
            fprintf(fp3,"\n");
        }
    }
    fclose(fp3);
    freemat(ages);
    freemat(s);
    freemat(beta);
    freemat(f1);
    freemat(age);
    return;
}
```

```
void fillemat(matrix knots, matrix E)
/* {fillemat} fills the non-zero part of the penalty matrix (E), which
   is the same for the birth and the death rate, using knots (knots).
*/
{
  int i, j, K;
  double part, theta;
  matrix diff;
  K = knots.r;
  diff = initmat(3L, 1L);
  theta = 1.0/(96.0*PI);
  for(i=0; i<K; i++){
    for(j=0; j<K; j++){
      diff.V[0] = knots.M[i][0] - knots.M[j][0];
      diff.V[1] = knots.M[i][1] - knots.M[j][1];
      diff.V[2] = knots.M[i][2] - knots.M[j][2];
      part = enorm(diff);
      part = theta*pow(part,3.0);
      E.M[i][j] = part; // E_ij = theta * ||s_i - s_j||^3
    }
  }
  freemat(diff);
  return;
}

void adjustJ(matrix J, matrix Jz, matrix Z)
/* {adjustJ} transforms the jacobian matrix (J) to the adusted
   version (Jz), using the orthogonal matrix (Z).
*/
{
  int i, j, K, np;
  matrix bigZ;
  K = Z.r;
  np = J.c;
  bigZ = initmat((long) np, (long) 2*K);
  for(i=0; i<bigZ.r; i++){
    for(j=0; j<bigZ.c; j++) bigZ.M[i][j] = 0.0;
  }
  for(i=0; i<K; i++){
    for(j=0; j<K-10; j++){
      bigZ.M[i][j] = Z.M[i][j];
      bigZ.M[i+K+10][j+K] = Z.M[i][j];
    }
  }
  for(i=0; i<10; i++){
    bigZ.M[i+K][i+K-10] = 1.0;
    bigZ.M[i+2*K+10][i+2*K-10] = 1.0;
  }
}
/* bigZ is a matrix consisting of
   Z 0 0 0 , Jz = J * bigZ
   0 I 0 0
   0 0 Z 0
   0 0 0 I
```

```
*/
  matmult(Jz, J, bigZ, 0,0);
  freemat(bigZ);
  return;
}

void adjustp(matrix param, matrix paramz, matrix Z)
/* {adjustp} transforms the constrained parameters (paramz) to the
   usable model parameters (param) using the orthogonal matrix (Z).
*/
{
  int i, j, K, np;
  matrix bigZ;
  K = Z.r;
  np = param.r;
  bigZ = initmat((long) np, (long) 2*K);
  for(i=0; i<bigZ.r; i++){
    for(j=0; j<bigZ.c; j++) bigZ.M[i][j] = 0.0;
  }
  for(i=0; i<K; i++){
    for(j=0; j<K-10; j++){
      bigZ.M[i][j] = Z.M[i][j];
      bigZ.M[i+K+10][j+K] = Z.M[i][j];
    }
  }
  for(i=0; i<10; i++){
    bigZ.M[i+K][i+K-10] = 1.0;
    bigZ.M[i+2*K+10][i+2*K-10] = 1.0;
  }
  matmult(param, bigZ, paramz, 0,0);
  freemat(bigZ);
  return;
}

void checkcons(matrix param, matrix T)
/* This function makes sure that the equality constraints are satisfied,
   i.e. T*param=0. If this is not the case, the user is informed
   and the run terminated.
*/
{
  FILE *fp1;
  int i, j, K;
  double Tp, ep1, ep2;
  matrix p1, p2;
  K = (param.r-20)/2;
  p1 = initmat((long) K, 1L);
  p2 = initmat((long) K, 1L);
  for(i=0; i<K; i++){
    p1.V[i] = param.V[i];
    p2.V[i] = param.V[i+K+10];
  }
  ep1 = enorm(p1);
  ep2 = enorm(p2);
}
```

```

for(i=0; i<T.r; i++){
// ith row of T times jth gamma parameter (for birth spline)...
  Tp = 0.0;
  for(j=0; j<T.c; j++) Tp += T.M[i][j]*p1.V[j];
  if((fabs(Tp)/ep1)>MAXCHECK){
    printf("\nCondition on birth rate parameters not ");
    printf("satisfied\n %d %lg",j,Tp);
    if((fp1=fopen("log", "a"))==NULL){
      printf("Error opening file \"log\" \n");
      exit(1);
    }
    fprintf(fp1,"\nCondition on birth rate parameters not ");
    fprintf(fp1,"satisfied\n %d %lg",j,Tp);
    fflush(fp1);
    fclose(fp1);
    exit(1);
  }
// ith row of T times jth delta parameter (for death spline)...
  Tp = 0.0;
  for(j=0; j<T.c; j++) Tp += T.M[i][j]*p2.V[j];
  if((fabs(Tp)/ep2)>MAXCHECK){
    printf("\nCondition on death rate parameters not ");
    printf("satisfied\n %d %lg",j,Tp);
    if((fp1=fopen("log", "a"))==NULL){
      printf("Error opening file \"log\" \n");
      exit(1);
    }
    fprintf(fp1,"\nCondition on death rate parameters not ");
    fprintf(fp1,"satisfied\n %d %lg",j,Tp);
    fflush(fp1);
    fclose(fp1);
    exit(1);
  }
}
freemat(p1);
freemat(p2);
return;
}

void constraints(matrix Z, matrix cov, matrix knots, matrix Ain,
matrix b)
/* {constraints} fills the inequality constraints matrices (Ain) and (b)
using the orthogonal matrix (Z), the grid of points (cov) (3 dimensions),
and the knots (knots).
The constraints here are that the birth and death functions must be
non-negative: Ain * paramz >= b. Note Ain * pz = (birth, death)^T
*/
{
int i, j, K, n;
matrix X, XZ, diff;
double part, theta;
theta = 1.0/(96.0*PI);
K = knots.r;

```

```

n = b.r/2;
X = initmat((long) n, (long) K); // design matrix of a tps
XZ = initmat((long) n, (long) K-10); // adjusted design matrix
diff = initmat(3L, 1L);
for(i=0; i<n; i++){
  for(j=0; j<K; j++){ // radial basis part:
    diff.V[0] = cov.M[i][0] - knots.M[j][0];
    diff.V[1] = cov.M[i][1] - knots.M[j][1];
    diff.V[2] = cov.M[i][2] - knots.M[j][2];
    part = enorm(diff);
    X.M[i][j] = theta*pow(part,3.0);
  } // polynomial part:
  Ain.M[i][K-10] = 1.0;
  Ain.M[i][K-9] = cov.M[i][0];
  Ain.M[i][K-8] = cov.M[i][1];
  Ain.M[i][K-7] = cov.M[i][2];
  Ain.M[i][K-6] = cov.M[i][0]*cov.M[i][0];
  Ain.M[i][K-5] = cov.M[i][1]*cov.M[i][1];
  Ain.M[i][K-4] = cov.M[i][2]*cov.M[i][2];
  Ain.M[i][K-3] = cov.M[i][0]*cov.M[i][1];
  Ain.M[i][K-2] = cov.M[i][0]*cov.M[i][2];
  Ain.M[i][K-1] = cov.M[i][1]*cov.M[i][2];
  b.V[i] = 0.0;
  b.V[i+n] = 0.0;
}
matmult(XZ, X, Z, 0, 0); // adjust the radial basis part
for(i=0; i<n; i++){
  for(j=0; j<K-10; j++){
    Ain.M[i][j] = XZ.M[i][j];
    Ain.M[i+n][j+K] = XZ.M[i][j];
    Ain.M[i][j+K] = 0.0;
    Ain.M[i+n][j] = 0.0;
  }
  for(j=K-10; j<K; j++){
    Ain.M[i+n][j+K] = Ain.M[i][j];
    Ain.M[i][j+K] = 0.0;
    Ain.M[i+n][j] = 0.0;
  }
}
freemat(X);
freemat(XZ);
freemat(diff);
return;
}

void inequalcons(matrix paramz, double *smoothp, matrix Ain, matrix b,
matrix Jz, matrix w, matrix z, matrix Pa0)
/* {inequalcons} minimises ||w*longJz*pz - longz||^2 subject to
Ain*paramz>=b
where
longJz = [ w*Jz ], longz = [z],
[(smoothp0*Pa0)^0.5] [0]
[(smoothp1*Pa0)^0.5] [0]

```

```
    This function should be called after {MultiSmooth} and before
    {goldensmooth}
*/
{
    FILE *fp1;
    int i, j, nr, n, K, ns;
    int *active;
    matrix longJz, longz, Z, Af;
    ndata(&n, &K, &ns);
    nr = n*ns;
    active = (int *) calloc((size_t)b.r+1, sizeof(int));
    if(!active) {
        printf("allocation failure - aborting");
        exit(1);
    }
    longJz = initmat((long) nr+4*K, (long) 2*K);
    longz = initmat(longJz.r, 1L);
    Z = initmat(OL, OL); // empty matrices to feed into QPCLS because
    Af = initmat(OL, OL); // have dealt with own equality constraints
    // set bottom part of longJz and longz to 0...
    for(i=nr; i<longz.r; i++){
        for(j=0; j<longJz.c; j++) longJz.M[i][j] = 0.0;
        longz.V[i] = 0.0;
    }
    // set top part of longJz and longz to W*Jz and W*z...
    for(i=0; i<nr; i++){
        for(j=0; j<longJz.c; j++){
            longJz.M[i][j] = sqrt(w.V[i])*Jz.M[i][j];
        }
        longz.V[i] = sqrt(w.V[i])*z.V[i];
    }
    // set bottom part of longJz to be the penalty part...
    for(i=0; i<K-10; i++){
        for(j=0; j<K-10; j++){
            longJz.M[i+nr][j] = sqrt(smoothp[0])*Pa0.M[j][i];
            longJz.M[i+nr+3*K][j+K] = sqrt(smoothp[1])*Pa0.M[j][i];
        }
    }
    if((fp1=fopen("log", "a"))==NULL){
        printf("Error opening file \"log\" \n");
        exit(1);
    }
    printf("\n Doing QPCLS (inequality constraints)...");
    fprintf(fp1, "\n Doing QPCLS (inequality constraints)...");
    fflush(fp1);
    // quadratic programming (in qp.c):
    QPCLS(&Z, &longJz, &paramz, &longz, &Ain, &b, &Af, active);
    printf(" done QPCLS");
    fprintf(fp1, " done QPCLS");
    fflush(fp1);
    fclose(fp1);
    freemat(longJz);
    freemat(longz);
}
```

```
freemat(Z);
freemat(Af);
free(active);
return;
}

double object(matrix pz, matrix Z, matrix Pa0, double *smoothp)
/* {object} evaluates the objective function (object) for the model
   parameters(pz), smoothing parameters (smoothp), penalty matrix
   (Pa0) using the orthogonal matrix (Z).
*/
{
  int i, n, K, ns;
  double yminusg, penalty;
  matrix p, z, cov, knots, J, w, y, pb, pd, pPp;
  matrix g;
  ndata(&n, &K, &ns);
  p = initmat((long) 2*K+20, 1L);
  z = initmat((long) n*ns, 1L);
  cov = initmat((long) n, 3L);
  knots = initmat((long) K, 3L);
  g = initmat(z.r, 1L);
  J = initmat(z.r, p.r);
  w = initmat(z.r, 1L);
  y = initmat(z.r, 1L);
  pb = initmat((long) K-10, 1L);
  pd = initmat(pb.r, 1L);
  pPp = initmat(1L, 1L);
  getdata(cov, y, knots);
  adjustp(p, pz, Z);
  // calculate sum of squares part:
  getJg(p, smoothp, cov, knots, g, J, w, -1, 99);
  mad(z,g,y,1.0,-1.0);
  for(i=0; i<z.r; i++) z.V[i]=sqrt(w.V[i])*z.V[i];
  yminusg = enorm(z);
  // calculate wiggleness penalty part:
  for(i=0; i<pb.r; i++){
    pb.V[i] = pz.V[i];
    pd.V[i] = pz.V[i+K];
  }
  multi(3,pPp,pb,Pa0,pb,1,0,0);
  penalty = *smoothp * pPp.V[0];
  multi(3,pPp,pd,Pa0,pd,1,0,0);
  penalty += *(smoothp+1) * pPp.V[0];
  freemat(p);
  freemat(z);
  freemat(cov);
  freemat(knots);
  freemat(g);
  freemat(J);
  freemat(w);
  freemat(y);
  freemat(pb);
```

```
freemat(pd);
freemat(pPp);
return(yminusg+penalty);
}

void goldensmooth(matrix oldpz, matrix plz, matrix Z, matrix Pa0,
double *smoothp, double *ob)
/* {goldensmooth} performs a golden section search to choose the
model parameters which minimise the true objective function
(||y-g||+ penalty) in the subspace containing (oldpz) and (plz)
when the smoothing parameters are fixed at (smoothp). The
objective function of (plz) is assumed to be less than the
objective function of (oldpz). This function does not use
steepest descent to find the best searching direction, it simply
searches in the direction oldpz -> plz. The best parameters are
put in (plz), and the objective function of these parameters in
(ob). (Z) and (Pa0) are required for the calculation of the
objective function.
*/
{
FILE *fp1;
int i, n, K, ns, check=1;
double f0, f1, fa, fb, fx, fo, ratio=0.6180339, epsilon=1.0e-3;
matrix p0z, paz, pbz, diffz;
p0z = initmat(plz.r, 1L);
paz = initmat(plz.r, 1L);
pbz = initmat(plz.r, 1L);
diffz = initmat(plz.r, 1L);
mcopy(&oldpz,&p0z);
mad(diffz,plz,p0z,1.0,-1.0); // plz-p0z
// find objective for the old and the new parameters
f0 = object(p0z,Z,Pa0,smoothp);
f1 = object(plz,Z,Pa0,smoothp);
// if the gap between the old and new parameters is small enough
// don't need to do anything so return...
if(enorm(diffz)<1.0e-4) return;
// Get on with the golden section search:
mad(paz,p0z,diffz,1.0,ratio);
mad(pbz,plz,diffz,1.0,-ratio);
fa = object(paz,Z,Pa0,smoothp);
fb = object(pbz,Z,Pa0,smoothp);
for(i=0; i<10; i++){ // max 10 steps allowed
if(fb<=fa){
mcopy(&paz,&plz);
f1 = fa;
mcopy(&pbz,&paz);
fa = fb;
mad(diffz,plz,p0z,1.0,-1.0);
mad(pbz,plz,diffz,1.0,-ratio);
fb = object(pbz,Z,Pa0,smoothp);
check=1;
}
else if(fb>fa){
```

```
        mcopy(&pbz,&p0z);
        f0 = fb;
        mcopy(&paz,&pbz);
        fb = fa;
        mad(diffz,p1z,p0z,1.0,-1.0);
        mad(paz,p0z,diffz,1.0,ratio);
        fa = object(paz,Z,Pa0,smoothp);
        check=2;
    }
    if(fabs(fb-fa)<1.0e-5) break;
}
if(check==1){
    mcopy(&pbz,&p1z);        // fb<fa so b is best
    f1 = fb;
}
else if(check==2){
    mcopy(&paz,&p1z);        // fa<fb so a is best
    f1 = fa;
}
if((fp1=fopen("log", "a"))==NULL){
    printf("Error opening file \"log\" \n");
    exit(1);
}
fprintf(fp1,"\t objective function is now %lg\n",f1);
fflush(fp1);
fclose(fp1);
printf("\t objective function is now %lg\n",f1);
*ob = f1;
freemat(p0z);
freemat(paz);
freemat(pbz);
freemat(poz);
freemat(diffz);
return;
}

int criteria(matrix oldg, matrix g)
/* {criteria} finds out if the fitted model has converged, i.e.
the distance from (oldg) to (g) is sufficiently small. Returned value
is 0 if there is no convergence, 1 if some elements have converged,
greater than 1 if convergence has occurred. If the parameters are
diverging then the user is informed and the program terminated.
*/
{
    FILE *fp1;
    int checkg=1, i;
    double diff, sumdiff, meang;
    if((fp1=fopen("log", "a"))==NULL){
        printf("Error opening file \"log\" \n");
        exit(1);
    }
    sumdiff=0.0;
    for(i=0; i<g.r; i++){
```

```
diff = fabs(oldg.V[i] - g.V[i]);
meang = fabs(oldg.V[i]+g.V[i])/2.0;
diff /= (1.0+meang);
if(diff>MAXDIFF){
    printf("\n Model fitting has failed! ");
    printf("\n The difference between g^k and g^(k-1) is");
    printf(" too large for convergence");
    fprintf(fp1,"\n Model fitting has failed! ");
    fprintf(fp1,"\n The difference between g^k and g^(k-1)");
    fprintf(fp1," is too large for convergence\n");
    fflush(fp1);
    exit(0);
}
if(diff>CONVERGE) checkg = 0;
sumdiff += diff;
}
printf("\n Sum of \"distance\" from old to new ");
printf("fitted values: %lg",sumdiff);
fprintf(fp1,"\n Sum of \"distance\" from old to new ");
fprintf(fp1,"fitted values: %lg",sumdiff);
fflush(fp1);
if(checkg>0){
    printf("\n The fitted values have converged");
    fprintf(fp1,"\n The fitted values have converged");
    fflush(fp1);
}
fclose(fp1);
return(checkg);
}

double rsq(matrix y, matrix g, int K)
/* {rsq} finds R-squared of (g) fitted to (y) given (K), the number
   of knots (which is used to give the number of parameters).
*/
{
    int i;
    double ypart=0.0, rpart=0.0, ybar, rbar, rsq;
    matrix r;
    r = initmat(y.r,1L);
    ybar = meanmat(y);
    for(i=0; i<y.r; i++) ypart += (y.V[i] - ybar)*(y.V[i] - ybar);
    ypart /= y.r;
    mad(r,y,g,1.0,-1.0);
    rbar = meanmat(r);
    for(i=0;i<r.r;i++){
        rpart += (r.V[i]-rbar)*(r.V[i]-rbar);
    }
    rpart /= (g.r - 2*K - 20);
    rsq = 1 - rpart/ypart;
    freemat(r);
    return(rsq);
}
```

```

void sumbirth(matrix param, matrix knots, double dimsize, matrix ans)
/* {sumbirth} calculates the total birth rate and other integrated values
   (ans) by adding up the functions evaluated at points on a fine grid
   (in file "sumbXY"), using (knots) and (param), then multiplying by the
   grid step size. The grid through time consists of (dimsize) time steps.
*/
{
  FILE *fp;
  int i, j, l, n, K, ns, nr;
  double h1,h2,h3, minx1, minx2, mint, maxx1, maxx2, maxt, x,y, b;
  double integral;
  matrix s, age, ages, beta, f1;
  ndata(&n, &K, &ns);
  nr = 2*MAXSIMP+1;
  s = initmat(3L, 1L); // one point
  ages = initmat((long) n, (long) ns+1);
  age = initmat((long) nr, 1L);
  beta = initmat((long) nr, 1L);
  f1 = initmat((long) nr, 1L);
  stageage(ages);
  // initialise all answers to zero:
  for(i=0; i<ans.r; i++){ ans.V[i] = 0.0;}
  if((fp=fopen("minmax", "r"))==NULL){
    printf("Error opening file 'minmax' \n"); getc(stdin);
    exit(1);
  }
  fscanf(fp,"%d%d%d",&i,&j,&l);
  fscanf(fp,"%lg%lg%lg%lg%lg%lg",&minx1,&maxx1,&minx2,&maxx2,&mint,&maxt);
  fclose(fp);
  h1 = (maxx1-minx1)/(dimsize-1.0);
  h2 = (maxx2-minx2)/(dimsize-1.0);
  h3 = (maxt-mint)/(dimsize-1.0);
  if((fp=fopen("sumbXY", "r"))==NULL){
    printf("Error opening file 'sumbXY' \n"); getc(stdin);
    exit(1);
  }
  printf("Starting sumbirth summation loop...\n");
  for(i=0; i<SIZEXY; i++)
  { fscanf(fp,"%lg%lg",&x,&y);
    s.V[0] = x;
    s.V[1] = y;
    for(j=0; j<dimsize; j++)
    { s.V[2] = mint+j*h3;
      if((s.V[2]<(-25.5))&&(s.V[1]>20.0)){ans.V[0]+=0.0;}
      else
      { b = birth(param, s, knots);
        if(b < 0.0){ b = 0.0;}
        ans.V[0] += b;
      }
    }
  }
  // this part is specific to the sole egg data:
  // at sample times -45.5 -25.0 -1.5 15.5 45.5:
  //naiff but quick to think of: age only differs for time

```

```

s.V[2] = -45.5;
for(j=0; j<ns; j++){
    age.V[0] = ages.M[1][j];
    age.V[age.r-1] = ages.M[1][j+1];
    intpopn(s,knots,param,age,beta,f1,&integral);
    if(integral < 0.0){ integral = 0.0;}
    ans.V[1+j]+= integral;
}
s.V[2] = -25.0;
for(j=0; j<ns; j++){
    age.V[0] = ages.M[61][j];
    age.V[age.r-1] = ages.M[61][j+1];
    intpopn(s,knots,param,age,beta,f1,&integral);
    if(integral < 0.0){ integral = 0.0;}
    ans.V[ns+1+j]+= integral;
}
s.V[2] = -1.5;
for(j=0; j<ns; j++){
    age.V[0] = ages.M[142][j];
    age.V[age.r-1] = ages.M[142][j+1];
    intpopn(s,knots,param,age,beta,f1,&integral);
    if(integral < 0.0){ integral = 0.0;}
    ans.V[ns*2+1+j]+= integral;
}
s.V[2] = 15.5;
for(j=0; j<ns; j++){
    age.V[0] = ages.M[226][j];
    age.V[age.r-1] = ages.M[226][j+1];
    intpopn(s,knots,param,age,beta,f1,&integral);
    if(integral < 0.0){ integral = 0.0;}
    ans.V[ns*3+1+j]+= integral;
}
s.V[2] = 45.5;
for(j=0; j<ns; j++){
    age.V[0] = ages.M[311][j];
    age.V[age.r-1] = ages.M[311][j+1];
    intpopn(s,knots,param,age,beta,f1,&integral);
    if(integral < 0.0){ integral = 0.0;}
    ans.V[ns*4+1+j]+= integral;
}
}
printf("DONE \n");
fclose(fp);
/*
// use this function for simulated data with regular data space:
for(i=0; i<dimsize; i++){
    for(j=0; j<dimsize; j++){
        for(l=0; l<dimsize; l++){
            s.V[0] = minx1+l*h1;
            s.V[1] = minx2+j*h2;
            s.V[2] = mint+i*h3;
            sumb += birth(param, s, knots);
        }
    }
}

```

```
    }
}
sumb = sumb*h1*h2*h3;
*/
for(i=0; i<ans.r; i++){ ans.V[i] = ans.V[i]*h1*h2;}
ans.V[0] = ans.V[0]*h3;
freemat(s);
return;
}

void fit(void)
/* {fit} is the central function: it initialises what needs
   initialised, iterates through the model fitting procedure and stops
   when convergence has occurred.
*/
{
FILE *fp1, *fp2, *fp;
int n, K, m=NPENALTY, i, j, np, ns, nr, count, check, r;
long *off;
double sig2=0.0, *oldsp, *smoothp;
double ybar, Rsq, gap, ob, sumb, z1, z2, t;
matrix resid;
matrix cov, knots, y, param, paramz, dummpyz, oldpz, Z, Q, T, T2;
matrix *P, J, Jz, z, Jpz, g, w, E, Ez, Cons, oldg,rP;
matrix Initial, ages, Ain, b, resid1, resid2;
matrix covlong, blong, Ainlong;
matrix ans;
double trA, conv_tol=1e-6, min_edf=20.0, target_edf=-1.0;
msctrl_type msctrl;
msrep_type msrep;
int direct_mesh=100, ms_max_half=15;
ndata(&n, &K, &ns);
np = 2*K+20; // number of parameters
nr = ns*n; // number of rows (stage counts)
// initialise arrays...
P = (matrix *) calloc((size_t)m, sizeof(matrix));
if(!P) {
printf("allocation failure - aborting");
exit(1);
}
off = (long *) calloc((size_t)m, sizeof(long));
if(!off) {
printf("allocation failure - aborting");
exit(1);
}
oldsp = (double *) calloc((size_t)m, sizeof(double));
if(!oldsp) {
printf("allocation failure - aborting");
exit(1);
}
smoothp = (double *) calloc((size_t)m, sizeof(double));
if(!smoothp) {
printf("allocation failure - aborting");
```

```

    exit(1);
}
// initialise matrices...
cov = initmat((long) n, 3L); // data points (covariates)
covlong = initmat((long) NCONSTRAINTS, 3L); // inequality constraint locations
knots = initmat((long) K, 3L); // knots
y = initmat((long) nr, 1L); // stage counts
param = initmat((long) np, 1L); // model parameters
paramz = initmat((long) 2*K, 1L); // constrained parameters
dummpz = initmat((long) 2*K, 1L); // dummy constrained parameters
oldpz = initmat((long) 2*K, 1L); // previous constrained parameters
Z = initmat((long) K, (long) K-10); // transforming matrix
Q = initmat((long) K, (long) K); // for QT decomposition
T = initmat(10L, (long) K); // equality constraints matrix
T2 = initmat(10L, (long) K); // copy of T
*P = initmat(Z.c, Z.c); // constrained penalty matrix for birth
*(P+1) = initmat(Z.c, Z.c); // constrained penalty matrix for death
J = initmat((long) nr, (long) np); // Jacobian matrix
Jz = initmat((long)nr, (long)2*K); // constrained Jacobian matrix
z = initmat((long) nr, 1L); // pseudo data
Jpz = initmat((long) nr, 1L); // Jz * paramz
g = initmat((long) nr, 1L); // fitted stage counts
oldg = initmat((long) nr, 1L); // previous fitted stage counts
w = initmat((long) nr, 1L); // weights
E = initmat((long) K, (long) K); // non-zero part of penalty matrix
Ez = initmat(E.r, Z.c); // partially adjusted E
Cons = initmat(0L, 0L); // constraints for MultiSmooth
Initial = initmat((long) 2*n, 1L); // initial birth and death rates
ages = initmat((long)n, (long)ns+1); // ages limits for each stage
Ainlong = initmat((long) 2*NCONSTRAINTS, (long) 2*K); // inequality
// constraints matrix
blong = initmat((long) 2*NCONSTRAINTS, 1L); // inequality
// constraints matrix
resid1 = initmat(g.r, 1L); // Jpz-z
resid2 = initmat(g.r, 1L); // y-g
ans = initmat((long) 5*ns+1, 1L); // answers e.g. egg production
// remove old log file and open a new one:
remove("log");
if((fp1=fopen("log", "a"))==NULL){
    printf("Error opening file \"log\" \n");
    exit(1);
}
/*
*****
* INITIALISE VARIABLES *
*****
*/
getdata(cov, y, knots); // get covariates, knots and y
stageage(ages); // get limits for age stages
// off indicates which row of P each penalty matrix begins on...
*off = 0; // 1st penalty corresponds to birth
*(off+1) = K; // 2nd penalty corresponds to death
*smoothp = 0.0; // initialise smoothing parameters

```

```

*(smoothp+1) = 0.0;
sig2 = 0.0;           // sig2=estimate(error) by MultiSmooth

for(i=0; i<K; i++){   // Constrain T*gamma=0, T*delta=0
    T.M[0][i] = 1;
    T.M[1][i] = knots.M[i][0];
    T.M[2][i] = knots.M[i][1];
    T.M[3][i] = knots.M[i][2];
    T.M[4][i] = knots.M[i][0]*knots.M[i][0];
    T.M[5][i] = knots.M[i][1]*knots.M[i][1];
    T.M[6][i] = knots.M[i][2]*knots.M[i][2];
    T.M[7][i] = knots.M[i][0]*knots.M[i][1];
    T.M[8][i] = knots.M[i][0]*knots.M[i][2];
    T.M[9][i] = knots.M[i][1]*knots.M[i][2];
}
// QT changes T, but need T later to check constraints so:
mcopy(&T, &T2);
QT(Q,T2,1);          // TQ=(0 Tr), Tr=reverse lower triangle
for(i=0; i<K; i++){ // Z=first K-10 columns of Q (Z'*Z = I)
    for(j=0; j<K-10; j++) Z.M[i][j] = Q.M[i][j];
}
if((fp=fopen("constraintsgrid", "r"))==NULL){
    printf("Error opening file \"constraintsgrid\" \n");
    getc(stdin);
    exit(1);
}
for(i=0; i<NCONSTRAINTS; i++){
    fscanf(fp, "%lg%lg%lg", &z1, &z2, &t);
    covlong.M[i][0]=z1;
    covlong.M[i][1]=z2;
    covlong.M[i][2]=t;
}
fclose(fp);
printf("Calculating constraints matrices...");
constraints(Z,covlong,knots,Ainlong,blong);
printf("DONE\n");
fillemat(knots,E); // non-zero part of penalty matrices
matmult(Ez,E,Z,0,0);
matmult(P[0],Z,Ez,1,0); // penalty matrix for birth
mcopy(&P[0],&P[1]); // penalty matrix for death
rP=svdroot(P[0],0.0); // square root of penalty matrix rP*rP'=P[0]
// initialize model parameters:
// only the ones relating to constant polynomial terms are non-zero:
for(i=0; i<paramz.r; i++){
    paramz.V[i] = 0.0;
}
ybar = meanmat(y);
paramz.V[K-10] = 1.1*ybar; // birth
paramz.V[2*K-10] = 0.1; // death
/*
*****
*                               WORK THROUGH MODEL                               *
*                               FITTING ITERATIONS                               *
*****

```

```

*****
*/
printf("Entering Loop\n");
for(count = 0; count<MAXCOUNT; count++){
// put something reasonable into dummypz for passing to MultiSmooth:
  mcopy(&paramz,&dummypz);
// parameters from previous iterate...
  mcopy(&paramz,&oldpz);
  *oldsp = *smoothp;
  *(oldsp+1) = *(smoothp+1);
  adjustp(param, paramz, Z); // convert paramz->param
  checkcons(param, T);      // check constraints satisfied

// get J, g, w & write "yparam[count]" for this iterate:
  printf("\nUpdating g=E(Y), the jacobian and weights.... \n");
  fprintf(fp1,"\n Updating g=E(Y), the jacobian and weights.\n");
  fflush(fp1);
  getJg(param, smoothp, cov, knots, g, J, w, 1, count);
  printf("DONE\n");
// uncomment if don't want to write "yparam[count]":
// getJg(param, smoothp, cov, knots, g, J, w, 1, 99);
  mcopy(&g, &oldg);
// constrain jacobian...
  printf("Projecting J into constraint null space... \n");
  adjustJ(J, Jz, Z);      // adjust jacobian Jz = J*bigZ
  printf("DONE\n");
// sort out pseudo data...
  matmult(Jpz,Jz,paramz,0,0); // Jpz = Jz*paramz
  mad(z,y,g,1.0,-1.0);
  mcopy(&z,&resid1);        // resid1 = y - g
  mad(z,z,Jpz,1.0,1.0);    // z = y - g + Jpz

// R-squared and other handy information...
  mad(resid2,Jpz,z,1.0,-1.0); // Jpz - z
  gap = enorm(resid1)-enorm(resid2);
  Rsq = rsq(y,g,np);
  printf("\n iterate no %d\n\t ||y-g|| = %g\n",count+1,enorm(z));
  printf("\t ||y-g|| - ||Jp-z|| = %lg\n",gap);
  printf("\t R-squared = %lg\n",Rsq);
  printf("\t smoothing parameters = %lg , %lg\n",*smoothp,*(smoothp+1));
  printf("\t enorm of model params = %lg \n",enorm(param));
  fprintf(fp1,"\n iterate no %d\n\t ||y-g|| = ",count+1);
  fprintf(fp1,"%g\n",enorm(resid1));
  fprintf(fp1,"\t ||y-g|| - ||Jp-z|| = %lg\n",gap);
  fprintf(fp1,"\t R-squared = %lg\n",Rsq);
  fprintf(fp1,"\t smoothing params = %lg , %lg\n",*smoothp,*(smoothp+1));
  fprintf(fp1,"\t enorm of model params = %lg \n",enorm(param));
  fflush(fp1);

// update smoothing parameters:
  sig2=-1.0; //need to set to -1 each time to do GCV not UBRE
  init_msrep(&msrep,m,direct_mesh);
  msctrl.conv_tol= conv_tol;

```

```

msctrl.max_step_half= ms_max_half;
msctrl.min_edf= min_edf;
msctrl.target_edf= target_edf;
printf("Calling MultiSmooth (to get smoothing params)... \n");
fprintf(fp1, "\n Doing MultiSmooth (to get smoothing params)... \n");
fflush(fp1);
if (count==3)
printf("At iteration 4\n");
smoothp[0]= -1.0;
MultiSmooth(&z, &Jz, &Cons, &w, P, &dummyspz, smoothp, off, m, &sig2,
            &msctrl, &msrep, direct_mesh, &trA);
printf(" DONE\n");
fprintf(fp1, " done MultiSmooth\n");
fflush(fp1);

// get model parameters satisfying inequality constraints:
inequalcons(paramz, smoothp, Ainlong, blong, Jz, w, z, rP);

// perform golden section search for "best" parameters:
printf("Doing golden section search... \n");
fprintf(fp1, "\n Doing golden section search\n");
fflush(fp1);
goldensmooth(oldpz, paramz, Z, P[0], smoothp, &ob);
printf("DONE\n");
// get latest update of g
adjustp(param, paramz, Z); // convert paramz->param
getJg(param, smoothp, cov, knots, g, J, w, -1, 99);

// check if convergence/divergence has occurred, if so stop iterating:
check=criteria(oldg, g);
if(check>0){
    fprintf(fp1, " after %d iterations\n", count+1);
    fflush(fp1);
    printf(" after %d iterations\n", count+1);
    if(check>0) break;
}
}
// END OF MODEL FITTING ITERATIONS
/*
*****
*           FINISH UP EVERYTHING           *
*****
*/
// convert paramz->param, get model estimates and write output:
printf("Getting final fitted values... \n");
adjustp(param, paramz, Z);
getJg(param, smoothp, cov, knots, g, J, w, -1, count+1);
// uncomment if writing first "yparam0":
// getJg(param, smoothp, cov, knots, g, J, w, -1, 0);
if((fp2=fopen("output", "a"))==NULL){
    printf("Error opening file \"output\" \n");
    exit(1);
}
}

```

```
printf("DONE\nCalling sumbirth ..... \n");
Rsq = rsq(y, g, np);
sumbirth(param, knots, GRIDSIZE, ans);
printf("DONE\n");
fprintf(fp2, "%lg\t%lg\t%d", Rsq, ob, count);
for(i=0; i<ans.r; i++){
    fprintf(fp2, "\t%lg", ans.V[i]);
}
fprintf(fp2, "\n");
fclose(fp2);
freemat(cov);
freemat(covlong);
freemat(knots);
freemat(y);
freemat(param);
freemat(paramz);
freemat(dummypz);
freemat(oldpz);
freemat(Z);
freemat(Q);
freemat(T);
freemat(T2);
freemat(*P);
freemat(*(P+1));
freemat(J);
freemat(Jz);
freemat(z);
freemat(Jpz);
freemat(g);
freemat(w);
freemat(E);
freemat(Ez);
freemat(Cons);
freemat(Initial);
freemat(ages);
freemat(Ain);
freemat(b);
freemat(Ainlong);
freemat(blong);
freemat(resid1);
freemat(resid2);
freemat(ans);
free(P);
free(off);
free(oldsp);
free(smoothp);
return;
}

void main(void)
/* {main} simply calls the function {fit}
*/
{
```

```
    fit();  
}
```

C.2 Programs used in the application of the SPSP model

The following R routines were used to manipulate the simulated and sole egg data as explained in chapter 3. Specifically, the programs obtain the age limits for the simulated data; add noise to the simulated 'true' data; translate the sole data sample locations from degrees to kilometres from 0°N and from 0°E; give the sample points for the sole data the same range in each dimension and make them symmetric in range about zero; and create bootstrap re-samples to which the SPSP model was fitted for confidence interval estimation. The routine `converttokm.f` is an adaption of a function `greatcircledist` supplied to me by Charles Paxton.

```
# agelims.f calculates the age limits for time t and stage stage  
# for simulated data  
agelims.f <- function(t,stage){  
  ans <- (t^2)/1600 + stage*1.8  
  ans  
}  
  
# addnoise adds noise to simulated data  
# arguments: g is non-noisy data, percent is amount of noise  
# when distribution is normal, type is n for normal, p for poisson  
addnoise.f <- function(g,percent,type){  
  distn <- pmatch(c("n","p"),type)  
  if(is.na(distn[1])==FALSE){  
    noise <- rnorm(length(g),0,g*percent)  
    noisyg <- noise+g  
  }  
  else if(is.na(distn[2])==FALSE){  
    noisyg <- rpois(length(g),g)  
  }  
  noisyg  
}  
  
# greatcircledist.f calculates the great circle distance from x to y,  
# converting the answer to kilometres
```

```
# It hasn't been tried over the equator
# x and y must be dataframes with [,1]=long, [,2]=lat
# R-trigonometry is in radians
greatcircledist.f <- function(x,y){
  cosdist <- sin(pi*x[,2]/180)*sin(pi*y[,2]/180)
    + cos(pi*x[,2]/180)*cos(pi*y[,2]/180)*cos(pi*abs(x[,1]-y[,1])/180)
  dist <- acos(cosdist)
  dist <- dist*180/pi # get back to degrees from radians
  dist <- dist*60     # convert to nautical miles
  dist <- dist*1.852  # convert to km
  dist
}

# converttokm.f converts the spatial variates (in degrees) to
# flat distance (km) from GMT and the equator
# data is a data frame with columns (long,lat,time,population counts),
# knots is a data frame with columns (long,lat,time),
# coastline is a data frame with (long,lat) columns,
# directory is a character string of the address for writing files
converttokm.f <- function(data,knots,coastline,directory){
  datakm <- data
  knotskm <- knots
  coastkm <- coastline
  gmt <- data[,1:2]
  gmt[,1] <- 0
  equator <- data[,1:2]
  equator[,2] <- 0
  # sole data points are west of GMT so following is negative:
  datakm[,1] <- -greatcircledist.f(data[,1:2],gmt)
  datakm[,2] <- greatcircledist.f(data[,1:2],equator)
  gmt <- knots[,1:2]
  gmt[,1] <- 0
  equator <- knots[,1:2]
  equator[,2] <- 0
  knotskm[,1] <- -greatcircledist.f(knots[,1:2],gmt)
  knotskm[,2] <- greatcircledist.f(knots[,1:2],equator)
  gmt <- coastline[,1:2]
  gmt[,1] <- 0
  equator <- coastline[,1:2]
  equator[,2] <- 0
  coastkm[,1] <- -greatcircledist.f(coastline[,1:2],gmt)
  coastkm[,2] <- greatcircledist.f(coastline[,1:2],equator)
  write(t(datakm),file=paste(directory,"datakm",sep=""),ncol=4)
  write(t(knotskm),file=paste(directory,"knotskm",sep=""),ncol=3)
  write(t(coastkm),file=paste(directory,"coastkm",sep=""),ncol=3)
  return(list(datakm=datakm,knotskm=knotskm,coastkm=coastkm))
}

# scale.f scales the covariates so that they are symmetric in range
# about zero and they all have the same range.
# data is a data frame with columns (long,lat,time,population counts),
# knots is a data frame with columns (long,lat,time),
# coastline is a data frame with (long,lat) columns,
```

```

# directory is a character string of the address for writing files
scale.f <- function(data,knots,coast,directory){
  ranget <- max(data[,3]) - min(data[,3])
  rangex1 <- max(data[,1]) - min(data[,1])
  rangex2 <- max(data[,2]) - min(data[,2])
  scaledat <- data
  scaleknots <- knots
  scalecoast <- coast
  # centre about zero:
  scaledat[,3] <- data[,3] - ranget/2 - min(data[,3])
  scaleknots[,3] <- knots[,3] - ranget/2 - min(data[,3])
  scaledat[,1] <- data[,1] - rangex1/2 - min(data[,1])
  scaleknots[,1] <- knots[,1] - rangex1/2 - min(data[,1])
  scalecoast[,1] <- coast[,1] - rangex1/2 - min(data[,1])
  scaledat[,2] <- data[,2] - rangex2/2 - min(data[,2])
  scaleknots[,2] <- knots[,2] - rangex2/2 - min(data[,2])
  scalecoast[,2] <- coast[,2] - rangex2/2 - min(data[,2])
  # give things the same range as t:
  scaledat[,1] <- scaledat[,1]*ranget/rangex1
  scaleknots[,1] <- scaleknots[,1]*ranget/rangex1
  scalecoast[,1] <- scalecoast[,1]*ranget/rangex1
  scaledat[,2] <- scaledat[,2]*ranget/rangex2
  scaleknots[,2] <- scaleknots[,2]*ranget/rangex2
  scalecoast[,2] <- scalecoast[,2]*ranget/rangex2
  write(t(scaledat),file=paste(directory,"datascale",sep=""),ncol=4)
  write(t(scaleknots),file=paste(directory,"knotsscale",sep=""),ncol=3)
  write(t(scalecoast),file=paste(directory,"coastscale",sep=""),ncol=2)
  return(scaledat)
}

# bootdata.f creates non-parametric re-samples of the data
# b is the number of re-samples,
# cov are the sample points, a data frame (long, lat, time)
# data is a data frame with columns (long,lat,time,population counts),
# ages is a data frame of stage age limits (rows correspond to rows of cov),
# directory is a character string of the address for writing files
# start is index start number
bootdata.f <- function(b,cov,data,ages,directory,start=1){
  file.remove(paste(directory,"boot.ind",sep=""))
  n <- length(cov[,1])
  C <- length(ages[1,])-1
  for(j in start:(b+start-1)){
    ind <- sample(1:n,n,replace=T)
    newdata <- NULL
    for(i in 1:C){
      newdata <- rbind(newdata,data[ind+(i-1)*n,])
    }
    newages <- ages[ind,]
    write(t(newdata),file=paste(directory,"datasim",j,sep=""),ncol=4)
    write(t(newages),file=paste(directory,"agessim",j,sep=""),ncol=C+1)
    write(t(ind),file=paste(directory,"boot.ind",sep=""),append=T,ncol=n)
  }
}

```

Appendix D

Results of the application of a smooth physiologically-structured population model

The following table lists the results obtained for the SPSP model fitted to the 100 sets of simulated normally distributed data with 10% noise. The data are described in chapter 3.

Index	R-sq	Objective function	Steps to converge	Total birth estimate	90% Confidence interval	Bias
Original				13,162,343		
No noise	1	0.004249	4	13 162 300		
10	0.9467	61.06	5	12 286 200	(11 980 800, 12 941 800)	-0.0665644
49	0.9608	59.04	43	12 492 400	(12 439 200, 13 202 300)	-0.0508985
44	0.9514	56.41	4	12 513 000	(12 099 400, 13 141 000)	-0.0493334
6	0.9513	61.33	5	12 513 300	(12 122 500, 13 280 900)	-0.0493106
97	0.9651	55.01	74	12 528 700	(12 132 200, 13 101 600)	-0.0481406
42	0.9562	65.74	49	12 531 000	(12 308 200, 13 095 600)	-0.0479659

continued on next page

D. Results of the application of the SPSP model

continued from previous page

Index	R-sq	Objective function	Steps to converge	Total birth estimate	90% Confidence interval	Bias
76	0.9591	69.11	3	12 581 100	(11 929 300, 13 015 300)	-0.0441595
7	0.9601	61.17	3	12 605 700	(12 470 600, 13 301 400)	-0.0422906
85	0.9566	60.01	4	12 633 300	(12 441 400, 13 294 400)	-0.0401937
18	0.9481	66.08	3	12 652 000	(12 209 700, 13 133 400)	-0.0387730
14	0.9601	62.53	4	12 672 100	(12 315 100, 13 477 100)	-0.0372459
30	0.9483	58.91	3	12 677 600	(12 371 300, 13 222 100)	-0.0368280
4	0.9655	56.99	101	12 704 700	(12 394 900, 13 155 700)	-0.0347691
50	0.9680	51.56	73	12 718 900	(12 510 200, 13 174 000)	-0.0336903
32	0.9525	57.38	4	12 721 400	(12 617 300, 13 295 400)	-0.0335003
28	0.9587	53.60	8	12 727 800	(12 586 300, 13 364 200)	-0.0330141
75	0.9526	61.44	78	12 731 100	(12 389 900, 13 531 400)	-0.0327634
13	0.9659	53.48	4	12 736 100	(12 673 100, 13 329 400)	-0.0323835
62	0.9653	53.55	3	12 737 000	(12 512 800, 13 117 500)	-0.0323151
64	0.9545	58.08	40	12 737 200	(12 392 500, 13 424 900)	-0.0322999
80	0.9464	63.36	3	12 739 000	(12 442 300, 13 247 100)	-0.0321632
70	0.9486	58.86	3	12 739 200	(12 255 600, 12 953 500)	-0.0321480
83	0.9470	58.88	4	12 761 600	(12 429 000, 13 382 700)	-0.0304462
71	0.9593	55.40	3	12 764 800	(12 561 800, 13 504 400)	-0.0302031
46	0.9528	56.38	8	12 769 700	(12 612 200, 13 274 800)	-0.0298308
58	0.9622	55.91	22	12 793 100	(12 532 000, 13 550 500)	-0.0280530
26	0.9371	65.57	4	12 802 200	(12 671 500, 13 555 700)	-0.0273616
52	0.9541	57.94	6	12 813 000	(12 637 200, 13 331 100)	-0.0265411
34	0.9439	61.59	4	12 816 900	(12 451 500, 13 336 400)	-0.0262448
63	0.9618	53.99	4	12 817 700	(12 532 800, 13 166 700)	-0.0261840
95	0.9623	55.72	61	12 825 100	(12 526 200, 13 522 900)	-0.0256218
87	0.9619	58.96	3	12 843 000	(12 583 000, 13 506 400)	-0.0242619
12	0.9499	56.98	4	12 845 000	(12 686 900, 13 445 100)	-0.0241099
79	0.9577	57.09	3	12 856 700	(12 383 700, 13 176 400)	-0.0232210
89	0.9412	62.60	3	12 860 100	(12 480 300, 13 389 700)	-0.0229627
8	0.9571	59.34	125	12 864 000	(12 655 100, 13 393 000)	-0.0226664
33	0.9495	62.09	6	12 870 200	(12 275 400, 13 142 400)	-0.0221954
86	0.9528	57.45	3	12 876 500	(12 490 800, 13 318 100)	-0.0217167
65	0.9422	62.60	6	12 893 000	(12 588 900, 13 550 500)	-0.0204632
37	0.9325	64.23	3	12 898 400	(12 559 100, 13 476 400)	-0.0200529
2	0.9591	54.66	4	12 901 700	(12 688 600, 13 415 900)	-0.0198022

continued on next page

D. Results of the application of the SPSP model

continued from previous page

Index	R-sq	Objective function	Steps to converge	Total birth estimate	90% Confidence interval	Bias
84	0.9362	66.33	4	12 910 100	(12 519 500, 13 345 800)	-0.0191640
38	0.9412	62.56	5	12 912 400	(12 529 600, 13 283 600)	-0.0189892
16	0.9375	69.53	7	12 915 000	(12 341 800, 13 640 300)	-0.0187917
17	0.9495	57.18	5	12 918 300	(12 689 800, 13 665 900)	-0.0185410
43	0.9577	58.45	5	12 919 500	(12 545 100, 13 565 800)	-0.0184498
90	0.9442	61.74	5	12 928 500	(12 583 400, 13 478 800)	-0.0177661
9	0.9530	58.57	4	12 941 200	(12 761 600, 13 456 400)	-0.0168012
77	0.9547	60.00	4	12 941 800	(12 618 600, 13 370 600)	-0.0167556
98	0.9536	60.95	84	12 952 700	(12 789 500, 13 629 700)	-0.0159275
60	0.9357	67.16	4	12 954 000	(12 491 600, 13 542 100)	-0.0158287
57	0.9532	63.70	3	12 960 000	(12 487 000, 13 348 000)	-0.0153729
81	0.9510	60.38	3	12 964 900	(12 343 500, 13 305 800)	-0.0150006
74	0.9433	65.74	3	12 967 400	(12 516 100, 13 552 200)	-0.0148107
39	0.9584	57.38	4	12 969 900	(12 674 700, 13 485 200)	-0.0146207
47	0.9491	61.64	4	12 973 300	(12 524 600, 13 310 100)	-0.0143624
3	0.9516	57.69	4	12 974 400	(12 380 300, 13 102 700)	-0.0142788
40	0.9484	65.41	3	12 998 800	(12 472 800, 13 485 600)	-0.0124251
56	0.9494	62.50	8	13 020 800	(12 514 800, 13 564 600)	-0.0107536
78	0.9554	62.92	3	13 021 600	(12 758 100, 13 569 400)	-0.0106929
92	0.9493	57.44	3	13 034 800	(12 591 000, 13 315 800)	-0.0096900
51	0.9461	65.55	3	13 038 600	(12 690 700, 13 529 300)	-0.0094013
82	0.9482	64.81	4	13 051 000	(12 564 100, 13 636 300)	-0.0084592
72	0.9496	64.96	4	13 062 100	(12 522 800, 13 549 300)	-0.0076159
68	0.9661	54.88	15	13 064 300	(12 942 000, 13 653 000)	-0.0074487
23	0.9435	60.94	4	13 067 300	(12 436 900, 13 717 800)	-0.0072208
29	0.9641	57.61	3	13 077 200	(12 736 100, 13 446 500)	-0.0064687
94	0.9580	57.54	60	13 083 900	(12 868 400, 13 764 500)	-0.0059597
45	0.9521	58.14	4	13 084 500	(12 888 200, 13 728 800)	-0.0059141
55	0.9455	62.21	5	13 085 900	(12 736 500, 13 488 800)	-0.0058077
25	0.9483	64.42	3	13 086 600	(12 696 200, 13 376 900)	-0.0057545
91	0.9659	53.12	3	13 094 300	(12 612 900, 13 556 100)	-0.0051695
41	0.9614	59.57	4	13 114 400	(12 818 400, 13 448 100)	-0.0036424
66	0.9455	64.99	3	13 118 500	(12 641 300, 13 562 300)	-0.0033309
21	0.9629	59.51	4	13 129 600	(12 342 500, 13 499 700)	-0.0024876
96	0.9523	58.77	3	13 145 900	(12 672 500, 13 529 800)	-0.0012492

continued on next page

D. Results of the application of the SPSP model

continued from previous page

Index	R-sq	Objective function	Steps to converge	Total birth estimate	90% Confidence interval	Bias
15	0.9663	53.52	4	13 163 500	(12 535 300, 13 554 400)	0.0000879
59	0.9661	50.48	3	13 166 800	(12 699 400, 13 635 200)	0.0003386
19	0.9713	53.30	3	13 189 000	(12 562 800, 13 598 000)	0.0020252
11	0.9552	58.51	45	13 195 000	(12 916 000, 13 803 300)	0.0024811
54	0.9503	61.88	4	13 249 500	(12 839 900, 13 611 700)	0.0066217
88	0.9550	61.96	3	13 255 200	(12 810 100, 13 555 100)	0.0070547
35	0.9559	67.67	3	13 257 600	(12 811 900, 13 866 100)	0.0072371
5	0.9592	62.04	3	13 267 600	(12 843 100, 13 687 000)	0.0079968
1	0.9583	57.43	6	13 284 600	(12 865 000, 13 951 600)	0.0092884
73	0.9743	45.54	4	13 286 300	(13 014 200, 13 641 200)	0.0094175
27	0.9589	58.38	44	13 286 400	(13 049 800, 13 981 200)	0.0094251
100	0.9504	59.75	4	13 291 400	(12 873 500, 13 573 600)	0.0098050
67	0.9507	58.05	4	13 301 600	(12 803 400, 13 537 100)	0.0105800
99	0.9512	66.97	3	13 311 600	(12 770 800, 13 830 600)	0.0113397
48	0.9614	63.27	4	13 317 800	(12 683 200, 13 856 100)	0.0118107
22	0.9355	73.99	5	13 318 200	(12 759 600, 13 838 800)	0.0118411
93	0.9483	61.69	3	13 327 600	(12 856 000, 14 116 000)	0.0125553
61	0.9799	40.75	4	13 360 600	(12 485 900, 13 471 200)	0.0150624
24	0.9461	63.95	4	13 365 900	(12 947 300, 13 977 300)	0.0154651
53	0.9481	61.60	4	13 388 900	(13 128 000, 13 881 300)	0.0172125
36	0.9576	62.99	3	13 419 900	(12 959 300, 13 894 500)	0.0195677
20	0.9492	65.69	5	13 426 900	(12 940 800, 13 844 300)	0.0200995
31	0.9575	63.66	4	13 448 100	(12 919 000, 13 865 100)	0.0217102
69	0.9522	64.28	6	13 618 400	(12 901 300, 14 009 100)	0.0346486

Table D.1: Results of simulation study for the normal data with 10% CV with estimates of the total number of individuals born (P_0) in ascending order. For comparison, the first row of the table contains the results for the model fitted to the original data without noise.

Appendix E

Simulating Data Using the McKendrick von Foerster Equation

Simulated data were required to demonstrate the existence of plausible alternative birth, growth and death rates which all solve the McKendrick von Foerster equation 4.9 in chapter 4. The demonstration is given in section 4.4.4.

The task of simulating population densities is the opposite of the inverse problem in that population densities are produced from a pre-defined growth rate $g(m, t)$, death rate $\mu(m, t)$, birth rate $R(t)$ and initial size distribution $\eta(m, t_0)$. Given these functions, it is sometimes possible to solve equation 4.9 analytically for the population density $\eta(m, t)$, using the method of characteristics. This method will produce a set of ordinary differential equations in a dummy variable s involving $\eta(m, t) = \eta(s)$, $g(m, t) = g(s)$ and $\mu(m, t) = \mu(s)$. The set of equations may be solvable depending on the form of these functions. However, this analytic method is not practical when we wish to simulate data from many different sets of vital rates.

For each different choice, we would need to solve a different set of characteristic equations and in some cases, an analytic solution might not be possible. Instead, equation 4.9 can be approximated and solved numerically.

We can discretize equation 4.9 by slicing the population density function η into size sections using the upwind difference scheme (Gurney & Nisbet 1998, page 243). This is achieved by approximating $\eta(m, t)$ using $N + 1$ functions, $\eta_i(t) = \eta(m^*, t)$ for $i = 0, \dots, N$, where $m_{min} + i\Delta m \leq m^* \leq m_{min} + (i + 1)\Delta m$. Δm is chosen to be sufficiently small so that $\frac{\partial \eta(m^*, t)}{\partial m} \approx 0$ and N must be sufficiently large so that none of the population survives to reach size $m_{min} + N\Delta m = m_{max}$. This approximation results in the system of ordinary differential equations

$$\frac{d\eta_i}{dt} + g(m_{min} + i\Delta m, t) \left(\frac{\eta_i(t) - \eta_{i-1}(t)}{\Delta m} \right) + \eta_i(t) \frac{\partial g}{\partial m} + \mu(m_{min} + i\Delta m, t)\eta_i(t) = 0$$

$$i = 1, \dots, N \quad (\text{E.1})$$

The scheme is illustrated in figure E.1. The derivative $\frac{d\eta_0}{dt}$ is not required for the estimation of $\eta_0(t)$ since $\eta_0(t) = \frac{R(t)}{g(m_{min}, t)}$.

Although the upwind difference scheme involves loss of accuracy by discretizing the population density function in the size dimension, the system of ordinary differential equations it produces can be used for finding $\eta(m, t)$ given any choice of vital rates. The system can be solved analytically or numerically, but to maintain generality of application, a numerical method - the midpoint method - is used here.

The midpoint method (Press, Flannery, Teukolsky & Vetterling 1988, pp 610–611) is used to solve each of the equations in the system E.1 separately. For each $i = 1, \dots, N$, we need

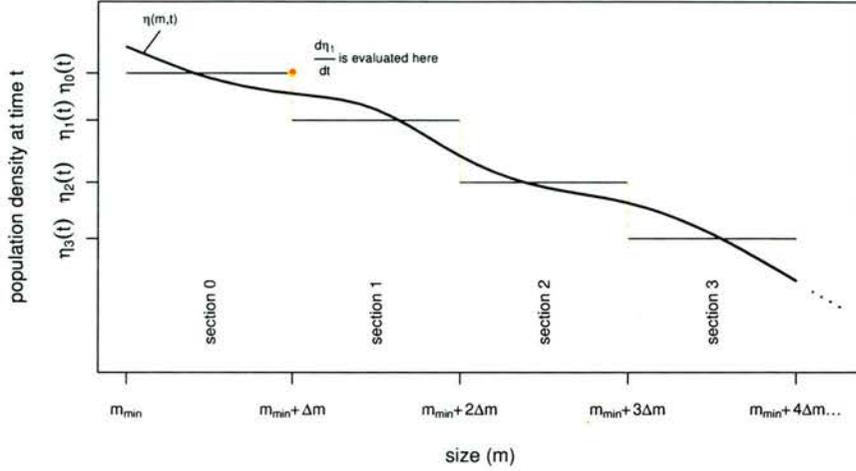


Figure E.1: Illustration of the approximation of the population density function $\eta(m, t)$ at time t using the upwind difference scheme (see equation E.1).

to know the growth rate ($g(m_{\min} + i\Delta m, t)$), death rate ($\mu(m_{\min} + i\Delta m, t)$) and initial size distribution ($\eta_i(t_0)$). A time-step Δt is chosen to be sufficiently small for obtaining a good discrete estimate of the smooth function η_i of time. The midpoint method is usually used to estimate the value of $\eta_i(t_j)$ at $t_j = \max(t)$, but for the purpose of generating the grid surface points $\eta_i(t_j)$ for $t_j \in \{t_0, t_0 + \Delta t, t_0 + 2\Delta t, \dots, t_0 + M\Delta t = \max(t)\}$, the value of $\eta_i(t_j)$ will be recorded at each intermediary step.

Firstly, the gradient $\frac{d\eta_i}{dt}$ is evaluated at $\eta_i(t_0)$, and this is used to take a half time-step to $\eta_i(t_0 + 0.5\Delta t)$. The gradient at the midpoint time $\frac{d\eta_i}{dt}(t_0 + 0.5\Delta t)$ is then estimated and used to get from η_i at the start point t_0 to η_i at the next time point, $(t_0 + \Delta t)$. This value is recorded and the process is repeated until finally $\eta(\max(t))$ is reached. The method is illustrated in figure E.2.

Originally functions were evaluated at $N + 1 = 1001$ different sizes ranging from $m_{\min} =$

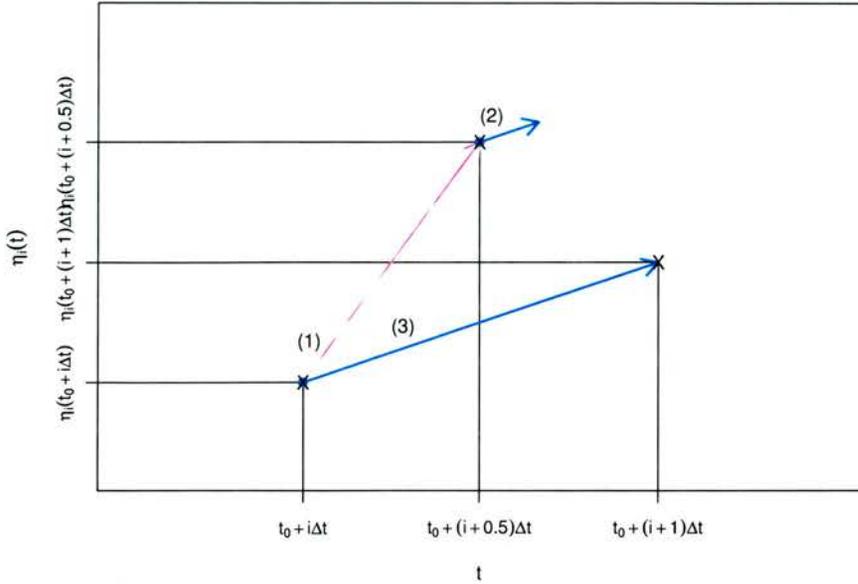


Figure E.2: One step of the midpoint method: (1) calculate $\frac{d\eta_i}{dt}(t_0 + i\Delta t)$ and use this gradient to find $\eta_i(t_0 + (i + 0.5)\Delta t)$, (2) calculate $\frac{d\eta_i}{dt}(t_0 + (i+0.5)\Delta t)$, (3) use the gradient from (2) to find $\eta_i(t_0 + (i + 0.5)\Delta t)$.

$120\mu\text{m}$ to $m_{max} = 2603\mu\text{m}$ (obtained from Campbell et al. (2001)), and $M = 10\,000$ different times from day $t_0 = 0$ to day $\max(t) = 365$, so $\Delta m = 2.483\mu\text{m}$ and $\Delta t = 0.0365$ days. For the simulated data set, some intermediary steps were not recorded to save disk space, resulting in 101 different size sections and 371 different times being used for obtaining the graphs in figures 4.4 and 4.5.

Alternative growth, death and birth rates were found using the methods described in section 4.4.4.