# Random Regression Models

## L. R. Schaeffer

## 2016

# Acknowledgements

The idea for analyzing test day records instead of 305-d records came from Dr John Moxley in 1974. Ewa Ptak helped me with my first analyses of test day records around 1992. An early application was to dairy records of Canadian goats with Brian Sullivan. Jack Dekkers put me onto the phrase “random regressions”, opening my eyes to the possibilities. Janusz Jamrozik contributed greatly to the implementation of random regression models and exploring many of the thorny issues. Then there was Karin Meyer and Bill Hill relating random regressions to covariance functions and showing how the covariance function matrices could be estimated with REML. Reinhard Reents and Gerrit Kistemaker participated in random regression model analyses of dairy cattle production and somatic cell score data at Guelph. Julius van der Werf co-taught a course with me on random regressions. I was pleased at how readily the Canadian dairy industry adopted the test day model, and I was surprised at how readily random regression models influenced research in animal breeding at many institutions around the world in the time after the 1994 WCGALP presentations. Hector Uribe studied random regression models using US type classification data. Fiona Galbraith applied random regression models to survival data, much like what is presented in this book. It is interesting how one thought ripples through time and through the lives of so many people. There have been many others that have subsequently shaped my current understanding of this and other problems. I extend my thanks to everyone above, and those I have inadvertently omitted.

# Contents

# Chapter 1

# Longitudinal Data

## 1.1 Introduction

A simple example of *longitudinal* data is the weight of an animal taken at different ages. Meat animals, like beef cattle, pigs, and sheep, are weighed two or three times from birth to market age, generally at birth, at weaning, and at market age. Weighing animals takes time and labour. Birth is always day 1, but weaning and market ages are not the same for every animal. Weights get larger over time because animals grow, and the variance of weights also increases with age.

Another example is the lactation yield of dairy cows, sheep, or goats. Dairy animals are milked two or more times daily for up to a year after they give birth. Typically, 24-h production increases shortly after the animal gives birth, peak at a few weeks after parturition, then slowly decreases until the animal dries up in preparation for the next parturition. Milk recording programs send supervisors to herds once a month or less frequently to weigh the milk and take samples for lab analyses of content. Thus, an animal might give milk for over 300 days, but there might only be seven to ten supervised weighings during that period. Herds with robotic milking machines can have daily weighings.

Traits measured at various times during the life of an animal are known as *longitudinal* data. Because the weights or yields occur at different ages or times, they are not the same trait. Weights at birth and weaning may have a positive correlation, but it is less than unity. Milk weights at day 10 and day 300 may also be correlated, but again that correlation is much less than unity. Thus, the weight of an animal on every day of life is a 'different' trait. Every milk weight from the start of lactation to the end is a 'different' trait. There is a continuum of points in time when an animal could be observed for a trait. These traits have also been called *infinitely dimensional* traits.

Instead of age or time, observations could be based on degree of maturity or weight. For example, fat content of an animal would change depending on an animal's weight or amount of feed ingested, regardless of age.

In general, there is a starting point, $t_{min}$, e.g. birth or parturition, at which observations start to be taken. The observations are made either at specific intervals or at random intervals, and the number of observations can vary from animal to animal. Then there is the end point, $t_{max}$, beyond which no more observations are made, or are not of interest. Each observation, $y_{ti}$, has an associated $t_i$. For simplicity, $t_i$, are whole integer numbers. There could be a dozen or so points, or there could be 400 points. The number depends on the trait and situation.

Orthogonal polynomials have been suggested for use with longitudinal data to model the shape of a growth curve or a lactation curve. The reason being that orthogonal polynomials would be less correlated to each other than would be the correlation between polynomials of age. One simple type of orthogonal polynomials are Legendre polynomials, discovered in 1797. In order to use Legendre polynomials or other kinds of orthogonal polynomials, the time values (whole integer numbers) must be scaled to range from -1 to +1. The scaling formula is

$$q_i = -1 + 2\left(\frac{t_i - t_{min}}{t_{max} - t_{min}}\right).$$

The $q_i$ are decimal numbers.

Plotting $y_{ti}$ against $t_i$ (or against $q_i$) gives a shape that is called the *trajectory*. This could be a lactation curve, or a growth curve, or an S-curve. The goal is to find a function that fits this trajectory as closely as possible and to study the amount of animal variation around the trajectory from $t_{min}$ to $t_{max}$. This type of study involves covariance functions and random regression models.

Covariance functions help to predict the change in variation from $t_{min}$ to $t_{max}$ for the population. Random regression models provide a way to estimate covariance functions, and to determine individual differences in trajectories.

## 1.2 Collect Data

The first step in the study of longitudinal data, is to collect data. In order to illustrate a few basic concepts, consider the following experiment. Two hundred female mice were sampled every hour after an injection with glucose to observe the change in blood insulin levels over the next nine hours. This gave a total of 1800 observations, on two hundred unrelated individuals. A small sample of the data are shown in Table 1.1.

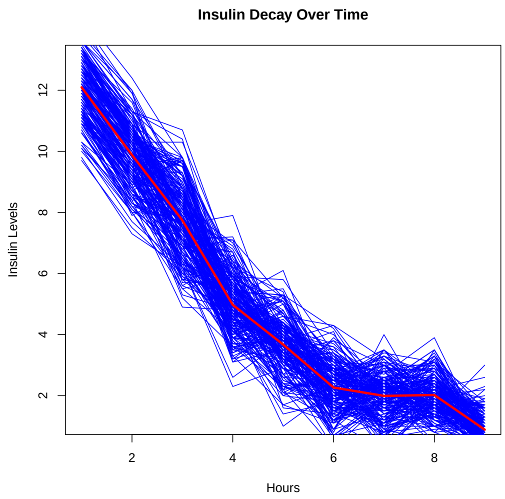
**Table 1.1**

Insulin levels in female mice.

| Mouse | Time After Injection of Glucose, min | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 60 | 120 | 180 | 240 | 300 | 360 | 420 | 480 | 540 |
| 1 | 11.9 | 9.7 | 8.7 | 4.5 | 5.3 | 1.9 | 2.3 | 1.6 | 1.0 |
| 2 | 12.9 | 10.0 | 7.5 | 3.3 | 1.7 | 2.3 | 2.3 | 2.1 | 0.5 |
| 3 | 12.2 | 10.0 | 6.0 | 4.2 | 4.4 | 2.7 | 2.2 | 2.9 | 0.2 |
| 4 | 12.6 | 10.1 | 9.5 | 5.9 | 5.8 | 3.4 | 0.9 | 0.5 | 0.7 |
| 5 | 12.7 | 10.5 | 8.2 | 5.4 | 4.7 | 2.1 | 1.9 | 3.2 | 0.2 |

A plot of all 200 mouse insulin decay trajectories are shown in Figure 1.1.

**Figure 1.1**



Note the general shape of the decay trajectory for all mice. Also, note the variability that exists around the average curve (shown in red). Thus, mice have different decay trajectories. We want to study the variability between mice.

Using data on the 200 mice the covariance matrix of the insulin amounts at each hour (a 9 × 9 matrix) can be calculated as follows.

$$\mathbf{V} = \begin{pmatrix} 0.8852 & 0.8352 & 0.6916 & 0.0350 & 0.1089 & -0.0050 & -0.0417 & -0.0476 & -0.0156 \\ 0.8352 & 0.9574 & 0.4621 & 0.0399 & 0.1607 & -0.0281 & -0.0433 & -0.0203 & 0.0197 \\ 0.6916 & 0.4621 & 1.4005 & 0.1154 & 0.1556 & 0.0270 & -0.1918 & -0.1106 & -0.0052 \\ 0.0350 & 0.0399 & 0.1154 & 0.9331 & 0.0454 & -0.0723 & -0.0362 & -0.0030 & 0.0117 \\ 0.1089 & 0.1607 & 0.1556 & 0.0454 & 0.7993 & 0.1204 & -0.0518 & -0.0414 & 0.0558 \\ -0.0050 & -0.0281 & 0.0270 & -0.0723 & 0.1204 & 0.6833 & 0.0015 & -0.0086 & -0.0211 \\ -0.0417 & -0.0433 & -0.1918 & -0.0362 & -0.0518 & 0.0015 & 0.5807 & 0.0474 & -0.0035 \\ -0.0476 & -0.0203 & -0.1106 & -0.0030 & -0.0414 & -0.0086 & 0.0474 & 0.4409 & -0.0291 \\ -0.0156 & 0.0197 & -0.0052 & 0.0117 & 0.0558 & -0.0211 & -0.0035 & -0.0291 & 0.2855 \end{pmatrix}$$

$$\mathbf{V} = \{\sigma_{t_i,t_j}\}.$$

This matrix is automatically a positive definite matrix by virtue of the way it was calculated, but a good practice is to always check each matrix. The eigenvalues ($EV$) were all positive, as shown below, and therefore $\mathbf{V}$ is positive definite.

$$EV = \begin{pmatrix} 2.48452319 \\ 0.96233482 \\ 0.86962324 \\ 0.80221637 \\ 0.59747351 \\ 0.50820761 \\ 0.42359840 \\ 0.27266640 \\ 0.04525645 \end{pmatrix}.$$

## 1.3 Covariance Functions

Kirkpatrick et al.(1991) proposed the use of covariance functions for longitudinal data. A covariance function (CF) is a way to model the variances and covariances of a longitudinal trait. Orthogonal polynomials are used in this model and Legendre polynomials are the easiest to apply. Each element of $\mathbf{V}$ is modeled as

$$\sigma_{t_i,t_j} = \phi(q_i)'\mathbf{K}\phi(q_j),$$

where $\phi(q_i)$ is a vector of functions of time, $q_i$, and $\mathbf{K}$ is a matrix of constants necessary to estimate $\sigma_{t_i,t_j}$. The matrix $\mathbf{K}$ must be estimated.

In the example, $\mathbf{V}$, above, there are $m = 9$ time periods, and therefore, there are $m*(m+1)/2 \ = \ 45$ parameters in $\mathbf{V}$. In Table 1.2 are the time variables and the standardized time values. Note that $t_{min} = 60$, and $t_{max} = 540$.

**Table 1.2**

Time Variables of Mouse Data.

| Item | $t_i$ (minutes) | $q_i$ |
|---|---|---|
| 1 | 60 | -1.00 |
| 2 | 120 | -0.75 |
| 3 | 180 | -0.50 |
| 4 | 240 | -0.25 |
| 5 | 300 | 0.00 |
| 6 | 360 | 0.25 |
| 7 | 420 | 0.50 |
| 8 | 480 | 0.75 |
| 9 | 540 | 1.00 |

Legendre polynomials, $P_k(x)$, are defined as follows, for $x$ being one of the $q_i$.

$$\begin{aligned} P_0(x) &= 1, \quad \text{and} \\ P_1(x) &= x, \end{aligned}$$

then, in general, the $(n+1)^{st}$ polynomial is described by the following recursive equation:

$$P_{n+1}(x) = \frac{1}{n+1}\left((2n+1)xP_n(x) - nP_{n-1}(x)\right).$$

These quantities are "normalized" using

$$\phi_n(x) = \left(\frac{2n+1}{2}\right)^{.5} P_n(x).$$

This gives the following series,

$$\begin{aligned} \phi_0(x) &= \left(\frac{1}{2}\right)^{.5} P_0(x) = .7071 \\ \phi_1(x) &= \left(\frac{3}{2}\right)^{.5} P_1(x) \\ &= 1.2247x \\ P_2(x) &= \frac{1}{2}(3xP_1(x) - 1P_0(x)) \\ \phi_2(x) &= \left(\frac{5}{2}\right)^{.5} (\frac{3}{2}x^2 - \frac{1}{2}) \\ &= -.7906 + 2.3717x^2, \end{aligned}$$

and so on.

Because **V** is 9×9, then to model all of the $\sigma_{t_i,t_j}$ we need 9 orthogonal polynomials. Thus, we need Legendre polynomials of order 8, where 8 is the highest order of polynomials of time. Order of 8 means there are 9 covariables (including time to the power of 0).

**Table 1.3**
Legendre Polynomials of Order 8.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.7071 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 1.2247 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | -0.7906 | 0.0 | 2.3717 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | -2.8062 | 0.0 | 4.6771 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.7955 | 0.0 | -7.9550 | 0.0 | 9.2808 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 4.3973 | 0.0 | -20.5206 | 0.0 | 18.4685 | 0.0 | 0.0 | 0.0 |
| 6 | -0.7967 | 0.0 | 16.7312 | 0.0 | -50.1935 | 0.0 | 36.8086 | 0.0 | 0.0 |
| 7 | 0.0 | -5.9907 | 0.0 | 53.9164 | 0.0 | -118.6162 | 0.0 | 73.4291 | 0.0 |
| 8 | 0.7972 | 0.0 | -28.6992 | 0.0 | 157.8457 | 0.0 | -273.5992 | 0.0 | 146.571 |

A simple R function that will give you the above table follows.

```
LPOLY = function(no) {
  if(no > 9 ) no = 9
  nom = no - 1
  phi = matrix(data=c(0),nrow=9,ncol=9)
  phi[1,1]=1
  phi[2,2]=1
 for(i in 2:nom){
   ia = i+1
   ib = ia - 1
   ic = ia - 2
   c = 2*(i-1) + 1
   f = i - 1
   c = c/i
   f = f/i
 for(j in 1:ia){
   if(j == 1){ z = 0 }
   else {z = phi[ib,j-1]}
   phi[ia,j] = c*z - f*phi[ic,j]
  }
 }
```

```
    for( m in 1:no){
       f = sqrt((2*(m-1)+1)/2)
       phi[m, ] = phi[m, ]*f
     }
   return(phi[1:no,1:no])
  }
```

Let the matrix in Table 1.3 be denoted as $\Lambda'$. Now define another matrix, **M**, as a matrix containing the polynomials of standardized time values.

Therefore,

$$
\begin{aligned}
\mathbf{M} &= \left\{ \begin{array}{ccccccccc} 1 & q_i & q_i^2 & q_i^3 & q_i^4 & q_i^5 & q_i^6 & q_i^7 & q_i^8 \end{array} \right\} \\
&= \left( \begin{array}{rrrrrr}
1 & -1.00 & 1.0000 & -1.000000 & 1.00000000 & -1.0000000000 \\
1 & -0.75 & 0.5625 & -0.421875 & 0.31640625 & -0.2373046875 \\
1 & -0.50 & 0.2500 & -0.125000 & 0.06250000 & -0.0312500000 \\
1 & -0.25 & 0.0625 & -0.015625 & 0.00390625 & -0.0009765625 \\
1 & 0.00 & 0.0000 & 0.000000 & 0.00000000 & 0.0000000000 \\
1 & 0.25 & 0.0625 & 0.015625 & 0.00390625 & 0.0009765625 \\
1 & 0.50 & 0.2500 & 0.125000 & 0.06250000 & 0.0312500000 \\
1 & 0.75 & 0.5625 & 0.421875 & 0.31640625 & 0.2373046875 \\
1 & 1.00 & 1.0000 & 1.000000 & 1.00000000 & 1.0000000000
\end{array} \right. \\
&\qquad \left. \begin{array}{rrr}
1.000000 & -1.0000000 & 1.000000 \\
0.177979 & -0.1334839 & 0.100113 \\
0.015625 & -0.0078125 & 0.003906 \\
0.000244 & -0.0000610 & 0.000015 \\
0.000000 & 0.0000000 & 0.000000 \\
0.000244 & 0.0000610 & 0.000015 \\
0.015625 & 0.0078125 & 0.003906 \\
0.177979 & 0.1334839 & 0.100113 \\
1.000000 & 1.0000000 & 1.000000
\end{array} \right).
\end{aligned}
$$

This gives

$$
\begin{aligned}
\Phi &= \mathbf{M}\Lambda, \\
&= \left( \begin{array}{rrrrr}
0.7071068 & -1.2247449 & 1.5811388 & -1.8708287 & 2.1213203 \\
0.7071068 & -0.9185587 & 0.5435165 & 0.1315426 & -0.7426693 \\
0.7071068 & -0.6123724 & -0.1976424 & 0.8184876 & -0.6131942 \\
0.7071068 & -0.3061862 & -0.6423376 & 0.6284815 & 0.3345637 \\
0.7071068 & 0.0000000 & -0.7905694 & 0.0000000 & 0.7954951 \\
0.7071068 & 0.3061862 & -0.6423376 & -0.6284815 & 0.3345637 \\
0.7071068 & 0.6123724 & -0.1976424 & -0.8184876 & -0.6131942 \\
0.7071068 & 0.9185587 & 0.5435165 & -0.1315426 & -0.7426693 \\
0.7071068 & 1.2247449 & 1.5811388 & 1.8708287 & 2.1213203
\end{array} \right.
\end{aligned}
$$

$$\left.\begin{array}{rrrr}
-2.3452079 & 2.54950976 & -2.73861279 & 2.9154759 \\
0.9765020 & -0.71584364 & 0.09361538 & 0.5761252 \\
-0.2107023 & 0.82410911 & -0.61110647 & -0.2146925 \\
-0.7967180 & 0.06189377 & 0.76658885 & -0.4444760 \\
0.0000000 & -0.79672180 & 0.00000000 & 0.7972005 \\
0.7967180 & 0.06189377 & -0.76658885 & -0.4444760 \\
0.2107023 & 0.82410911 & 0.61110647 & -0.2146925 \\
-0.9765020 & -0.71584364 & -0.09361538 & 0.5761252 \\
2.3452079 & 2.54950976 & 2.73861279 & 2.9154759
\end{array}\right),$$

which can be used to specify the elements of $\mathbf{V}$ as

$$\begin{aligned}
\mathbf{V} &= \Phi\mathbf{K}\Phi' \\
&= \mathbf{M}(\Lambda\mathbf{K}\Lambda')\mathbf{M}' \\
&= \mathbf{MHM}'.
\end{aligned}$$

Note that $\Phi$, $\mathbf{M}$, and $\Lambda$ are matrices defined by the Legendre polynomial functions and by the standardized time values and do not depend on the data or values in the matrix $\mathbf{V}$. One can estimate either $\mathbf{K}$ or $\mathbf{H}$,

Estimate $\mathbf{K}$ using

$$\begin{aligned}
\mathbf{K} &= \Phi^{-1}\mathbf{V}\Phi^{-T}, \\
&= \left(\begin{array}{rrrrr}
0.51696 & -0.10761 & 0.51424 & 0.01598 & 0.34633 \\
-0.10761 & 0.35040 & -0.00961 & 0.02374 & 0.04881 \\
0.51424 & -0.00961 & 1.20888 & -0.10372 & 0.66167 \\
0.01598 & 0.02374 & -0.10372 & 0.32922 & -0.05996 \\
0.34633 & 0.04881 & 0.66167 & -0.05996 & 0.59757 \\
-0.03278 & 0.02711 & -0.07196 & -0.00723 & -0.04682 \\
-0.13492 & -0.02044 & -0.28443 & 0.05744 & -0.24611 \\
0.02114 & -0.12437 & 0.13495 & -0.24125 & 0.10218 \\
-0.49907 & -0.05387 & -1.02176 & 0.06434 & -0.70061
\end{array}\right.
\end{aligned}$$

$$\left.\begin{array}{rrrr}
-0.03278 & -0.13492 & 0.02114 & -0.49907 \\
0.02711 & -0.02044 & -0.12437 & -0.05387 \\
-0.07196 & -0.28443 & 0.13495 & -1.02176 \\
-0.00723 & 0.05744 & -0.24125 & 0.06434 \\
-0.04682 & -0.24611 & 0.10218 & -0.70061 \\
0.12477 & 0.01695 & -0.12536 & 0.08352 \\
0.01695 & 0.17716 & -0.05564 & 0.22522 \\
-0.12536 & -0.05564 & 0.35359 & -0.12008 \\
0.08352 & 0.22522 & -0.12008 & 1.03220
\end{array}\right),$$

and estimate $\mathbf{H}$ using

$$
\begin{aligned}
\mathbf{H} &= \mathbf{M}^{-1}\mathbf{V}\mathbf{M}^{-T} \\
&= \left(\begin{array}{rrrrr}
0.7993 & 0.3759 & -16.2282 & -4.1121 & 85.2585 \\
0.3759 & 18.8458 & -17.7140 & -140.9800 & 116.0053 \\
-16.2282 & -17.7140 & 565.2491 & 145.2098 & -3353.8067 \\
-4.1121 & -140.9800 & 145.2098 & 1244.7194 & -1032.9673 \\
85.2585 & 116.0053 & -3353.8068 & -1032.9673 & 20823.0945 \\
8.2512 & 279.3968 & -275.2019 & -2605.3471 & 2024.2377 \\
-149.0388 & -221.6913 & 6130.4997 & 2044.5953 & -38807.8169 \\
-4.5415 & -157.4900 & 148.5089 & 1505.9029 & -1117.0747 \\
79.2916 & 123.1336 & -3327.8998 & -1156.0341 & 21270.5182
\end{array}\right. \\
&\qquad \left.\begin{array}{rrrr}
8.2512 & -149.0388 & -4.5415 & 79.2916 \\
279.3968 & -221.6913 & -157.4900 & 123.1336 \\
-275.2019 & 6130.4997 & 148.5089 & -3327.8998 \\
-2605.3471 & 2044.5953 & 1505.9029 & -1156.0341 \\
2024.2377 & -38807.8169 & -1117.0747 & 21270.5182 \\
5566.7003 & -4064.5381 & -3249.7079 & 2313.7405 \\
-4064.5381 & 72970.3381 & 2262.0269 & -40177.7306 \\
-3249.7079 & 2262.0269 & 1906.4858 & -1292.3600 \\
2313.7405 & -40177.7306 & -1292.3600 & 22174.7101
\end{array}\right).
\end{aligned}
$$

Note the difference in magnitude of elements in $\mathbf{K}$ compared to $\mathbf{H}$. Now calculate the correlations among the elements in the two matrices.

$$
Cor(\mathbf{K}) = \left(\begin{array}{rrrrrrrrr}
1.00 & -0.25 & 0.65 & 0.04 & 0.62 & -0.13 & -0.45 & 0.05 & -0.68 \\
-0.25 & 1.00 & -0.01 & 0.07 & 0.11 & 0.13 & -0.08 & -0.35 & -0.09 \\
0.65 & -0.01 & 1.00 & -0.16 & 0.78 & -0.19 & -0.61 & 0.21 & -0.91 \\
0.04 & 0.07 & -0.16 & 1.00 & -0.14 & -0.04 & 0.24 & -0.71 & 0.11 \\
0.62 & 0.11 & 0.78 & -0.14 & 1.00 & -0.17 & -0.76 & 0.22 & -0.89 \\
-0.13 & 0.13 & -0.19 & -0.04 & -0.17 & 1.00 & 0.11 & -0.60 & 0.23 \\
-0.45 & -0.08 & -0.61 & 0.24 & -0.76 & 0.11 & 1.00 & -0.22 & 0.53 \\
0.05 & -0.35 & 0.21 & -0.71 & 0.22 & -0.60 & -0.22 & 1.00 & -0.20 \\
-0.68 & -0.09 & -0.91 & 0.11 & -0.89 & 0.23 & 0.53 & -0.20 & 1.00
\end{array}\right),
$$

and

$$
Cor(\mathbf{H}) = \left(\begin{array}{rrrrrrrrr}
1.00 & 0.10 & -0.76 & -0.13 & 0.66 & 0.12 & -0.62 & -0.12 & 0.60 \\
0.10 & 1.00 & -0.17 & -0.92 & 0.19 & 0.86 & -0.19 & -0.83 & 0.19 \\
-0.76 & -0.17 & 1.00 & 0.17 & -0.98 & -0.16 & 0.95 & 0.14 & -0.94 \\
-0.13 & -0.92 & 0.17 & 1.00 & -0.20 & -0.99 & 0.21 & 0.98 & -0.22 \\
0.66 & 0.19 & -0.98 & -0.20 & 1.00 & 0.19 & -1.00 & -0.18 & 0.99 \\
0.12 & 0.86 & -0.16 & -0.99 & 0.19 & 1.00 & -0.20 & -1.00 & 0.21 \\
-0.62 & -0.19 & 0.95 & 0.21 & -1.00 & -0.20 & 1.00 & 0.19 & -1.00 \\
-0.12 & -0.83 & 0.14 & 0.98 & -0.18 & -1.00 & 0.19 & 1.00 & -0.20 \\
0.60 & 0.19 & -0.94 & -0.22 & 0.99 & 0.21 & -1.00 & -0.20 & 1.00
\end{array}\right).
$$

In **H** many of the correlations round off to +1 or -1, which means that **H** is very close to being singular. This is not a good property for using **H** to construct mixed model equations. This could lead to poor estimation of effects in the model.

By contrast, the largest correlation in **K** is only -0.91. **K** is not close to singularity, and should be safe to invert. The signs of the correlations are often opposite to those in **H**. **K** is a much preferred matrix for use in mixed model equations.

### 1.3.1 Predictions of Covariances

Once there is an estimate of **K**, then the covariance function model can be used to calculate variances and covariances between other time points (between $t_{min}$ and $t_{max}$). For example, let $t_{150} = 150$ minutes and $t_{400} =$ 400 minutes, neither of which were actually observed or recorded in the 200 mice, but both points are within the upper and lower bounds of the experimental period.

First, calculate the standardized time equivalents (between -1 to +1).

$$\begin{aligned} q_{150} &= -0.6250 \\ q_{400} &= +0.4167 \end{aligned}$$

Set up the matrix **M** for these two points,

$$\mathbf{M}' = \begin{pmatrix} 1 & 1 \\ -0.6250000 & 0.4166667 \\ 0.3906250 & 0.1736111 \\ -0.24414062 & 0.07233796 \\ 0.15258789 & 0.03014082 \\ -0.09536743 & 0.01255867 \\ 0.059604645 & 0.005232781 \\ -0.037252903 & 0.002180325 \\ 0.0232830644 & 0.0009084689 \end{pmatrix},$$

The Legendre polynomials are

$$\begin{aligned}\Phi' &= (\mathbf{M}\Lambda)' \\ &= \begin{pmatrix} 0.7071068 & 0.7071068 \\ -0.7654655 & 0.5103104 \\ 0.1358791 & -0.3788145 \\ 0.6120387 & -0.8309381 \\ -0.8957736 & -0.3058426 \\ 0.5003195 & 0.5797175 \\ 0.2739309 & 0.7877318 \\ -0.84232235 & 0.7767490 \\ 0.7767490 & -0.7262336 \end{pmatrix}.\end{aligned}$$

$$\Phi \ \mathbf{K} \ \Phi' = \begin{pmatrix} 2.767230 & -0.4789920 \\ -0.478992 & 0.5497332 \end{pmatrix}.$$

Thus, the variance at 150 minutes is expected to be 2.767, and for 400 minutes is 0.550.

Suppose the variance for 700 minutes was needed. This could not be predicted or calculated because $t_{max}$ is only 540 minutes. Do not predict variances for time periods outside the observed range.

## 1.4 Reduced Order of Fit

In the previous example, the matrix $\mathbf{V}$ was $9 \times 9$, and the Legendre polynomials were generated for a *Full* fit, with 9 covariates. Thus, the covariance function model resulted in no errors. All of the calculated variances and covariances were exactly the same as those in the original $\mathbf{V}$.

Kirkpatrick et al.(1990) suggested looking at the eigenvalues of the matrix $\mathbf{K}$ from a full rank fit. Below are the values. The sum of all the eigenvalues was 4.690745, and also shown is the percentage of that total.

**Table 1.4**
Eigenvalues of **K**.

| **K** | |
|---|---|
| Eigenvalue | Percentage |
| 2.980498752 | 63.5 |
| 0.624171898 | 13.3 |
| 0.433933695 | 9.3 |
| 0.208716099 | 4.4 |
| 0.195913819 | 4.2 |
| 0.135701396 | 2.9 |
| 0.102963163 | 2.2 |
| 0.007215617 | 0.2 |
| 0.001630618 | 0.03 |

The majority of change in elements in **K** is explained by the first three eigenvalues at 86.1 %. The first seven explain 99.8 %. If a cut-off is set to 95%, then the first 5 eigenvalues would be important.

Covariance functions can be based on fewer than 9 covariates. Thus, the orders of fit can be 8, 7, 6, 5, 4, 3, 2, 1, or 0. Order 0 implies that all of the elements in **V** are the same, which is obviously not true.

Use the subscript $r$ to indicate a reduced order of fit, that is, $r < 8$, then

$$\mathbf{V} = \Phi_r \mathbf{K}_r \Phi'_r + \mathbf{E},$$

for $r < 8$, and where $\Phi_r$ is a rectangular matrix of rank $r$ composed of the first $r$ columns of $\Phi$, and **E** is a matrix of residuals because any lower order fit will not be perfect. Thus, $\Phi_r$ does not have an inverse, but we can obtain an estimate of $\mathbf{K}_r$. To determine $\mathbf{K}_r$, first pre-multiply **V** by $\Phi'_r$ and post-multiply that by $\Phi_r$ as

$$\begin{aligned} \Phi'_r \mathbf{V} \Phi_r &= \Phi'_r(\Phi_r \mathbf{K}_r \Phi'_r)\Phi_r \\ &= (\Phi'_r \Phi_r)\mathbf{K}_r(\Phi'_r \Phi_r). \end{aligned}$$

Now pre- and post- multiply by the inverse of

$$(\Phi'_r \Phi_r) = \mathbf{P}$$

to determine $\mathbf{K}_r$,

$$\mathbf{K}_r = \mathbf{P}^{-1}\Phi'_r\mathbf{V}\Phi_r\mathbf{P}^{-1}.$$

$\mathbf{K}_r$ will be square with $r$ rows and columns.

With $\mathbf{K}_r$ we can estimate $\mathbf{V}_r$ as

$$\mathbf{V}_r = \Phi_r\mathbf{K}_r\Phi'_r.$$

This matrix is symmetric with 45 unique elements, but only has rank $r$. The half-store function in R is a way of changing a matrix to a vector of the unique elements.

```
hsmat <- function(vcvfull) {
   mord = nrow(vcvfull)
   np = (mord *(mord + 1))/2
 desg = rep(0,np)
   k = 0
   for(i in 1:mord){
     for(j in i:mord){
    k = k + 1
       desg[k] = vcvfull[i,j] }  }
return(desg)    }
```

Let

$$ss_r = ||\text{hsmat}(\mathbf{V} \ - \ \mathbf{V}_r)||/||\text{hsmat}(\mathbf{V})||$$

be the goodness of fit statistic. The $||\mathbf{X}||$ is defined as the sum of squares of the elements of a half-stored symmetric matrix $\mathbf{X}$. Thus, $ss_r$ measures the amount of difference between two matrices scaled by the square of the values in the original matrix. This statistic is like a convergence criterion for solving a set of equations. The smaller is $ss_r$, then the less difference there is between the two matrices.

To illustrate, for order 7, there are 8 covariates. The calculated $\mathbf{K}_7$ is as follows.

$$\mathbf{K}_7 = \begin{pmatrix} 0.28382 & -0.13845 & 0.02516 & 0.05280 & 0.00783 & 0.01502 & -0.02732 & -0.04759 \\ -0.13845 & 0.35040 & -0.06584 & 0.02374 & 0.01210 & 0.02711 & -0.00793 & -0.12437 \\ 0.02516 & -0.06584 & 0.20045 & -0.03657 & -0.03170 & 0.01522 & -0.06227 & 0.00961 \\ 0.05280 & 0.02374 & -0.03657 & 0.32922 & -0.01611 & -0.00723 & 0.04249 & -0.24125 \\ 0.00783 & 0.01210 & -0.03170 & -0.01611 & 0.12203 & 0.01010 & -0.09328 & 0.02035 \\ 0.01502 & 0.02711 & 0.01522 & -0.00723 & 0.01010 & 0.12477 & -0.00245 & -0.12536 \\ -0.02732 & -0.00793 & -0.06227 & 0.04249 & -0.09328 & -0.00245 & 0.12822 & -0.02775 \\ -0.04759 & -0.12437 & 0.00961 & -0.24125 & 0.02035 & -0.12536 & -0.02775 & 0.35359 \end{pmatrix},$$

which is used to calculate $\mathbf{V}_7$, and the goodness of fit statistic is

$$\begin{aligned} ss_7 &= \frac{||\mathbf{V}_7 - \mathbf{V}||}{||\mathbf{V}||} \\ &= 0.1384349 \end{aligned}$$

To determine the probability of finding a smaller value of $ss_r$ one can use simulation, as shown in the following R script.

```
N=10000
can=c(1:N)*0
VR=V
nocov = 2 # order of fit + 1
phr = PH[ ,c(1:nocov)]
PVP = t(phr)%*%VR%*%phr
PP = t(phr)%*%phr
PPI=ginv(PP)
Kr = PPI%*%PVP%*%PPI
```

```
ndf=199

for(ko in 1:N){
   Ka = rWishart(1,ndf,Kr)/ndf
   Kb = Ka[, ,1]
   Vr = phr%*%Kb%*%t(phr)
   DEL = Vr - VR
   er = hsmat(DEL)
   vh = hsmat(VR)
   vv = sum(vh*vh)
   ssr = sum(er*er)/vv
   can[ko] = ssr
} #end of samples

hist(can,breaks=50)
```

Then compare the $ss_r$ to the histogram to find the probability of obtaining a smaller statistic. With an order of fit equal to 1, $ss_r = 0.4592$. In R one can use

```
kb = order(-can)
ncan = can[kb]
ncan[1:10]
kc=which(ncan < 0.4592)
prob = 0
if(length(kc)>0)prob = 1 - (kc[1]/length(ncan))
prob
```

This gives 0.4379 which is a pretty large probability. Similarly for the other orders, one gets the results in Table 1.5.

**Table 1.5**
Test statistics for
reduced order of fit models.

| Order | Covariates | $ss_r$ | Probability |
|---|---|---|---|
| 1 | 2 | 0.4591629 | 0.4379 |
| 2 | 3 | 0.4112741 | 0.2854 |
| 3 | 4 | 0.3132577 | 0.2699 |
| 4 | 5 | 0.2564594 | 0.1864 |
| 5 | 6 | 0.1992198 | 0.1138 |
| 6 | 7 | 0.1384349 | 0.0389 |
| 7 | 8 | 0.0495289 | 0.0001 |

Orders 1, 2, 3, 4, and 5, gave $\mathbf{V}_r$ that were significantly different from $\mathbf{V}$ (i.e. probabilities greater than 0.05) while orders 6 and 7 were not different from $\mathbf{V}$ (i.e. less than 0.05). Order 6 would be a sufficient minimal fit for the mice insulin decay data.

The mouse data example is entirely fictitious.

## 1.5 Starting V

The most frequent situation is that one is faced with a file of data on growth or lactation production, not from any experiemt, and where animals are measured at many different ages or many different days in milk. The researcher does not have any $\mathbf{V}$ matrix with which to start. What can or should be done?

### 1.5.1 Literature Search

The first option is to check the literature to see if random regression or covariance function studies have been made in the same species or others. Possibly one of these papers has published a correlation matrix for different time periods or ages. The correlation results may be used as the initial values for your current data. Then estimate new values from your own data.

### 1.5.2 Covariances among Phenotypes

Stratify your data into 9 or 10 artificial groups on the basis of age or days in milk, and calculate the phenotypic variances within each group and covariances between groups for animals that have a record in that pair of groups. This matrix is very likely not positive definite because each element will be based on different numbers of animals, and not the same animals. To make a matrix positive definite, the following R script could be used.

```
# Let A be the matrix to be made p.d.
#     no be the order of the matrix
E = eigen(A)
ev = E$values
U = E$vectors
no = dim(A)[1]
nev = which(ev < 0)
wr = 0
k=length(nev)
if(k > 0){
   p = ev[no - k]
   B = sum(ev[nev])*2.0
   wr = (B*B*100.0)+1
   val = ev[nev]
   ev[nev] = p*(B-val)*(B-val)/wr
   A = U%*%diag(ev)%*%t(U)
   }
```

To illustrate making a matrix positive definite, take

$$\mathbf{A} = \begin{pmatrix} 9 & 7 & 2 & 1 \\ 7 & 6 & 3 & 4 \\ 2 & 3 & 11 & 7 \\ 1 & 4 & 7 & 9 \end{pmatrix},$$

which has the following eigenvalues,

```
    E = eigen(A)
    ev = E$values
    ev
      [1] 20.9576366
      [2] 11.1578605
      [3]  3.3289730
      [4] -0.4444701

    U = E$vectors
    no = dim(A)[1]
    nev = which(ev < 0)
    nev
#      = ( 4  )
    wr = 0
    k=length(nev)   #  k=1
    if(k > 0){
       p = ev[no - k]
         #  = 3.3289730  lowest positive eigenvalue
       B = sum(ev[nev])*2.0  # = -0.8889403
       wr = (B*B*100.0)+1    # = 80.02148
       val = ev[nev]         # = -0.4444701
       ev[nev] = p*(B-val)*(B-val)/wr
          # = 0.00821843
       A = U%*%diag(ev)%*%t(U)
       }
```

The new eigenvalues are used with the original eigenvectors to regenerate the new, positive definite **A** matrix.

$$\mathbf{A} = \left( \begin{array}{rrrr} 9.135600 & 6.812468 & 1.973774 & 1.084506 \\ 6.812468 & 6.259353 & 3.036270 & 3.883131 \\ 1.973774 & 3.036270 & 11.005072 & 6.983656 \\ 1.084506 & 3.883131 & 6.983656 & 9.052664 \end{array} \right).$$

Once there is a positive definite **V** matrix, then a covariance function analysis to find the best minimum order of fit can be carried out.

**V** can be scaled to a genetic covariance matrix by multiplying by heritability. Or multiply times $(r - h^2)$ for a permanent environmental covariance matrix. All of these matrices are for the purpose of giving suitable starting or initial values of those matrices for the statistical model.

The best method of estimation is a Bayesian approach using Gibbs sampling, which would allow all of the data to be utilized.

## 1.6 Data Requirements

Suppose a fixed regression model for growth of sheep is to be studied. Each lamb has, at most, three weight measurements, and from that coefficients for 5 covariates are to be estimated. With three data points we can only estimate coefficients for 3 covariates because there would be no degrees of freedom remaining. However, because we have many lambs weighed at various ages, it is possible to estimate coefficients for 5 covariates across lambs, but not for individual lambs.

The same logic must have some effect on a random regression model, even though with animal genetic effects are random, it is computationally possible to estimate 5 coefficients per lamb. However, the quality of those estimates might be questionable. In early test day models, researchers required cows to have 7 or more test day records before an attempt was made to estimate covariance matrices with orders of fit equal to 5 or 6 (Jamrozik et al. 1998).

A general recommendation is, if the number of weights per animal is three or less, then a multiple trait animal model with each weight as a separate trait is the preferred mode of analysis. With more than three weights per animal (on average), then a random regression model could be employed. The appropriate order of fit should not be greater than the average number of weights per animal.

# Chapter 2

# The Models

Random regression models (RRM) are used to estimate matrices that are part of the covariance functions to estimate a larger array of variances and covariances. RRM are also used to model the shape or trajectory of observations taken over time. Phenotypically, the trajectory has to be fitted, and at the same time the variation along the trajectory needs to be considered.

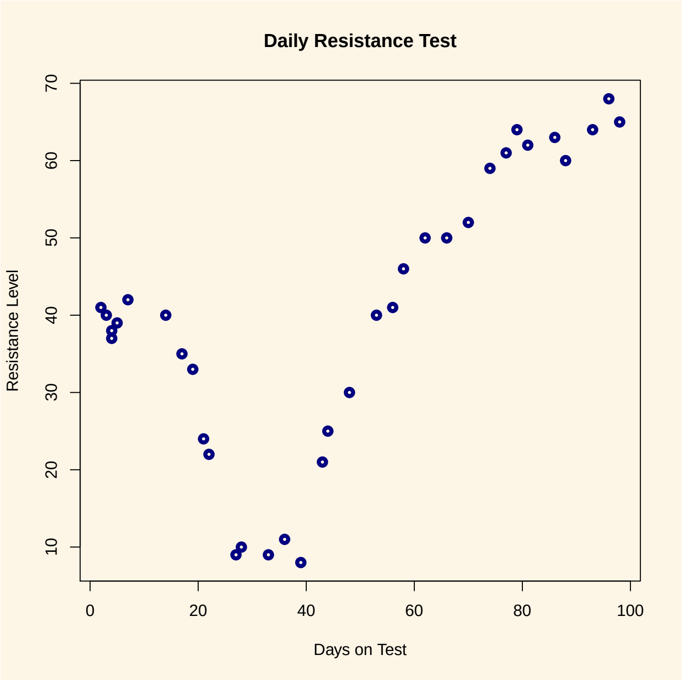
## 2.1 Fitting The Trajectory

Most trajectories are smooth, continuous, and can be fit with very few covariates. Sometimes, however, the trajectory is unknown and may be undefined with ups and downs over time. There could be different trajectories for males versus females, or for different breeds. Over the years, the trajectories could shift due to selection of animals for faster growth or higher milk yields. All of these factors must be considered.

The first course of action is to plot data against the time scale of interest. Observations can be partitioned by gender, by breed, by age at start, or by years. One should look at all aspects of the data, before commiting to one model for analysis. Below are some fictitious data on animals over a period of 1 to 100 days. The trait measured is the amount of resistance to a bacteria from first day of spring to fall. Figure

2.1 represents data on 6 animals with from 4 to 7 observations per animal, a total of 34 observations.

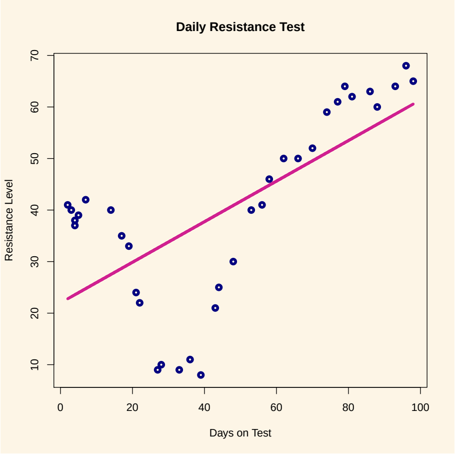
**Figure 2.1**



The data show an initial resistance that becomes less up to day 35, then improves again until fall. Fitting the trajectory of this curve could be problematic.

We will fit ordinary linear regressions on days on test (divided by 100) from linear to sextic equations. Legendre polynomials could be used from order 1 to order 6, but at this stage of model building, simple regressions suffice.

**Figure 2.2**
**Linear Fit**



A linear regression does not fit the data very well. The predicted **y** is correlated with the original **y** at 0.66. Including a squared term for days on test gave a correlation of 0.85.

**Figure 2.3**
**Quadratic Fit**

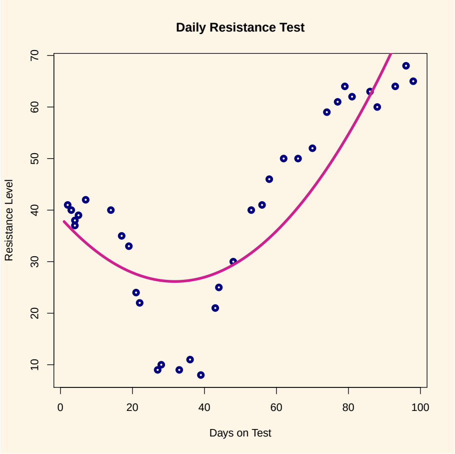


**Figure 2.4**
**Cubic Fit**

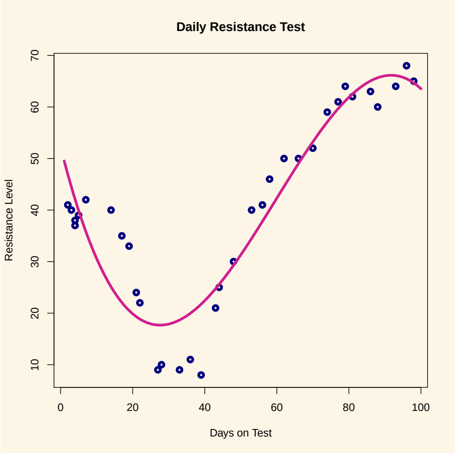


**Figure 2.5**
**Quartic Fit**

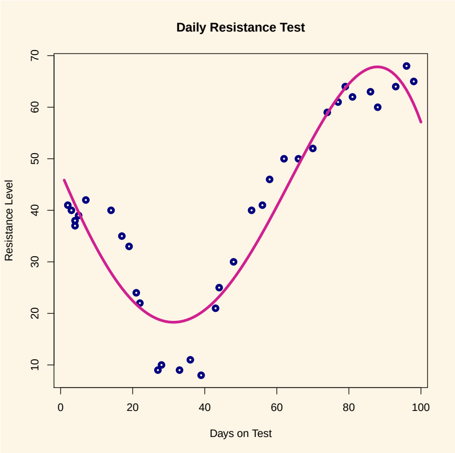


**Figure 2.6**
**Quintic Fit**

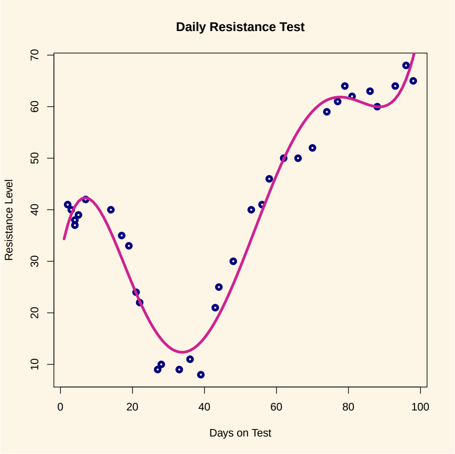
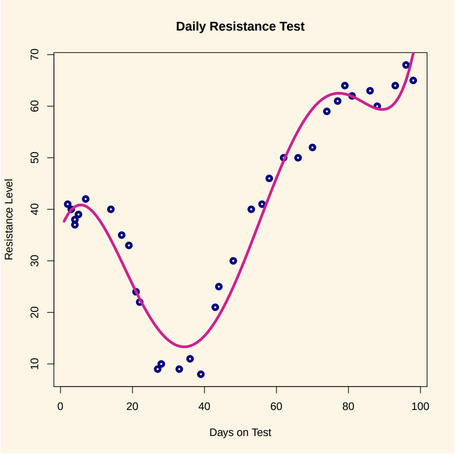


**Figure 2.7**
**Sextic Fit**



A summary of the fit of regressions of different powers of days (divided by 100) are given in Table 2.1.

**Table 2.1**

Correlations of $\widehat{\mathbf{y}}$ with $\mathbf{y}$.

| Fit | Correlation |
|---|---|
| Linear | 0.66 |
| Quadratic | 0.85 |
| Cubic | 0.9356 |
| Quartic | 0.9423 |
| Quintic | 0.9789 |
| Sextic | 0.9740 |

The fit of the data improves with an increase in power of the time variable, but none of the regressions adequately fit the low observations from day 30 to 40. Going from Quintic to Sextic the correlation actually decreased, so that the fit is starting to become worse.

In the quintic equation, there are waves in the first five days, and at the very end from day 90 to 100. The lack of fit for the lower values in days 30 to 40 persists.

What happens when there are 2000 observations rather than just 34? Does the fit become worse or better? These functions are not entirely adequate for fitting the trajectories.

### 2.1.1 Classification Approach

One hundred days can be grouped into 20 periods of 5 days each. A linear model with time period groups could be used to model the trajectory. That is,

$$y_{ij} = \mu \ + \ T_i \ + \ e_{ij},$$

where

$y_{ij}$ is the $j^{th}$ observation within the $i^{th}$ time group (twenty groups),

$\mu$ is an overall mean,

$T_i$ is the effect of the $i^{th}$ time group, and

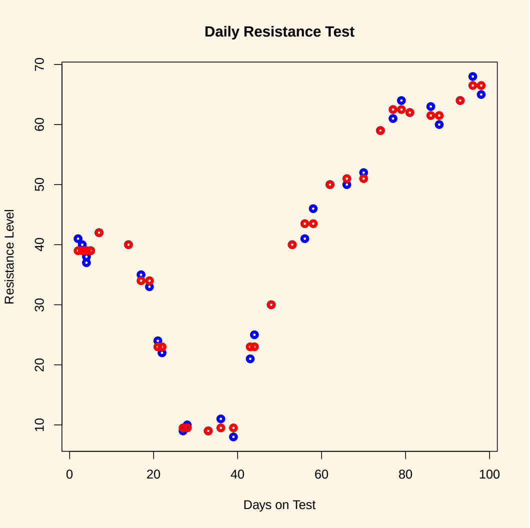
$e_{ij}$ is a residual effect.

This model will fit the data very well, but at the expense of needing to estimate 20 parameters rather than 6 (Quintic function) or 7.

The time groups do not need to be equal in size. Time periods of 10 or 20 days might be appropriate if the observations have about the same magnitude over all 10 or 20 days. That would not be true in the example data, because the values are decreasing sharply at the beginning, then increasing quickly. The last 10 or 15 days of the 100 day period might be able to be combined into one group. However, there is little harm in keeping the 20 periods of 5 days each. There is no major computing problem in doing so.

The fit of the classification model gave a correlation of 0.9975 between $\widehat{\mathbf{y}}$ (red points) and $\mathbf{y}$ (blue points) (Figure 2.8).

**Figure 2.8**
**Fitting Time Groups**



The time group means give a non-smooth tragectory, but it fits the data very well. Also, one does not need to define the type of curve or the shape. A drawback is that days are being combined, so if resistance is changing (increasing or decreasing) a lot from the first day to the fifth within a group, then the group mean or $T_i$ effect will not account for that, and there will be errors for the predicted observations farthest away from the middle day in the group. Forming time groups must be handled judiciously.

In this example there were not enough observations to make smaller time groups (e.g. 3 days each or 2 days each). The classification approach is good when you do not know what kind of function describes the trajectory. Usually researchers are dealing with many thousands of observations so that having 20 time groups or a 6 parameter regression does not make very much difference in terms of estimation difficulty. One cannot really go wrong with the classification approach over the regression approach for modelling the trajectories unless there are not sufficient numbers of observations.

### 2.1.2 Spline Functions

Another approach would be to divide the trajectory into parts, such that the parts are best fit by linear or quadratic functions. For the example, the parts might be days 1 to 20, days 21 to 50, and days 51 to 100. A model might be

$$y_{kj} = \mu + b_1 X_{kj} + b_2 X_{kj}^2 + b_3 U_{kj} + b_4 U_{kj}^2 + b_5 W_{kj} + b_6 W_{kj}^2 + e_{kj},$$

where

$y_{kj}$ is the $j^{th}$ observation on the $k^{th}$ animal,

$\mu$ is an overall mean,

$X_{kj}$ is the days on test (divided by 100) corresponding to the observation,

$U_{kj}$ is zero, unless days on test are greater than 20, then it is equal to $(X_{kj} - 0.2)$,

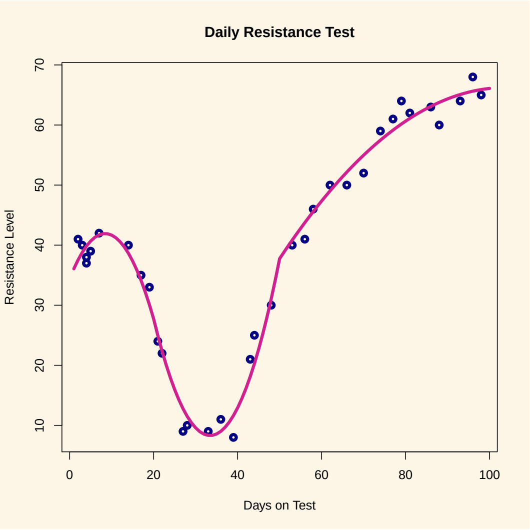
$W_{kj}$ is zero, unless days on test are greater than 50, then it is equal to $(X_{kj} - 0.5)$,

$b_\ell$ for $\ell = 1$ to 6 are regression coefficients,

$e_{kj}$ are residual effects.

The fit of the spline function model gave a correlation of 0.9923, which is slightly less than that of the classification model, but having only seven parameters to estimate rather than 20. The agreement of $\hat{\mathbf{y}}$ and $\mathbf{y}$ is shown in Figure 2.9.

**Figure 2.9**
**Fitting Splines**



In practice, one would first try the classification model approach in order to get an idea what the trajectory might be. Then pick an appropriate regression model with fewer parameters that gives a good fit. Having a good fit for the trajectory is important to the random regression model. Keep in mind, that at this stage we are only concerned with the phenotypic fit of the curve, and not with the covariance structures of observations deviated from the curves.

## 2.2 Random Variation in Curves

Assume with the fictitious resistance data, that the logical factors in the statistical model are gender, year of test, contemporary group, animal additive genetic, animal permanent environment, and residual effects. Assume that there are no interactions among these factors. Suppose also that we have several thousand animals and their data. The fixed factors of the model are gender and year of test, and all other factors are random factors. The fixed factors are modelled with the function that we have chosen to fit the phenotypic trajectory, e.g. the spline function with seven parameters. Thus, there will be a separate spline function for each gender, and for each year of test. If there are two levels of the gender effect, and five years, then we will have 7 curves to be estimated, each with 7 parameters (i.e. 49 total parameters).

The random parts of the model will be modelled with Legendre polynomials of order 4. (Suppose that order 4 was determined to be the best fit). Thus, there are five parameters to be estimated for each curve to fit the covariance functions. There will be one curve for each animal additive genetic effect, one curve for each contemporary group effect, and one curve for each animal permanent environmental effect. The residual effects will be discussed in the next section.

A comparison of a typical linear model and a random regression model are shown in Table 2.2.

**Table 2.2**

Usual Linear Model versus Random Regression Model (RRM).

| Factor | Linear Elements | RRM Elements |
|---|---|---|
| Gender | $g_i$ | $g_{i0} + g_{i1}X_t + g_{i2}X_t^2$<br>$+g_{i3}U_t + g_{i4}U_t^2 + g_{i5}W_t + g_{i6}W_t^2$ |
| Year | $h_j$ | $h_{j0} + h_{j1}X_t + h_{j2}X_t^2$<br>$+h_{j3}U_t + h_{j4}U_t^2 + h_{j5}W_t + h_{j6}W_t^2$ |
| Contemporary Group | $c_k$ | $c_{k0}z_{t0} + c_{k1}z_{t1} + c_{k2}z_{t2}$<br>$+c_{k3}z_{t3} + c_{k4}z_{t4}$ |
| Animal Additive | $a_\ell$ | $a_{\ell 0}z_{t0} + a_{\ell 1}z_{t1} + a_{\ell 2}z_{t2}$<br>$+a_{\ell 3}z_{t3} + a_{\ell 4}z_{t4}$ |
| Animal PE | | $p_{\ell 0}z_{t0} + p_{\ell 1}z_{t1} + p_{\ell 2}z_{t2}$<br>$+p_{\ell 3}z_{t3} + p_{\ell 4}z_{t4}$ |
| Days on Test | | $X_t = t/100$, $U_t = (t - 20)/100$,<br>and $W_t = (t - 50)/100$<br>$U_t = 0$ if $t < 21$, and $W_t = 0$ if $t < 51$ |
| Legendre Polynomials | | $z_{tm}$ for $m = 0$ to 4 |

Notice that every fixed and random factor in the linear model has been expanded to be a regression function of days on test. The fixed factors are regressions that fit the phenotypic trajectory well, and the random factors involve Legendre polynomials and attempt to fit the covariance functions. The animal permanent environmental effects are not in the linear model, because that assumes each animal is observed only once. However, in the RRM, animals are expected to be observed several times, and hence there are permanent environmental effects. Every RRM for any species and trait can be set up in the same manner. The RRM is about visualizing curves and covariance functions over time. The observations are changing with time, and the effects of the model also change with time.

The curves that fit the trajectory may require more covariates than those that fit the covariance functions. They do not need to be of the same order. Computationally, there could be advantages to using the same order of fit for the fixed and random factors. The important point is to have a good fit of the trajectories amongst the fixed effects, and an appropriate order for the Legendre polynomials with the random factors.

Some research has reported different orders of fit of the Legendre polynomials for each of the random factors. This is not necessary and can complicate the analysis of the RRM, especially when dealing with multiple traits. I prefer to use the same order of Legendre polynomials for all random factors.

## 2.3 Residuals

Residuals are the difference between predicted observations (using the estimates of parameters from the RRM) and the actual observations.

$$\widehat{\mathbf{e}} = \widehat{\mathbf{y}} - \mathbf{y}.$$

Now partition the residuals into the 20 time periods as in the classification model,

$$\widehat{\mathbf{e}} = \left( \begin{array}{c} \widehat{\mathbf{e}}_1 \\ \widehat{\mathbf{e}}_2 \\ \widehat{\mathbf{e}}_3 \\ \vdots \\ \widehat{\mathbf{e}}_{20} \end{array} \right).$$

Then estimate the residual variance for the $i^{th}$ time group as

$$\sigma^2_{e_i} = \widehat{\mathbf{e}}'_i \widehat{\mathbf{e}}_i / n_i,$$

where $n_i$ are the number of observations in the $i^{th}$ time group.

The overall residual covariance matrix, $\mathbf{R}$, is assumed to be diagonal with 20 different residual variances according to time group. The mixed model equations for the RRM therefore involve $\mathbf{R}^{-1}$ which means that observations are inversely weighted by the magnitude of their residual variance. Larger residual variances lead to lesser weight in the equations.

The residual variances can be plotted on a graph relative to time group number. Either a pattern of the residual variances can be observed and a function used to determine the residual variance for each observation, or some collapsing of the groups into larger time groups may be possible. The possibility exists for each of the 20 time group variances to be different, and thus, one may always have to use 20 different variances.

## 2.4 Complete Model

### 2.4.1 Fixed Factors

The fixed factors are for modelling the shape of the phenotypic trajectories. Let

$$\mathbf{f}'(t) \;=\; \begin{pmatrix} t^0 & t^1 & t^2 & \cdots & t^{m-1} \end{pmatrix}$$

be a vector of covariates of time of length $m$, these may or may not be Legendre polynomials depending on the function that best fits the trajectories, OR

$$\mathbf{f}'(t) \;=\; \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

a vector indicating which time group an observation belongs as in the classification approach (of length $m$). The desgin matrix for gender effects, for example, would be written as

$$\mathbf{Xg} \;=\; \begin{pmatrix} \mathbf{f}'(t_1) & \mathbf{0}' \\ \mathbf{f}'(t_2) & \mathbf{0}' \\ \mathbf{0}' & \mathbf{f}'(t_3) \\ \vdots & \vdots \\ \mathbf{0}' & \mathbf{f}'(t_N) \end{pmatrix} \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix}$$

where $N$ is the total number of animals observed. The first two animals belonged to gender 1, the third and the $N^{th}$ animals belonged to gender two. Also,

$$\mathbf{g}_i = \begin{pmatrix} g_{i1} \\ g_{i2} \\ \vdots \\ g_{im} \end{pmatrix},$$

is a vector of length $m$ for each gender which represent the fixed regression coefficients which give the trajectory of the responses for gender $i$. Instead of one number for each gender effect, there will be a vector of $m$ numbers.

Similarly, the fixed effects of years can be represented as

$$\mathbf{Wh} = \begin{pmatrix} \mathbf{f}'(t_1) & \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{0}' \\ \mathbf{f}'(t_2) & \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{0}' \\ \mathbf{0}' & \mathbf{f}'(t_3) & \mathbf{0}' & \cdots & \mathbf{0}' \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}' & \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{f}'(t_N) \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \vdots \\ \mathbf{h}_{n_y} \end{pmatrix}$$

where $n_y$ is the number of years in the data. If $m = 7$, for example, and $N = 1000$, then $\mathbf{X}$ has 1000 rows and $2 \times m = 14$ columns. If $n_y = 10$, then $\mathbf{W}$ has 1000 rows and $10 \times m = 70$ columns.

### 2.4.2 Random Factors

The random factors are for modelling the covariance functions and make use of Legendre polynomials of order $r$. The analysis also gives different curves for every level of each random factor. A curve for each contemporary group, for each animal's genetic effect, and for each animal's permanent environmental effect. Let

$$\mathbf{z}'(q) = \begin{pmatrix} \phi(q)_0 & \phi(q)_1 & \phi(q)_2 & \cdots & \phi(q)_r \end{pmatrix}.$$

The design matrices are constructed in the same manner as those for the fixed factors, only using $\mathbf{z}'(q)$ rather than $\mathbf{f}'(t)$. Assume that $r = 4$, for this discussion. The design matrices are

$\mathbf{Z}_c$ for contemporary groups,

$\mathbf{Z}_a$ for animal additive genetic effects, and

$\mathbf{Z}_p$ for animal permanent environmental effects.

Contemporary groups are groups of animals usually born within a few days or weeks of each other, which are reared together through much of their early life, and who are kept together during the time that the observations were collected. If $N = 1000$ observations, and the number of contemporary groups is $n_c = 50$, then $\mathbf{Z}_c$ has 1000 rows and $5 \times n_c = 250$ columns. The covariance function matrix for contemporary groups is $\mathbf{K}_c$ of dimension $5 \times 5$. Normally, the covariance matrix for contemporary group effects is $\mathbf{I}\sigma_c^2$, but in a RRM it is a block diagonal matrix,

$$Var(\mathbf{c}) \ = \ \mathbf{I} \bigotimes \mathbf{K}_c$$

of dimension $250 \times 250$, where $\otimes$ is the direct product operation.

The additive relationship matrix, $\mathbf{A}$, is used in all animal models, and includes ancestors who may not have been observed or measured in the data itself. The design matrix for animal additive genetic effects must, therefore, have additional columns of zeros to accommodate the ancestors. Let $n_w$ be the number of animals with observations in the data, and let $n_a$ be the total number of animals including ancestors. Hence, $n_a \geq n_w$. The matrix, $\mathbf{Z}_a$, has 1000 rows, and $n_a \times 5$ columns. If $n_a = 200$, then $\mathbf{Z}_a$ has 1000 columns. If $\mathbf{K}_a$ is the covariance function matrix for genetic variances, then

$$Var(\mathbf{a}) \ = \ \mathbf{A} \bigotimes \mathbf{K}_a$$

of dimension $1000 \times 1000$.

The animal permanent environmental (PE) covariance function matrix is $\mathbf{K}_p$. The design matrix, $\mathbf{Z}_p$ has 1000 rows and $n_w \times 5$ columns. Let $n_w = 140$, then that is 700 columns.

$$Var(\mathbf{p}) \ = \ \mathbf{I} \bigotimes \mathbf{K}_p$$

of dimension $700 \times 700$.

The assumption is that there are no covariances between levels of different random factors, e.g. between contemporary groups and animal additive genetic effects.

The residual matrix, **R** was shown to be diagonal, but with different residual variances depending on the day on test.

### 2.4.3 Mixed Model Equations

Once a model is specified, then Henderson's *best linear unbiased prediction* methodology is employed. This requires constructing and solving the *Mixed Model Equations*(MME). These equations are

$$\begin{pmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{W} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z}_c & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z}_a & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z}_p \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{W} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{Z}_c & \mathbf{W}'\mathbf{R}^{-1}\mathbf{Z}_a & \mathbf{W}'\mathbf{R}^{-1}\mathbf{Z}_p \\ \mathbf{Z}_c'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}_c'\mathbf{R}^{-1}\mathbf{W} & \mathbf{Z}_c'\mathbf{R}^{-1}\mathbf{Z}_c + \mathbf{I}\bigotimes\mathbf{K}_c^{-1} & \mathbf{Z}_c'\mathbf{R}^{-1}\mathbf{Z}_a & \mathbf{Z}_c'\mathbf{R}^{-1}\mathbf{Z}_p \\ \mathbf{Z}_a'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}_a'\mathbf{R}^{-1}\mathbf{W} & \mathbf{Z}_a'\mathbf{R}^{-1}\mathbf{Z}_c & \mathbf{Z}_a'\mathbf{R}^{-1}\mathbf{Z}_a + \mathbf{A}^{-1}\bigotimes\mathbf{K}_a^{-1} & \mathbf{Z}_a'\mathbf{R}^{-1}\mathbf{Z}_p \\ \mathbf{Z}_p'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}_p'\mathbf{R}^{-1}\mathbf{W} & \mathbf{Z}_p'\mathbf{R}^{-1}\mathbf{Z}_c & \mathbf{Z}_p'\mathbf{R}^{-1}\mathbf{Z}_a & \mathbf{Z}_p'\mathbf{R}^{-1}\mathbf{Z}_p + \mathbf{I}\bigotimes\mathbf{K}_p^{-1} \end{pmatrix}$$

$$\cdot \begin{pmatrix} \widehat{\mathbf{g}} \\ \widehat{\mathbf{h}} \\ \widehat{\mathbf{c}} \\ \widehat{\mathbf{a}} \\ \widehat{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}_c'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}_a'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}_p'\mathbf{R}^{-1}\mathbf{y} \end{pmatrix}.$$

These equations are solved by using iteration on data routines (Chapter 7), in which coefficients of the MME are calculated as they are needed. The following procedure describes how to estimate the covariance function matrices using a pseudo-Bayesian method.

1. First, go through the observations, one at a time, and calculate deviations,
$$\mathbf{y} - \mathbf{X}\widehat{\mathbf{g}} - \mathbf{W}\widehat{\mathbf{h}} - \mathbf{Z}_c\widehat{\mathbf{c}} - \mathbf{Z}_a\widehat{\mathbf{a}} - \mathbf{Z}_p\widehat{\mathbf{p}}$$
2. Accumulate the deviations for gender effects, then solve for new gender regression coefficients.
3. Repeat for year effects.
4. Repeat for contemporary groups. For each contemporary group calculate
$$\widehat{\mathbf{c}}_i\widehat{\mathbf{c}}_i'$$

and accumulate these 5 × 5 matrices over all contemporary groups. Then

$$\widehat{\mathbf{K}}_c = \sum_i \widehat{\mathbf{c}}_i \widehat{\mathbf{c}}_i' / \chi^2(n_c + 2)$$

to estimate the covariance function matrix, where $\chi^2(s)$ is a random Chi-square variate having $s$ degrees of freedom.

5. Repeat for animal additive genetic effects. This step involves elements of $\mathbf{A}^{-1}$. For each animal that has observations, calculate

$$\mathbf{m}_\ell = \widehat{\mathbf{a}}_\ell - 0.5(\widehat{\mathbf{a}}_{sire} + \widehat{\mathbf{a}}_{dam})$$

From $\mathbf{A}^{-1}$ there will be $b_{ii} = (0.5 - 0.25(F_{sire} + F_{dam}))^{-1}$ for each animal. The new covariance function matrix is

$$\widehat{\mathbf{K}}_a = \sum_i \mathbf{mm}' b_{ii} / \chi^2(n_w + 2).$$

Only animals with records are used because they are the only ones who contribute to the estimation of variances and covariances directly.

6. Repeat for animal permanent environmental effects. $\mathbf{K}_p$ is estimated in the same manner as that for contemporary groups.

7. Estimate new residual variances as before.

8. The new covariance function matrices are used in the next iteration, and should be saved in a file so that they may be averaged to give a final estimate of each.

9. Go back to step 1 and repeat.

# Chapter 3

# RRM Calculations

This chapter is about the calculations necessary to apply a random regression model. The R language is used to illustrate, but R is only useful for small examples. To analyze millions of records one would need to write programs in either Fortran or C++ (Chapter 7).

Below are completely fictitious data on six animals for resistance to a bacteria during the first 25 days of spring (Table 3.1), and the pedigree of those six animals (Table 3.2). All data were from one year, so year effects were removed from the previous model. There were two contemporary groups.

**Table 3.1**
Example Data for Resistance to Bacteria
in the first 25 days of spring.

| Animal | Gender | CG | Day,Resistance | | | |
|---|---|---|---|---|---|---|
| 7 | 1 | 1 | 4,38 | 7,37 | 16,35 | 25,27 |
| 8 | 2 | 1 | 2,40 | 11,40 | 21,28 | |
| 9 | 2 | 1 | 6,42 | 17,37 | 25,22 | |
| 10 | 2 | 2 | 5,39 | 15,36 | 19,33 | 24,25 |
| 11 | 1 | 2 | 3,41 | 14,38 | 22,23 | |
| 12 | 1 | 2 | 4,37 | 9,35 | 17,30 | 23,24 |

**Table 3.2**
Pedigree of Example Animals.

| Animal | Sire | Dam | $b_{ii}$ |
|---|---|---|---|
| 1 | - | - | 1.0 |
| 2 | - | - | 1.0 |
| 3 | - | - | 1.0 |
| 4 | - | - | 1.0 |
| 5 | - | - | 1.0 |
| 6 | - | - | 1.0 |
| 7 | 1 | 4 | 0.5 |
| 8 | 2 | 4 | 0.5 |
| 9 | 3 | 5 | 0.5 |
| 10 | 1 | 5 | 0.5 |
| 11 | 2 | 6 | 0.5 |
| 12 | 3 | 6 | 0.5 |

## 3.1 Data Prep

```
   y = c(38,37,35,27,40,40,28,42,37,22,39,36,33,
         25,41,38,23,37,35,30,24)
   days=c(4,7,16,25,2,11,21,6,17,25,5,15,19,24,
          3,14,22,4,9,17,23)
   dgrp=c(1,2,4,5,  1,3,5,  2, 4, 5,1, 3, 4, 5,1,
          3, 5,1,2,4,5)
# Animals with records
   anw = c(7,7,7,7,8,8,8,9,9,9,10,10,10,10,11,11,
           11,12,12,12,12)
# Gender codes for each observation
   gend = c(1,1,1,1,2,2,2,2,2,2,2,2,2,2,1,1,1,1,
            1,1,1)
# Contemporary group levels
   cg = c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,
          2,2)
# Sires and dams of all animals (first 6 are unknown)
  sirs=c(0,0,0,0,0,0,1,2,3,1,2,3)
  dams=c(0,0,0,0,0,0,4,4,5,5,6,6)
```

```
# bii values  = 0.5 - 0.25*(Fsire + Fdam)
  bii=c(1,1,1,1,1,1,.5,.5,.5,.5,.5,.5)
# Initial residual variances
  vare=c(1,1,1,1,1,.97,.97,.97,.97,.97,.95,.95,.95,.95,.95,
        .93,.93,.93,.93,.93,.90,.90,.90,.90,.90)
  length(y)
  length(days)  # check that lengths are the same
```

## 3.2 Covariates for Fixed and Random Regressions

Because the example data covers the first 25 days of the test period, the curve that fits the trajectory is a simple quadratic function. The assumption is that a Legendre polynomial of order 2 will fit the random regressions.

For the fixed curves functions of days are used, but days are divided by 100 so that time is a decimal number. Otherwise the covariates when squared could become large numbers, especially when accumulated over many animals. The primary reason for dividing by 100 is to avoid rounding problems with large datasets. Legendre polynomials are already decimal numbers.

```
# divide days by 100 to reduce magnitude
  alld=c(1:25)/100
  all2=alld*alld
  alle=alld*0 + 1
  fT=cbind(alle,alld,all2)
# Legendre polynomials
  LAM=LPOLY(3)
  LAM
  ti=c(1:25)
  tmin=1
  tmax=100  # you could also use 25, in this case
  qi = 2*(ti - tmin)/(tmax - tmin) - 1
  qi
```

```
x1=qi
x0=x1*0 + 1
x2=x1*x1
M=cbind(x0,x1,x2)
PH = M %*% t(LAM)
PH
```

## 3.3 Design Matrices for Model

Each of the design matrices for the factors in the model, i.e. gender effects, contemporary groups, animal additive genetic, and animal permanent environmental effects, needs to be created. A simple function was written to create these matrices, making use of fT and PH created in the previous subsection.

```
rdesgn = function(v,tim,fT,nc,no){
  if(is.numeric(v)){
    va = v
    mrow = length(va)
    mcol = max(va)
    if(nc > mcol)mcol = nc }
 if(is.character(v)){
    vf = factor(v)
    va = as.numeric(vf)
    mrow = length(v)
    mcol = length(levels(vf))
    if(nc > mcol)mcol = nc }
 mcc = mcol*no
 X = matrix(data=c(0),nrow=mrow,ncol=mcc)
 for(i in 1:mrow){
    ic = (va[i]-1)*no
    jc = c((ic+1):(ic+no))
    X[i,jc] = fT[tim[i], ] }
return(X)  }

# design matrix for gender effects
```

```
   Xg=rdesgn(gend,days,fT,2,3)
   Xg
# design matrix for random factors using
#  Legendre polynomials in PH
   Zc=rdesgn(cg,days,PH,2,3)
   Za=rdesgn(anw,days,PH,12,3)
   Zp=Za[ ,c(19:36)]  # PE design matrix is a
       # subset of design matrix for genetic
# Check the dimensions of the matrices
```

## 3.4 Initial Covariance Matrices

If this were a new experiment, then we would not have any covariance function matrices, or residual matrices to start a mixed model analysis. We also need to construct the inverse of the additive relationship matrix. Some parameters were chosen arbitrarily to illustrate the calculations of the example. The example data are not sufficiently numerous with which to obtain adequate estimates of the parameters.

```
# Routine to set up inverse of additive relationship
#  matrix from list of sires and dams and bii-values
 AINV = function(sid,did,bi){
# IDs assumed to be consecutively numbered, and
# parents come before progeny
   rules=matrix(data=c(1,-0.5,-0.5,
        -0.5,0.25,0.25,
        -0.5,0.25,0.25),
        byrow=TRUE,nrow=3)
   nam = length(sid)
   np = nam + 1
   ss = sid + 1
   dd = did + 1
  LAI = matrix(data=c(0),nrow=np,ncol=np)
  for(i in 1:nam){
      ip = i + 1
      X = 1/bi[i]
```

```
       k = cbind(ip,ss[i],dd[i])
       LAI[k,k] = LAI[k,k] + rules*X
     }
   k = c(2:np)
   C = LAI[k,k]
   return(C) }

  AI = AINV(sirs,dams,bii)
# AI is 12 by 12 for the example
```

Begin with the following matrices, which may come from the literature or from other creative means. With a much larger data set, then we could estimate more legitimate covariance matrices.

```
Kc = matrix(data=c(.51696, -.1623, -.0895,
 -.1623, .3504, .1135,
 -.0895, .1135, 1.20888),byrow=TRUE,ncol=3)
> Kc
         [,1]    [,2]     [,3]
[1,]  0.51696 -0.1623 -0.08950
[2,] -0.16230  0.3504  0.11350
[3,] -0.08950  0.1135  1.20888

Ka = matrix(data=c(.0522, -.00170,-.00142,
-.00170,.0350,-.00149,
-.00142,-.00149,.121),byrow=TRUE,ncol=3)
> Ka
         [,1]     [,2]     [,3]
[1,]  0.05220 -0.00170 -0.00142
[2,] -0.00170  0.03500 -0.00149
[3,] -0.00142 -0.00149  0.12100

Kp = matrix(data=c(.06,-.00635,.003753,
  -.00635,.04,-.00106,
  .003753,-.00106,.15),byrow=TRUE,ncol=3)
> Kp
          [,1]     [,2]      [,3]
  [1,]  0.060000 -0.00635  0.003753
  [2,] -0.006350  0.04000 -0.001060
  [3,]  0.003753 -0.00106  0.150000
# Invert the covariance matrices
   Kci=ginv(Kc)
   Kai=ginv(Ka)
   Kpi=ginv(Kp)
#  Residual variances
   R = diag(vare[days])
   RI=ginv(R)
#  Set up covariance matrices for each factor

#  Contemporary groups (2 of them
   C=id(2)
   CI = C %x% Kci  # direct product, order 6 x 6

#  Additive genetic (12 animals
   dim(AI)
   GI = AI %x% Kai  # order 36 x 36
#  Permanent Environmental
   P=id(6)
   PI = P %x% Kpi  # order 18 x 18

   W = cbind(Zc,Za,Zp)
   X = Xg
   HI=block(CI,GI,PI)
 # Uses block function, or direct sum
```

## 3.5 Mixed Model Equations

Another R function was made to construct the mixed model equations of Henderson, using the previously created design matrices and covariance matrices. The function is as follows.

```
#  Function to form MME
MME = function(X,Z,GI,RI,y){
        XX = t(X) %*% RI %*% X
        XZ = t(X) %*% RI %*% Z
        ZZ = t(Z) %*% RI %*% Z
        Xy = t(X) %*% RI %*% y
        Zy = t(Z) %*% RI %*% y
        N = length(y)
        R1 = cbind(XX,XZ)
        R2 = cbind(t(XZ),(ZZ+GI))
        LHS = rbind(R1,R2)
        RHS = rbind(Xy,Zy)
    #  now solve
        C = ginv(LHS)
        bhat = C %*% RHS
    #  estimate residual variance
        SSR = t(bhat) %*% RHS
        VPE = diag(C)
        sep = matrix(data=VPE,ncol=1)
  return(list(LHS=LHS,RHS=RHS,SSR=SSR,C=C,
        VPE=sep,SOLNS=bhat))  }
```

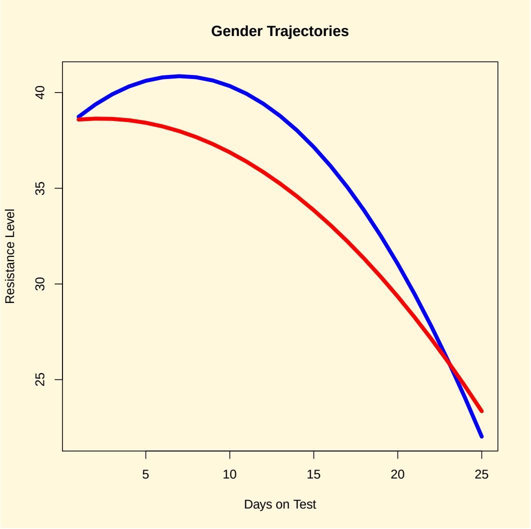
Below are the statements to create and solve the MME, and then to separate the solutions by factor. Remember there will be 3 solutions for each gender (the fixed curves or trajectories), and for each contemporary group, and each animal (genetic and PE).

```
      SA = MME(Xg,W,HI,RI,y)
      bh = SA$SOLNS
      ghat = bh[c(1:6),]
      chat = bh[c(7:12),]
      ahat = bh[c(13:48),]
      phat = bh[c(49:66),]
    # must reformat the solutions
      gh = matrix(data=ghat,byrow=TRUE,ncol=3)
    > gh  # GENDER EFFECTS
              [,1]     [,2]      [,3]
     [1,] 38.48489 13.58843 -296.4775
     [2,] 37.97264 82.02404 -583.2681
```

```
  ch = matrix(data=chat,byrow=TRUE,ncol=3)
> ch  # CONTEMPORARY GROUP EFFECTS
           [,1]       [,2]       [,3]
[1,]  0.6131888 -0.4489996 -0.5207769
[2,] -0.6131888  0.4489996  0.5207769

  ah = matrix(data=ahat,byrow=TRUE,ncol=3)
> ah   # ANIMAL ADDITIVE GENETIC
               [,1]         [,2]        [,3]
  [1,]  0.077419331 -0.029556261 -0.19075749
  [2,] -0.025356727 -0.010108294  0.22883726
  [3,] -0.052062604  0.039664554 -0.03807977
  [4,]  0.009288360  0.004802451 -0.08615141
  [5,]  0.041504861 -0.026617064 -0.01809211
  [6,] -0.050793221  0.021814613  0.10424352
  [7,]  0.094147066 -0.034191518 -0.24269798
  [8,] -0.049539044  0.023964143  0.08943504
  [9,]  0.009599878 -0.012351672  0.04033591
 [10,]  0.086088207 -0.035828310 -0.19093877
 [11,] -0.021926840 -0.030872198  0.37728555
 [12,] -0.118369267  0.089279555 -0.07341975

  ph = matrix(data=phat,byrow=TRUE,ncol=3)
> ph  # ANIMAL PERMANENT ENVIRONMENTAL
      #  corresponding to animals 7 through 12
             [,1]        [,2]       [,3]
 [1,]  0.11292716 -0.05915988 -0.2491713
 [2,] -0.10021776  0.06779429  0.0375444
 [3,]  0.04467334 -0.04492373  0.1719830
 [4,]  0.05554442 -0.02287056 -0.2095274
 [5,]  0.06423224 -0.08458156  0.5245844
 [6,] -0.17715940  0.14374144 -0.2754131
```

With the solutions we can look at the phenotypic trajectories for the two genders, in Figure 3.1 below.

**Figure 3.1**

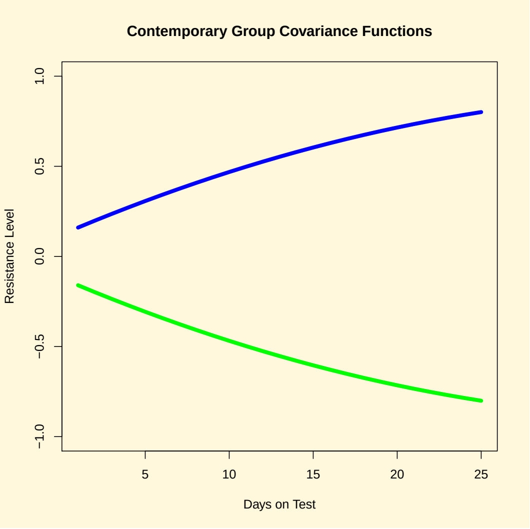


There is definitely a difference in phenotypic curves between the two genders. The plot was made, as follows.

```
gsoln = gh%*%t(fT) # points along 25 days
gs1 = gsoln[1,]
gs2 = gsoln[2,]
par(bg="cornsilk")
plot(gs2,col="blue",lwd=5,type="l",xlab="Days on Test",
  ylab="Resistance Level")
title(main="Gender Trajectories")
lines(gs1,col="red",lwd=5)
```

Similarly for contemporary groups, a plot is made only using Legendre polynomials.

**Figure 3.2**



Note that the solutions for contemporary groups must add to zero, thus, the mirror image in their curves. The next two figures show the curves for animal additive genetic and permanent environmental effects, but only for the animals with records (i.e. animals 7 through 12).

**Figure 3.3**

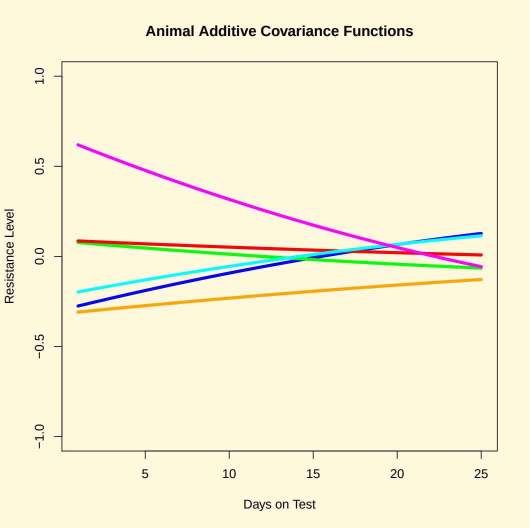
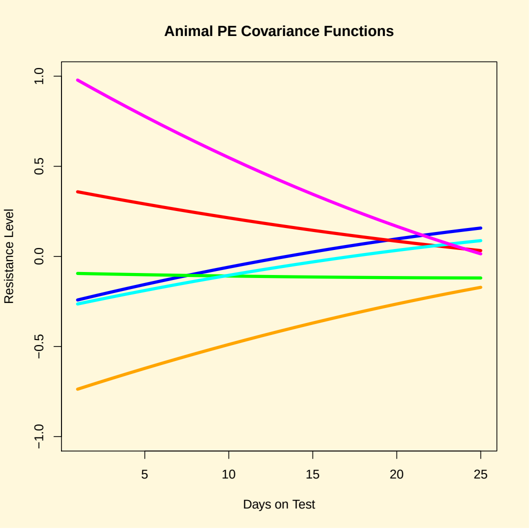


**Figure 3.4**



Looking at the curves for the animal genetic effects, for example, determining the best animal is difficult. Below are the estimated breeding values (EBV) for the animals with records for day 5 of the test and day 20.

**Table 3.3**

EBV for resistance for animals 7 to 12.

| Animal | Day 5 | Day 20 |
|---|---|---|
| 7 | -0.18940620 | 0.06571111 |
| 8 | 0.04650564 | -0.04328822 |
| 9 | 0.06963370 | 0.02054059 |
| 10 | -0.13046155 | 0.06693418 |
| 11 | 0.47701849 | 0.04924207 |
| 12 | -0.27329008 | -0.15913973 |

Assuming high values are good, then animal 11 would be the better animal at day 5 and animals 10 and 7 would be better at day 20. Having a higher value at day 20 means the animal's resistance is not decreasing as much during the low part of the 100 day trajectory, and this may be a good thing. Having animals whose resistance is decreasing much more rapidly at day 20 may not be ideal for the animal.

Thus, one has to decide how best to present the results of curves. It may be one point on the curve, or it could be the area under the curve, or it could be one of the curve parameters that summarizes everything. Each situation or problem is different. The choice must also be readily explainable to producers who may not understand curves and regressions. Producers usually want numbers that are interpretted on their day to day level, such as kilograms of weight, or kilograms of milk yield.

## 3.6 Estimation of Covariance Matrices

The example data are not suitable for estimating covariance matrices due to the small number of observations (21) and animals (12). However, it is useful to illustrate the calculations that would be necessary. The procedure is a pseudo-Bayesian method, where sums of squares of solutions to the MME are calculated and assumed to follow an inverted Wishart distribution, or an inverted Chi-square distribution. The solutions themselves are assumed to follow normal distributions and random noise would be added to solutions during each sample. Many samples would be generated, i.e. 50,000 or more. Allowing for a suitable "burn-in" period, then the remaining samples would be averaged. This section describes one sampling for the covariance matrices.

### 3.6.1 Contemporary Group Matrix

For the contemporary group covariance matrix, calculate the sum of squares and crossproducts of the 3 solutions per contemporary group (there should really be more than 3 contemporary groups for this to work properly, because the new Kc has a rank of only 2). Instead of an inverted Wishart distribution, the whole matrix was divided by a random Chi-square variate with 4 degrees of freedom (number of contemporary groups plus 2). Hence the reason this is called a pseudo-Bayesian approach.

```
pchi=rchisq(1,4)
```

```
#  pchi=5.14983
Kc =t(ch)%*%ch/pchi
Kc
           [,1]        [,2]        [,3]
[1,]  0.1460244 -0.10692449 -0.12401750
[2,] -0.1069245  0.07829408  0.09081022
[3,] -0.1240175  0.09081022  0.10532719
```

### 3.6.2 Additive Genetic Matrix

The sum of squares and crossproducts for the additive genetic covariance matrix involves the inverse of the additive relationship matrix (AI). Again, a random Chi-square variable with number of animals plus 2 degrees of freedom (rather than a Wishart distribution).

```
 pchi=rchisq(1,14)
 # pchi=10.42511
 Ka =t(ah)%*%AI%*%ah/pchi
 Ka
             [,1]         [,2]         [,3]
[1,]  0.003231216 -0.001984457 -0.001825579
[2,] -0.001984457  0.001583572 -0.001872834
[3,] -0.001825579 -0.001872834  0.025616399
```

### 3.6.3 Permanent Environmental Matrix

The calculations are similar to those for the contemporary groups. The degrees of freedom are the number of animals with records plus 2.

```
pchi=rchisq(1,8)
# pchi = 8.717388
Kp =t(ph)%*%ph/pchi
Kp
            [,1]         [,2]         [,3]
[1,]  0.007271474 -0.005466108  0.005349242
```

```
[2,] -0.005466108  0.004411046 -0.007984768
[3,]  0.005349242 -0.007984768  0.055982019
```

### 3.6.4 Residual Variances

One can calculate an overall residual variance for fitting the entire model.

```
 T = cbind(Xg,W)
 res = y - T%*%bh  # residuals
 sse = sum(res*res)
 sse/19
 2.609571  # overall variance
```

In setting up the MME, we used 5 different residual variances depending on the day on test. Thus, there were five groups of five days each.

```
Q = desgn(dgrp,5)
idual=Q*cbind(res,res,res,res,res)

D = t(Q)%*%Q
DI = ginv(D)
ee = t(idual)%*%idual
RR=ee*DI
RR
           [,1]      [,2]     [,3]     [,4]     [,5]
 [1,] 0.5812101 0.0000000 0.000000 0.000000 0.000000
 [2,] 0.0000000 0.9033761 0.000000 0.000000 0.000000
 [3,] 0.0000000 0.0000000 4.025413 0.000000 0.000000
 [4,] 0.0000000 0.0000000 0.000000 1.315003 0.000000
 [5,] 0.0000000 0.0000000 0.000000 0.000000 4.438237
```

### 3.6.5 Heritabilities

Legendre polynomials of order 2 were used in the example, and we can calculate the heritability of each regression coefficient. Assuming the estimates above are the converged final estimates (which they are not), the heritability of the intercept is

$$\begin{aligned} h^2 &= .00323/(.00323\ +\ 0.00727\ +\ .14602\ +\ 2.60957) \\ &= .00117. \end{aligned}$$

We may also do the slope and the quadratic coefficients in the same way. For the slope, the heritability was

$$\begin{aligned} h^2 &= .00158/(.00158 + .07829 + .00441 + 2.60957) \\ &= .00059, \end{aligned}$$

and for the quadratic term was

$$\begin{aligned} h^2 &= .02562/(.02562 + .10533 + .05598 + 2.60957) \\ &= .00916. \end{aligned}$$

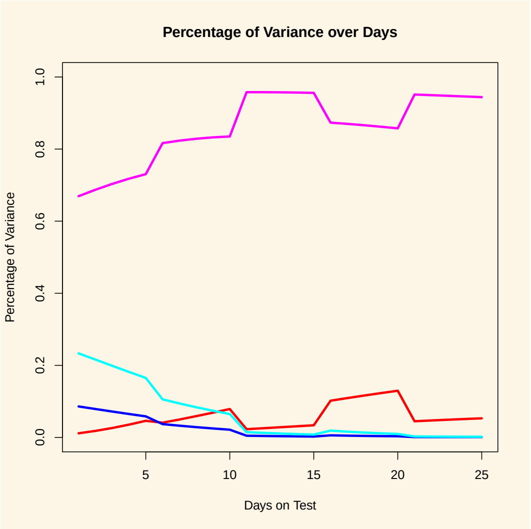
Individually, the parameters are not very heritable. However, this is probably the best way to look at heritability in a random regression model. Unfortunately, many researchers want to use the random regression results to estimate variances for every day in the test period, and thereby estimate daily heritabilities. The first thing is to calculate matrices of dimension 25 by 25 for contemporary group, genetic, and permanent environmental variances and covariances as predicted by the covariance functions, as follows.

```
# CONTEMPORARY GROUPS
  varc = PH%*%Kc%*%t(PH)
  vc = diag(varc)  # 25 variances, each day
# ADDITIVE GENETIC
  vara = PH%*%Ka%*%t(PH)
  va = diag(vara)  # 25 variances, each day
```

```
# PERMANENT ENVIRONMENTAL
  varp = PH%*%Kp%*%t(PH)
  vp = diag(varp)  # 25 variances, each day
# combine into one table
  vtab = cbind(vc,va,vp)
# RESIDUAL VARIANCES
  vres = diag(RR)  # from earlier section
  v1=vres[1]
  v2=vres[2]
  v3=vres[3]
  v4=vres[4]
  v5=vres[5]
  R = c(v1,v1,v1,v1,v1, v2,v2,v2,v2,v2,
      v3,v3,v3,v3,v3,
      v4,v4,v4,v4,v4,  v5,v5,v5,v5,v5)
  vtt = cbind(vtab,R)
# total sum of individual variances by day
  pvar = vtt[ ,1]+vtt[ ,2]+vtt[ ,3]+vtt[ ,4]
  D=diag(pvar)
  DI=ginv(D)
# Convert absolute values to percentages of pvar
  HH=DI%*%vtt
  Hc = HH[ ,1]  # contemporary group
  Ha = HH[ ,2]
  Hp = HH[ ,3]
  Hr = HH[ ,4]
  par(bg="oldlace")
  plot(Hc,type="l",lwd=3,col="red",xlab="Days on Test",
          ylab="Percentage of Variance",ylim=c(0,1))
  title(main="Percentage of Variance over Days")
  lines(Ha,lwd=3,col="blue")
  lines(Hp,lwd=3,col="cyan")
  lines(Hr,lwd=3,col="magenta")
```

Plotting the daily percentage values over the 25 day test period shows how variation changes over time. The additive genetic and permanent environmental decrease over time and are small throughout. Contemporary group variance is relatively much higher. I caution the reader again, to remember that the estimates obtained here for this small example are not totally converged, and they are quite meaningless. There should be many thousands of animals and observations in order to obtain any appropriate estimates.

**Figure 3.5**



The distinct dips and hikes are due to the residual variances changing substantially from one group to the next (the magenta line in Figure 3.5). Again, there were not sufficient data to obtain good smooth estimates of residual variances.

# Chapter 4

# Lactation Production

When a female dairy animal gives birth this begins her lactation period. In an average dairy cow, the lactation period runs for 305 days. The amount of milk produced is greatest after calving and peaks about 40 days with daily production decreasing thereafter. Around 100 to 120 days after calving, the cow is impregnated again through artificial insemination, and the growth of the new fetus begins to pull lactation production downwards even more. About 60 days before the cow is due to give birth again, she is stopped from milking (if she has not already stopped on her own) and *dried off*. The whole process is repeated as soon as she has the next calf, approximately 13 month intervals.

Some cows have their peak production right at birth of a calf and it decreases continually thereafter. Other cows continue to increase in yield from birth to day 90, before starting to decrease. Another group of cows gives milk at a continuous level for many days. Some cows stop milking around 280 days, while others are kept milking to 365 days or more. Thus, there are many different shapes of daily production trajectories between cows.

Besides milk yield, there are also components of milk, i.e. percentages of fat, protein, and lactose, somatic cell scores, milk urea nitrogen, and betahydroxybutylase. All of these have their own interrelated trajectories over the lactation period. Consequently, multiple trait analyses are favoured to make use of genetic correlations among the traits. Luckily, the same model equation is generally assumed for all of these traits. That is, the same factors are assumed to influence each lactation trait.

## 4.1 Measuring Yields

In the early days of milk recording in Canada, the federal government would record the daily milk yield of every cow in the herd. The trait that was analyzed was called the 305-day yield. This was the sum of the daily yields of cows from day 5 to 305 in the lactation period (or whenever the cow stopped milking). Daily yields were defined as the amount of milk given in a 24 hour period. This was usually two milkings per day, morning or AM milking, and evening or PM milking.

However, it was very costly in terms of people time to measure the amount of milk per cow every day, and someone had to add up the milk weights over the lactation period. The program ended before 1970.

At the same time there was a program of supervised testing, in which a milk supervisor would visit a farm, at approximately one month intervals. He would measure yields in the PM and following AM, and collect milk samples of each cow, which were sent to laboratories to be analyzed for fat content. The data were accumulated by the milk recording program, which was called Record of Performance (ROP). Later, provincial programs arose called Dairy Herd Improvement (DHI) programs. The monthly milk weights were combined using the Test Interval Method (TIM), which estimated the amount of milk produced between two visits by linear interpolation. There were tables of special factors to adjust the first test day (TD) visit, and another table for *projecting* the yield after the latest TD visit to 305-days. Both tables were based on the assumption of a standard or average lactation trajectory. There was no allowance for the fact that cows could vary drastically in their trajectories. The factors worked well for most cows, but gave biased results for cows with atypical trajectories.

A dairy technical committee existed in Canada, run by Agriculture Canada and composed of scientists from different universities across Canada. The purpose of the committee was to advise the ROP program on the best statistical procedures to adjust milk yields and to evaluate dairy bulls. The committee met twice a year in various locations across Canada. During one of these meetings when it was time to develop new tables of factors for adjusting the first TD yields and the latest TD yields to get 305-day production, there was debate over who would do this and how often it needed to be done. Dr John Moxley, who worked for the Quebec DHI equivalent, DHAS at the time, made a remark in 1974 that "it would be better if we could analyze TD milk yields directly rather than combining them into a 305-d yield." In 1974, however, we were still trying to get a linear sire model adopted to evaluate dairy bulls, so the computing power of the day was not capable of handling models for test day records. However, the idea of a linear model for test day yields was planted solidly in my head that day. I thought about it all the time. By 1990 the dairy world had advanced to using animal models, and computer hardware had caught up, so that it was feasible to begin working on TD models.

My first TD model did not have any curves in it. The model assumed that the trajectories of the curves were the same for all cows. Trajectories only differed by the height at peak yield. Thus, there was still only one variable to estimate per animal. The problem with analyzing TD records was that each cow had 7 to 10 TD records compared to only one 305-d record. There was much more data to process.

Jack Dekkers said to me one day, "there should be a different lactation curve for every cow" to which I readily agreed. Then I thought of the problem of estimating a curve for each cow, and how I could build in the additive genetic relationships. Then Jack said "random regressions" as a passing thought, regressions but they would be a random factor in the model. I immediately knew that was the "obvious" solution. I looked through Dr Henderson's 1984 book that same day, and was astounded to find a section on the topic of "random regressions." Unfortunately, there was only one paragraph and nothing about use with TD models. Henderson's son published a paper in *Biometrics* in 1982 on random regression models and the analysis of covariance.

Working with Ewa Ptak of Poland we tried different TD models. In 1994 I presented the idea of TD models using random regression models to the WCGALP meetings in Guelph. Four years later, everyone was studying random regression models for many situations. By 2000, Canada had adopted a TD model for its genetic evaluations of dairy bulls and cows.

## 4.2 Curve Fitting

A host of different models have been used to fit lactation curves in different species of dairy cattle. Kistemaker (1996) compared almost 20 different models that had been studied in the literature previously. His results are shown in Table 4.1.

**Table 4.1**
Correlations (r) between Predicted and Actual
Test Day Yields and Mean Absolute Error (MAE)
when applied to 5409 cows with at least 9 TD yields.

| No. | Model[1] | r | MAE |
|---|---|---|---|
| 1 | $\ln(y/t) = a + b\dot{t}$ | .717 | 4.780 |
| 2 | $\ln(y) = a + b \cdot \ln(t) + c \cdot t$ | .951 | 1.290 |
| 3 | $\ln(y) = a + b \cdot \ln(t) + c \cdot t + d \cdot t^{.5}$ | .963 | 1.084 |
| 4 | $\ln(y) = a + b \cdot \ln(t) + c \cdot t + d \cdot t^2$ | .964 | 1.079 |
| 5 | $\ln(y) = a + b \cdot t^{-1} + c \cdot t + d \cdot t^2$ | .964 | 1.063 |
| 6 | $\ln(y) = a + b \cdot \ln(t) + c \cdot t * d \cdot t^{.5} + f \cdot t^2$ | .973 | 0.888 |
| | | | |
| 7 | $1/y = a + b \cdot t^{-1} + c \cdot t$ | .102 | 2.050 |
| 8 | $1/y = a + b \cdot t^{-1} + c \cdot t + d \cdot t^2$ | .766 | 1.269 |
| 9 | $1/y = a + b \cdot t^{-1} + c \cdot t + d \cdot t^2 + f \cdot t^3$ | .378 | 1.078 |
| | | | |
| 10 | $y = a$ | .646 | 3.466 |
| 11 | $y = a + b \cdot t + c \exp(-.5(\log(t) - 1)/.6)^2 \cdot t^{-1}$ | .953 | 1.229 |
| 12 | $y = a + b \cdot t^{.5} + c \ln(t)$ | .955 | 1.230 |
| 13 | $y = a + b \cdot t + c \exp(-.05 \cdot t)$ | .953 | 1.232 |
| 14 | $y = a + b \cdot t^{.5} + c \ln(t) + d \cdot t^4$ | .967 | 1.032 |
| 15 | $y = a + b(t/305) + c(t/305)^2 + d \ln(305/t) + f \ln^2(305/t)$ | .975 | 0.857 |
| 16 | $y = a + b \cdot t + c \sin(.01)t^2 + d \sin(.01)t^3 + f \exp(-.055t)$ | .974 | 0.878 |
| 17 | $y = a + b \cdot t + c \cdot t^2 + d \cdot t^3 + f \ln(t)$ | .975 | 0.864 |
| 18 | $y = a + b \cdot t + c \cdot t^2 + d \cdot t^3 + f \cdot t^4$ | .974 | 0.905 |
| 19 | $y = a + b \cdot t + c \cdot t^2 + d \cdot t^3 + f \cdot t^4 + g \cdot t^5 + h \cdot t^6$ | .987 | 0.581 |

Wood's model (1967) has been used to study groups of cows and is equation 2 in the table. Equation 13 is known as Wilmink's function

(1987) which has been applied in many studies. Equation 15 is known as the Ali and Schaeffer function (1987), which gives the second smallest mean absolute error and the second largest correlation. Equation 19 appears to be the best, but has the most parameters to be estimated. The first 9 equations use the natural log of the test day yields or the inverse of yield. Equations 10 through 19 use the actual TD yields.

As you increase the number of covariates in the model, the better is the fit of the model, in general. Subsequent work showed that Legendre polynomials of order 4 were similar to the Ali and Schaeffer function, but had the advantage of having much lower correlations among the parameter estimates. Thus, Legendre polynomials of order 4 have been used for both fixed and random regressions in test day models in Canada.

A classification approach could be used for the fixed factor regressions for at least one of the fixed factors. There have probably been a hundred different studies that investigated the best curve function for fitting lactation curves in dairy cows, dairy goats, dairy sheep, and water buffalos. The conclusions have not been unanimous, depending on the amount of data in the analyses. The majority of studies found test day models gave higher correlations of estimated breeding values with true breeding values, and recommended their use for genetic evaluations of lactation production.

## 4.3 Factors in a Model

### 4.3.1 Observations

A multiple trait model will be described. Traits will be defined within parity number. Parities one and two are separate, and parity three includes third parity and all subsequent parities. The assumption is that cows in third parity or later are mature and the shape of their curves are similar. In some cases parity two might also be considered mature. No matter what, there are definite shape differences between parity 1 heifers and all later parities. In some situations it might be better to limit analyses to the first three parities only.

After parity number come several traits, depending on the country. These include milk yield, fat yield, protein yield, lactose yield, and somatic cell scores (SCS). There could also be milk urea nitrogen and betahydroxybutylase. Finally there could be fatty acid components. Deciding which factors to analyze depends on how many cows have data. In the United States of America, for example, there are too many cows and too many test days, such that a TD model is impractical to apply, even for one trait. The initial Canadian Test Day Model had the first three parities, and milk, fat, and protein yields plus SCS or a total of 12 traits.

### 4.3.2 Year-Month of Calving

In all animal models it is critically important to account for time trends in phenotypes. For lactation production this means putting in a factor for the year and month of calving. If data begin in 1986, then that means 30 years (it is now 2016), times 12 months per year, gives 360 levels or 360 different lactation curves for one parity. Then assume 72 five-day periods within each lactation and that gives 25,920 parameters to be estimated using the classification approach. Hopefully, there are many more test day records with which to estimate those parameters. If there are not, then maybe 36 ten-day periods could be used. If data are limiting, then Legendre polynomials of order 4 could be used, i.e. 1800 parameters to estimate.

### 4.3.3 Age-Season of Calving

Age at calving (parturition) is known to have a significant effect on milk production, as does month of calving. However, month of calving has already been considered in the Year-Month of Calving effects. However, there is an interaction of month of calving with age at calving. To avoid some confounding, months can be combined into seasons either six or four seasons per year. These can be formed on the basis of phenotypic averages, so that consecutive months that are similar in yield levels can be grouped together.

Age groups would differ depending on the parity number, and there could be different numbers of age groups per parity. First parity heifers start calving at 18 months of age, and can extend to 30 months. Again, if there are lots of data, then 13 age groups by 6 seasons would only be 78 subclasses in parity one. Later parities extend over a much wider age range, and thus, some groupings of ages may be necessary too. Legendre polynomials would be used for this factor.

If the data cover several decades, then age-season differences could change over time as production increases. Thus, time periods of 5 to 10 years should be made and the model expanded to have Time-Age-Season of Calving subclasses. This allows the age-season differences to change over time.

### 4.3.4 Days Pregnant

Once a cow becomes pregnant, part of her feed intake goes towards the growth of the fetus, and therefore, less energy goes towards milk production. Groups of 5 or ten days can be created, perhaps 30 groups altogether, to measure the decrease in yield. The assumption is that the decrease in yield is the same regardless of number of days in milk when the cow becomes pregnant. As the number of days pregnant becomes larger, so does the amount of decrease in yield. Legendre polynomials of days in milk would be used within each days pregnant group.

Determining the time of conception is not immediate, and therefore, test day records need to be continuously updated when pregnancy is validated. Canada has opted to multiplicatively pre-adjust TD yields for number of days pregnant rather than to put a factor into the model.

### 4.3.5 Herd-Test-Day

This is probably the biggest mistake I made with the original TD models. The purpose of this factor was to account for the environmental effects on the cows that were tested on the same day. This is very messy because cows (in the same parity) would have calved at different ages and months of the year. Thus, some cows would be just starting a lactation, and other cows (in the same HTD subclass) could be ending their lactation. Thus, the yields would be all over the place. There would only be one parameter to estimate for each HTD subclass. The contemporaries would be constantly changing from TD to TD.

After much thinking about this factor, it did not make any sense to me. Thus, I DO NOT RECOMMEND this factor in TD models. Instead one should use Parity-Herd-Year-Season of calving contemporary groups.

### 4.3.6 Parity-Herd-Year-Season of Calving

The random factor of Parity-Herd-Year-Season of Calving, (PHYS), each with its own curve (not one parameter but five), should be used to account for contemporaries. Contemporaries are cows that share the same environmental effects throughout their lactation, from birth to being dried off. They encounter the same weather and management variables through-out. They likely also have the same test days during their lactations. I also suggest using 4 seasons per year of 3 months each. However, if number of cows per subclass is small, then maybe larger season groups (4 months or 6 months) may be necessary in some herds, especially for the less numerous breeds.

Legendre polynomials of order 4 should be used with this factor, and hence a covariance function matrix needs to be estimated for it. As a random factor in the model, it is less critical to have a minimum number of records per subclass because just one test day record will suffice.

### 4.3.7 Additive Genetic Effects

Every animal with TD records has both parents identified. If a parent is unknown, then a phantom parent group is assigned. Ancestors, without TD records, also have unknown parents replaced by phantom parent groups. The groups are based on year of birth of the animal and whether it is a male or female animal. Phantom groups represent the four pathways of selection, in dairy cattle, and years of birth. Phantom groups are necessary in order to properly estimate genetic trends, unbiasedly.

Each animal additive genetic effect is fitted by a Legendre polynomial of order 4. A covariance function matrix must also be estimated.

### 4.3.8 Permanent Environmental Effects

Because cows have more than one TD record per lactation, permanent environmental effects are modeled for each parity by Legendre polynomials of order 4. A covariance matrix is needed for this factor too.

### 4.3.9 Number Born

The number of offspring born at a parturition, in litter bearing species such as dairy goats and sheep, can have an effect on the milk yield of the female. A female carrying four young apparently “knows” this is happening and the body prepares by increasing the amount of milk that will be needed after birth to feed that number of young. This is a fixed environmental effect and might differ depending on parity number of the dam, but it can be fit by Legendre polynomials of order 4.

### 4.3.10 Residual Effects

In the Canadian Test Day Model, the lactation is divided into 4 periods of various numbers of days, such that the residual variance is similar across days within a period, but different between periods. One should begin using many groups, perhaps 30 of ten days each, in an initial analysis to determine the best grouping of days. The point is, the residual variance changes throughout the lactation.

Table 4.2 contains residual variances for milk yields in the first three parities for a small subset of Canadian Holstein dairy cattle born from 2005 through 2009.

**Table 4.2**

Residual variances for a TD model.

| Days in Milk | Parity 1 | Parity 2 | Parity 3 |
|---|---|---|---|
| 1-45 | 7.86 | 13.96 | 16.42 |
| 46-115 | 5.01 | 8.12 | 9.33 |
| 116-265 | 3.95 | 5.41 | 6.24 |
| 266-365 | 3.57 | 4.36 | 3.60 |

## 4.4 Covariance Function Matrices

Many of the early studies of random regression models focussed on the estimation of the covariance function matrices, and the subsequent graphs that could be made. Let $\mathbf{a}_i$ represent the vector of random regression coefficients of an animal for parity $i$. This vector is order 5 by 1 (order 4 Legendre polynomial). Then an analysis of 3 parities gives a covariance function matrix of order 15 by 15. The parts of this matrix are shown below in order 5 by 5 subgroups.

$$Var(\mathbf{a}_1) = \left( \begin{array}{rrrrr} 8.1910 & 0.2880 & -0.6694 & 0.2360 & -0.1407 \\ 0.2880 & 1.4534 & -0.1327 & 0.4590 & 0.4926 \\ -0.6694 & -0.1327 & 0.5108 & -0.1512 & 0.0713 \\ 0.2360 & 0.4590 & -0.1512 & 0.1855 & -0.0524 \\ -0.1407 & 0.4926 & 0.0713 & -0.0524 & 0.0766 \end{array} \right),$$

$$Cov(\mathbf{a}_1, \mathbf{a}_2) = \left( \begin{array}{rrrrr} 8.3749 & 0.4892 & -0.5377 & 0.2686 & -0.0281 \\ 1.486 & 1.343 & -0.1682 & -0.0111 & -0.0232 \\ -0.7102 & 0.2314 & 0.2984 & -0.2155 & 0.0487 \\ 0.3355 & -0.0659 & -0.1044 & 0.1136 & -0.0136 \\ -0.1365 & 0.0979 & 0.0705 & -0.0456 & 0.0072 \end{array} \right),$$

$$Cov(\mathbf{a}_1, \mathbf{a}_3) = \left( \begin{array}{rrrrr} 8.1921 & 0.8613 & -0.5547 & 0.2076 & -0.0466 \\ 1.6933 & 1.1975 & -0.2810 & 0.0614 & -0.0956 \\ -0.6192 & 0.1654 & 0.2865 & -0.1721 & 0.0372 \\ 0.3242 & -0.0281 & -0.0950 & 0.1168 & -0.0478 \\ -0.1003 & 0.1086 & 0.0856 & -0.0424 & 0.0194 \end{array} \right),$$

$$Var(\mathbf{a}_2) = \begin{pmatrix} 12.0818 & 1.6093 & -0.6870 & 0.4367 & -0.1217 \\ 1.6093 & 3.0648 & 0.0456 & -0.2809 & 0.0247 \\ -0.6870 & 0.0456 & 0.5917 & -0.1626 & 0.0166 \\ 0.4367 & -0.2809 & -0.1626 & 0.4004 & -0.1092 \\ -0.1217 & 0.0247 & 0.0166 & -0.1092 & 0.1696 \end{pmatrix},$$

$$Cov(\mathbf{a}_2, \mathbf{a}_3) = \begin{pmatrix} 11.4893 & 1.9730 & -0.9016 & 0.4398 & -0.2730 \\ 1.8533 & 2.7023 & -0.1320 & -0.2323 & -0.0117 \\ -0.6313 & -0.0365 & 0.2919 & -0.2010 & 0.8696 \\ 0.2867 & -0.2212 & -0.1646 & 0.2461 & -0.0805 \\ -0.0897 & 0.0631 & 0.0654 & -0.0579 & 0.0185 \end{pmatrix},$$

and

$$Var(\mathbf{a}_3) = \begin{pmatrix} 13.5971 & 1.7354 & -0.8951 & 0.3615 & -0.3546 \\ 1.7354 & 3.8151 & -0.2634 & -0.1821 & 0.0662 \\ -0.8951 & -0.2634 & 0.9197 & -0.2324 & 0.0873 \\ 0.3615 & -0.1821 & -0.2324 & 0.5535 & -0.1615 \\ -0.3546 & 0.0662 & 0.0873 & -0.1615 & 0.2291 \end{pmatrix}.$$

A plot of the genetic variances within parities and across the lactation period can be obtained as shown below.

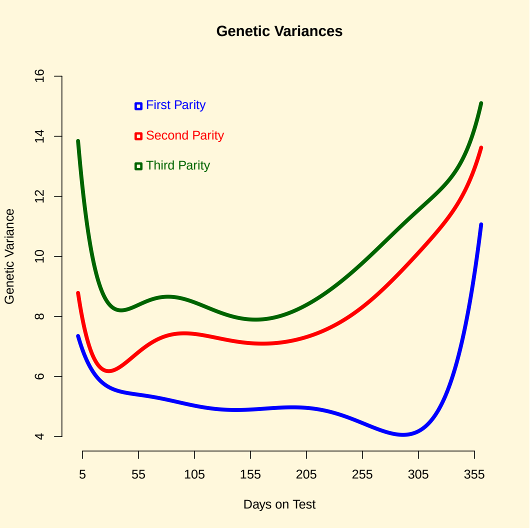
```
# Legendre polynomials
  LAM=LPOLY(5)
  ti=c(5:365)
  tmin=5
  tmax=365
  qi = 2*(ti - tmin)/(tmax - tmin) - 1
  x=qi
  x0=x*0 + 1
  x2=x*x
  x3=x2*x
  x4=x3*x
  M=cbind(x0,x,x2,x3,x4)
```

```
       PH = M %*% t(LAM)
      Ka1 = matrix(data=c(8.1910, 0.2880,-0.6694,0.2360, -0.1407,
       0.2880, 1.4534, -0.1327, 0.4590, 0.4926,
      -0.6694, -0.1327, 0.5108, -0.1512, 0.0713,
       0.2360, 0.4590, -0.1512, 0.1855, -0.0524,
      -0.1407, 0.4926, 0.0713, -0.0524, 0.0766),byrow=TRUE,ncol=5)
     Va1 = PH%*%Ka1%*%t(PH)  # order 361 x 361
     vg1 = diag(Va1)
   #  similar arrays for vg2, vg3, Ka2, Ka3 (not shown)
     par(bg="cornsilk")
     plot(vg1,col="blue",lwd=5,type="l",axes=FALSE,xlab="Days on Test",
       ylab="Genetic Variance",ylim=c(4,16))
     axis(1,days)
     axis(2)
     title(main="Genetic Variances")
     lines(vg2,col="red",lwd=5)
     lines(vg3,col="darkgreen",lwd=5)
     points(55,15,pch=0,col="blue",lwd=3)
     text(55,15,"First Parity",col="blue",pos=4)
     points(55,14,pch=0,col="red",lwd=3)
     text(55,14,"Second Parity",col="red",pos=4)
     points(55,13,pch=0,col="darkgreen",lwd=3)
     text(55,13,"Third Parity",col="darkgreen",pos=4)
```

The plot is shown in Figure 4.1. An obvious observation is that there are distinct differences in the variance curves across the lactation between parities. Also, the genetic variance is highest at the beginning of lactation and at the end of lactation, for each parity. This implies that there are great differences between cows in the amount of milk produced at the start of lactation, then after 55 days the variances are smaller by nearly half, but tend to increase upwards again towards day 365.

Some researchers have interpretted the high variances at the start and end of tests as artifacts of the Legendre polynomials. However, similar shapes are obtained using other polynomials (e.g. Ali and Schaeffer, 1987) of order 4. Spline functions tend to flatten the beginning and end a little more, but the general shape persists.

**Figure 4.1**



The only way to determine the correct shape of the variances is to use a multiple trait model where yields are divided into 36 ten-day periods, then genetic variances may be estimated for each period, and also covariances between periods. Then a Legendre polynomial of order 4 could be fit to the 36 by 36 covariance matrix, and compared to the estimates from the test-day model.

My opinion is that the shape of these variance curves is not important, but rather the entirety of the results which includes the estimated breeding values. The residual variances are greatly reduced. The variances that need to be correct are $Cov(\mathbf{a}_i, \mathbf{a}_j)$ for all pairs of parities.

## 4.5 Expression of EBVs

Estimated breeding values (EBV) in random regression models, come in vectors of length equal to the order of the Legendre polynomials. The problem was how to condense 5 breeding values for a curve into one value for a single trait, like milk yield. Dairy cattle producers were used to a standard called “305-day yields”. The solution was to calculate the daily milk yield per day of lactation, and then to sum those daily yields from day 5 through 305. (The first 4 days of yield were typically used to feed the newborn calf and provide its colostrum, or immunization.) Let the solutions for one animal's additive genetic value for first parity milk yield be

$$\mathbf{a}_{1i} = \left( \begin{array}{ccccc} a_{1i0} & a_{1i1} & a_{1i2} & a_{1i3} & a_{1i4} \end{array} \right),$$

then daily yield $(DY)_{ij}$ for animal $i$ on the $j^{th}$ day would be

$$DY_{ij} = \phi_{j0}a_{1i0} + \phi_{j1}a_{1i1} + \phi_{j2}a_{1i2} + \phi_{j3}a_{1i3} + \phi_{j4}a_{1i4},$$

where $\phi_{jm}$ is a Legendre polynomial covariate. The 305-d milk yield, $M305_i$, is the sum of the daily yields,

$$M305_i = \sum_{j=5}^{305} DY_{ij}.$$

Because the breeding values are constant for the calculation of every daily yield, then

$$M305_i = (\sum_{j=5}^{305} \phi_{j0})a_{1i0} + (\sum_{j=5}^{305} \phi_{j1})a_{1i1} + (\sum_{j=5}^{305} \phi_{j2})a_{1i2} + (\sum_{j=5}^{305} \phi_{j3})a_{1i3} + (\sum_{j=5}^{305} \phi_{j4})a_{1i4}$$

or

$$M305_i = c_0a_{1i0} + c_1a_{1i1} + c_2a_{1i2} + c_3a_{1i3} + c_4a_{1i4},$$

where the $c_j$ are constants, and represent the sum of the Legendre polynomial coefficients, which can be obtained by the following R script.

```
 PH
 ka=c(1:301)
 P305 = PH[ka, ]
 C305 = t(P305)%*%jd(301,1)
 C305
[1,] 212.839141
[2,] -61.441368
[3,] -51.778637
[4,] -29.763643
[5,]  -1.346922
```

Now multiply the constants times the EBVs of the random regression coefficient solutions for each animal, and you have 305-d EBVs for ranking animals.

One can question if 305 days should be the standard length of lactation. In 2016 many cows lactate for longer than 305 days, and the analysis was for test day yields up to 365 days. So a new standard could be 365 day yields. The constants to use for that standard would be

```
[1,]  255.2655
[2,]    0.0000
[3,]    1.5855
[4,]    0.0000
[5,]    2.1410
```

For dairy sheep and goats the standard length might be less than 305 days because those two species do not lactate as long as cattle.

Note that in the dairy cattle example, it is not valid to calculate EBV for daily yields beyond 365 days because only test day yields from days 5 to 365 were analyzed.

### 4.5.1 Other Expressions

One of the first new EBV in dairy cattle as a result of random regressions was persistency. Persistency is the ability of a cow to milk at a high level over much of the lactation period. This would allow for better feeding of animals, which could be sub-housed according to high, medium, or low persistency. The trouble was how to define persistency in a random regression model setting.

The variable, $a_{1i1}$, was itself a measure of persistency, but it did not have any units. Animals with high values were more persistent. Because dairy producers could not relate to this number, other measures were proposed. The idea was to have some number that represented the downward slope of the curve after the peak yield of lactation. Cows differed in the day on which they expressed peak yield, so the initial point had to be well after the day of peak yield. The measure was also desired to be independent of peak yield or total 305-d yield.

Suppose the yield on day 60 of an average, first parity cow was 90 kg of milk and on day 260 was 68 kg. Then calculate

$$Persist = \frac{DY_{260} + 68}{DY_{60} + 90}$$

which should be a number from 0 to 1, in most situations. The higher is the value, then the more persistent is the cow. The average, first parity cow would have a value of (68/90) = 0.756. Later parity cows tend to have lower persistency than first parity heifers. Note that it is possible for a cow to have a persistency value greater than one, but that should happen very infrequently.

# Chapter 5

# Growth

Growth curves have been studied in many species of plants and animals, but usually with non-linear models. Growth is the accumulation of size and mass of an organism over time. For most agricultural species, growth to maturity takes only 3 to 4 years at most, but for humans and other larger mammals, growth can take decades.

In beef cattle, growth is important from birth until the animal reaches market age, or often only the period from weaning to one year of age is of interest. In the latter period, growth can be considered almost linear, with a slight quadratic shape. Early growth from birth to weaning is often ignored.

A non-linear mathematical model that describes growth from birth to maturity is the Gompertz function, where weight at time $t$, $WT_t$, is given by the following equation.

$$WT_t \ = \ BW \ + \ A \cdot [1.0 - \exp(-\exp(B) \cdot (t^C))]$$

where

$t =$ unit of time, usually in days,

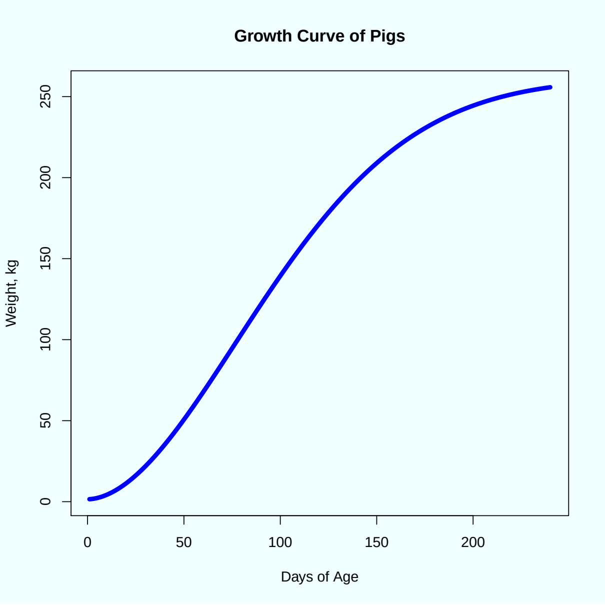
$BW$ is average birthweight,

$A$, $B$, and $C$ are parameters that define the shape of the growth curve.

$A$ is related to mature weight, $B$ is related to the day of change from increasing growth rate to decreasing growth rate, and $C$ is related to the steepness of growth, or how quickly an animal grows to maturity.

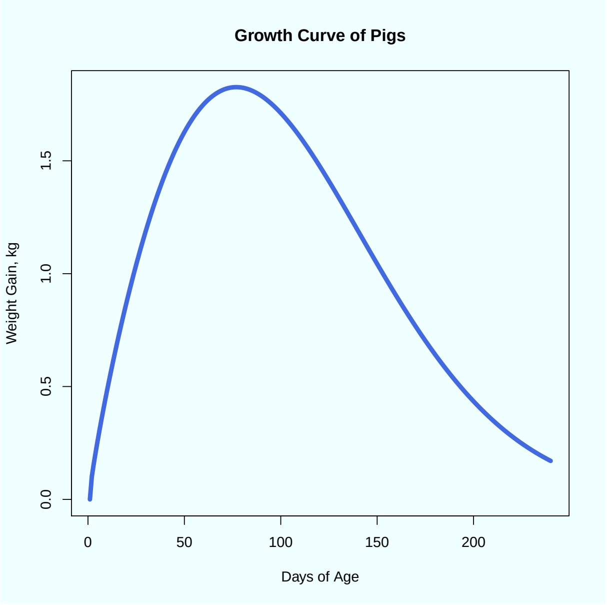
Predicted body weights are positive at all ages, and weights never decrease, unless an animal is being starved. Another advantage is that there are only 4 parameters to estimate, which mean we need 5 or more weights per animal. Unfortunately, we need to solve a nonlinear system. A *differential evolutionary algorithm* can be used to solve. Figure 5.1 shows the growth curve of a pig from birth to maturity, where $A = 272$, $B = -12.8$, and $C = 2.65$, and we assume the birthweight is $1.5kg$.

**Figure 5.1**



The figure emphasizes that growth is cumulative. With the curve we can look at the amount of weight gained each day, as in Figure 5.2. This is known as average daily gain, ADG. As can be seen in the figure, ADG is not constant over the growth period.

**Figure 5.2**



Hence from birth to about 100 days of age, pigs are putting on weight faster and faster. After 100 days, their rate of gain declines. There are problems with measuring ADG. Firstly, the magnitude of ADG is small, only 1 to 2.5 kg per day, so that weigh scales must be precise. Secondly, weight gain depends on the time of day in which it is taken. Did the pig just defecate or just eat breakfast? The amount eaten or lost could be as much as 1 to 2.5 kg. Lastly, you need to weigh pigs every day and this would be very labour intensive, unless it was computerized and automated. The amount of variation in ADG from day to day would be large for one animal.

Cumulative weights keep getting larger as the animal ages. Total weights can be off 1 to 2.5 kg without changing the growth curve dramatically, and the pigs do not need to be weighed daily, but obviously there are key times when pigs should be weighed. Birthweights, tend to be small relative to mature weights. Thus, whether it is 1.5 kg or 3 kg at birth, does not alter the growth curve substantially, but weights at 200 days of age can differ by 10 to 20 kilograms between animals giving very different growth curves.

## 5.1 Curve Fitting

### 5.1.1 Spline Function

Random regression models are linear models, thus the nonlinear Gompertz function needs to be approximated by linear regressions. The phenotypic shape may be approximated by a spline function. Let $t_{max}$ be the maximum age, and in terms of the pig growth curve, let that be 240 days of age. $t_{min}$ is day 1, and let $T = t/t_{max}$, and $U = (t-100)/t_{max}$ for $t > 100$ otherwise $U = 0$. Day 100 is when growth rate starts to decrease with age (Figure 5.2). The phenotypic curve might be

$$y_t = b_0 + b_1 T + b_2 T^2 + b_3 T^3 + b_4 U + b_5 U^2 + b_6 U^3.$$

Estimates of the regression coefficients from the data in Figure 5.1 are

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{pmatrix} = \begin{pmatrix} 3.77839 \\ -96.82071 \\ 1038.66035 \\ -393.28000 \\ 81.43243 \\ -1306.77963 \\ 579.37098 \end{pmatrix}.$$

A seven covariate function to model the trajectory of growth seems too large to be practical. There are places along the curve that are not fit well. At the beginning of the growth curve, the spline function will predict that weights actually decrease after birth, and then turn upwards. Also, weights do not plateau at maturity, but actually begin to decrease. The errors in the prediction are at the beginning and end. Inbetween weights are predicted relatively accurately. The inflection point of 100 days was assumed known, but this point would not be 100 days for every animal, and would need to be estimated. Thus, the spline function is not totally suitable.

### 5.1.2 Classification Approach

Given the problems with the spline function, the classification approach could possibly work much better, for all groups animals, without making any assumptions about the shape of the growth curve, or the position of the inflection point. Over the 240 day age range, make 48 five-day periods and estimate the mean yield within each period. Unfortunately, that requires estimating 48 means per curve, and thus there needs to be a lot of data points within each mean.

## 5.2 Model Factors

### 5.2.1 Observations

Growth observations can be weight, height, length, feed intake, backfat thickness, or loin eye area. Depending on the situation, the growth period could be from birth to weaning, weaning to slaughter weight, or birth to maturity. If the period is short term, often growth is linear during this period. A lifetime curve would look the same as in Figure 5.1. This determines the order of the random regression covariates that are required. If growth is after weaning, then maternal genetic effects may be unnecessary and safely ignored. So the factors listed in this section may or may not be needed, but should at least be considered in developing a working model for growth.

On a per animal basis there should probably be five or more measures of growth. Management systems where animals can be weighed automatically every day should be considered, or where feed intake can be recorded daily. However, if the management system does not allow weighing more than four times during the life of the animal, then random regression models should not be applied, but multiple trait models should be considered as an alternative, where each weight is a different trait, like birth, weaning, and end of test weights.

### 5.2.2 Breed-Year-Month of Birth-Gender

The first fixed factor in the model needs to account for time trends in growth curves for each breed and gender separately. Breeds are usually analyzed separately, but sometimes if breeds are not very numerous, they might be analyzed together because several breeds may appear under one owner. This happens in sheep in Canada. There are about four main breeds and 55 others represented in small numbers. Separate breed analyses are not viable, so all breeds are analyzed as one breed. Indeed the majority of Canadian sheep are crossbreds, of many different combinations of breeds. In the model we need to account for different breed growth curves. It may be necessary to group breeds together according to their growth similarities. There are 25 breed groupings in the Canadian sheep data.

In some species the male is sometimes neutered, and so a third gender is needed for these animals, even if the neutering occurs later, after weaning for example.

The classification approach will be used for this factor. Thus, 48 period means within each subclass implies there should be more than 48 observations within the subclass. Assuming one breed only, then 20 years of data, times 12 months of birth per year, times 3 genders, gives 720 subclasses. Assuming a minimum of 50 observations per subclass, then there should be more than 36,000 weight measures.

### 5.2.3 Maternal Genetic Effects

Growth, in mammals, is a trait that is influenced by *maternal genetic* effects (Willham, 1960s). That is, the female that gives birth provides an environment during the early growth period of that offspring. Maternal effects decrease as the animal ages and becomes more independent. However, some maternal effects can persist a long time. The female provides this environment to every offspring. Her genetic maternal ability is passed along to her progeny (male and female), but is only expressed when her female progeny have their own offspring.

Usually, maternal genetic effects are genetically correlated with the direct genetic effects, and this is often a negative correlation. The maternal genetic effects complicate random regression models for growth. In studies that look at growth after weaning, maternal genetic effects are sometimes ignored, which makes the analysis by random regressions easier, but perhaps biased. Maternal effects are funneled into the direct genetic and residual effects.

If one is working with a species in which embryo transfer and cross-fostering are employed, then it is possible that an animal could have three different dams affecting their growth.

The first dam is the *genetic-dam*, the female ancestor that provides DNA to the offspring. If the fertilized egg is put into another unrelated female (i.e. recipient), then the *birth-dam* is the female that carries the fetus until it is born. Once the progeny is born, it may be necessary to cross-foster to another dam, known as the *raise-dam*. The different dams are associated with maternal genetic effects. The raise-dam should receive the maternal credits for raising a progeny after it has been born. Most of the time the three dams are the same individual, but there can be substantial numbers of animals with two or three different dams affecting their growth to maturity. It is possible to account for all three dam types in a random regression model, but the programming becomes very complicated. I have found it necessary to ignore the direct-maternal genetic covariance and assume that it is zero to accommodate three types of dams.

A third order Legendre polynomial would be used for this random factor too. The additive genetic relationship matrix, and phantom parent groups are also utilized for this factor.

### 5.2.4 Year - Breed - Age of Dam - Gender

The birth-dam can be either the genetic-dam or a recipient dam, in the case of an embryo transfer. Offspring from older birth-dams often outgrow offspring of first time mothers. This might be because offspring from young mothers are smaller than those of older dams, or because young mothers do not provide enough nutrients in the milk as do older females. Age of birth-dam is usually defined within parity groups. So parity 1 with 2 or 3 age groups, parity 2 with 4 or 5 age groups, and so on. The interactions with year of birth and gender of offspring probably exist, so it is best to account for them. Years of birth may be grouped together if there are not enough data.

Legendre polynomials of order 3 can be used for this factor. Hence we are estimating deviations from the standard curves defined by the Breed-Year-Month of Birth-Gender subclasses.

If multiple breeds are analyzed together, then age of dam by gender effects should be nested within breed groups, based on breed grouping of the birth-dam.

### 5.2.5 Contemporary-Management Groups

During growth, animals are usually moved to different management groups as they get bigger or older. Thus, animals belong to a different contemporary group each time they are weighed. In pigs there is the farrowing barn during which a dozen or more sows give birth within the same week. All of the piglets could be one contemporary group, separated by gender. After 20 days, the piglets are moved to growing pens where pigs of different litters are merged and become competitors for feed and water. Later those animals are moved to finishing pens where they are fed to market weight. Some could be selected for potential herd replacements and moved to a different facility. The contemporaries of a pig are, therefore, constantly changing. Contemporary - Management groups are defined as pigs of roughly the same age and gender within the same physical environment at the time of weighing. The contemporary-management group accounts for the environmental effects at one point in time for a group of similarly treated individuals. We do not estimate a growth curve for each contemporary-management group, but only the effect on weights of pigs at one point in time. Contemporary-management groups are a random factor in the model, and there are many of these groups. The number of animals within a contemporary group is not critical.

### 5.2.6 Direct Genetic Effects

In growth data, the additive genetic effects are known as *direct genetic* effects (Willham, 1960s). The usual additive genetic relationship matrix, **A**, is used, as are phantom parent groups for animals with unknown parents.

Legendre polynomials of order 3 are used to model the animal deviations from the fixed trajectories, and hence, four parameters per animal for additive genetic effects to be estimated.

### 5.2.7 Animal Permanent Environmental Effects

Because animals are weighed several times, permanent environmental effects must be taken into account. Legendre polynomials of order 3 are used for this factor, which only exists for animals with records.

### 5.2.8 Maternal Permanent Environmental Effects

Because dams have more than one progeny in the data, there are non-genetic permanent environmental effects associated with each dam. Legendre polynomials of order 3 are used for this factor too.

### 5.2.9 Litter Effects

In litter bearing species, such as sheep, goats, and swine, there is a common litter effect of the group of full-sibs. This has to be matched to the birth-dam or the raise-dam, if an animal is cross-fostered to another dam after birth. This is also a random factor in the model and can be modeled by Legendre polynomials of order 3.

### 5.2.10 Residual Variances

There should be a different residual variance for every day of age, and these variances should be getting larger over time, as weights increase. One can calculate phenotypic variances for each of the 48 five-day periods, separately for each gender. Then express all of the variances relative to the variance at birth. The assumption is that the residual variances will follow that same relative pattern. Residual variances can be estimated for each five-day period.

### 5.2.11 Summary

Growth is a very complicated trait. The main problem is having enough weight measurements on an animal to be able to estimate the trajectories and covariance functions. Maternal genetic effects and the correlation of those with direct genetic effects adds a degree of difficulty to the model analyses. Also, if each animal can have up to three different dams influencing its growth, this too can make the analysis difficult.

If there are only 3 or 4 weights per animal, it may be much easier to analyze them with a multiple trait model, where each weight is taken at roughly the same age in all animals. The shape of the trajectory is then not important, and analyses can be simplified.

## 5.3 Covariance Function Matrices

A study of pigs on test from day 40 to 250 at a Quebec test station was conducted on 10,000 pigs. Two pigs per litter were represented in the trial. Litter effects and maternal effects were ignored. Quadratic random regressions (using Legendre polynomials) were for contemporary groups, animal additive genetic, and animal permanent environmental. Each pig had 7 or more weight measurements during the test, and almost daily feed intakes.

The covariance function matrices were estimated using Gibbs sampling on a Bayesian approach to a six-trait model. Other traits were number of times visiting the feeder (daily), time spent eating (daily), feed intake (daily), weight, fat thickness, and loin thickness. Below are the submatrices for weights only.

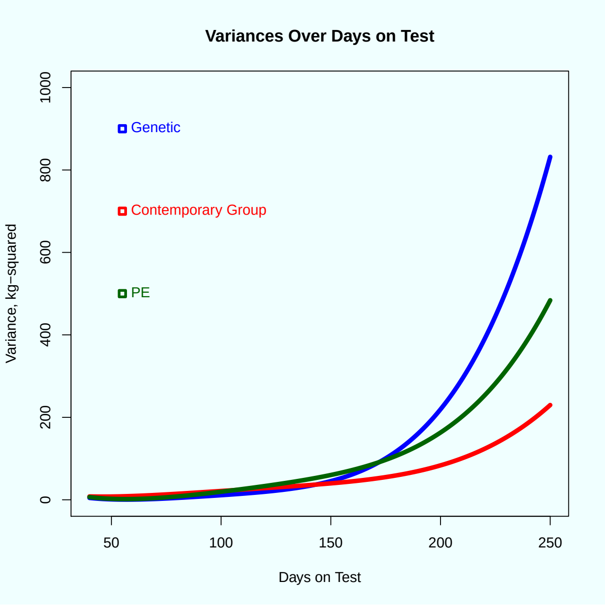
```
> Kaa  # ADDITIVE GENETIC
       [,1]   [,2]  [,3]
[1,] 139.47 126.60 42.25
[2,] 126.60 125.49 50.19
[3,]  42.25  50.19 26.30
> Kpe  # PERMANENT ENVIRONMENT
       [,1]  [,2]  [,3]
[1,] 117.77 86.97 13.04
[2,]  86.97 76.79 22.74
[3,]  13.04 22.74 16.81
> Kcg  # CONTEMPORARY GROUPS
       [,1]  [,2]  [,3]
[1,] 80.85 38.39  7.13
[2,] 38.39 27.97 11.46
[3,]  7.13 11.46  8.26
```

Using the above matrices, variances for each day on test were calculated, and then plotted (Figure 5.3).

```
# Legendre polynomials
  LAM=LPOLY(3)
  ti=c(40:250)
  tmin=40
  tmax=250
  qi = 2*(ti - tmin)/(tmax - tmin) - 1
  x=qi
  x0=x*0 + 1
  x2=x*x
  M=cbind(x0,x,x2)
  PH = M %*% t(LAM)
  Vpe = PH%*%Kpe%*%t(PH)
  vgpe = diag(Vpe)
```

```
Vcg = PH%*%Kcg%*%t(PH)
vgcg = diag(Vcg)
Vaa = PH%*%Kaa%*%t(PH)
vgaa = diag(Vaa)

par(bg="aquamarine")
plot(ti,vgaa,col="blue",lwd=5,type="l",xlab="Days on Test",
 ylab="Variance, kg-squared",ylim=c(0,1000))
title(main="Variances Over Days on Test")
lines(ti,vgcg,col="red",lwd=5)
lines(ti,vgpe,col="darkgreen",lwd=5)
points(55,900,pch=0,col="blue",lwd=3)
text(55,900,"Genetic",col="blue",pos=4)
points(55,700,pch=0,col="red",lwd=3)
text(55,700,"Contemporary Group",col="red",pos=4)
points(55,500,pch=0,col="darkgreen",lwd=3)
text(55,500,"PE",col="darkgreen",pos=4)
```

**Figure 5.3**



Because of the quadratic regression the variances all increased as days on test increased, but the larger increases did not occur until after

150 days. Higher order polynomials were not appropriate for these data. Residual variances were divided into 23 periods of 8 or 9 days each. The residual variances ranged from 3.5 kg$^2$ to 17.18 kg$^2$, and so were much smaller than the other components.

## 5.4 Expression of EBVs

With weight as the growth trait, there are two options for expressing the breeding value of an animal and ranking them. One option is for choosing a particular age and ranking animals on the basis of their EBV for weight at a given age. The other option is for determining the number of days for an animal to reach a particular weight, for example, 110 kg. The latter option is essentially a growth rate. You want to select animals with the smaller age.

For swine and some other species, growth has to be combined with feed intake. Which animals grew the fastest and ate the least amount of feed? So an index must be constructed to select for optimum growth. In addition, a fat carcass is usually not desired, and so carcass quality also needs to be included in the index. Increasing weight and growth rate could also have adverse consequences on ease of birth through larger birthweights. Growth is more than a single trait selection problem.

# Chapter 6

# Survival

The lifetime of a light bulb is the number of hours that it provides light. The lifetime of an animal is the age when it dies. For agricultural livestock animals, however, humans often determine when an animal dies. Some animals are *voluntarily culled* because the owner perceives that they are of lesser value than other animals. At the same time, some animals are *involuntarily culled* due to old age, accident, or disease. Producers generally want animals that are robust and hardy, and which could live a long time. It costs money to feed and raise an animal to maturity. Animals should have "longevity" or "stayability". Animals should be functional, either at producing offspring or producing milk, meat, eggs. or wool.

The date of an animal dying or leaving the herd (flock) for any reason gives an *uncensored* record of survival. An animal's record is said to be *censored* when it has not yet died or been culled due to a lack of adequate opportunities. All current, active animals are censored. When analyzing survival there are two possible situations.

1. Censored data are removed from the analysis, or
2. Censored data are included in the analysis.

Animals can relocate from one herd to another through sales. Such animals may be considered culled from the original herd, but are actually still alive and productive in the second herd. Reasons for disposal from herds are important to determine if records are censored. The analysis of survival should include censored data, in an appropriate manner.

The age of an animal at the time it is culled is the observation, measured in days, months, or years. This trait is not normally distributed. For censored animals, a prediction of length of productive life is usually made based on probabilities estimated from past data. Thus, if an animal has lived to time $t$, then the probability that it will live to the next time, $t+1$, is used as the observation.

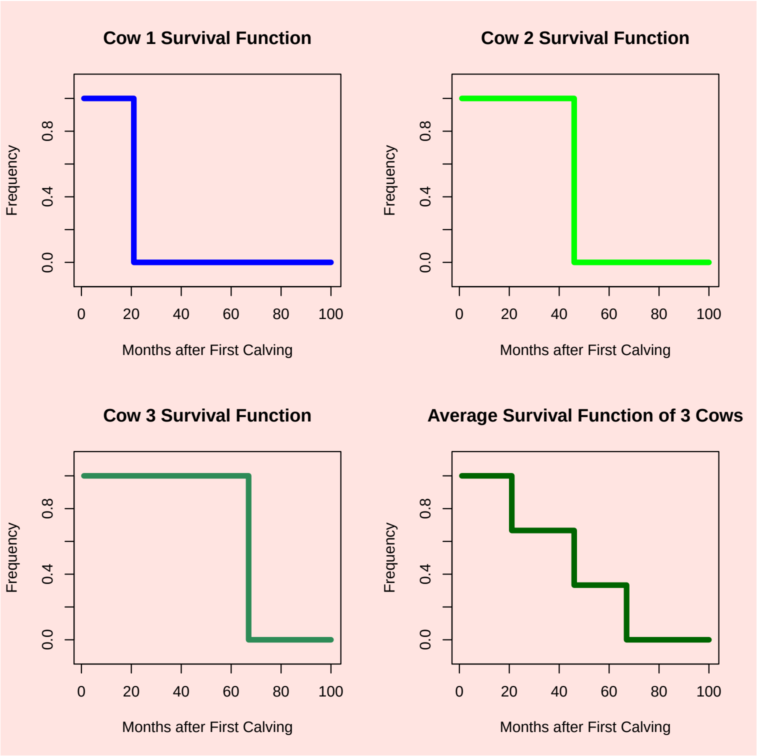
A different approach to survival analyses is to define a fixed time period, such as survival to 60 months of age, yes or no. Then survival to 75 months of age as another binary trait.

A non-linear approach is where time to failure is modelled. Censored data can be included. A survival function is derived and from this a hazard function is created, which is influenced by time dependent variables, and time independent variables.

## 6.1 Survival Function

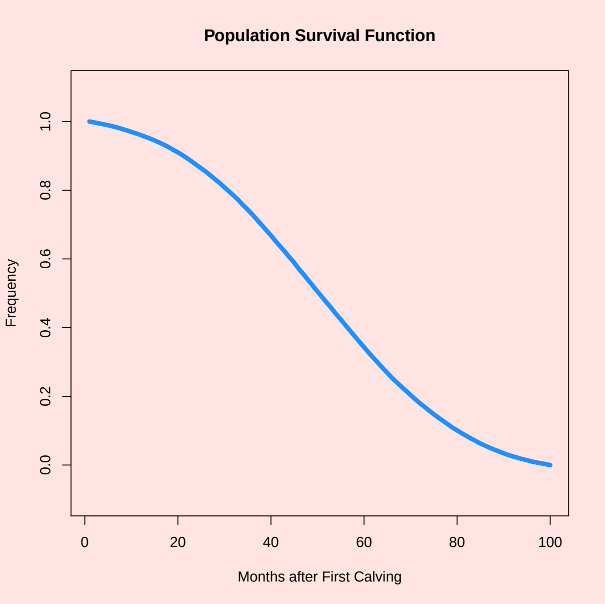
Consider 100 months after first calving as the productive life for a dairy cow. A survival function goes from 1 for an animal that is alive to 0 when the animal is dead or culled. A vertical line from 1 to 0 indicates the moment in the productive period when the animal's function changes, i.e. when the animal is removed from production. The survival function for one animal is a one-step function. Figure 6.1 shows the survival functions of 3 cows, where one has died at 20 months after first calving, one at 45 months, and one at 66 months. The fourth graph in the lower right of Figure 6.1 is the average step function for the three cows combined.

**Figure 6.1**



As more and more cows are accumulated and averaged together, the survival function for the population becomes a smooth curve as in Figure 6.2. The values on the curve give the expected probability of an animal being alive in $x$ months after first calving. By the time a cow reaches 100 months, it has a pretty high probability of being culled in the next month.

**Figure 6.2**



The approach of Veerkamp et al. (1999) and Galbraith (2003) was to apply a random regression model. For each cow there would be 100 observation points of 0 or 1. A cow that has lived 30 months past first calving and which has not yet been culled, is a censored record. If a cow was censored, then the step function would be just ones up to the point of being censored (e.g. 30 months), and the next seventy values would be not known, or not observed.

The survival function in Figure 6.2, for this example, is

$$S_t = \frac{n - d_t}{n}$$

where $t$ is the month in which an animal was last alive, $n$ is the total number of live animals that had the opportunity to live for 100 months, and $d_t$ is the number that have died up to and including period $t$. Eventually $d_t$ comes closer to $n$.

## 6.2 Model Factors

A population survival function is shaped similar to a lactation curve, and so using Legendre polynomials of order 4 (5 covariates) may be appropriate for fitting the general shape. However, because the scale goes from 1 down to 0, at the beginning of the curve many animals are alive, so that the variation in the first months after calving is very small. In general, the variance is the frequency times one minus the frequency, which has the greatest value when frequency is 0.5. The variance becomes smaller again at the end when most animals are dead. Legendre polynomials of order 4 (5 covariates) will be used to model the random animal additive genetic, and permanent environmental effects.

### 6.2.1 Year-Season of Birth-Gender

The classification approach can be taken to model the fixed time factor curves for animals born in the same year and season of the year (perhaps months). If both genders are being analyzed together, then the additional interaction with gender is needed.

In the dairy cow example, there would be 100 categories of months alive after first calving. That is a lot of levels (i.e. parameters) to be estimated, and requires a lot of animals. At the same time, there are 100 observations for all uncensored animals.

If one was studying mice, then the time scale has to be altered, and survival might be related to time after being infected with a deadly virus. Or there could be a study of bacteria and their survial to different antibiotics measured in hours or minutes. In some cases, there might be only one overall fixed curve rather than several.

### 6.2.2 Age at First Calving

For dairy cows, the age at first calving could be important to survival after calving. For mice and bacteria an important variable might be the length of exposure time before the trial begins.

### 6.2.3 Production Level

Dairy cows that produce at a high level, and therefore make more profit for the owner, tend to have a higher survival advantage. Cows should therefore, be divided in 3 or 5 categories of production levels based upon their EBV for milk yields or protein yields. These groups could also be modelled by classification variables or with order 4 Legendre polynomials. A study should be conducted to see which alternative is more suitable. Adjusting for production level makes the survival evaluations free of production level, and this is called *functional* survival.

### 6.2.4 Conformation Level

Another important factor in dairy cows is their conformation scores. More favourable looking cows (scoring Good Plus or better) have a higher survival than cows scoring Good, Fair or Poor. Making six levels of conformation and fitting classification variables of order 4 Legendre polynomials is necessary. Thus, the survival EBV would be free of both production and conformation considerations.

### 6.2.5 Unexpected Events

An unexpected event which may have a short or long term impact on animal survival are things like outbreaks of disease or drought. Animals have to be culled, that would not normally be culled, to guarantee the survival of the herd. This may affect certain types of animals (e.g. low producers, older animals) more than others. A simple year-month-age of cow subclass effect (not a curve, but an average percentage survival) could be used to model routine and unexpected downturns in survival. This could be across all herds or within provinces or regions of a country. Besides increases in culling, this factor would also identify periods when it was difficult to find cows such that culling is at below normal levels.

### 6.2.6 Contemporary Groups

Contemporary groups are random effects in the model, and hence modelled with order 4 Legendre polynomials. The definition of a contemporary for survival analyses would be animals born in the same year-season, of the same gender, and undergoing the same or similar management practices up to first calving. Because survival looks at animals over many months and years, animals will move around and be placed in different environments with different managers, and therefore, under different decision processes. Accounting for all of these possibilities is difficult, and therefore, the easy option is to leave animals in their original contemporary group throughout their lifetime. All subsequent changes cause variation that goes into the residual effects.

### 6.2.7 Animal Additive Genetic Effects

Animal additive genetic effects are random, also modelled by order 4 Legendre polynomial. The heritability of survival is generally low due to all of the environmental influences on the decisions to keep or cull animals.

### 6.2.8 Animal Permanent Environment Effects

Animal permanent environmental effects are random and account for some of the environmental influences on each animal. Legendre polynomials of order 4 could be used for this factor too.

### 6.2.9 Residual Variances

For dairy cows, looking at 100 months after first calving, this period could be divided into twenty subgroups of five months each. Some trial and error is needed to get the groupings correct.

## 6.3 Example

Because this method of survival analysis is not common, a small example will be used to illustrate. Consider a beef cattle situation and we want to look at the survival of cows, as indicated by number of calvings, where the maximum is set at nine. Thus, there are just nine categories, each representing about 12 months. Assume the data are from two years, and six contemporary groups for a total of 30 cows. Including ancestors without survival data, there are a total of 53 animals. None of the animals were inbred. The data are shown in Table 6.1. Note that four of the records in year 2 were censored, which means those animals are still active, i.e. not yet culled.

**Table 6.1**
Example Beef Cow Survival Data.

| Year | CG | Cow | Sire | Dam | Calvings | Year | CG | Cow | Sire | dam | Calvings |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 24 | 1 | 9 | 7 | 2 | 4 | 39 | 5 | 9 | 5 |
| 1 | 1 | 25 | 1 | 10 | 2 | 2 | 4 | 40 | 5 | 10 | 7* |
| 1 | 1 | 26 | 2 | 11 | 5 | 2 | 4 | 41 | 5 | 12 | 5 |
| 1 | 1 | 27 | 2 | 12 | 6 | 2 | 4 | 42 | 6 | 13 | 6* |
| 1 | 1 | 28 | 2 | 13 | 8 | 2 | 4 | 43 | 6 | 29 | 4 |
| 1 | 2 | 29 | 2 | 14 | 3 | 2 | 4 | 44 | 6 | 30 | 6* |
| 1 | 2 | 30 | 2 | 15 | 2 | 2 | 5 | 45 | 5 | 14 | 2 |
| 1 | 2 | 31 | 3 | 16 | 1 | 2 | 5 | 46 | 6 | 17 | 6 |
| 1 | 2 | 32 | 3 | 17 | 4 | 2 | 5 | 47 | 7 | 18 | 8* |
| 1 | 2 | 33 | 3 | 18 | 4 | 2 | 5 | 48 | 7 | 19 | 2 |
| 1 | 2 | 34 | 3 | 19 | 6 | 2 | 5 | 49 | 7 | 35 | 4 |
| 1 | 3 | 35 | 3 | 20 | 6 | 2 | 6 | 50 | 5 | 20 | 4 |
| 1 | 3 | 36 | 4 | 21 | 6 | 2 | 6 | 51 | 7 | 22 | 1 |
| 1 | 3 | 37 | 4 | 22 | 9 | 2 | 6 | 52 | 8 | 23 | 3 |
| 1 | 3 | 38 | 4 | 23 | 3 | 2 | 6 | 53 | 8 | 25 | 5 |

* indicates censored records

The data can be set up in R as follows.

```
#  Example data for RRM of survival
  cg=c(rep(1,5),rep(2,6),rep(3,4),rep(4,6),
       rep(5,5),rep(6,4))  # contemporary groups
  YR = c(rep(1,15),rep(2,15))  # Two years
```

```
# Pedigrees
  aid=c(1:53)
  sid=c(rep(0,23),1,1,2,2,2,2,2,3,3,3,3,3,4,4,
     4,5,5,5,6,6,6,5,6,7,7,7,5,7,8,8)
  did=c(rep(0,23),c(9:23),9,10,12,13,29,30,14,
         17,18,19,35,20,22,23,25)
  bi=c(rep(1,23),rep(0.5,30))
# Inverse of additive relationship matrix
  AI=AINV(sid,did,bi)

  y = c(7,2,5,6,8, 3,2,1,4,4,6, 6,6,9,3,
        5,7,5,6,4,6, 2,6,8,2,4, 4,1,3,5)
  yb= c(9,9,9,9,9, 9,9,9,9,9,9, 9,9,9,9,
        9,7,9,6,9,6, 9,9,8,9,9, 9,9,9,9)
  sum(yb)  # total number of observations
```

The vector y contains the number of calvings completed, and yb contains the number of calvings that could have been observed up to the current date.

The covariance function matrices for the random effects were as follows.

$$\mathbf{K}_a = \begin{pmatrix} .36814 & -.17200 & .32359 & .00000 & -.01844 \\ -.17200 & .35300 & -.24600 & -.03448 & .00000 \\ .32359 & -.24600 & .55338 & .00000 & -.04292 \\ -.00000 & -.03448 & .00000 & .06567 & .00000 \\ -.01844 & .00000 & -.04292 & .00000 & .06466 \end{pmatrix},$$

for the additive genetic effects,

$$\mathbf{K}_p = \begin{pmatrix} .31894 & -.13760 & .25719 & .00000 & -.01844 \\ -.13760 & .30380 & -.19680 & -.03448 & .00000 \\ .25719 & -.19680 & .46318 & .00000 & -.04292 \\ .00000 & -.03448 & .00000 & .06567 & .00000 \\ -.01844 & .00000 & -.04292 & .00000 & .06466 \end{pmatrix},$$

for animal permanent environmental effects, and

$$\mathbf{K}_c = \begin{pmatrix} .68214 & -.04600 & -.03141 & .00000 & -.01844 \\ -.04600 & .99000 & .00900 & -.03448 & .00000 \\ -.03141 & .00900 & .14638 & .00000 & -.04292 \\ .00000 & -.03448 & .00000 & .06567 & .00000 \\ -.01844 & .00000 & -.04292 & .00000 & .06466 \end{pmatrix},$$

for contemporary groups. Therefore, the assumed heritabilities by number of calvings are shown in Table 6.2.

**Table 6.2**
Heritabilities by Number of Calvings.

| Number of Calvings | Heritability |
|---|---|
| 1 | 0.420 |
| 2 | 0.349 |
| 3 | 0.231 |
| 4 | 0.153 |
| 5 | 0.191 |
| 6 | 0.185 |
| 7 | 0.175 |
| 8 | 0.218 |
| 9 | 0.309 |

The Legendre polynomials for the random factors were order 4, and set up as

```
reglp = jd(9,5)*0
tmin=1
tmax=9
no=5
for(i in 1:9){
   reglp[i, ] = LPTIME(i,tmin,tmax,no)
}
```

The design matrices need to be constructed for years, contemporary groups, and animal genetic and animal permanent environment factors.

Recall that each cow with survival data has 9 observations, unless their record is censored, then they have less than 9 observations. In total in this example, there were 261 observations. We also need to create the observation vector, YOB, of zeros and ones, and the residual variances for each of the nine categories. For simplicity, let the residual variances be equal to the numbers given in pq below, and this script makes the design matrix for years. The year effects make use of the classification approach, so that there are nine parameters for each year.

```
X = jd(261,18)*0
YOB = rep(0,261)
ri=YOB
nly=length(y)
 pq=c(1:9); pq[1]=0.09; pq[2]=0.16;
 pq[3]=0.21; pq[4]=0.24; pq[5]=0.25;
 pq[6]=0.24; pq[7]=0.21; pq[8]=0.16;
 pq[9]=0.09
 pq = 1/pq  # residuals inverted
 k=0
 for(i in 1:nly){
     my = YR[i]
     loff=(my-1)*9
     ja = y[i]  # number of ones
     jb = yb[i]  # number of obs for animal
    for(j in 1:jb){
      k=k+1
      X[k,j+loff]=1
      YOB[k]=1
      ri[k]=pq[j]
      if(j > ja)YOB[k]=0
     }
  }
```

Similarly, the design matrix for contemporary groups is generated as follows. Because contemporary groups are random, they are modelled by order 4 Legendre polynomials (from reglp ).

```
# Contemporary Groups Zc
```

```
Zc=jd(261,30)*0
k=0
for(i in 1:nly){
    mc = cg[i]
    loff = (mc-1)*5 +1
    lofl = loff + 4
    ja = y[i]
    jb = yb[i]
  for(j in 1:jb){
      k=k+1
      Zc[k,c(loff:lofl)] = reglp[j, ]
    }
 }
```

Animal additive genetic effects are also modelled by order 4 Legendre polynomials, as are the animal permanent environmental effects, but which are a subset of the columns for the animal additive genetic effects.

```
# Animal Additive
 mcol = 53*5
 manc = 23*5 + 1
 Za = jd(261,mcol)*0
 k=0
 for(i in 1:nly){
     ma = anwr[i]
     loff = (ma-1)*5 + 1
     lofl = loff + 4
     ja = y[i]
     jb = yb[i]
    for(j in 1:jb){
        k=k+1
        Za[k,c(loff:lofl)] = reglp[j, ]
      }
   }
 Zp = Za[ ,c(manc:mcol)]
```

After the design matrices, one sets up matrices for the mixed model equations, then solve them.

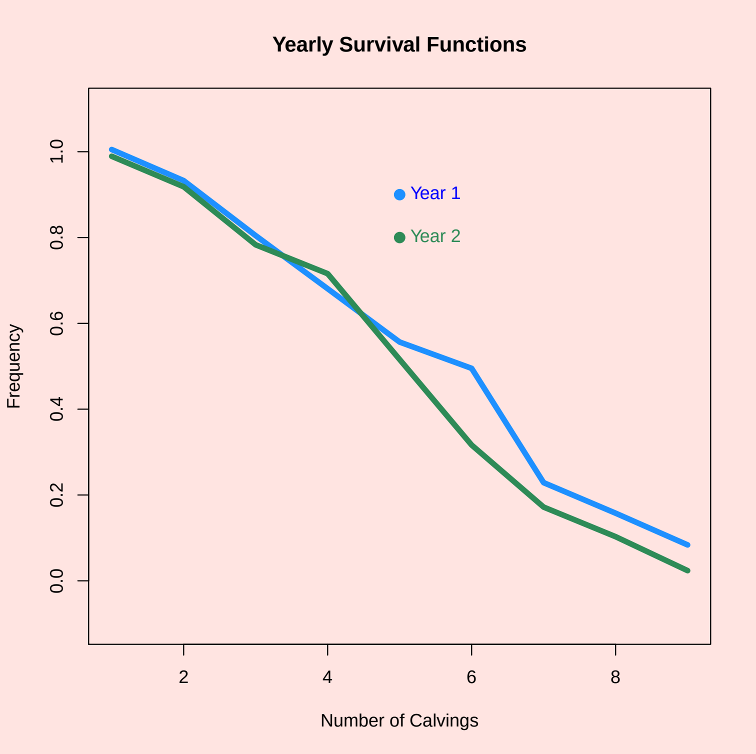
```
# setup MME and solve
  ZZ=cbind(Zc,Za,Zp)
  RI=diag(ri)
# make covariance matrices
  Gai=solve(Ga) # Ga = Ka
  Gpi=solve(Gp) # Gp = Kp
  Gci=solve(Gc) # Gc = Kc
  HI= id(6) %x% Gci
  GI=AI %x% Gai
  PI=id(30) %x% Gpi
  QI=block(HI,GI,PI)

# solve MME
  RRS = MME(X,ZZ,QI,RI,YOB)
```

MME is a routine for setting up mixed model equations and solving them. See Chapter 2 for details of MME and AINV.

### 6.3.1 Year Trajectories

The next step is to look at the solutions and make sense of the results. The first thing is to look at the year solutions and plot them in a graph.

**Figure 6.3**



Note that in year 1 all of the animal records were uncensored, and therefore, none of them were being observed any longer because they have all been culled (some years after these data were obtained). In year 2, however, there were 4 censored records, and therefore, the line for year 2 is not fully completed, and will not be until all the animals in year 2 have been culled. The line for year 2 could still change, but the line for year 1 is essentially complete and not likely to change very much in future analyses. It will change a little due to adding relatives information in later years.

With only 261 observations, the fixed curves (i.e. trajectories) are not very smooth. If there were several thousand records per year, then the curves might be more smooth looking.

### 6.3.2 Contemporary Groups

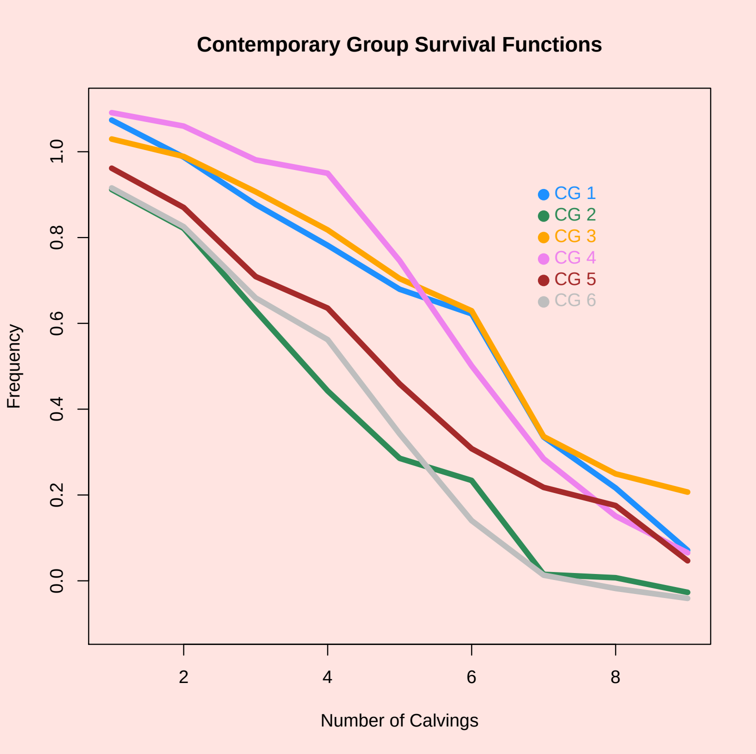
Contemporary groups were modelled by order 4 Legendre polynomials. The solutions are shown in Table 6.3.

**Table 6.3**
Random regression solutions for contemporary groups.

| Group Number | $c_0$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 1 | 0.12264 | -0.00160 | -0.04552 | -0.02469 | 0.00526 |
| 2 | -0.28856 | -0.01639 | 0.09221 | 0.00770 | -0.02311 |
| 3 | 0.16592 | 0.01799 | -0.04669 | 0.01699 | 0.01784 |
| 4 | 0.24412 | -0.02660 | -0.07702 | 0.00605 | 0.01125 |
| 5 | -0.03720 | 0.05555 | 0.03736 | -0.01745 | -0.01820 |
| 6 | -0.20692 | -0.02895 | 0.03966 | 0.01140 | 0.00695 |

From the values in Table 6.3, it is not easy to know which contemporary groups had greater or lesser survival rates. One needs to calculate survival differences for each of the nine categories using the Legendre polynomials, and then one must add the year trajectories for the years in which those contemporary groups were nested. Thus, year 1 trajectory is added to contemporary groups 1, 2, and 3, and year 2 trajectory is added to contemporary groups 4, 5, and 6. Then those values can be plotted as shown in Figure 6.4.

**Figure 6.4**

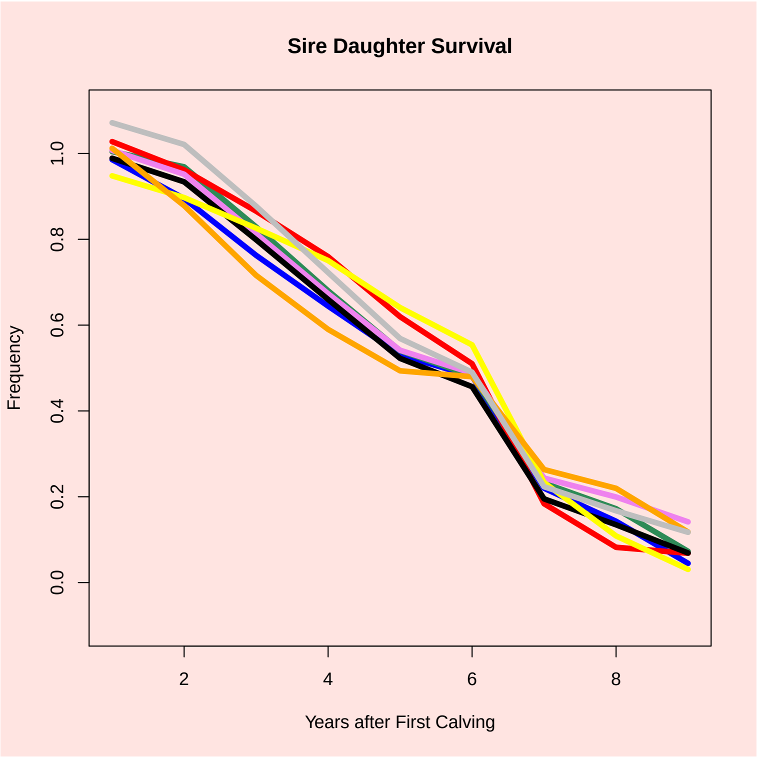


From Figure 6.4, contemporary groups 1, 3, and 4 had the better survival rates, and these corresponded to positive values for $c_0$, and negative values for $c_2$. Usually, the survival rates of contemporary groups are not of interest, but they need to be taken into account in calculating animal EBVs.

### 6.3.3 Animal Estimated Breeding Values

With the trait of survival, interest is primarily in sires and how they rank on daughter survival. As with the contemporary groups, the solutions for the regression coefficients are not informative on their own. Multiply times the Legendre polynomials for the nine categories, and add the year 1 trajectory to those numbers. Year 1 was chosen because all animals in that year have been culled (uncensored data). Usually one would take the latest year in which all animal records are uncensored. The results for eight sires are given in Figure 6.5.

**Figure 6.5**



Notice the difference from Figure 6.4. There were greater differences among contemporary groups than among sires. To pick up differences among sires one has to look at the end of the trajectories, or category 9. Sires rank differently depending on which number of calvings you want to consider as the ranking criteria. The sires and their ranks at the $1^{st}$, $5^{th}$, and $9^{th}$ calvings are given in Table 6.4.

**Table 6.4**

Sire rankings at $1^{st}$, $5^{th}$, and $9^{th}$ calvings.

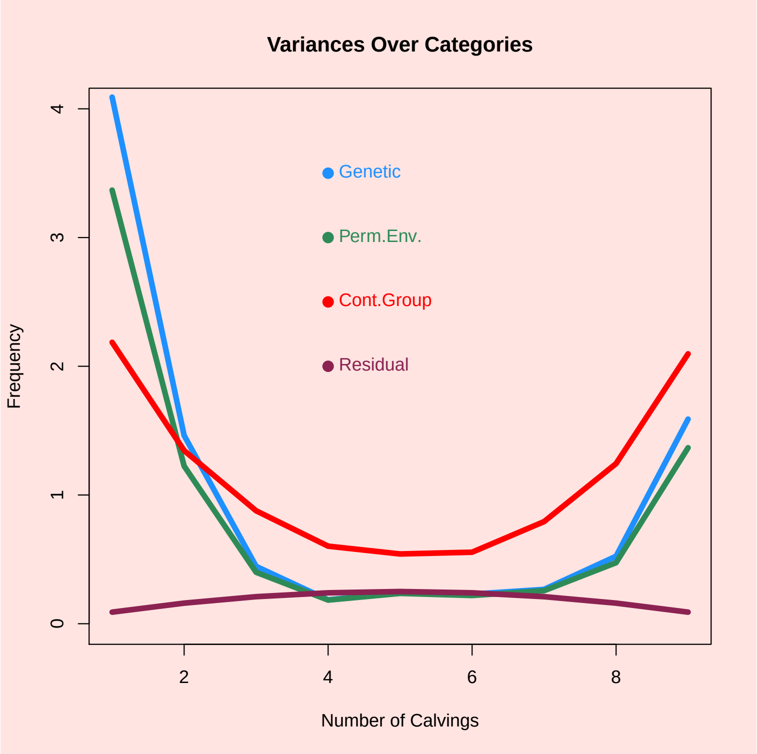
| Sire | $1^{st}$ | $5^{th}$ | $9^{th}$ |
|---|---|---|---|
| 1 | 7 | 6 | 7 |
| 2 | 4 | 4 | 4 |
| 3 | 2 | 2 | 6 |
| 4 | 5 | 5 | 1 |
| 5 | 6 | 7 | 5 |
| 6 | 8 | 1 | 8 |
| 7 | 3 | 8 | 2 |
| 8 | 1 | 3 | 3 |

Which sire would you choose to use in future matings?

### 6.3.4 Variances

The covariance function matrices used in the example were not estimated from real data, but were concocted for illustration purposes. Still, by using an order 4 Legendre polynomial for the random factors, the variances at the first and nineth calvings were artificially high (Figure 6.6). As already mentioned the variances at the first and nineth calvings should be the smallest, and the largest should occur at the fifth calving. A full study using a very large data set needs to be conducted. The random regression model approach to survival analyses seems appropriate and useful. Comparisons to other methods may be warranted (Jamrozik et al. 2008).

**Figure 6.6**


Variances Over Categories
● Genetic
● Perm.Env.
● Cont.Group
● Residual
Frequency
0
1
2
3
4
2
4
6
8
Number of Calvings

# Chapter 7

# Fortran Programs

My first programming course was in 1969 at Purdue University. The language was FORTRAN, and the course was very exciting to me. Programs had to be punched on cards, and then submitted to be run. Each card held one line of FORTRAN code. So a 10-line program required ten cards plus two or three header cards to identify user accounts and provide information about the amount of memory and time required to run the program, and then an ending card to separate different jobs (other people's programs). I worked for Prof Wayne Keim at the time, and one of his grad students, Lane Lester, knew some FORTRAN, and we discussed my assignments and FORTRAN at times. I wanted to learn quickly, and Lane helped me to forge ahead of the instructor's lecture schedule to get to the more interesting topics of arrays and do-loops. Many of the rules of FORTRAN stem from having to use punch cards to store programs.

In June of 1969 I went to Cornell University for graduate school and all grad students got to use the IBM 360 machine with 128K of memory. During the day the computer was fully utilized by the Dairy Records Processing Laboratory, but from 11:00 pm to 7:00 am anyone could run their programs. This was when most of the research was conducted. There would be three or four of us running programs. Programs were prepared during the day. If your program ran the first time it was run, then you wanted to run the next program. If the next program was not ready, everyone learned to compose their program in the computer room, punch it from your head directly onto cards, and then run it, get a printout (with all of your errors), and then fix it up and try again. One could often get help in debugging from the other grad students.

One could time their programs by looking at the printer. By printing output after each cow or herd you could pinpoint the lines in your code that required the most time. Then you could work on improving your algorithms to save time. For example, division often took longer to execute than multiplication, and so instead of dividing you could save time by multiplying, or re-arrange your program so that the number of division operations was reduced. It was a great time for experimentation and testing different algorithms.

Shortly after I arrived at Cornell, I was given a catalogue of FORTRAN subroutines that were written by Dr Henderson and Bob Everett. These were handy routines for inverting matrices, generating random numbers, looking up things in arrays, perpetual day routines, and others. The inversion routine is the only one that I still use from those days.

I am a dinosaur when it comes to writing programs. I still use FORTRAN, and also one of the basic versions of FORTRAN. If I were starting over as a grad student, I would learn and use C++, but C++ did not exist in 1969. Now it is too late for me to learn a new language, plus there is no need for me to do so. Thus, this chapter is more for ideas on how to program random regression models rather than to provide a finely honed tool for people to use. The necessity for writing efficient programs has lessened due to today's compilers, large amounts of memory and disk space, and the much faster processors. Surprisingly, however, efficient code can still have an impact on computing times.

This chapter gives ideas on how to write a FORTRAN program to perform a multiple trait random regression model. The example will be lactation production of dairy cows in the first three lactations for milk yield. The linux servers that I use now (in 2016) employ the Intel compiler and utilize a few Intel math libraries of programs (sorting and calculating eigenvalues and eigenvectors).

## 7.1 Main Program

My strategy is to have a main program that is only one or two pages in length. Almost every line of the main program is a call to a subroutine. As such it is easy to follow what the program is doing. The subroutines are also kept as simple as possible, but some can be very long.

The model assumes that all of the factors except Year-month of calving are fitted by an order 4 set of Legendre polynomials. So every level of these factors has 5 covariates to estimate. The Year-Month of Calving (within parities) has 36 ten day periods to estimate.

**Figure 7.1**
Main Program

```
c  Random regression model production test day milk records
c   one trait, 3 lactations, order 4 Legendre polynomials
c
c  y = YM(36) + RAS(5) + CG(15) +  a(15) + p(15) + e(3,4)

      include 'SShd.f'
c
c #############################################################
c  read in parameters
c
      itest = 0
      igibb = 0
      if(igibb.gt.0)then
c Two files for storing Gibbs samples
       open(17,file='animalVCV.d',form='unformatted',
     x status='unknown')
       open(19,file='cgVCV.d',form='unformatted',
     x status='unknown')
       open(20,file='resVC.d',form='formatted',
     x status='unknown')
      endif
      call params
```

```
c
c ############################################################
c read in pedigree info
c
      call peddys
c
c ############################################################
c  read in data
c
      call datum
c
c ############################################################
c  iterations on equations to solve
c
 800  iter = iter + 1
      if(iter.gt.6000)go to 9901
c
      ccn = 0.d0
      ccd = 0.d0
c
      call facYM
      call facRAS
      call facCG
      call permenv
      call genetic
c
      if(igibb.gt.0)then
        call facRES
        endif
      ccc = 100.0*(ccn/ccd)
      if(mod(iter,100).eq.0)print *,iter,ccc
      if(ccc.gt.0.1d-09)go to 800
c
c  #############################################################
c  finished, save solutions
c
 9901 call finis
      if(igibb.gt.0)then
```

```
  close(17)
  close(19)
  close(20)
 endif
 stop
 end

 include 'SSparams.f'
 include 'SSpeddys.f'
 include 'SSdatum.f'
 include 'SSfini.f'
 include 'SSYM.f'
 include 'SSRAS.f'
 include 'SSCG.f'
 include 'SSanm.f'
 include 'SSape.f'
 include 'SSres.f'
 include 'dkmvhf.f'
```

The program is only a few steps.

1. `call params`, to read in the necessary covariance function matrices, and set up their inverses for use it the mixed model equations.

2. `call peddys`, to read in the pedigree information and to set up the diagonals of $\mathbf{A}^{-1}$ for each animal.

3. `call datum`, to read and store the data for the iteration on data procedure. Some arrays need to be sorted.

4. Iterate solutions by call subroutines, one for each factor in the model.

5. `call finis`, to write out and save the solutions to the factors that are of interest.

All of the subroutines are joined together by `include 'SShd.f'`. This is a file that defines the variables in the program that need to be shared between subroutines. It specifies which variables are double precision and which are integer. The big arrays are put into COMMON areas so that they are stored consecutively in memory and thereby take a little less space. Thus, during the initial programming, if an array needs to be increased in length, this can be done in this file, and it therefore, occurs in all other subroutines. There is no chance of forgetting to make a change in the other subroutines.

With COMMON areas one has to worry about boundary alignments. Thus, this problem is avoided by having separate COMMON areas for double precision arrays and integer arrays. The boundary alignment problems occur when integer and double precision arrays are mixed together in one COMMON area. If the two types are to be in the same COMMON area, then the double precision arrays should precede the integer arrays.

### Figure 7.2

SShd.f

```
      Parameter(no=15,nop=120,nam=200000,ncg=12000,nras=200,
     x ndim=365,nrec=1270000,nped=500000, nym=500,mcov=5,
     x ntg=36,ntim=4)

c   no = 5 covariates times 3 lactations = 15
c   nop = (no*(no+1))/2, half-stored matrix array size
c   nam = maximum number of animals
c   ncg = maximum number of contemporary groups
c   nras = number of region-age-season of calving subclasses
c   ndim = number of days in milk (maximum)
c   nrec = maximum number of test day records
c   nped = maximum number of pedigree elements to store
c   nym = number of year-month subclasses
c   mcov = number of covariates

      Common /recs/lp(ndim,5),obs(nrec),anid(nrec),cgid(nrec),
     x rasid(nrec),ymid(nym),pari(nrec),days(nrec),timg(nrec),
     x mrec,mcgid,mras,mym,iseed
```

```
c   lp = legendre polynomials of order 4 for days 1 to 365 in milk
c   obs = test day milk yields
c   anid = animal ID associated with each obs
c   cgid = contemporary group associated with each obs
c   rasid = region-age-season for each obs
c   ymid = year-month for each obs
c   pari = parity number for each obs
c   days = days in milk for each obs
c   timg = 1 to 36 time groups within YM subclasses
c   mrec = actual total number of test day records
c   mcgid = max id of contemporary groups
c   mras = max id of ras subclasses
c   mym = max id of year-month subclasses

      Common /peds/bii(nam),adiag(nam),sir(nam),dam(nam),
     x cpa(nped),cpc(nped),cps(nped),cpd(nped),
     x jped(nped),mam,mped

c   bii = elements needed for A-inverse
c   adiag = diagonal elements for each animal
c   sir = sire ID (consecutively numbered and chronological
c   dam = dam ID (consecutively numbered and chronological
c   cpa = coded pedigree record, animal id
c   cpc = code (0,1,2)
c   cps = sire or progeny ID
c   cpd = dam or mate ID
c   mam = total number of animals < nam
c   mped = pedigree records < nped

      Common /parms/gi(nop),pi(nop),ci(3,15),res(4,3),
     x ri(ndim,3),wv(nras)

c   gi = genetic covariance function matrix
c   pi = permanent environmental covariance function matrix
c   ci = contemporary group covariance function matrix
c   res = residual variances, 3 parities, 4 periods
c   ri = inverses for each day in milk
c   wv = work vector, used for many things
```

```
      Common /diags/pcg(nrec),pras(nrec),pymid(nrec),
     x panid(nrec),wr(nam),iwv(nras),itest,igibb

c  pcg = CGID sorted order
c  pras = rasid sorted order
c  pymid = YMID sorted order
c  panid = anid sorted order
c  wr = number of records per animal (many are zero
c  iwv = integer work vector
c  itest = 0 for good run, not zero during initial programming
c  igibb not zero, means to estimate covariance matrices

      Common /solns/sanm(nam,no),sape(nam,no),scg(ncg,mcov),
     x sras(nras,mcov),sym(nym,ntg),ccn,ccd,ccc

c  solution arrays for animal genetic, PE, cont. groups,
c  and region-age-season, ccn, ccd, and ccc for
c   convergence criteria

      real*8 lp,obs,adiag,gi,pi,res,ri,sanm,sape,scg,sras,
     x sym,rhs,ccn,ccd,ccc,wv

      integer anid,cgid,ras,ymid,pari,days,timg,pcg,pras,
     x pymid,panid,mped,mrec,sir,dam,cpa,cpc,cps,cpd,jped,
     x wr,iwv,itest,mam,mrec,igibb,mras,mcgid,mym,iseed
```

The above lines are included in every subroutine that is part of the main program. Subroutines may have some of their own variables, which are only used within that subroutine.

Define all of the variables in this file as either `real*8` or `integer`. Do not rely on default rules.

## 7.2 Call Params

The first subroutine to be called is `params`, which reads in the covariance matrices that will be used to start the iteration process. The random factors of the model are contemporary groups, and animal additive genetic and animal permanent environmental effects.

**Figure 7.3**
SSparams.f

```
      subroutine params
      include ’SShd.f’

      real*8 varc,varg,varp,x,z(5),hh(15)

      open(10,file=’GP4.d’,form=’formatted’,status=’old’)

      gi = 0.d0
      pi = 0.d0
      ri = 0.d0
      ci = 0.d0
      res = 0.d0
      lp=0.0d0
c
 10   read(10,1001,end=20)kr,kc,varg,varp
 1001 format(1x,2i4,8d20.10)
      if(kr.eq.0)go to 20
      m = ihmssf(kr,kc,no)
      gi(m) = varg
      pi(m) = varp
      go to 10
c
 20   read(10,1002,end=21)kp,kr,kc,varc
 1002 format(1x,3i4,d20.10)
      m = ihmssf(kr,kc,mcov)
      ci(kp,m)=varc
      go to 20
 21   close(10)
c
c  residual variances, by parity and days groups
c
      open(11,file='RES4.d',form='formatted',status='old')
      do 22 i=1,3
         do 22 j=1,4
            read(11,*,end=30)ka,x
            res(j,i)=x
 22   continue
 30   close(11)
c
      call dkmvhf(gi,no,wv,iwv)
      call dkmvhf(pi,no,wv,iwv)
      do 81 kp=1,3
        hh = 0.d0
        do 82 m=1,15
          hh(m)=ci(kp,m)
 82    continue
      call dkmvhf(hh,mcov,wv,iwv)
      do 83 m=1,15
        ci(kp,m)=hh(m)
 83   continue
 81   continue
c
      ri = 0.0d0
      do 31 i=1,45
         ri(i,1) = 1.d0/res(1,1)
         ri(i,2) = 1.d0/res(1,2)
         ri(i,3) = 1.d0/res(1,3)
 31   continue
      do 32 i=46,115
         ri(i,1) = 1.d0/res(2,1)
         ri(i,2) = 1.d0/res(2,2)
         ri(i,3) = 1.d0/res(2,3)
 32   continue
      do 33 i=116,265
```

```
         ri(i,1) = 1.d0/res(3,1)
         ri(i,2) = 1.d0/res(3,2)
         ri(i,3) = 1.d0/res(3,3)
 33   continue
      do 34 i=266,365
         ri(i,1) = 1.d0/res(4,1)
         ri(i,2) = 1.d0/res(4,2)
         ri(i,3) = 1.d0/res(4,3)
 34   continue
c
c  read in Legendre polynomials, order 4
c
      open(12,file='LPOLY4.d',form='formatted',status='old')
      lp = 0.0d0
  40  read(12,1201,end=55)kdim,z
 1201 format(2x,i5,2x,5f20.10)
       do 42 k=1,5
       lp(kdim,k)=z(k)
  42   continue
      go to 40
  55  close(12)
c
c  read in a random number seed, initialize random number
c    generators
c
      open(13,file='seedno.d',form='formatted',status='old')
      read(13,1301,end=65)iseed
 1301 format(1x,i10)
      call firan(iseed)
      close(13)
      return
      end
```

Four input files are needed, a) one for the covariance matrices for genetic, permanent environmental, and contemporary groups, b) one for the residual variances, c) one for the Legendre polynomials, and d) one for the random number seed. Remember to create the appropriate files, and make sure the format statements are in agreement with the data files.

The inverses of the covariance matrices are obtained using `dkmvhf.f`. This is Henderson's inversion routine that he wrote back in the 1960's. His version was called `djnvhf.f`, but Karin Meyer found a way to improve its speed. Henderson's version physically re-arranges rows and columns during the inversion process. Meyer's version merely kept an array of indexes of the rows to be re-arranged, but did not actually re-arrange them physically until the end. This increased the speed very much, and so the new version became `dkmvhf.f` where the `km` is for Karin Meyer. One advantage of both routines is that the matrix can have rows and columns that are all zeros. Many inversion routines only want to invert matrices that are non-singular, so that you must remove the zero rows and columns before calling the subroutine. This routine is given in the Appendix.

Note that in the version of FORTRAN that I am using that I can set an entire array to zero with one statement, `gi = 0.d0`. It appears to be important to use `0.d0` rather than `0.0` in this version of FORTRAN. The later results in 0, but only to 7 or so decimal places, which can be critical to some programs. Thus, I always use the `0.d0` option in my programs.

The subroutine `firan` is specialized software for initializing a series of random number generators for different distributions. The Mersene twister is used as the algorithm in these routines which has a very long cycle time, $(2^{19937} - 1)$. The cycle time is how many numbers it takes before the sequence starts to repeat itself. When using Gibbs sampling it is good to have a long cycle time.

## 7.3 Call Peddys

The following subroutine reads in the pedigree with the `bii` values needed for the inverse of the additive relationship matrix. These values were computed by another series of programs which order the animals chronologically, and then calculate the inbreeding coefficients, as long as parents are processed before their progeny.

The subroutine also reads in a ‘coded’ pedigree file, which has an animal, then all of its progeny following, and the mate that produced that progeny. This is so additive relationships can be accounted for easily in the iteration program.

**Figure 7.4**
SSpeddys.f

```
     subroutine peddys
     include 'SShd.f'
     character*8  oid
     real*8 x,v,z

     open(10,file='PARTES.d',form='formatted',
    x status='old')

     adiag = 0.d0
     sir = 0
     dam = 0
     bii = 0.d0
     mam=0

10   read(10,1001,end=50)ka,ks,kd,x,z,oid
1001 format(1x,3i10,1x,d20.10,2x,d20.10,1x,a8)

     mam = mam + 1

     sir(ka) = nam
       if(ks.gt.0)sir(ka) = ks
     dam(ka) = nam
       if(kd.gt.0)dam(ka) = kd
     bii(ka) = x

     adiag(ka) = adiag(ka) + x
     v = 0.25d0*x
```

```
      if(ks.gt.0)adiag(ks)=adiag(ks)+v
      if(kd.gt.0)adiag(kd)=adiag(kd)+v
c
      go to 10
 50   close(10)
      print *,'peddys, mam= ',mam
c
c read and store coded pedigree file
c
      open(11,file='CARTES.d',form='formatted',
     x status='old')
      mped = 0
      jped = 0
      cpa = 0
      cpc = 0
      cps = 0
      cpd = 0
 60   read(11,1101,end=90)ia,ic,is,id
 1101 format(1x,i10,i3,1x,2i10)
        mped = mped + 1
        if(mped.gt.nped)go to 89
        cpa(mped)=ia
        cpc(mped)=ic
          if(is.lt.1)is = nam
          if(id.lt.1)id = nam
        cps(mped)=is
        cpd(mped)=id
        if(ic.eq.0)jped(ia)=mped
        go to 60
c
 89   print *,'nped limit exceeded in SSpeddys.f'
 90   close(11)
      print *,'peddys, mped = ',mped
c
      return
      end
```

## 7.4 Call Datum

The data have been prepared by other programs, and the levels of each factor have been converted to a consecutive number from 1 to the maximum number of levels for that factor.

One could also calculate means of the milk yields by days in milk, year months, or whatever may be of interest.

At the end of the routine, the levels of the factors were sorted so that levels of each factor could be processed one at a time, in sequence. Thus, the diagonal blocks for each factor can be constructed at the same time as accumulating the right hand sides of the mixed model equations (MME). Thus, there is no need to save the diagonal blocks on disk or in memory. This is also handy if Gibbs sampling is to be used to estimate the covariance function matrices in the same program, because the covariance function matrices would change with each sampling.

Lastly, the solution vectors are set to zeros before the iterations begin. Otherwise there could be unknown information in those arrays that might cause problems with convergence.

## Figure 7.5

SSdatum.f

```
      subroutine datum
      include 'SShd.f'
      real*8 p(3)
      open(11,file='MILKTDM.d',form='formatted',
     x status='old')

      mrec = 0
      obs=0.d0
      nerr = 0
 11   read(11,1101,end=20,err=88)iam,iym,iras,icg,
     x jdim,jtim,p
 1101 format(1x,6i10,3f9.2)
c
      if(jdim.lt.5)go to 11
      if(jdim.gt.ndim)go to 11
      mrec = mrec + 1
      if(mrec.gt.nrec)go to 19
      anid(mrec) = iam
      cgid(mrec) = icg
      rasid(mrec) = iras
      ymid(mrec) = iym
      timg(mrec) = jtim
      if(icg.gt.mcgid)mcgid=icg
      if(iras.gt.mras)mras=iras
      if(iym.gt.mym)mym=iym
      pari(mrec) = 1
        if(p(2).gt.-9000.0)pari(mrec)=2
        if(p(3).gt.-9000.0)pari(mrec)=3
      kp=pari(mrec)
      obs(mrec) = p(kp)
      days(mrec) = jdim
      go to 11
 19   print *,'Too many records'
       go to 20
 88     print *,'Err rec',mrec
      go to 11
 20   close(11)
      print *,'  datum, mrec= ',mrec

 35   write(30,3005)mrec,nrec,mcgid,ncg,mras,nras,
     x mym,nym
 3005 format(1x,2i10,' recs'/1x,2i10,' mcgid'/1x,2i10,
     x ' mras'/1x,2i10,' mym')
      close(11)
c
c  sort data by levels of different factors
c   IPSORT is an Intel math library function
      kflag = 1
      ier = 0
      call IPSORT(ymid,mrec,pymid,kflag,ier)
      call IPSORT(rasid,mrec,pras,kflag,ier)
      call IPSORT(cgid,mrec,pcg,kflag,ier)
      call IPSORT(anid,mrec,panid,kflag,ier)
c
c  set all solution vectors to zero
c
      sanm = 0.d0
      sape = 0.d0
      scg = 0.d0
      sras = 0.d0
      sym = 0.d0
      return
      end
```

## 7.5 Iteration Subroutines

The main program can be thought of as 'modular'. There are fixed factors, random factors, and the animal additive genetic factor. Fixed factors do not have any covariance function matrices. There are two types of fixed factors in this model. The Year-Month of calving subclasses each have 36 ten-day periods associated with them - to model the trajectory of test day milk yields. The other type is the region-age-season subclasses which are modelled by order 4 Legendre polynomials, and thus, there are 5 parameters for each subclass.

### 7.5.1 Year-Month Factor

**Figure 7.6**
SSym.f

```
      subroutine facYM
      include 'SShd.f'
c
      real*8 diags(ntg),vnois(ntg),ay(ntg),
     x XRY(ntg),c,y,z,w,x,ddif,xad

      integer levs(nrec),iork(nop),mfac
c #######################################################
c  determine number of observations per
c   level of the factor, store in levs
      levs = 0
      mfac = 0
      do 8 krec=1,mrec
         iym = ymid(krec)
         if(iym.gt.mfac)mfac = iym
         levs(iym) = levs(iym) + 1
   8  continue
      kend=0
c #######################################################
c   For each level of the factor
c    adjust observations for all other solutions
c    save in XRY, make diags of MME
      do 11 iym = 1,mfac
        jrec = levs(iym)
        XRY = 0.d0
        diags = 0.d0
        if(jrec.lt.1)go to 11
        kstr = kend+1
        kend = kend+jrec
        do 10 lrec = kstr,kend
          krec = pras(lrec)
c
         iam = anid(krec)
         icg = cgid(krec)
         iras = rasid(krec)
         jdim = days(krec)
         jtim = timg(krec)
         kp = pari(krec)
         ja = (kp-1)*5
         c = ri(jdim,kp)
         y = obs(krec)
         do 15 j=1,mcov
            ka=ja+j
            y = y - (scg(icg,j) + sanm(iam,ka)
     x  + sras(iras,j) + sape(iam,ka))*lp(jdim,j)
 15      continue
c
         xad = y*c
         XRY(jtim)=XRY(jtim) + xad
         diags(jtim)=diags(jtim)+ c
c
 10    continue
c ####################################################
c  solve for new solution for this level of factor
c
      vnois = 0.d0
c  ###################################################
c   if estimating covariance matrices then
c     generate sampling variance to
c     add to solutions
      if(igibb.gt.0)then
        do 17 i=1,ntg
          call fgnor3(z)
          if(diags(i).gt.0.d0)diags(i)=1.d0/diags(i)
          vnois(i)=z*diags(i)
  17    continue
      endif
c
      do 25 j=1,ntg
         z = XRY(j)*diags(j)
c   add vnois, compute convergence criteria
         z = z + vnois(j)
         ddif = z - sym(iym,j)
         ccn = ccn + ddif*ddif
         ccd = ccd + z*z
         sym(iym,j) = z
 25      continue
 11   continue
      if(itest.gt.0)then
         jj=6
         print 5003,jj,(sym(jj,L),L=1,5)
 5003 format('  PYM',i4,5f12.4)
      endif
      return
      end
```

### 7.5.2 Region-Age-Season

The above statements were for the Year-Month subclasses, modelling the trajectories of lactation curves for milk yield using 36 ten-day periods. Now we compare this routine to one for region-age-seasons which are modelled by order 4 Legendre polynomials as covariates, but only within a parity. Subclasses were numbered consecutively across region-age-seasons. Ages are nested within parities and thus, only 5 covariates per subclass. Dealing with the covariates requires different coding.

**Figure 7.7**
SSras.f

```
      subroutine facRAS
      include 'SShd.f'
c
      real*8 diags(200),vnois(200),ay(20),
     x XRY(20),work(200),hh(200),c,y,z,w,x,ddif,xad

      integer levs(nrec),iork(200),mfac
c #######################################################
c  determine number of observations per
c   level of the factor, store in levs
      levs = 0
      mfac = 0
      do 8 krec=1,mrec
         iras = rasid(krec)
         if(iras.gt.mfac)mfac = iras
         levs(iras) = levs(iras) + 1
   8  continue
      kend=0
c #######################################################
c   For each level of the factor
c    adjust observations for all other solutions
c    save in XRY, make diags of MME
      do 11 iras = 1,mfac
        jrec = levs(iras)
        XRY = 0.d0
        diags = 0.d0
        if(jrec.lt.1)go to 11
        kstr = kend+1
        kend = kend+jrec

        do 10 lrec = kstr,kend
          krec = pras(lrec)
         iam = anid(krec)
         icg = cgid(krec)
         iym = ymid(krec)
         jdim = days(krec)
         jtim = timg(krec)
         kp = pari(krec)
         ja = (kp-1)*5
         c = ri(jdim,kp)
         y = obs(krec) - sym(iym,jtim)
         do 15 j=1,mcov
            ka=ja+j
            y = y - (scg(icg,j) + sanm(iam,ka)
     x  + sape(iam,ka))*lp(jdim,j)
 15      continue
c
         xad = y*c
       do 17 j=1,mcov
          kb=ja+j
         XRY(j)=XRY(j) + xad*lp(jdim,j)
         do 19 m=j,mcov
           kc=ja+m
           ma=ihmssf(j,m,mcov)
           diags(ma)=diags(ma)+lp(jdim,j)*c*lp(jdim,m)
 19      continue
 17    continue
c
 10    continue
c ####################################################
c  solve for new solution for this level of factor
c
      call dkmvhf(diags,mcov,work,iork)
      vnois = 0.d0
c  ###################################################
c   if estimating covariance matrices then
c     do cholesky decomposition on diags
c     generate sampling variance (vnois) to
c     add to solutions
      if(igibb.gt.0)then
        call cholsk(diags,work,mcov)
        call vgnor(vnois,work,hh,mcov)
      endif
c
      do 25 j=1,mcov
          z = 0.d0
          do 27 k=1,mcov
```

```
            m=ihmssf(j,k,mcov)
            z = z + diags(m)*XRY(k)
 27        continue
c   add vnois, compute convergence criteria
            z = z + vnois(j)
            ddif = z - sras(iras,j)
            ccn = ccn + ddif*ddif
            ccd = ccd + z*z
            sras(iras,j) = 0.5d0*(z + sras(iras,j))
 25      continue
 11   continue
      if(itest.gt.0)then
         jj=6
         print 5003,jj,(sras(jj,L),L=1,5)
 5003 format('  RAS',i4,5f12.4)
      endif
      return
      end
```

### 7.5.3 Contemporary Groups

Contemporary groups (CG) are defined as cows in the same parity number calving within a few months of each other in the same herd and year. CG are modelled by order 4 Legendre polynomials. Because three parities are being analyzed together, it is possible for there to be 3 covariance function matrices for CG effects, i.e. one for each parity. We have assumed this would be true. However, if we find that the three covariance matrices are similar, then we could assume the same covariance function matrix for all parities.

The subroutine for CG is different from that for RAS because CG is a random factor, and we are allowing for three separate covariance function matrices, and the coding has to allow for the estimation of new matrices, and saving them in a file.

**Figure 7.8**

## SScg.f

```
      subroutine facCG
      include 'SShd.f'
c
      real*8 diags(200),vnois(200),ay(20),
     x XRY(20),work(200),hh(200),c,y,z,w,x,ddif,xad

      real*8 ssc(3,15),VIc(15)
      integer levs(nrec),iork(200),levp(nrec),mfac
c #######################################################
c  determine number of observations per
c   level of the factor, store in levs
      levs = 0
      kop = 15
      levp = 0
      ssc=0.d0
      mfac = 0
      do 8 krec=1,mrec
         icg = cgid(krec)
         if(icg.gt.mfac)mfac = icg
         levs(icg) = levs(icg) + 1
         levp(icg) = pari(krec)
   8  continue
      kend=0
c #######################################################
c   For each level of the factor
c    adjust observations for all other solutions
c    save in XRY, make diags of MME
      do 11 icg = 1,mfac
        jrec = levs(icg)
        XRY = 0.d0
        diags = 0.d0
        if(jrec.lt.1)go to 11
        kstr = kend+1
        kend = kend+jrec

        do 10 lrec = kstr,kend
```

```
          krec = pcgid(lrec)
         iam = anid(krec)
         iras = rasid(krec)
         iym = ymid(krec)
         jdim = days(krec)
         jtim = timg(krec)
         kp = pari(krec)
         ja = (kp-1)*5
         c = ri(jdim,kp)
         y = obs(krec) - sym(iym,jtim)
         do 15 j=1,mcov
            ka=ja+j
            y = y - (sras(iras,j) + sanm(iam,ka)
     x  + sape(iam,ka))*lp(jdim,j)
 15      continue
c
         xad = y*c
       do 17 j=1,mcov
          kb=ja+j
         XRY(j)=XRY(j) + xad*lp(jdim,j)
         do 19 m=j,mcov
           kc=ja+m
           ma=ihmssf(j,m,mcov)
           diags(ma)=diags(ma)+lp(jdim,j)*c*lp(jdim,m)
 19      continue
 17    continue
c
 10    continue
c ####################################################
c  Add inverse of covariance function matrix to diags
c   before inverting (one of three possible inverses)
c
      m=0
      kp = levp(icg)
      do 61 ir=1,mcov
      do 61 ic=ir,mcov
       m=m+1
       diags(m)=diags(m)+ci(kp,m)
```

```
  61  continue
      call dkmvhf(diags,mcov,work,iork)
      vnois = 0.d0
c  ###################################################
c   if estimating covariance matrices then
c     do cholesky decomposition on diags
c     generate sampling variance (vnois) to
c     add to solutions
      if(igibb.gt.0)then
        call cholsk(diags,work,mcov)
        call vgnor(vnois,work,hh,mcov)
      endif
c
      do 25 j=1,mcov
          z = 0.d0
          do 27 k=1,mcov
            m=ihmssf(j,k,mcov)
            z = z + diags(m)*XRY(k)
 27        continue
c   add vnois, compute convergence criteria
            z = z + vnois(j)
            ddif = z - scg(icg,j)
            ccn = ccn + ddif*ddif
            ccd = ccd + z*z
            scg(icg,j) = z
 25      continue
c
c  if estimating covariance matrices - must accumulate
c    sum of squares
      if(igibb.gt.0)then
       m=0
       ndf(kp)=ndf(kp)+1
       do 71 ir=1,mcov
         z = scg(icg,ir)
         do 71 ic=ir,mcov
           m=m+1
           ssc(kp,m)=ssc(kp,m)+scg(icg,ic)*z
 71    continue
```

```
      endif
 11   continue
c
c  Estimate new ci matrices
c
       if(igibb.gt.0)then
       kop=15
       do 217 kp=1,3
       nde = ndf(kp) + 2
       call fgchi1(nde,w)
       z=1.d0/w
       do 215 k=1,kop
          VIc(k)=ssc(kp,k)*z
 215   continue
      write(17)iter,VIc
      call dkmvhf(VIc,mcov,work,iork)
      do 216 k=1,kop
        ci(kp,k)=VIc(k)
 216   continue
 217  continue
      endif
      if(itest.gt.0)then
         jj=6
         print 5003,jj,(scg(jj,L),L=1,5)
 5003 format('  CGS',i4,5f12.4)
      endif
      return
      end
```

### 7.5.4 Animal Permanent Environmental

Animal permanent environmental (APE) effects concern all three parities, and are correlated between parities, so that there are 15 covariates to estimate for each animal. Each parity is modelled by order 4 Legendre polynomials. The covariance matrix is therefore, 15 by 15, and there is only one covariance matrix.

## Figure 7.9
SSape.f

```
      subroutine facAPE
      include 'SShd.f'
c
      real*8 diags(nop),vnois(nop),ay(no),
     x XRY(no),work(nop),hh(nop),c,y,z,w,x,ddif,xad

      real*8 ssp(nop),VIp(nop)
      integer levs(nrec),iork(nop),mfac
c #######################################################
c  determine number of observations per
c   level of the factor, store in levs
      levs = 0
      ssp=0.d0
      mfac = 0
      do 8 krec=1,mrec
         iam = anid(krec)
         if(iam.gt.mfac)mfac = iam
         levs(iam) = levs(iam) + 1
   8  continue
      kend=0
c #######################################################
c   For each level of the factor
c    adjust observations for all other solutions
c    save in XRY, make diags of MME
      do 11 iam = 1,mfac
        jrec = levs(iam)
        XRY = 0.d0
        diags = 0.d0
        if(jrec.lt.1)go to 11
        kstr = kend+1
        kend = kend+jrec

        do 10 lrec = kstr,kend
          krec = pcgid(lrec)
         icg = cgid(krec)
         iras = rasid(krec)
         iym = ymid(krec)
         jdim = days(krec)
         jtim = timg(krec)
         kp = pari(krec)
                  ja = (kp-1)*5
         c = ri(jdim,kp)
         y = obs(krec) - sym(iym,jtim)
         do 15 j=1,mcov
            ka=ja+j
            y = y - (sras(iras,j) + sanm(iam,ka)
     x  + scg(icg,j) )*lp(jdim,j)
 15      continue
c
         xad = y*c
       do 17 j=1,mcov
          kb=ja+j
         XRY(kb)=XRY(kb) + xad*lp(jdim,j)
         do 19 m=j,mcov
           kc=ja+m
           ma=ihmssf(kb,kc,no)
           diags(ma)=diags(ma)+lp(jdim,j)*c*lp(jdim,m)
 19      continue
 17    continue
c
 10    continue
c #####################################################
c  Add inverse of covariance function matrix to diags
c   before inverting (one of three possible inverses)
c
      m=0
      do 61 ir=1,no
      do 61 ic=ir,no
       m=m+1
       diags(m)=diags(m)+pi(m)
  61  continue
      call dkmvhf(diags,no,work,iork)
      vnois = 0.d0
c  ###################################################
c   if estimating covariance matrices then
c     do cholesky decomposition on diags
c     generate sampling variance (vnois) to
c     add to solutions
      if(igibb.gt.0)then
        call cholsk(diags,work,no)
        call vgnor(vnois,work,hh,no)
      endif
c
      do 25 j=1,no
          z = 0.d0
          do 27 k=1,no
            m=ihmssf(j,k,no)
            z = z + diags(m)*XRY(k)
 27        continue
c   add vnois, compute convergence criteria
            z = z + vnois(j)
            ddif = z - sape(iam,j)
            ccn = ccn + ddif*ddif
            ccd = ccd + z*z
            sape(icg,j) = z
 25      continue
 c
c  if estimating covariance matrices - must accumulate
c    sum of squares
      if(igibb.gt.0)then
       m=0
       ndf=ndf+1
       do 71 ir=1,no
         z = sape(iam,ir)
         do 71 ic=ir,no
           m=m+1
           ssp(kp,m)=ssp(kp,m)+sape(iam,ic)*z
 71    continue
      endif
 11   continue
```

```
c
c  Estimate new pi matrix
c
       if(igibb.gt.0)then
       nde = ndf + 2
       call fgchi1(nde,w)
       z=1.d0/w
       do 215 k=1,nop
          VIp(k)=ssp(k)*z
 215   continue
       nm=1
      write(19)iter,nm,VIp
      call dkmvhf(VIp,no,work,iork)
      do 216 k=1,nop
        pi(k)=VIp(k)
 216   continue
 217  continue
      endif
      if(itest.gt.0)then
         jj=6
         print 5003,jj,(sape(jj,L),L=1,5)
 5003 format(’  APE’,i4,5f12.4)
      endif
      return
      end
```

### 7.5.5 Animal Additive Genetic

Animal additive genetic (ANM) effects concern all three parities, like the APE, and are correlated between parities, so that there are 15 covariates to estimate for each animal. Each parity is modelled by order 4 Legendre polynomials. The covariance matrix is therefore, 15 by 15, and there is only one covariance matrix.

However, the big difference from APE are the additive genetic relationships that must be taken into account amongst all animals. This accounts for the extra length of the following subroutine.

## Figure 7.10

SSanm.f

```
      subroutine facANM
      include 'SShd.f'
c
      real*8 diags(nop),vnois(nop),ay(no),
     x XRY(no),work(nop),hh(nop),c,y,z,w,x,ddif,xad

      real*8 ssa(nop),VIa(nop),tcc(no),dg
      integer levs(nrec),iork(nop),mfac
c #######################################################
c  determine number of observations per
c   level of the factor, store in levs
      levs = 0
      ssa=0.d0
      mfac = mam
      do 8 krec=1,mrec
         iam = anid(krec)
         levs(iam) = levs(iam) + 1
   8  continue
      kend=0
c #######################################################
c   For each level of the factor
c    adjust observations for all other solutions
c    save in XRY, make diags of MME
      do 11 iam = 1,mfac
        jrec = levs(iam)
        XRY = 0.d0
        diags = 0.d0
        if(jrec.lt.1)go to 11
        kstr = kend+1
        kend = kend+jrec

        do 10 lrec = kstr,kend
          krec = pcgid(lrec)
         icg = cgid(krec)
         iras = rasid(krec)
         iym = ymid(krec)
         jdim = days(krec)
         jtim = timg(krec)
         kp = pari(krec)
                  ja = (kp-1)*5
         c = ri(jdim,kp)
         y = obs(krec) - sym(iym,jtim)
         do 15 j=1,mcov
            ka=ja+j
            y = y - (sras(iras,j) + sape(iam,ka)
     x  + scg(icg,j) )*lp(jdim,j)
 15      continue
c
         xad = y*c
       do 17 j=1,mcov
          kb=ja+j
         XRY(kb)=XRY(kb) + xad*lp(jdim,j)
         do 19 m=j,mcov
           kc=ja+m
           ma=ihmssf(kb,kc,no)
           diags(ma)=diags(ma)+lp(jdim,j)*c*lp(jdim,m)
 19      continue
 17    continue
c
 10    continue
c
c  Must account for genetic relationships among animals
c
 50   uped = jped(iam)
      tcc=0.d0
      if(uped.lt.1)go to 11
      iam = cpa(uped)
      if(iam.ne.ianm)print *,'xxxxx',iam,ianm
c
 850  jcode = cpc(uped)
      if(cpa(uped).ne.ianm)go to 432
      if(jcode.eq.0)then
```

```
         js = cps(uped)
         jd = cpd(uped)
         c = bii(ianm)*0.5d0
         do 406 jc=1,ntr
            tcc(jc)=tcc(jc)+c*(sanm(js,jc)+sanm(jd,jc))
 406     continue
      else
         jp = cps(uped)
         jm = cpd(uped)
         d = bii(jp)*0.5d0
         do 412 ja=1,ntr
            tcc(ja)=tcc(ja)+d*(sanm(jp,ja)-0.5d0*sanm(jm,ja))
 412     continue
      endif
c
 405  uped = uped + 1
      if(iam.ne.cpa(uped))go to 432
      if(uped.gt.mped)go to 432
      go to 850
c
 432  do 435 jr=1,no
         s=0.d0
         do 437 jc=1,no
            s=s + gi(ihmssf(jr,jc,ntr))*tcc(jc)
 437     continue
         XRY(jr)=XRY(jr)+s
 435  continue
c ####################################################
c  Add inverse of covariance function matrix to diags
c   before inverting (one of three possible inverses)
c
      dg = adiag(iam)
      m=0
      do 61 ir=1,no
      do 61 ic=ir,no
       m=m+1
       diags(m)=diags(m)+gi(m)*dg
  61  continue
```

```
      call dkmvhf(diags,no,work,iork)
      vnois = 0.d0
c  ###################################################
c   if estimating covariance matrices then
c     do cholesky decomposition on diags
c     generate sampling variance (vnois) to
c     add to solutions
      if(igibb.gt.0)then
        call cholsk(diags,work,no)
        call vgnor(vnois,work,hh,no)
      endif
c
      ay=0.d0
      js = sir(iam)
      jd = dam(iam)
      do 25 j=1,no
          z = 0.d0
          do 27 k=1,no
            m=ihmssf(j,k,no)
            z = z + diags(m)*XRY(k)
 27         continue
c   add vnois, compute convergence criteria
            z = z + vnois(j)
            ddif = z - sanm(iam,j)
            ccn = ccn + ddif*ddif
            ccd = ccd + z*z
            sanm(iam,j) = z
          ay(j)=sanm(iam,j)-0.5d0*(sanm(ja,j)+sanm(jd,j))
 25       continue
c
c  if estimating covariance matrices - must accumulate
c    sum of squares of Mendelian sampling terms
      if(igibb.gt.0)then
       if(jrec.gt.0)then
       m=0
       ndf=ndf+1
       d = bii(iam)
       do 71 ir=1,no
```

```
         z = ay(ir)*d
         do 71 ic=ir,no
           m=m+1
           ssa(m)=ssa(m)+ay(ic)*z
 71    continue
      endif
      endif
 11   continue
c
c  Estimate new gi matrix
c
       if(igibb.gt.0)then
       nde = ndf + 2
       call fgchi1(nde,w)
       z=1.d0/w
       do 215 k=1,nop
          VIa(k)=ssa(k)*z
 215   continue
       nm=2
      write(19)iter,nm,VIa
      call dkmvhf(VIa,no,work,iork)
      do 216 k=1,nop
        gi(k)=VIa(k)
 216   continue
 217  continue
      endif
      if(itest.gt.0)then
         jj=6
         print 5003,jj,(sanm(jj,L),L=1,5)
 5003 format('  ANM',i4,5f12.4)
      endif
      return
      end
```

### 7.5.6 Residual Effects

If the program is set to estimate covariance matrices, then a subroutine is needed to estimate the residual variances by parity and by periods within a lactation. If igibb=0, then this subroutine is skipped, and no residual variances are calculated.

**Figure 7.11**
SSres.f

```
     subroutine facRES
      include 'SShd.f'
c
      real*8 diags(nop),vnois(nop),ay(no),
     x XRY(no),work(nop),hh(nop),c,y,z,w,x,ddif,xad

      real*8 sse(3,ntim),VI(nop),ndf(3,ntim)
      integer levs(nrec),iork(nop),mfac
c #######################################################
c  determine number of observations per
c   level of the factor, store in levs
      levs = 0
      sse=0.d0
      mfac = 0
      do 8 krec=1,mrec
         itim = timg(krec)
         levs(itim) = levs(itim) + 1
   8  continue
      kend=0
c #######################################################
c   For each level of the factor
c    adjust observations for all other solutions
c    save in XRY, make diags of MME
      do 11 itim = 1,mfac
        jrec = levs(itim)
        if(jrec.lt.1)go to 11
        kstr = kend+1
        kend = kend+jrec

        do 10 lrec = kstr,kend
          krec = pcgid(lrec)
         iam = anid(krec)
         icg = cgid(krec)
         iras = rasid(krec)
         iym = ymid(krec)
         jdim = days(krec)
         jtim = timg(krec)
         kp = pari(krec)
           ja = (kp-1)*5
         c = ri(jdim,kp)
         y = obs(krec) - sym(iym,jtim)
         do 15 j=1,mcov
            ka=ja+j
            y = y - (sras(iras,j) + sape(iam,ka)
     x  + scg(icg,j)+sanm(iam,ka) )*lp(jdim,j)
 15      continue
c
       sse(jtim,kp)=sse(jtim,kp)+y*y
       ndf(jtim,kp)=ndf(jtim,kp)+1.d0
c
 10    continue
c
      do 31 jtim=1,4
      do 32 kp=1,3
        nde = ndf(jtim,kp)+2
        call fgchi1(nde,w)
        res(jtim,kp) = sse(jtim,kp)/w
 32    continue
 31   continue
      ri=0.d0
      do 41 i=1,45
        do 141 kp=1,3
       ri(i,kp)=1.d0/res(1,kp)
 141    continue
```

```
  41  continue
      do 42 i=46,115
        do 142 kp=1,3
       ri(i,kp)=1.d0/res(2,kp)
 142    continue
  42  continue
      do 43 i=116,265
         do 143 kp=1,3
         ri(i,kp)=1.d0/res(3,kp)
 143     continue
  43  continue
      do 44 i=266,365
         do 144 kp=1,3
         ri(i,kp)=1.d0/res(4,kp)
 144     continue
  44   continue
c
c  save new estimates in file with sample number
c
      write(20,1235)iter,(res(i,j),i=1,4),j=1,3)
 1235 format(1x,i10,12f15.5)
c
      return
      end
```

### 7.5.7 Finish Off

The last subroutine is `call finis`, which is to save solutions for the important factors, usually just the genetic evaluations of animals. However, some may want to save all of the solutions for all factors.

With the genetic evaluations one may also want to save information about the number of records each animal had (by parity number), and perhaps number of progeny, and sire and dam identifications. This information could be used to approximate the reliabilities of the EBVs.

Thus, this last subroutine depends on the wishes of the user to decide what information should be saved and how it should be saved. Thus, no coding will be provided for this subroutine.

## 7.6 Other Items

If Gibbs sampling was performed then there will be three files of sample estimates for each covariance matrix and the residuals. The burn-in period needs to be determined, then the remaining samples need to be averaged in some manner. Either all of the samples, after burn-in, could be sampled, or every $m^{th}$ sample could be averaged, where $m$ is a number like 7 or 17 or 19. Consecutive samples are known to be dependent on the previous sample, and by averaging every $m^{th}$ sample this dependency is lessened considerably. Often the same results are obtained either way.

After new covariance matrices are available, then another run is made where the new parameters are inputted and `igibb = 0` is imposed. This is to obtain solutions to the mixed model equations (MME).

Having the EBV, then one can compute the 305-d breeding values and persistency in a follow-up program. There may be other necessities to calculate for users of the EBVs. Note also that none of the preliminary programs have been provided. Programs for preparing the data, numbering the levels of each factor, editting out the error records, and ordering the animals chronologically for calculating inbreeding coefficients have also not been shown.

The programs shown in this chapter are not available for downloading. If you want to use them, then you should type them in from these pages. Why? Because it will help you to learn what the programs are doing, and you might find that I have some errors in them. I hope there are few errors, but you could find some. As I mentioned at the beginning of this chapter, the programs are merely to give you an idea about writing code for a random regression model.

# Chapter 8

# DKMVHF - Inversion Routine

Matrix inversion routine of C. R. Henderson, as modified by Karin Meyer in 1983. Input is a half-stored symmetric matrix. There can be zero rows and columns present in the matrix.

```
       SUBROUTINE DKMVHF(A,N,V,IFLAG)
C                                             KARIN MEYER
C                                             NOVEMBER 1983
C-----------------------------------------------------------------------
      DOUBLE PRECISION A(1),V(1),XX,DMAX,AMAX,ZERO,DIMAX
      INTEGER IFLAG(1)
      zero=1.D-12
      IF(N.EQ.1)THEN
      XX=A(1)
      IF(DABS(XX).GT.ZERO)THEN
      A(1)=1.D0/XX
      ELSE
      A(1)=0.D0
      END IF
      RETURN
      END IF
      N1=N+1
      NN=(N*N1)/2
      DO 1 I=1,N
    1 IFLAG(I)=0
C
C      SET MINIMUM ABSOLUTE VALUE OF DIAGONAL ELEMENTS FOR
C      NON-SINGULARITY (MACHINE SPECIFIC!)
       ZERO=1.D-12
C-----------------------------------------------------------------------
C      START LOOP OVER ROWS/COLS
C-----------------------------------------------------------------------
      DO 8 II=1,N
C      ... FIND DIAGONAL ELEMENT WITH BIGGEST ABSOLUTE VALUE
          DMAX=0.D0
          AMAX=0.D0
          KK=-N
          DO 2 I=1,N
C       ... CHECK THAT THIS ROW/COL HAS NOT BEEN PROCESSED
              KK=KK+N1-I
              IF(IFLAG(I).NE.0)GO TO 2
              K=KK+I
              IF(DABS(A(K)).GT.AMAX)THEN
                  DMAX=A(K)
                  AMAX=DABS(DMAX)
                  IMAX=I
              END IF
    2     CONTINUE
C      ... CHECK FOR SINGULARITY
          IF(AMAX.LE.ZERO)GO TO 11
C      ... ALL ELEMENTS SCANNED,SET FLAG
          IFLAG(IMAX)=II

C      ... INVERT DIAGONAL
          DIMAX=1.D0/DMAX
C      ... DEVIDE ELEMENTS IN ROW/COL PERTAINING TO THE BIGGEST
C      DIAGONAL ELEMENT BY DMAX
          IL=IMAX-N
          DO 3 I=1,IMAX-1
```

```
              IL=IL+N1-I
              XX=A(IL)
              A(IL)=XX*DIMAX
          IF(DABS(XX).LT.0.1D-17)XX=0.D0
    3     V(I)=XX
C      ... NEW DIAGONAL ELEMENT
          IL=IL+N1-IMAX
          A(IL)=-DIMAX
          DO 4 I=IMAX+1,N
              IL=IL+1
              XX=A(IL)
              A(IL)=XX*DIMAX
          IF(DABS(XX).LT.0.1D-17)XX=0.D0
    4     V(I)=XX
C      ... ADJUST THE OTHER ROWS/COLS :
C      A(I,J)=A(I,J)-A(I,IMAX)*A(J,IMAX)/A(IMAX,IMAX)
          IJ=0
          DO 7 I=1,N
              IF(I.EQ.IMAX)THEN
      IJ=IJ+N1-I
      GO TO 7
      END IF
              XX=V(I)
              IF(XX.NE.0.D0)THEN
                  XX=XX*DIMAX
                  DO 5 J=I,N
                      IJ=IJ+1
                      IF(J.NE.IMAX)A(IJ)=A(IJ)-XX*V(J)
    5             CONTINUE
              ELSE
    6             IJ=IJ+N1-I
              END IF
    7     CONTINUE
C      ... REPEAT UNTIL ALL ROWS/COLS ARE PROCESSED
    8 CONTINUE
C-----------------------------------------------------------------------
C      END LOOP OVER ROWS/COLS
C-----------------------------------------------------------------------
```

```
C       ... REVERSE SIGN
      DO 9 I=1,NN
    9 A(I)=-A(I)
C       ... AND THAT'S IT !
C     PRINT 10,N
   10 FORMAT(' FULL RANK MATRIX INVERTED, ORDER =',I5)
C       RETURN RANK AS LAST ELEMENT OF FLAG VECTOR
       IFLAG(N)=N
      RETURN
C-----------------------------------------------------------------------
C       MATRIX NOT OF FULL RANK, RETURN GENERALISED INVERSE
C-----------------------------------------------------------------------
   11 IRANK=II-1
      IJ=0
      DO 14 I=1,N
          IF(IFLAG(I).EQ.0)THEN
C       ... SET REMAINING N-II ROWS/COLS TO ZERO
              DO 12 J=I,N
                  IJ=IJ+1
                  A(IJ)=0.D0
   12         CONTINUE
          ELSE
              DO 13 J=I,N
                  IJ=IJ+1
                  IF(IFLAG(J).NE.0)THEN
C       ... REVERSE SIGN FOR II-1 ROWS/COLS PREVIOUSLY PROCESSED
                      A(IJ)=-A(IJ)
                  ELSE
                      A(IJ)=0.D0
                  END IF
   13         CONTINUE
          END IF
   14 CONTINUE
C     PRINT 15,N,IRANK
C  15 FORMAT(' GENERALISED INVERSE OF MATRIX WITH ORDER =',I5,
C    1  '   AND RANK =',I5)
      IFLAG(N)=IRANK
      RETURN
      END
c
c  half-stored matrix subscripting function
c
      FUNCTION IHMSSF(I,J,N)
      IF(I-J)1,1,2
 1    IHMSSF=((N+N-I)*(I-1))/2+J
      RETURN
 2    IHMSSF=((N+N-J)*(J-1))/2+I
      RETURN
      END
```

# Chapter 9

# CHOLSK - Cholesky Decomposition

```
C
C  SUBROUTINE FOR CALCULATING Cholesky decomp
C
      SUBROUTINE cholsk(a,b,n)
      REAL*8 a(1),b(1),x,y,z,w
C
C a IS SYMETRIC, HALF STORED
C b triangular storeD COLUMN-WISE
C n IS THE ORDER OF a
C
C  COMPUTE b FIRST
C
      b = 0.d0
      DO 1 i=1,n
         m = ihmssf(i,i,n)
         x = a(m)
       if(x.gt.0.d0)then
         im = i - 1

           if(i.gt.1)then
            do 2 j=1,im
```

```
               y = b(ihmssf(j,i,n))
               x = x - y*y
 2          continue
           endif

       else
         x=0.d0
       endif

   20  ma=ihmssf(i,i,n)
       b(ma)=0.d0
       z=0.d0
       if(x.gt.0.d0)then
         b(ma) = dsqrt(x)
         z = 1.d0/b(ma)
       endif
      ip = i+1
      if(ip.gt.n)go to 1
         do 3 j = ip,n
            mb = ihmssf(j,i,n)
            x = a(mb)
         if(i.gt.1)then
            do 4 k=1,im
               y = b(ihmssf(j,k,n))
               w = b(ihmssf(i,k,n))
               x = x - y*w
 4          continue
         endif
       b(mb) = x*z
 3     continue
 1    continue
C
      return
      end
```

Below is a routine for creating a vector of random normal deviates with a particular covariance structure, for Gibbs sampling usage.

```
      subroutine vgnor(v,w,t,n)
      real*8 v(1),w(1),t(1),u,z
      integer n
       v=0.d0
      do 5 i=1,n
         call fgnor3(u)
         t(i)=u
5     continue
      do 7 i=1,n
         z = 0.d0
         do 6 j=1,i
            m=ihmssf(i,j,n)
            z = z + w(m)*t(j)
6        continue
        v(i) = z
7     continue
      return
      end
```

# Chapter 10

# References

Albuquerque, L. G., Meyer, K. 2001. Estimates of covariance functions for growth from birth to 630 days of age in Nelore cattle. J. Anim. Sci. 79:2776-2789.

Ali, T. E., L. R. Schaeffer. 1987. Accounting for covariances among test day milk yields in dairy cows. Can. J. Anim. Sci. 67:637-644.

Anang, A., Mielenz, N., Schuler, L. 2000. Genetic and phenotypic parameters for monthly egg production in White Leghorn hens. J. Anim. Breed. Genet. 117:407-415.

Andersen, S., Pedersen, B. 1996. Growth and food intake curves for group-housed gilts and castrated male pigs. Animal Science 63:457-464.

Ducrocq, V. 1987. An analysis of productive life in dairy cattle. Ph.D. Diss., Cornell University, Ithaca, NY.

Ducrocq, V., R. L. Quaas, E. J. Pollak, G. Casella. 1988. Length of productive life of dairy cows. 1. Justification of a Weibull model. J. Dairy Sci. 71:3061-3070.

Ducrocq, V., R. L. Quaas, E. J. Pollak, G. Casella. 1988. Length of productive life of dairy cows. 2. Variance component estimation and sire evaluation. J. Dairy Sci. 71:3071-3079.

Ducrocq, V. 1994. Statistical analysis of length of productive life for dairy cows in the Normande breed. J. Dairy Sci. 77:855-865.

Ducrocq, V., J. Solkner. 1994. The Survival Kit, a Fortran package for the analysis of survival data. In Proc. $5^{th}$ World Cong. on Genet. Appl. To Livest. Prod.

Ducrocq, V., J. Solkner. 1998. Implementation of a routine breeding value evaluation for longevity of dairy cows using survival analysis techniques. In Proc. $6^{th}$ World Cong. on Genet. Appl. To Livest. Prod. p. 359-362.

Galbraith, F.] 2003. Random regression models to evaluate sires for daughter survival. Master's Thesis, University of Guelph, Ontario, Canada, August.

Henderson, Jr., C. R. 1982. Analysis of covariance in the mixed model: Higher level, nonhomogeneous, and random regressions. Biometrics 38:623-640.

Henderson, C. R. 1984. *Applications of Linear Models in Animal Breeding.* University of Guelph.

Jamrozik, J., Schaeffer, L. R. 1997. Estimates of genetic parameters for a test day model with random regressions for production of first lactation Holsteins. J. Dairy Sci. 80:762-770.

Jamrozik, J., L. R. Schaeffer, J. C. M. Dekkers. 1997. Genetic evaluation of dairy cattle using test day yields and random regression model. J. Dairy Sci. 80:1217-1226.

Jamrozik, J., L. R. Schaeffer. 2000. Comparison of two computing algorithms for solving mixed model equations for multiple trait random regression test day models. Livest. Prod. Sci. 67:143-153.

Jamrozik, J., Schaeffer, L. R., Weigel, K. A. 2002. Genetic evaluation of bulls and cows with single- and multiple-country test-day models. J. Dairy Sci. 85:1617-1622.

Jamrozik, J., J. Fatehi, L. R. Schaeffer. 2008. Comparison of models for genetic evaluation of survival traits in dairy cattle: a simulation study. J. Anim. Breed. Genet. 125:75-83.

Jamrozik, J., J. Bphmanova, L. R. Schaeffer. 2010. Selection of locations of knots for linear splines in random regression test-day models. J. Anim. Breed. Genet. 127:87-92.

Jones, H. E., White, I. M. S., Brotherstone, S. 1999. Genetic evaluation of Holstein Friesian sires for daughter condition score changes using a random regression model. Animal Science 68:467-475.

Kirkpatrick, M., Lofsvold, D., Bulmer, M. 1990. Analysis of the inheritance, selection and evolution of growth trajectories. Genetics 124:979-993.

Laird, N. M., Ware, J. H. 1982. Random effects models for longitudinal data. Biometrics 38:963-974.

Liu, X. 1998. The estimation of genetic parameters of test day dry matter intake, energy intake and milk yield of Holstein cows. M.Sc. Thesis, University of Guelph, Guelph, Ontario, Canada.

McKay, L. R., Schaeffer, L. R., McMillan, I. 2002. Analysis of growth curves in rainbow trout using random regression. $7^{th}$ World Congress of Genetics Applied to Livestock Production, paper 241.

Meyer, K., H.-U. Graser, K. Hammond. 1989. Estimates of genetic parameters for first lactation test day production of Australian Black and White cows. Livest. Prod. Sci. 21:177-199.

Meyer, K. 1999. Estimates of genetic and phenotypic covariance functions for postweaning growth and mature weight of beef cows. J. Anim. Breed. Genet. 116:181-205.

Meyer, K. 2000. Random regressions to model phenotypic variation in monthly weights of Australian beef cows. Livest. Prod. Sci. 65:19-38.

Meyer, K., Hill, W. G. 1997. Estimation of genetic and phenotypic covariance functions for longitudinal or 'repeated' records by restricted maximum likelihood. Livest. Prod. Sci. 47:185-200.

Meyer, K. 2005. Random regression analyses using B-splines to model growth of Australian Angus cattle. Genet. Sel. Evol. 37:473-500.

Misztal, I. 2006. Properties of random regression models using linear splines. J. Anim. Breed. Genet. 123:74-80.

Pool, M. H., Janss, L. L. G., Meuwissen, T. H. E. 2000. Genetic parameters of Legendre polynomials for first parity lactation curves. J. Dairy Sci. 83:2640-2649.

Ptak, Ewa, L. R. Schaeffer. 1993. Use of test day yields for genetic evaluation of dairy sires and cows. Livest. Prod. Sci. 34:23-34.

Reents, R.], J. C. M. Dekkers, L. R. Schaeffer. 1995. Genetic evaluation for somatic cell score with a test day model for multiple lactations. J. Dairy Sci. 78:2858-2870.

Schaeffer, L. R., E. B. Burnside. 1976. Estimating the shape of the lactation curve. Can. J. Anim. Sci. 56:157-170.

Schaeffer, L. R., Dekkers, J. C. M. 1994. Random regressions in animal models for test-day production in dairy cattle. Proc. $5^{th}$ World Congress of Genetics Applied to Livestock Production. Guelph, Ontario, Canada XVIII:443-446.

Schaeffer, L. R., Jamrozik, J., Kistemaker, G. J., Van Doormaal, B. J. 2000. Experience with a test-day model. J. Dairy Sci. 83:1135-1144.

Schaeffer, L. R. 2004. Application of random regression models in animal breeding. Livest. Prod. Sci. 86:35-45.

Schenkel, F. S., Miller, S. P., Jamrozik, J., Wilton, J. W. 2002. Two step and random regression analyses of weight gain of station tested beef bulls. J. Anim. Sci. 80:1497-1507.

Schnyder, U., Hofer, A., Kunzi, N. 2001. Impact of variation in length of individual testing periods on estimation of covariance components of a random regression model for feed intake of growing pigs. J. Anim. Breed. Genet. 118:235-246.

Stanton, T. L., L. R. Jones, R. W. Everett, S. D. Kachman. 1992. Estimating milk, fat, and protein lactation curves with a test day model. J. Dairy Sci. 75:1691-1700.

Strandberg, E., Kolmodin, R., Madsen, P., Jensen, J., Jorjani, H. 2000. Genotype by environment interaction in Nordic dairy cattle studied by use of reaction norms. Proc. Interbull Meeting, Bled, Slovenia. Bulletin 25:41-45.

Swalve, H. H. 1995. Test day models in the analysis of dairy production data: A review. Arch. Tierz. 38:591-612.

Swalve, H. H. 2000. Theoretical basis and computational methods for different test-day genetic evaluation methods. J. Dairy Sci. 83:1115-1124.

Uribe, H., Schaeffer, L. R., Jamrozik, J., Lawlor, T. J. 2000. Genetic evaluation of dairy cattle for conformation traits using random regression models. J. Anim. Breed. Genet. 117:247-259.

van der Werf, J. H. J., M. E. Goddard, K. Meyer. 1998. The use of covariance functions and random regressions for genetic evaluation of milk production based on test day records. J. Dairy Sci. 81:3300-3308.

Veerkamp, R. F., Thompson, R. 1999. A covariance function for feed intake, live weight, and milk yield estimated using a random regression model. J. Dairy Sci. 82:1565-1573.

Veerkamp, R. F., Brotherstone, S., Meuwissen, T. H. E. 1999. Survival analysis using random regression models. Proc. International Workshop on EU Concerted Action Genetic Improvement of Functional Traits in Cattle; Longevity. Interbull Bulletin 21:36-40.

Veerkamp, R. F., Koenen, E. P. C., DeJong, G. 2001. Genetic correlations among body condition score, yield, and fertility in first parity cows estimated by random regression models. J. Dairy Sci. 84:2327-2335.

Wilmink, J. B. M., 1987. Adjustment of test day milk, fat, and protein yield for age, season and stage of lactation. Livest. Prod. Sci. 16, 335-348.

Wood, P. D. P. 1967. Algebraic model of the lactation curve in cattle. Nature 216:164-165.

Wood, P. D. P. 1968. Factors affecting the shape of the lactation curve in cattle. Anim. Prod. 11:307-316.