GLAM: The IWSM Story

Iain Currie¹

¹ Heriot-Watt University, Edinburgh, UK

E-mail for correspondence: I.D.Currie@hw.ac.uk

Abstract: Data that can be arranged in an array are common in statistics and then models often have a row-column-depth-... structure. In such cases, data sets and models can be large, and can present computational problems even for modern computers, particularly when smoothing is carried out in a Generalized Linear Model (GLM) framework. A Generalized Linear Array Model or GLAM is a fast low-footprint method of computation for such data sets and models. The GLAM approach and algorithms were first described at the IWSM in Florence in 2004. We describe GLAM and then mention some of its many applications since that time. The IWSM conferences have played a key role in GLAM’s development. We tell the story of GLAM through these workshops, starting with the 1999 conference in Graz and finishing in Bristol in 2018.

Keywords: Arrays; Efficient Computation; GLAM; P-splines; Smoothing.

1 The Motivating Example

The forecasting of human mortality rates at individual ages is necessary for the correct pricing of annuity and pension products in the insurance business, the provision of state pensions and, more generally, care for the elderly in an ageing society. This is a hard problem since it is self-evident that the future course of mortality is dependent on many unknown factors, such as medical advances, drug resistance, obesity, and many more besides. Some measure of the reliability of any forecast is certainly required.

To fix ideas, we suppose we have mortality data for ages 50 to 90 and for years 1975 to 2015. These data lie naturally in a matrix with rows indexed by age and columns indexed by year. The data thus comprise two matrices

\[ D = (d_{x,y}) \], the matrix of the number of deaths at age \( x \) in year \( y \), and

\[ E = (e_{x,y}) \], the matrix of the total time lived at age \( x \) in year \( y \).

Plots of the observed log hazard rates, \( \log(d_{x,y}/e_{x,y}) \), show that mortality increases
We use the method of P-splines (Eilers and Marx, 1996). We define B-spline bases for age: \( B_a = \{B_{a,1}, \ldots, B_{a,c_a}\} \) and year: \( B_y = \{B_{y,1}, \ldots, B_{y,c_y}\} \). Let \( x_a = (x_1, \ldots, x_{n_a})^T \) and \( x_y = (y_1, \ldots, y_{n_y})^T \) denote the vectors of ages and years. Let \( B_a = [B_{a,1}(x_a) : \cdots : B_{a,c_a}(x_a)] \), \( n_a \times c_a \), and \( B_y = [B_{y,1}(x_y) : \cdots : B_{y,c_y}(x_y)] \), \( n_y \times c_y \), be the marginal regression matrices for age and year. Then

\[
B = B_y \otimes B_a, n_a n_y \times c_a c_y,
\]

is a suitable regression matrix for 2-dimensional modelling of the age-year log hazard surface; here \( \otimes \) denotes the Kronecker product. The left panel of Figure 1 illustrates the underlying 2-dimensional basis. As in one dimension, we can associate each regression coefficient with the summit of its associated B-spline. The right panel shows the location of both the data and coefficient grid; GLAM takes advantage of this data and coefficient structure.

We force smoothness of the fitted surface by smoothing (1) the columns of the coefficient grid, i.e. smoothing by age, and (2) the rows of the coefficient grid, i.e. smoothing by year. The 2-dimensional penalty is

\[
P = \lambda_a I_{c_y} \otimes D_y^T D_a + \lambda_y D_a^T D_y \otimes I_{c_a}
\]

where \( D_a \) and \( D_y \) are second order difference matrices, and \( I_{c_a} \) and \( I_{c_y} \) are identity matrices; see Currie et al. (2004) for further details.

We assume that \( D_{x,y} \sim \mathcal{P}(e_{x,y}\mu_{x,y}) \) where \( \mu_{x,y} \) is the hazard rate or force of mortality at age \( x \) in year \( y \). If there is no penalty we have a generalized
linear model. One of the attractive features of $P$-splines is the way the penalty $P$ is incorporated into the scoring algorithm for a GLM. We have

$$(B^t\tilde{W}_d B + P)\hat{\theta} = B^t\tilde{W}_d \tilde{z}$$

(3)

where $\tilde{W}_d$ is the diagonal matrix of weights and $\tilde{z}$ is the working-variable vector; here $\hat{\cdot}$ and $\tilde{\cdot}$ represent current and updated estimates respectively. In equation (3) there is no attempt to take advantage of the structure of the data and/or the model. The $n_a \times n_y$ data matrices $D$ and $E$ have been flattened to vectors $d$ and $e$, and the $c_a \times c_y$ coefficient matrix $\Theta$ has also been flattened to a vector $\theta$. In principle, this gives a simple solution but certainly fifteen years ago, when this work was first done, there were two worries: the model matrix was large and computational time was substantial. Importantly, there was little prospect of extending the computation to three or higher dimensions. This was the problem that GLAM set out to address.

2 Array Arithmetic

We consider a sequence of data, model matrices and coefficient structures:

1. One dimension: data vector $y$, $n_1 \times 1$, model matrix $X_1$, $n_1 \times c_1$, coefficient vector $\theta$, $c_1 \times 1$ (ordinary regression).

2. Two dimensions: data matrix $Y$, $n_1 \times n_2$, model matrix $X_2 \otimes X_1$, $n_1 n_2 \times c_1 c_2$, coefficient matrix $\Theta$, $c_1 \times c_2$ (our motivating example).

3. Three dimensions: data array $Y$, $n_1 \times n_2 \times n_3$, model matrix $X_3 \otimes X_2 \otimes X_1$, $n_1 n_2 n_3 \times c_1 c_2 c_3$, coefficient array $\Theta$, $c_1 \times c_2 \times c_3$.

Before we consider the efficient evaluation of the penalized scoring algorithm (3) we look at a simpler problem: the efficient evaluation of the fitted values $X\hat{\theta}$, where $\hat{\theta}$ is an estimate of $\theta$. In one dimension there is no special structure and we proceed as usual. In two dimensions with $X = X_2 \otimes X_1$ we have the beautiful and well-known formula

$$(X_2 \otimes X_1)\hat{\theta}, n_1 n_2 \times 1 \equiv X_1 \tilde{\Theta} X_2^T = (X_2 (X_1 \tilde{\Theta})^2)^T, n_1 \times n_2,$$  

(4)

where $\equiv$ indicates that each side contains the same values but not arranged in the same way. Searle (1982, p333) gives this formula but what he doesn’t mention are two very remarkable properties:

1. the right hand side (RHS) operates sequentially on $\tilde{\Theta}$ without the storage of the potentially large matrix $X_2 \otimes X_1$,

2. less obviously, the number of multiplications required to evaluate the RHS is substantially smaller than that required to evaluate the LHS.
The extension of the RHS of (4) to three dimensions requires: (1) the “multiplication” of the matrix $X_1$ onto the 3-dimensional coefficient array $\hat{\Theta}$, (2) the “transpose” of the result, and (3) the repetition of (1) and (2) with first $X_2$ and then $X_3$. We must define what we mean by multiplication and transpose in this situation.

We consider the “multiplication” of the matrix $X_1$, $n_1 \times c_1$, onto the array $A$, $c_1 \times c_2 \times c_3$. We think of $A$ as $c_3$ matrices of size $c_1 \times c_2$ and construct the $c_1 \times c_2 c_3$ matrix, $M$, as follows:

\[
\begin{array}{ccc}
1 & 2 & 3 \\
\hline
1 & & \\
2 & & \\
3 & & \\
\end{array}
\]

We form the ordinary matrix product $X_1 M$ with size $n_1 \times c_2 c_3$. Next we reinstate dimensions two and three of $X_1 M$ to give a $n_1 \times c_2 \times c_3$ array denoted $H(X_1, A)$, the $H$-transform of the array $A$ by the matrix $X_1$. The second requirement is the definition of the “transpose” of an array. We define the rotation of the array $A$, $c_1 \times c_2 \times c_3$, to be the array $R(A)$, $c_2 \times c_3 \times c_1$, obtained by permuting the indices of $A$.

It is convenient to combine these two operations into a single function: the rotated $H$-transform of the array $A$ by the matrix $X_1$ is

\[
\rho(X_1, A) = R\{H(X_1, A)\}. \tag{5}
\]

We note that $\rho(X_1, A)$ is $c_2 \times c_3 \times n_1$ so this array is ready to receive, $X_2$, a matrix with $c_2$ columns.

We return to our original problem: the efficient evaluation of $X\hat{\theta}$ in item 3 at the beginning of this section. We have

\[
(X_3 \otimes X_2 \otimes X_1)\hat{\theta}, n_1 n_2 n_3 \times 1 \equiv \rho(X_3, \rho(X_2, \rho(X_1, \hat{\theta}))), n_1 \times n_2 \times n_3. \tag{6}
\]

We obtain a nice corollary to (6) if we replace $\hat{\theta}$ with $y$, $\hat{\Theta}$ with $Y$ and $X$ with $X^T$. Then (6) becomes

\[
X^T y, c_1 c_2 c_3 \times 1 \equiv \rho(X_3^T, \rho(X_2^T, \rho(X_1^T, Y))), c_1 \times c_2 \times c_3. \tag{7}
\]

If we also assume that $X^T X = I$ and we have independent normal errors with common variance (as is the case in a classical factorial design) then $\hat{\theta} = X^T y$ and (7) is a general (and compact) statement of Yates’s algorithm (Yates, 1937), a popular method of computation in factorial experiments in pre-computer days.
3 The GLAM algorithms

We turn now to the efficient computation of first the scoring algorithm (3) and second the standard errors of fitted values. We need one further definition: the row tensor of a matrix $X$ with $c$ columns is

$$G(X) = (X \otimes 1^T) * (1^T \otimes X) \quad (8)$$

where $1$ is a vector of 1s of length $c$ and $*$ indicates element-by-element multiplication. We note that the columns of $G(X)$ contain all the pairwise products $x_j \times x_k$ of the columns of $X$. We now state the three GLAM algorithms in the 2-dimensional case $X_1, n_1 \times c_1, X_2, n_2 \times c_2$ and $\Theta, c_1 \times c_2$.

The general formulae are the obvious extensions of these formulae. Let $X = X_2 \otimes X_1$

1. **Linear function:**

$$X\theta, n_1 n_2 \times 1 \equiv \rho[X_2, \rho(X_1, \Theta)], n_1 \times n_2. \quad (9)$$

2. **Inner product function:** let $\tilde{W}$ be the $n_1 \times n_2$ matrix of weights, i.e. $\text{vec}(\tilde{W}) = \text{diag}(\tilde{W}d)$. We have

$$X^T \tilde{W} d X, c_1 c_2 \times c_1 c_2, \equiv \rho[G(X_2)^T, \rho(G(X_1)^T, \tilde{W})], c_1^2 \times c_2^2. \quad (10)$$

3. **Diagonal function:** we have $\text{Var}(\hat{\theta}) = (X^T \tilde{W} d X + P)^{-1} = S_m$, say. We seek the diagonal elements of $\text{Var}(X \hat{\theta}) = X S_m X^T$. This variance matrix is $n_1 n_2 \times n_1 n_2$ and is potentially huge; the GLAM formula calculates the $n_1 n_2$ diagonal elements only. We reorganize $S_m, c_1 c_2 \times c_1 c_2$, into the matrix $S, c_1^2 \times c_2^2$ (see section 5 of Currie et al. (2006) for details of this reorganization). Then

$$\text{diag}\{X S_m X^T\}, n_1 n_2 \times 1, \equiv \rho[G(X_2), \rho(G(X_1), S)], n_1 \times n_2. \quad (11)$$

A formal proof of the equivalences in (9), (10) and (11) is given in section 4 of Currie et al. (2006). Each GLAM algorithm requires element reorganization, notably from the output in array form of the inner product algorithm to the matrix form required by the scoring algorithm, and from the matrix form of $S_m = \text{Var}(\hat{\theta})$ to the array form of $S$. Computationally, reorganization is very efficient and R-code for all three GLAM algorithms is given in section 5 of Currie et al. (2006).

We conclude with four remarks on the algorithms:

1. A nice feature of the RHSs of (9) and (11) is that the output matrices/arrays of fitted values and variances of fitted values match those of the original data matrices/arrays.
2. Each of these formulae processes each dimension in turn, a process we call *marginal processing*; this automatically solves any storage problems.

3. GLAM offers two ways of computing the effective dimension, $ED$, of the fitted model. We have $ED = \text{tr}(H)$ where $H$ is the hat-matrix

$$H = X(X^T \hat{W}_d X + P)^{-1} X^T \hat{W}_d = Q \hat{W}_d,$$  \hspace{1cm} (12)

say. Now $\text{tr}(AB) = \text{tr}(BA)$ for conformable matrices so the second GLAM algorithm gives $ED$ immediately. Rather more elegantly, let $R$ be the array form of the diagonal elements of $Q$ given by the third GLAM algorithm, i.e. the RHS of (11), and let $W$ be the array form of the weights. Then, $ED$ is the sum of the elements of $R \ast W$.

4. The computations of $X \hat{\theta}$, $X^T W \hat{\theta} X$ and $\text{diag} \left( X \text{Var}(\hat{\theta}) X^T \right)$ are apparently very different. Despite this, the GLAM algorithms all have the *same functional form* and depend on the *same two functions*, $\rho(\cdot)$ and $G(\cdot)$. This strikes me as very remarkable; at any rate, the algorithms are highly efficient computationally and can lead to improvements in computer time of orders of magnitude, and may even bring some problems within reach. Tables 2 and 3 in the RSS paper provide details of the efficiency gains possible with GLAM.

### 4 Applications

#### 4.1 Modelling and forecasting mortality

The modelling of 2-dimensional mortality data with GLAM is described in Eilers et al. (2006) and Currie et al. (2006). Forecasting, considered in Currie et al. (2004) in pre-GLAM days, is easily incorporated into the GLAM system. The R-package MortalitySmooth (Camarda, 2012) for modelling and forecasting both 1- and 2-dimensional data makes full use of GLAM.

#### 4.2 An Example in Three Dimensions

The data comprise the number of deaths from a respiratory disease classified by age: 1, ..., 105, year of death: 1959, ..., 1998, and month: 1, ..., 12. This gives a 105 × 40 × 12 data array with 50400 entries. The data are smoothed across each dimension with $P$-splines and a $15 \times 10 \times 7$ coefficient array. This is a large problem: the regression matrix alone has over $5 \times 10^7$ elements. This example is described in Currie et al. (2006); the treatment of missing values, the choice of initial estimates and various devices for improving computational time still further are also discussed.
4.3 Scattered Data and Density Estimation

Surprisingly, some examples with scattered data can be fitted into the GLAM scheme. Eilers et al. (2006) described such an example in two dimensions. The method uses a nice trick: we enclose the data in a rectangle which is then divided into a large number of bins. For each bin we record (1) $W$, the number of observations and (2) $S$, the average value of these observations. The error introduced by this discretization is negligible and GLAM can be used. This approach was used in three papers on density estimation at the IWSM in Galway. The method of smoothing parameter selection was: AIC in Eilers and Marx (2006), mixed model in Durbán et al. (2006) and a Bayesian method in Lambert and Eilers (2006). All three papers exploited GLAM.

4.4 Spatio-temporal Smoothing

Dae-Jin Lee and María presented two papers on spatio-temporal smoothing, the first in 2008 in Utrecht and then in 2009 in Ithaca. These papers gave a practical solution to the problem of fitting smooth space-time interaction models. The spatial data were scattered over two dimensions while the temporal data were regularly spaced. We have a 2-dimensional GLAM where space is one GLAM dimension and time is the other. In Utrecht they used a mixed model representation which gave a decomposition of the model terms similar to ANOVA which they christened P-spline ANOVA or PS-ANOVA. The following year they introduced the idea of nested bases; here the time part of the space-time interaction uses a basis that is nested in the basis for the main effect of time. The use of GLAM and the nested bases gave substantial improvements in computer time while PS-ANOVA gave a clear interpretation of model terms. See Lee and Durbán (2008, 2009, 2011) and Lee et al. (2013).

4.5 Lee-Carter Model

The Lee-Carter model (Lee and Carter, 1992) is widely used in the modelling and forecasting of mortality (our motivating example). With the notation of section 1 we have $\log \mu_{x,y} = \alpha_x + \beta_x \kappa_y$, $x = x_1, \ldots, x_n$, $y = y_1, \ldots, y_n$. One problem with this model is that irregularities in the estimated $\beta$ terms and to a lesser extent the $\alpha$ terms lead to forecasts with unsatisfactory properties. This problem was initially addressed in Delwarde et al. (2007) and then more generally in Currie (2013). This second paper estimated smoothed and constrained regression coefficients in a GLM setting and made full use of GLAM. Unusually explicit formulae for GLAM’s expression for the inner products were given.
4.6 Additive GLAMs

GLAM is just a regression model so there is no reason why we should not have a model with more than one component. Our 2006 RSS paper presented an additive model with two GLAM components for the analysis of micro-array data. We could also have both GLAM components and regression components, a semi-GLAM. An example of this kind of model occurs in the analysis of variety trials. Fields are generally rectangular so fertility and any row-column effects can be modelled in a GLAM setting while variety effects are modelled in the usual way. In an important paper, Rodríguez-Álvarez et al. (2018) exploited earlier work to the full and used the mixed model decomposition of PS-ANOVA for the analysis of such trials. The work is supported by the R-package SpATS.

4.7 Components of Mortality

We return to our motivating example, the modelling of mortality data. We consider first 1-dimensional data and the smooth modelling of mortality by age for a given year. Let $B_a$, $n_a \times c_a$, be the regression matrix of $B$-splines with penalty matrix $D_a^T D_a$, $c_a \times c_a$, as before. In recent years, smoothing parameter selection has increasingly been through a mixed model representation. One method of achieving this is with the singular value decomposition (SVD). Let $U_a \Lambda_a U_a^T$ be the SVD of $D_a^T D_a$. With second order differences there are two zero eigenvalues so

$$\Lambda_a = \text{blockdiag} \{ 0_2, \Lambda_{a,s} \}$$

where $\Lambda_{a,s}$ contains the $c_a - 2$ non-zero eigenvalues. Let $U_a = [U_{a,n} : U_{a,s}]$ be the partition of $U_a$ into the eigenvectors corresponding to the zero and non-zero eigenvectors respectively. For simplicity, we will assume a normal approximation and take $y_{i,j} = \log(d_{i,j}/e_{i,j})$ to be independent $N(0, \sigma^2)$ variables. Let $y = (y_{1,j}, \ldots, y_{n_a,j})^T$ for some year $j$. Then the model $y = B_a \theta + \epsilon$, $\epsilon \sim N(0, \sigma^2 I)$ with penalty matrix $\lambda D_a^T D_a$ can be transformed into a standard form mixed model $y = X_a \beta + Z_a \alpha + \epsilon$ where

$$X_a = [1_a : x_a], Z_a = B_a U_{a,s} \Lambda_{a,s}^{-1/2}, \alpha \sim N(0, \tau^2 I_{c_a-2}).$$

We turn now to the 2-dimensional case and apply exactly the same method as in one dimension. We seek the SVD, $U \Lambda U^T$, of the 2-dimensional penalty $P(\lambda_a, \lambda_y)$ in (2). This can be expressed in terms of the eigenvectors and eigenvalues of the 1-dimensional penalties $\lambda_a D_a^T D_a$ and $\lambda_y D_y^T D_y$. We define $U$ as

$$[U_{y,n} \otimes U_{a,n} : U_{y,n} \otimes U_{a,s} : U_{y,s} \otimes U_{a,n} : U_{y,s} \otimes U_{a,s}]$$

and $\Lambda$ as the diagonal matrix

$$\text{blockdiag} \{ 0_4, I_2 \otimes \lambda_a \Lambda_{a,s}, \lambda_y \Lambda_{y,s} \otimes I_2, I_{c_y-2} \otimes \lambda_a \Lambda_{a,s} + \lambda_y \Lambda_{y,s} \otimes I_{c_a-2} \}.$$
It is straightforward to check that $U \Lambda U^T$ is indeed a singular value decomposition of the 2-dimensional penalty $P(\lambda_a, \lambda_y)$ in (2).

Lee et al. (2013) now come up with a clever, yet simple, idea. They make the following replacements in $\Lambda$

$$I_2 \otimes \lambda_a \Lambda_{a,s} \rightarrow \text{blockdiag}\{\lambda_1 \Lambda_{a,s}, \lambda_2 \Lambda_{a,s}\}$$

$$\lambda_y \Lambda_{y,s} \otimes I_2 \rightarrow \text{blockdiag}\{\lambda_3 \Lambda_{y,s}, \lambda_4 \Lambda_{y,s}\}$$

$$I_{c_y-2} \otimes \lambda_y \Lambda_{a,s} + \lambda_y \Lambda_{y,s} \otimes I_{c_a-2} \rightarrow \lambda_5 (I_{c_y-2} \otimes \Lambda_{a,s} + \Lambda_{y,s} \otimes I_{c_a-2})$$

and there are five smoothing parameters instead of two. We now apply the SVD transformation as in the 1-dimensional case with $U$ defined in (15) but with $\Lambda$ defined in (17). We obtain a standard mixed model $y = X\beta + Z\alpha + \epsilon$

with

$$X = [1_y \otimes 1_a : 1_y \otimes x_a : x_y \otimes 1_a : x_y \otimes x_a]$$

$$Z = [Z_1 : Z_2 : Z_3 : Z_4 : Z_5]$$

$$= [1_y \otimes Z_a : Z_y \otimes 1_a : x_y \otimes Z_a : Z_y \otimes x_a : Z_y \otimes Z_a].$$

Let $\alpha_i$ be the random effect corresponding to $Z_i$, $i = 1, \ldots, 5$. We take $\alpha_i \sim N(0, \tau^2_i I)$ and the model may be fitted with standard software such as the lme function in the R package nlme.

The original 2-dimensional model defined in (1) and (2) fits a general surface. The present model has the advantage that it decomposes the fit into six components and each component has an interpretation and associated degrees of freedom; this is the PS-ANOVA of Lee et al. (2013). The results for female mortality for ages 50 to 90 and years 1975 to 2015 for data from the UK’s Office for National Statistics are given in Table 1.

<table>
<thead>
<tr>
<th>Term</th>
<th>Interpretation</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>Bilinear surface in age and year</td>
<td>4</td>
</tr>
<tr>
<td>$Z_1$</td>
<td>Smooth in age</td>
<td>5.4</td>
</tr>
<tr>
<td>$Z_2$</td>
<td>Smooth in year</td>
<td>7.8</td>
</tr>
<tr>
<td>$Z_3$</td>
<td>Interaction: smooth in age, linear in year</td>
<td>4.1</td>
</tr>
<tr>
<td>$Z_4$</td>
<td>Interaction: linear in age, smooth in year</td>
<td>5.2</td>
</tr>
<tr>
<td>$Z_5$</td>
<td>Interaction: smooth in age, smooth in year</td>
<td>27.4</td>
</tr>
</tbody>
</table>

$[X : Z]$ Total 54.0

The six components of mortality just described lend themselves naturally to a graphical display. Here the contribution of each component to the final fitted mortality is given a graphical interpretation which corresponds to the decomposition of the degrees of freedom of PS-ANOVA. Some examples of this graphical decomposition can be seen at
From the GLAM perspective, the mixed model defined in (18) and (19) is an additive GLAM with six GLAM style components. Problems with storage or computational time for this model do not arise with current computers and \texttt{lme} completes in effect instantly. For a larger problem the GLAM algorithms might help. The normal model can be replaced with the original Poisson assumption in which case GLAM might well be useful; Lee et al. (2013) is an essential reference here. The PS-ANOVA model can be extended to three dimensions by following the template laid out above; in this case GLAM is likely to be needed.

5 GLAM: The IWSM Story

María went to the IWSM in New Orleans in 1998. On her return she had no difficulty in convincing me that I should go to this conference. The arguments were simple and persuasive: a beautiful and interesting location, a small and friendly group of delegates, and an atmosphere that encouraged scientific discussion. My first IWSM was in Graz the following year. Of course, Paul was already a seasoned IWSM man.

The 2000 conference in Bilbao was key. I wanted to meet the man who, along with Brian Marx, had invented \textit{P}-splines. There was something about the method that seemed right to me: it was just regression with a twist. I met Paul at the dinner. I remember it well: I was interested in smoothing and forecasting 2-dimensional mortality data; Paul was interested in extending \textit{P}-splines to two dimensions. Our first paper on smoothing such data was presented at the IWSM in Crete (Durbán et al., 2002); we added forecasting at the Leuven conference the next year (Currie et al., 2003). These two papers were combined in a \textit{Statistical Modelling} paper (Currie et al., 2004).

We had already started work on array methods. Paul made the initial breakthrough with algebraic expressions for the linear and inner product functions in two dimensions (Eilers et al. 2006). The problem was how to extend this work to three and higher dimensions. It was only when the work was cast in array form that the full beauty of the GLAM method was revealed. It was a time of frenzied effort and sleepless nights but by the time of the IWSM in Florence in 2004 we had broken the back of the problem. María produced the mixed model formulation and the complete work appeared in our RSS B paper of 2006. Since then the focus has been on applications and increasingly the mixed model representation is used where the detailed insight of analyses is provided by PS-ANOVA.

The IWSM continues to play a key role in the development of GLAM in particular and \textit{P}-splines more generally. The annual meetings provide the ideal forum for presenting work in progress, discussing future projects, receiving feedback and even recruiting new followers to the cause. GLAM
Currie 11

and P-splines may not be a religion but they certainly have their fanatics and for this we have the IWSM to thank!

Acknowledgments: It is a great pleasure to acknowledge the work of Paul Eilers and María Durbán. Our work together on GLAM was the most exciting time of my career. Thank you both!

References


