USGS
science for a changing world

Computing Ordinary Least-Squares Parameter Estimates for the National Descriptive Model of Mercury in Fish

Chapter 10 of
Section C, Computer Programs
Book 7, Annotated Data Processing and Computations

Techniques and Methods 7–C10

U.S. Department of the Interior
U.S. Geological Survey

Computing Ordinary Least-Squares Parameter Estimates for the National Descriptive Model of Mercury in Fish

By David I. Donato

Chapter 10 of
Section C, Computer Programs
Book 7, Annotated Data Processing and Computations

Techniques and Methods 7–C10

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
KEN SALAZAR, Secretary

U.S. Geological Survey
Marcia K. McNutt, Director

U.S. Geological Survey, Reston, Virginia: 2013

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment, visit http://www.usgs.gov or call 1–888–ASK–USGS.

For an overview of USGS information products, including maps, imagery, and publications, visit http://www.usgs.gov/pubprod

To order this and other USGS information products, visit http://store.usgs.gov

Suggested citation:
Donato, D.I., 2013, Computing ordinary least-squares parameter estimates for the National Descriptive Model of Mercury in Fish: U.S. Geological Survey Techniques and Methods, book 7, chap. C10, 9 p., http://pubs.usgs.gov/tm/07/c10.

Contents

This page intentionally left blank.

Computing Ordinary Least-Squares Parameter Estimates for the National Descriptive Model of Mercury in Fish

By David I. Donato

Abstract

A specialized technique is used to compute weighted ordinary least-squares (OLS) estimates of the parameters of the National Descriptive Model of Mercury in Fish (NDMMF) in less time using less computer memory than general methods. The characteristics of the NDMMF allow the two products $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{X}'\boldsymbol{y}$ in the normal equations to be filled out in a second or two of computer time during a single pass through the N data observations. As a result, the matrix $\boldsymbol{X}$ does not have to be stored in computer memory and the computationally expensive matrix multiplications generally required to produce $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{X}'\boldsymbol{y}$ do not have to be carried out. The normal equations may then be solved to determine the best-fit parameters in the OLS sense. The computational solution based on this specialized technique requires $O(8p^2+16p)$ bytes of computer memory for p parameters on a machine with 8-byte double-precision numbers. This publication includes a reference implementation of this technique and a Gaussian-elimination solver in preliminary custom software.

Introduction

The National Descriptive Model of Mercury in Fish (NDMMF) is a statistical model used to predict the concentration of methylmercury in fish tissue (Wente, 2004). This model is of interest in current research at the U.S. Geological Survey (USGS) because of its ability to explain much of the variation in fish-tissue methylmercury concentrations as variation by geographic location, variation over time, and variation by fish species and length.

Before the NDMMF can be used to predict fish-tissue methylmercury concentrations, its parameters must be fitted to a collection of observations of fish-tissue methylmercury concentrations. The statistical procedure used to fit the parameters of the NDMMF to observed data is that of **maximum-likelihood estimation** (MLE). The parameters of the NDMMF are fitted in the maximum-likelihood sense because the available national database of fish-tissue methylmercury concentrations includes a substantial number of **left-censored** observations[1], and the MLE method makes better use of the information contained in censored observations than other common statistical methods of parameter estimation, including the method of **ordinary least-squares estimation** (Helsel, 2004). Ordinary least-squares (OLS) parameter estimation is, however, a preliminary step of choice in computing maximum-likelihood estimates and, therefore, an essential part of the complete procedure for estimating the parameters of the NDMMF.

This publication presents a technique, along with a reference (baseline) implementation in custom computer software, for computing OLS parameter estimates for the NDMMF based on weighted observations. By exploiting the specific characteristics of the NDMMF, this technique enables faster computation of OLS parameter estimates for the NDMMF using less computer random-access memory (RAM) than is possible with generalized statistical computer software. Faster computation and use of less RAM are major improvements in fitting parameters to the NDMMF because some generalized software procedures may fail to run because they require more RAM than is available, and those that do run will require many hours or days of computation. The reference software computes the best-fit parameters for the NDMMF in the ordinary least-squares sense using Gaussian elimination with back substitution (Noble, 1969; Draper and Smith, 1966); a revised version of the software using an alternative computational method, such as LU or Cholesky decomposition, is planned (Press and others, 1992).

The preliminary custom software included with this publication is not intended for general use; rather, it is intended for research and development use in conjunction with other custom software developed at the U.S. Geological Survey (USGS) for computing maximum-likelihood estimates of the parameters of the NDMMF more quickly, and using less RAM, than is possible with generalized statistical software (Donato, 2012). A full understanding of this report requires knowledge of linear algebra and familiarity with statistical models.

[1]A left-censored observation is a value determined to be below a particular detection limit but otherwise unspecified. For example, if a laboratory procedure cannot detect methylmercury concentration values below 0.020 part per million, then a sample with an undetected concentration would be recorded as a nondetected and, thus, as a left-censored value with a detection limit of 0.020 part per million (Helsel, 2004).

The National Descriptive Model of Mercury in Fish as a Formal Statistical Model

The NDMMF is expressed formally and compactly as:

$$y_{ijk} = \alpha_j L_{ijk} + \beta_k + \varepsilon_{ijk} \quad (1)$$

where

y_{ijk} = $\ln\left(C_{ijk}+1\right)$ and C_{ijk} is the ith observed concentration value for species/cut j and sampling event k;

α_j = the parameter for the jth species/cut;

L_{ijk} = $\ln\left(length_{ijk}+1\right)$ and $length_{ijk}$ is the ith observed fish length in inches for species/cut j and sampling event k;

β_k = the parameter for the kth sampling event; and

ε_{ijk} = the random error for the ith observation for species/cut j and sampling event k and $\varepsilon_{ijk} \sim N(0,\sigma)$.

In the seminal paper on the NDMMF (Wente, 2004), the terms **species/cut** and **sampling event** are defined to have specialized meaning as applied to the NDMMF. Briefly, the species/cut is a combination of a species of fish and a method of preparing the tissue of the fish for laboratory analysis; a sampling event is a collection of samples from a particular geographic location within the same year. Wente's 2004 publication provides a full explanation of these terms.

Equation (1) implies that the number of parameters of the NDMMF equals the sum of the number of species/cut combinations and the number of sampling events. This is more readily apparent when the NDMMF is expressed less compactly as:

$$y_{ijk} = \alpha_1 SPC_1 L_{ijk} + \alpha_2 SPC_2 L_{ijk} + \cdots + \alpha_n SPC_n L_{ijk} + \beta_1 Event_1 + \beta_2 Event_2 + \cdots + \beta_m Event_m + \varepsilon_{ijk} \quad (2)$$

where

y_{ijk}, L_{ijk}, and ε_{ijk} are defined as for equation (1);

n = the number of species/cut combinations and parameters;

m = the number of sampling events and parameters;

$$\alpha_t SPC_t = \begin{cases} 0 & \text{if } t \neq j \\ \alpha_j & \text{if } t = j \end{cases}, \text{ and}$$

$$\beta_t Event_t = \begin{cases} 0 & \text{if } t \neq k \\ \beta_k & \text{if } t = k \end{cases}.$$

In equation (2), the sets of variables $\{SPC_t\}_{t=1}^{n}$ and $\{Event_t\}_{t=1}^{m}$ are **indicator variables** (Box, 1978). These indicator variables allow the NDMMF to be expressed in the form of the general linear model, a model linear with respect to its parameters (Mendenhall, 1968).

The Normal Equations for Ordinary Least-Squares Estimation

An advantage of expressing the NDMMF in the form of the general linear model is that an expression for its best-fit parameters in the ordinary least-squares sense is known in a general form. The system of equations that determines the ordinary least-squares (OLS) parameters for a general linear model is called the **normal equations** (Monahan, 2001; Press and others, 1992). In matrix form, the general linear model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (3)$$

and the normal equations for this model are:

$$\left(\boldsymbol{X}'\boldsymbol{X}\right)\boldsymbol{b} = \boldsymbol{X}'\boldsymbol{y} \quad (4)$$

where specifically for the NDMMF

$p = n + m$ is the total number of parameters (the number of species/cut parameters plus the number of sampling-event parameters);

N = the total number of observations (and generally $N \gg p$);

$\boldsymbol{y}$ = the $(N \times 1)$ vector of response observations;

$\boldsymbol{X}$ = an $(N \times p)$ matrix of values of observed data;

$\boldsymbol{X}'$ = the $(p \times N)$ matrix transpose of $\boldsymbol{X}$; and

$\boldsymbol{\beta}$ = a $(p \times 1)$ vector of parameters and $\boldsymbol{b}$ is the corresponding $(p \times 1)$ vector of least-squares estimates of the parameters.

The symbol for the vector $\boldsymbol{\beta}$ in the normal equations should not be confused with the symbol for the sampling-event parameters $\{\beta_k\}$ in the NDMMF. The vector $\boldsymbol{\beta}$ is a vector of all model parameters, so for the NDMMF, $\boldsymbol{\beta}$ includes all n of the $\{\alpha_j\}$ and all m of the $\{\beta_k\}$ parameters. The matrix $\boldsymbol{X}$ of data observations is sparse, containing only two nonzero values per row.

Matrix Form of Data for the National Descriptive Model of Mercury in Fish

Equation (4) includes all of the matrices involved in the computation of OLS parameter estimates: ***y***, ***b***, ***X***, and ***X′***. Since the internal structure of these matrices depends on features of parameters and data specific to the NDMMF, each of these matrices must be specialized to the NDMMF before the details of computation of OLS parameter estimates for the NDMMF can be finalized.

$$\boldsymbol{y} = \begin{bmatrix} y_{1,1,1} \\ y_{2,1,1} \\ \vdots \\ y_{1,52,104} \\ y_{2,52,104} \\ \vdots \\ y_{15,n,m} \end{bmatrix} \tag{5}$$

The response vector ***y***, which contains one row for each of the *N* observations, may and usually does contain multiple observations (rows) for each combination of species/cut and sampling event. Therefore, *N* is typically much larger than the total number of parameters, $p = n + m$. Each row in ***y*** corresponds to a row in ***X***.

$$\boldsymbol{b} = \begin{bmatrix} \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_n \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_m \end{bmatrix} \tag{6}$$

The vector of least-squares parameter estimates ***b*** must be put into a sequence according to some convention in order to establish a correspondence between each parameter's estimate in ***b*** and its relevant data observations in the matrices ***X*** and ***X′***. Let the convention be that the rows of ***b*** and ***X′*** and the columns of ***X*** must correspond to the *n* species/cut parameters $\{\alpha_j\}$ in sequence by species/cut parameter number, followed by the *m* sampling-event parameters $\{\beta_k\}$ in sequence by sampling-event parameter number.

$$\boldsymbol{X} = \left[\begin{array}{ccccc|ccccc} 0 & \cdots & \cdots & \cdots & 0 & 0\cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & SPC_j L_{ijk} & \cdots & 0 & 0\cdots & \cdots & Event_k & \cdots & 0 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & 0\cdots & \cdots & \cdots & \cdots & 0 \end{array}\right] \tag{7}$$

Equation (7) illustrates the general appearance of the sparse matrix ***X*** symbolically and equation (8) shows a numerical example of ***X*** in outline.

$$\boldsymbol{X} = \left[\begin{array}{ccccccc|ccccccc} 0 & \cdots & 1.4 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1.2 & 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & \cdots & 2.7 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \\ \alpha_1 & & \rightarrow & & \alpha_n & & & \beta_1 & & \rightarrow & & & & \beta_m \end{array}\right] \begin{array}{l} 1 \\ \\ \downarrow \\ \\ N \\ \\ \end{array} \tag{8}$$

The matrix ***X*** of observations of the independent variables contains one row for each of the *N* observations. Each row of ***X*** is an ordered list of the coefficients of the parameters $\{\alpha_t\}_{t=1}^{n}$ and then $\{\beta_t\}_{t=1}^{m}$ for one equation in the system of *N* equations expressed equivalently, but in different forms, by equations (2) and (3). The correspondence between rows and the *N* observations, and between columns and parameters, is illustrated in equation (8); and the transposed correspondences are illustrated in equation (9).

Each row of ***X*** provides an observation for exactly one species/cut *j* and exactly one sampling event *k*, **so each row contains exactly two nonzero values** determined by the set of observations: **first**, the value of L_{ijk} for the i^{th} observation for species/cut *j* and sampling event *k*, **then** a value of 1 for the single indicator variable for sampling event *k*. All other values in each row are zero (0). Thus in each row, all of the indicator variables $\{SPC_t\}_{t=1}^{n}$ and $\{Event_t\}_{t=1}^{m}$ are equal to zero, except for the one SPC_j indicator for the species/cut *j* associated with the observation for the row, and except for the one $Event_k$ indicator for the sampling event *k* associated with the observation for the row.

$$\boldsymbol{X}' = \begin{array}{c} \begin{matrix} 1 & \rightarrow & N \end{matrix} \\ \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 1.4 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 2.7 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \\ \hline 1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \end{array} \begin{array}{l} \alpha_1 \\ \\ \\ \downarrow \\ \\ \\ \alpha_n \\ \hline \beta_1 \\ \\ \\ \\ \\ \\ \beta_m \end{array} \qquad (9)$$

The matrix transpose $\boldsymbol{X}'$ contains one column for each observation, as shown above in equation (9), with the same illustrative values shown for $\boldsymbol{X}$ in equation (8). Here, each **column** contains all zero values except for the two values determined by the observation it represents. (Although the mathematical form of the NDMMF permits L_{ijk} to equal zero, in practice L_{ijk} is always strictly greater than zero.) The first n rows of each column in the transpose correspond to the species/cut parameters $\{\alpha_t\}_{t=1}^{n}$ in sequence, and the following m rows correspond to the sampling-event parameters $\{\beta_t\}_{t=1}^{m}$ in sequence.

Computing the Normal-Equation Matrix Products During Data Input

The normal equations in matrix form, $(\boldsymbol{X}'\boldsymbol{X})\boldsymbol{b} = \boldsymbol{X}'\boldsymbol{y}$, contain two matrix products: $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{X}'\boldsymbol{y}$. Because of the sparseness and particular configuration of the matrix $\boldsymbol{X}$ and its transpose $\boldsymbol{X}'$ for the NDMMF, the two matrix products can be created directly by software during the input of the N data records, thus making it unnecessary to store $\boldsymbol{X}$ or $\boldsymbol{X}'$ in computer memory and unnecessary to perform any computationally expensive matrix multiplications. **This is the essence of the technique described in this publication: that a computation that might take hours or days with generalized computational methods is avoided by using an alternative, fast input process that requires only a few seconds of computer processing time**.

The benefits of the alternative computation of $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{X}'\boldsymbol{y}$ during data input are substantial. Generalized computational methods must store the $(N \times p)$ matrix $\boldsymbol{X}$ in computer memory, and because $N \gg p$ this requires an amount of additional memory that is several times the amount of memory required for storing just the $(p \times p)$ product matrix $\boldsymbol{X}'\boldsymbol{X}$. Although some generalized software performs the multiplication of the sparse matrices $\boldsymbol{X}'$ and $\boldsymbol{X}$ with fewer computations or with less expensive computational operations than a full multiplication, the multiplication of $\boldsymbol{X}'\boldsymbol{X}$ still requires a substantial proportion of the $O(Np^2)$ floating-point multiplications and additions required in the general case. Thus generalized computational methods may require more time just to set up the normal equations in computer memory (by computing $\boldsymbol{X}'\boldsymbol{X}$) than is required to solve them.

To understand how the product matrix $\boldsymbol{X}'\boldsymbol{X}$ is created during data input, begin by applying the definition of matrix multiplication. Each element of the $(p \times p)$ product matrix $\boldsymbol{X}'\boldsymbol{X}$ is the sum of the N products of each element of a row multiplied by the corresponding element of a column of $\boldsymbol{X}$. Now, rather than considering what would be involved in multiplying these two matrices in full, consider instead how the values for a single data observation affect the final values of the elements of the product matrix $\boldsymbol{X}'\boldsymbol{X}$. Because of the characteristics of the NDMMF, a particular data observation (y_{ijk}, L_{ijk}) for species/cut j and sampling-event k will affect the values in only two of the rows of $\boldsymbol{X}'$: the row for the species/cut j and the row for the sampling-event k. Thus, during matrix multiplication, only these two rows from $\boldsymbol{X}'$ will have any role in the particular data observation's effect on the final values in the product matrix.

Now consider what happens as each of these two rows from $\boldsymbol{X}'$ is multiplied by columns from $\boldsymbol{X}$. There are four cases to consider:

1. **Species/cut row from $\boldsymbol{X}'$ and species/cut column from $\boldsymbol{X}$:** The row-column product will be zero for every species/cut column in $\boldsymbol{X}$, except the column for the same species/cut j. Note that the row from $\boldsymbol{X}'$ for species/cut j is the transpose of the column from $\boldsymbol{X}$ for species/cut j. Therefore, the element of the product matrix $\boldsymbol{X}'\boldsymbol{X}$ at row j and column j will be affected by the addition of $(L_{ijk})^2$.

2. **Species/cut row from $\boldsymbol{X}'$ and sampling-event column from $\boldsymbol{X}$:** The row-column product will be zero for every sampling-event column, except the column for sampling-event k. Therefore, the element of the product matrix $\boldsymbol{X}'\boldsymbol{X}$ at row j and column $n + k$ will be affected by the addition of $L_{ijk} \times 1$. (The column in $\boldsymbol{X}$ for sampling-event k is column $n + k$ because the n columns for species/cut parameters come first, so counting for sampling-event columns begins at $n + 1$.)

3. **Sampling-event row from $\boldsymbol{X}'$ and species/cut column from $\boldsymbol{X}$:** The row-column product will be zero for every species/cut column, except the column for species/cut j. Therefore, the element of the product matrix $\boldsymbol{X}'\boldsymbol{X}$ at row $n + k$ and column j will be affected by the addition of $1 \times L_{ijk}$.

4. **Sampling-event row from $\boldsymbol{X}'$ and sampling-event column from $\boldsymbol{X}$:** The row-column product will be zero for every sampling-event column, except the column for sampling-event *k*. Therefore, the element of the product matrix $\boldsymbol{X}'\boldsymbol{X}$ at row $n + k$ and column $n + k$ will be affected by the addition of $1 \times 1 = 1$.

Thus, as a result of the fact that each row of $\boldsymbol{X}$ contains exactly two nonzero values, any particular data observation (y_{ijk}, L_{ijk}) only influences four elements of the product matrix $\boldsymbol{X}'\boldsymbol{X}$. Let $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{P}$ and let $p_{a,b}$ denote the element of $\boldsymbol{P}$ in the *a*th row and *b*th column. Then any particular data observation (y_{ijk}, L_{ijk}) affects only four values in the product matrix $\boldsymbol{P}$ as follows:

1. $p_{j,j}$: increased by the addition of $(L_{ijk})^2$
2. $p_{j,n+k}$: increased by the addition of L_{ijk}
3. $p_{n+k,j}$: increased by the addition of L_{ijk}
4. $p_{n+k,\,n+k}$: increased by the addition of 1

This result allows the $(p \times p)$ product matrix $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{P}$ to be computed by adding in values as each data observation is read in from an input file. When data input is complete, so is the product matrix $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{P}$. Since $p_{j,n+k}$ and $p_{n+k,j}$ always receive the same additive contributions, $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{P}$ is a symmetric matrix.

The other product matrix, $\boldsymbol{X}'\boldsymbol{y}$, can also be computed during data input. Let $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$ and let q_a denote the element of $\boldsymbol{q}$ in the *a*th row. Note that the product matrix $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$ is a $(p \times 1)$ column matrix (also called a **column vector**) and that there are exactly two rows in $\boldsymbol{X}'$ with values for any particular data observation:

1. **Species/cut row from $\boldsymbol{X}'$:** The row-column product will be zero for every species/cut row in $\boldsymbol{X}'\boldsymbol{y}$ except the row for species/cut *j*. The element of the product matrix $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$ at row *j* will be affected by the addition of $L_{ijk} \times y_{ijk}$.
2. **Sampling-event row from $\boldsymbol{X}'$:** The row-column product will be zero for every sampling-event row in $\boldsymbol{X}'$ except the row for sampling-event *k*. The element of the product matrix $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$ at row $n + k$ will be affected by the addition of $1 \times y_{ijk}$.

Consequently, $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$ can be computed during input by observing that any particular data observation affects only two rows of $\boldsymbol{q}$ as follows:

1. q_j : increased by the addition of $L_{ijk} \times y_{ijk}$
2. q_{n+k} : increased by the addition of y_{ijk}

Computing With Weighted Observations

The data used for fitting parameters to the NDMMF (the "NDMMF data") include weights. The essential idea of a weight for a data observation is that an observation with a weight of 2.0 should have the same effect on the fitting of parameters as it would have if it were replaced by two identical observations, each with a weight of 1.0. The majority of observations in the NDMMF data are for the processing of tissue from a single fish; each such observation has a weight of 1.0.

Although the weights for the NDMMF data are sometimes adjusted nonlinearly into nonintegral values such as 1.65 or 2.8, all weights remain greater than or equal to 1.0 (and as previously mentioned, a majority of them equal 1.0 exactly). For the purpose of making use of the weights associated with NDMMF data observations when computing OLS parameter estimates, the essential idea of weights as simple integral multiples will be generalized to allow for nonintegral weights. In other words, an observation weighted by an integer should have the same effect that the corresponding number of identical observations would have, and nonintegral weights should have effects intermediate between the effects of the bracketing integral weights. For example, an observation with a weight of 2.4 should have an effect between the effect of a weight of 2 and a weight of 3. The previous section of this publication showed that any particular data observation (ignoring its weight for the moment) will contribute additively to exactly four elements of $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{P}$ and to exactly two elements of $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$. In principle, because two identical observations would make the same additive contributions twice, and three identical observations would make the same additive contributions three times, the additive contributions for a weighted observation are equal to each of the contributions shown in the preceding section multiplied by the weight. If w_{ijk} is a real number represented in computation as a floating-point number, then when the weight w_{ijk} is associated with an observation (y_{ijk}, L_{ijk}) the weighted effects of this observation on $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{P}$ and $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{q}$ are as follows:

1. $p_{j,j}$: increased by the addition of $(L_{ijk})^2 \times w_{ijk}$
2. $p_{j,n+k}$: increased by the addition of $L_{ijk} \times w_{ijk}$
3. $p_{n+k,j}$: increased by the addition of $L_{ijk} \times w_{ijk}$
4. $p_{n+k,\,n+k}$: increased by the addition of $1 \times w_{ijk}$
5. q_j : increased by the addition of $L_{ijk} \times y_{ijk} \times w_{ijk}$
6. q_{n+k} : increased by the addition of $y_{ijk} \times w_{ijk}$

Computational Methods for Solving the Normal Equations

There are several different, well-documented computational methods available for solving the normal equations for $\boldsymbol{b}$ once the $(p \times p)$ matrix $\boldsymbol{X}'\boldsymbol{X}$ and the $(p \times 1)$ column vector $\boldsymbol{X}'\boldsymbol{y}$ from the normal equations $(\boldsymbol{X}'\boldsymbol{X})\boldsymbol{b} = \boldsymbol{X}'\boldsymbol{y}$ are available in computer RAM. By analogy with standard algebra, the solution of the normal equations may be expressed as

$$\boldsymbol{b} = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{10}$$

in terms of the left multiplicative inverse of the $(p \times p)$ matrix $\boldsymbol{X}'\boldsymbol{X}$; and one way to compute the solution of the normal equations $\boldsymbol{b}$ is to invert the matrix $\boldsymbol{X}'\boldsymbol{X}$ and left-multiply this inverse with $\boldsymbol{X}'\boldsymbol{y}$. In general, however, it is not necessary in computations to invert the matrix $\boldsymbol{X}'\boldsymbol{X}$ in order to solve for $\boldsymbol{b}$, and when the matrix inverse is not required for additional computations, the inversion is often undesirable because it entails more computation than other methods and is, therefore, relatively slow (Monahan, 2001).

Well-known and widely used methods for solving the normal equations and other systems of linear equations computationally (Monahan, 2001; Press and others, 1992) include the following:

- Gaussian elimination with back substitution;
- Gauss-Jordan elimination with back substitution;
- Cholesky decomposition;
- LU decomposition with back substitution;
- Singular value decomposition; and
- QR decomposition.

Of these methods, no one stands out as the best for all problems though in general LU decomposition comes close to being the presumptive method of choice (Press and others, 1992). Each method has advantages and disadvantages that make it more suitable for some sets of normal equations and less suitable for others. Gauss-Jordan elimination is acceptably efficient when the matrix inverse is required for other operations, but it is three times slower than alternative methods for simply solving a system for a single solution vector $\boldsymbol{b}$. Although Gaussian elimination is faster than Gauss-Jordan elimination and comparable in speed to LU decomposition, it is seldom used in statistical packages because it is subject to round-off errors and other computational inaccuracies. Cholesky decomposition is the fastest of the commonly used methods (approximately twice as fast as Gaussian elimination or LU decomposition), but it can only be used when the matrix $\boldsymbol{X}'\boldsymbol{X}$ is symmetric and positive definite. Singular value decomposition is a "can't fail" method that effectively overcomes ill-conditioning and near-zero values in the matrix, but it is slow and, consequently, it is not suited for solving large systems of equations such as those encountered in computing OLS parameters for the NDMMF. Finally, QR decomposition requires about twice as much computation as LU decomposition and is not generally chosen, except in special cases (Press and others, 1992).

Detailed description of the derivation of the normal equations for OLS parameter estimation is beyond the scope of this publication, as are detailed descriptions of the various methods of solving systems of linear equations. These topics are covered thoroughly in a number of texts and other books (Mendenhall, 1968; Draper and Smith, 1966; Monahan, 2001; Press and others, 1992).

Reference Software

The reference custom software included with this publication is a single program module of structured and commented source code written in the C programming language. This code has been compiled using Version 4.4.4 of GCC and has been executed successfully on a computer workstation running under Version 2.6.33 of the Linux kernel. This reference software uses the method of Gaussian elimination with back substitution to solve the normal equations and find the best-fit parameter estimates for the NDMMF. For a computation involving about 15,400 parameters, the real run time (elapsed wall-clock time) on an otherwise lightly loaded workstation was about 16 hours.

A run time of 16 hours may seem long, especially to those who are accustomed to computing solutions for systems of linear equations with 100 or fewer parameters. The long run time is understandable, however, in view of the fact that the time required to solve a system of linear equations is approximately proportional to the cube of the number of parameters. More precisely, the computation required

for Gaussian elimination or LU decomposition is $O(\frac{5}{6}p^3)$ additions and multiplications where p is the number of parameters. Back substitution requires the comparatively modest, additional computation of $\frac{1}{2}p^2$ multiplications and additions (Press and others, 1992). Doubling p increases the computation required for Gaussian elimination or LU decomposition by a factor of approximately eight (8). Computing a solution for a model with 15,400 parameters will take about 3,650,000 times longer than computing a solution for a model with 100 parameters.

The reference software uses relatively little memory, except for what is required for the $(p \times p)$ matrix $\boldsymbol{X'X}$ and the two $(p \times 1)$ column vectors $\boldsymbol{b}$ and $\boldsymbol{X'y}$ where p is the number of parameters. Thus, the software requires $O(8p^2 \times 16p)$ bytes of computer memory for p parameters. For example, when the number of parameters is 15,400, then memory required is 1.767 gigabytes. The memory saved by not storing the $(N \times p)$ matrix of observations (with $N \cong 101,000$) is 11.589 gigabytes.

Also provided with the C source code are:

- a compilation script,
- a sample data input file, and
- samples of the program's three output files.

The software requires that there be no gaps in the species/cut and sampling-event parameter numbers. Each set of parameter numbers must begin with 1 and end with a number that does not exceed the value of its respective manifest constant—`NUMSPC` or `NUMEVENTS`. The values of these two manifest constants, along with the value of the manifest constant `MATRIXDIMENSION`, must be set correctly before compiling and running the program code. The value of `MATRIXDIMENSION` must equal or exceed the sum of the values of `NUMSPC` and `NUMEVENTS`.

This software does not perform checks on the validity of input data. The software will fail to produce valid results if inputs are not valid. Input concentrations are assumed to be in units of parts per million. Input lengths are assumed to be in inches.

This software is provided as a research tool, not as production code. Use of this code requires basic proficiency in reading, understanding, modifying, and compiling C program code. Setting up the input data file requires careful attention to its contents.

Summary

This publication describes a technique for efficient computation of ordinary least-squares (OLS) parameter estimates for the National Descriptive Model of Mercury in Fish (NDMMF). Included with this report is a preliminary reference implementation of the technique in software, using the method of Gaussian elimination. The technique enables rapid setup of the normal equations for the NDMMF so that the equations can then be solved for the OLS parameter estimates by any of several available methods.

The characteristic of the NDMMF that enables the rapid and direct setup of the normal equations is its restriction to exactly two parameters for each observation of methylmercury concentration. This characteristic allows the matrix products in the normal equations to be filled out quickly during data input so that the matrix $\boldsymbol{X}$ of data values for the independent variables does not need to be stored in computer memory and the full matrix multiplications implied by the normal equations do not actually have to be carried out.

It is not essential that all computation be performed using the provided reference software. The technique is also potentially usable with generalized statistical software. The reference software can be modified by a qualified programmer to omit the computation for Gaussian elimination and just write the two matrices created on data input ($\boldsymbol{X'X}$ and $\boldsymbol{b}$) to a file or files for subsequent input into any generalized statistical package or program capable of accepting the normal equations in matrix form as file input to a linear-system solver.

Computing the OLS parameter estimates for the NDMMF using the specialized technique described in this publication requires $O(8p^2 \times 16p)$ bytes of computer memory on a system with 8-byte double-precision numbers, where p is the number of parameters of the model. In recent usage, for illustration, the custom software, with $p \cong 15,400$, used about 1.767 gigabytes of computer memory and required about 16 hours for computation. The computer memory saved by not storing the $\boldsymbol{X}$ matrix in simplest form (with $N \cong 101,000$) was 11.589 gigabytes. The computational time saved by avoiding a full matrix multiplication to create $\boldsymbol{X'X}$ would vary by generalized statistical software but could range up to the time required for $O(Np^2)$ multiplications and additions; so the computational time saved could be a multiple of the $O(\frac{5}{6}p^3)$ multiplications and additions used to compute the ordinary least-squares parameter estimates by Gaussian elimination. With $p \cong 15,400$ and $N \cong 101,000$ the computational time saved could be up to ~7.87 times what is used for Gaussian elimination.

References Cited

Box, George E.P., and others, 1978, Statistics for experimenters; an introduction to design, data analysis, and model building: John Wiley & Sons, Inc., 653 p.

Donato, D.I., 2012, Computing maximum-likelihood estimates for parameters of the National Descriptive Model of Mercury in Fish: U.S. Geological Survey Open-File Report 2012–1181, 15 p. (Available only at http://pubs.usgs.gov/of/2012/1181.)

Draper, N.R. and Smith, Harry, 1966, Applied regression analysis: John Wiley & Sons, Inc., 407 p.

Helsel, D.R., 2004, Nondetects and data analysis; statistics for censored environmental data: Wiley-Interscience, 268 p.

Mendenhall, William, 1968, Introduction to linear models and the design and analysis of experiments: Duxbury Press, 465 p.

Monahan, J.F., 2001, Numerical methods of statistics: Cambridge University Press, 428 p.

Noble, Ben, 1969, Applied linear algebra: Prentice-Hall, Inc., 523 p.

Press, W.H., and others, 1992, Numerical recipes in C; the art of scientific computing (2d ed.): Cambridge University Press, 994 p.

Wente, S.P., 2004, A statistical model and national dataset for partitioning fish-tissue mercury concentration variation between spatiotemporal and sample characteristic effects: U.S. Geological Survey Scientific Investigations Report 2004–5199, 15 p. (Available only at http://pubs.usgs.gov/sir/2004/5199.)

Appendix

This appendix contains the C language source computer code (`NdmOls82.c`) for the reference software that implements and illustrates the technique described in this publication. Also included in this appendix are:

- a compilation script–`comp`,
- a sample data input file–`Hgdata.srt`, and
- samples of the three output files produced by the reference software–`EVENTparameters.txt`, `SPCparameters.txt`, and `log`.

Please take notice of the disclaimers included in the C language source-code file.

Appendix files are available for download at http://pubs.usgs.gov/tm/07/c10.

This page intentionally left blank.

Publishing support provided by:
U.S. Geological Survey Science Publishing Network,
Reston and Tacoma Publishing Service Centers

For more information concerning the research in this report, contact the
Eastern Geographic Science Center
U.S. Geological Survey
521 National Center
12201 Sunrise Valley Drive
Reston, VA 20192
http://egsc.usgs.gov

Donato, D.I.— **Computing Ordinary Least-Squares Parameter Estimates for the National Descriptive Model of Mercury in Fish**—Techniques and Methods 7–C10