CALIBRATION IN A NON-STATIONARY FIELD

David R. Fox
Department of Statistics
University of Wyoming

Abstract : In this paper we examine the problem of statistical calibration in a non-stationary field. A linear statistical model of the form $V = W\beta + \xi$ is assumed to exist at each of a number of locations. Furthermore, the vector of parameters, β at a particular location X is assumed to be parameterized as $\beta = AX$, where A is a matrix of deterministic scalars. Given V,W data we could estimate β at each location via OLS. The resulting collection of β's could then be used to similarly estimate A . An alternative to this 'two-stage regression' approach is to determine that A which minimizes an appropriate criterion such as $(V-\hat{V})^{\top}(V-\hat{V})$. The details of this latter method are given and it is shown to be equivalent to the two-stage regression approach in the case of a single location. The method is then extended to cater for the case of multiple sampling locations. In this case the covariance structure between locations is taken into consideration using Seemingly Unrelated Regression (SUR) procedures. The computations are illustrated with the use of an example and the programming language GAUSS.

Keywords : calibration,multivariate regression,simple regression,inverse regression, varying parameters model, seemingly unrelated regression.

1. Introduction

The results discussed herein represent a continuation (and a generalization) of work recently reported in a paper entitled "Calibration in a correlated field". We again consider the case of determining the matrix A in the system $\beta = AX$; $V = W\beta + \xi$, where V is an (nx1) vector of dependent variable values, W is a (nxp) matrix of independent variable values, β is a (px1) parameter vector for location X (νx1).

The extension to more than one location has been considered elsewhere and procedures were developed for the least-squares estimation of the matrix **A**. However, these previous methods implicitly assumed that observations between locations were independent. In many applications this will not be the case - particularly in a geostatistical context where some or all of the locations are within a zone of influence of each other. We now extend earlier results to allow for this between-location covariance effect.

2. Seemingly unrelated regression and generalized least-squares.

For the ith. location :

$$\underset{(nx1)}{\mathbf{V}_i} = \underset{(nxp)}{\mathbf{W}_i}\ \underset{(px1)}{\beta_i} + \underset{(nx1)}{\xi_i} \qquad (2.1)$$

where

$$\underset{(px1)}{\beta_i} = \underset{(px\nu)}{\mathbf{A}}\ \underset{(\nu x1)}{\mathbf{X}_i} \qquad (2.2)$$

A convenient way to write this system of equations is :

$$\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 & & & \\ & \mathbf{W}_2 & & \\ & & \ddots & \\ & & & \mathbf{W}_k \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_k \end{bmatrix} \qquad (2.3)$$

or

$$\mathbf{V} = \mathbf{Z}\,\mathbf{T} + \xi \qquad (2.4)$$

where **V** is (knx1) , **Z** is (knxkp) , and **T** is (kpx1).

Furthermore, we assume $\mathbb{E}[\xi_i] = 0$ and the covariance matrix of the joint disturbance vector is given by

$$\mathbb{E}[\xi\xi^{\top}] = \Sigma \otimes I_n = \psi, \text{ where} \tag{2.5}$$

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1k} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{bmatrix} \tag{2.6}$$

where σ_{ij} is the covariance between V in location i and V in location j.

If we regard the n observations in V_i as representing a different point in time, then the covariance assumption in equation (2.5) implies that the disturbances in different equations are correlated at a given point in time but are not correlated over time. In econometric theory this is known as a contemporaneous correlation.

2.1 Estimation.

When the system represented by equation (2.3) is viewed as the single equation (2.4), we can estimate Γ and hence all the β_i via generalized least squares (GLS). If Z is of rank kp and Σ is known and of rank k, the GLS estimator exists and is given by :

$$\hat{\Gamma} = (Z^{\top}\psi^{-1}Z)^{-1}Z^{\top}\psi^{-1}V \tag{2.1.1}$$

Within the class of all estimators that are unbiased and linear functions of V, this estimator is minimum variance and, if V is normally distributed, it is the maximum likelihood estimator and is minimum variance within the class of all unbiased estimators (Judge et.al.1980,p246).

Furthermore :

$$\mathbb{E}[\hat{\Upsilon}] = \Upsilon$$

and

$$\mathbb{E}[(\hat{\Upsilon} - \Upsilon)(\hat{\Upsilon} - \Upsilon)^{\top}] = (Z^{\top} \psi^{-1} Z)^{-1} = [Z^{\top}(\Sigma^{-1} \otimes I_n) Z]^{-1}$$

If interest centers only the i^{th} equation and only estimators of V_i are to be considered, the the OLS estimator $\hat{b}_i = (W_i^{\top} W_i)^{-1} W_i^{\top} V_i$ is the minimum variance linear unbiased estimator. However we can improve on this estimator by considering a wider class, namely linear unbiased estimators that are a function of V. Within this class $\hat{\beta}_i$, the i^{th} vector component of $\hat{\Upsilon}$, is better than $\hat{b}_i$ because it uses information on explanatory variables that are included in the system but are excluded from the i^{th} equation. The possible gain in efficiency obtained by jointly considering all the equations led Zellner (1962) to give equation (2.4) the title "a set of seemingly unrelated regression equations." Zellner indicates that if $\sigma_{ij} = 0$ for all $i \neq j$ or if $W_1 = W_2 = \ldots W_k$ the estimators $\hat{b}_i$ and $\hat{\beta}_i$ will be identical, and so there will be no gain in efficiency. Also, the efficiency gain tends to be higher when the explanatory variables in different equations are not highly correlated but the disturbance terms corresponding to different equations are highly correlated.

In most applications Σ is uknown, and so the estimator $\hat{\Upsilon}$ cannot be employed. However, one can utilize the estimated generalized least squares (EGLS) estimator :

$$\hat{\Upsilon} = [Z^{\top}(\hat{\Sigma}^{-1} \otimes I) Z]^{-1} Z^{\top}(\hat{\Sigma}^{-1} \otimes I) V \qquad (2.1.2)$$

where the estimator $\hat{\Sigma}$ is based on the OLS residuals $\hat{\xi}_i = V_i - W_i \hat{b}_i$ and has elements given by

$$\hat{\sigma}_{ij} = \frac{\xi_i^T \xi_i}{n} \quad , \qquad i,j = 1,2,\ldots k \tag{2.1.3}$$

2.2 Least squares estimation of matrix A.

2.2.1 Single location case

We now examine the problem of least squares estimation of matrix A appearing in equation (2.2). Consider a single location i, and thus

$$\beta_i = AX_i$$

where A is a (pxν) matrix of unknown constants and X_i is a (νx1) position vector. Our objective is to find that A which minimizes

$$Q = (V_i - \hat{V}_i)^T \psi^{-1} (V_i - \hat{V}_i).$$

(in this case ψ represents the covariance between observations within a single location).

Now,

$$Q = (V_i - W_i A X_i)^T \psi^{-1} (V_i - W_i A X_i)$$

$$= V_i^T \psi^{-1} V_i - 2V_i \psi^{-1} W_i A X_i + X_i^T A^T W_i^T \psi^{-1} W_i A X_i \tag{2.2.1}$$

Differentiating equation (2.2.1) with respect to the matrix A and setting the result equal to zero we have :

$$\frac{\partial Q}{\partial A} = 0 \Rightarrow \frac{\partial}{\partial A} \left[V_i^T \psi^{-1} V_i - 2V_i^T \psi^{-1} W_i A X_i + X_i^T A^T W_i \psi^{-1} W_i A X_i \right] = 0$$

$$\Rightarrow \quad -2\frac{\partial}{\partial A} \left[V_i^T \psi^{-1} W_i A X_i \right] + \frac{\partial}{\partial A} \left[X_i^T A^T W_i \psi^{-1} W_i A X_i \right] = 0$$

Now,

$$\frac{\partial}{\partial A}\left[V_i{}^{\top}\psi^{-1}W_iAX_i\right] = W_i^{\top}\psi^{-1}V_iX_i^{\top}$$

and

$$\frac{\partial}{\partial A}\left[X_i^{\top}A^{\top}W_i\psi^{-1}W_iAX_i\right] = \frac{\partial}{\partial A}\left[(W_iAX_i)^{\top}\psi^{-1}(W_iAX_i)\right]$$

$$= 2X_i^{\top} \otimes W_i^{\top}\psi^{-1}W_iAX_i$$

and therefore

$$\frac{\partial Q}{\partial A} = 0 \Rightarrow -2W_i^{\top}\psi^{-1}V_iX_i^{\top} + 2X_i^{\top} \otimes W_i^{\top}\psi^{-1}W_iAX_i = 0$$

$$\Rightarrow X_i^{\top} \otimes W_i^{\top}\psi^{-1}W_iAX_i = W_i^{\top}\psi^{-1}V_iX_i^{\top}$$

Taking transposes :

$$\Rightarrow X_i \otimes X_i^{\top}A^{\top}W_i^{\top}\psi^{-1}W_i = X_iV_i^{\top}\psi^{-1}W_i$$

Multiplying both sides by $X^{\top}$:

$$\Rightarrow X_i^{\top}X_i \otimes X_i^{\top}A^{\top}W_i^{\top}\psi^{-1}W_i = X_i^{\top}X_iV_i^{\top}\psi^{-1}W_i$$

$$\Rightarrow X_i^{\top}A^{\top}(W_i^{\top}\psi^{-1}W_i) = V_i^{\top}\psi^{-1}W_i \qquad \text{(since } X_i^{\top}X_i \text{ is scalar)}$$

Now W_i is full rank and hence :

$$X_i^{\top}A^{\top} = (W_i^{\top}\psi^{-1}W_i)^{-1}V_i^{\top}\psi^{-1}W_i$$

Finally, taking transposes we obtain:

$$AX_i = (W_i^T \psi^{-1} W_i)^{-1} W_i^T \psi^{-1} V_i$$

$$\Rightarrow \hat{T}_i = (W_i^T \psi^{-1} W_i)^{-1} W_i^T \psi^{-1} V_i \qquad (2.1.2)$$

Notice that , in the case of a single location,the least squares solution given by equation (2.1.2) is the same as the GLS estimator in equation (2.1.1) (Z in the case of a single location reduces to W_i).

2.2.2 Multiple locations.

For the case where we have more than one location we may express the quantity Q as follows :

$$Q = [V - Z\ \mathrm{vech}(AX)]^T \psi^{-1} [V - Z\ \mathrm{vech}(AX)] \qquad (2.2.1)$$

where V and Z are respectively (knx1) and (knxkp) matrices. A is (pxν) and X is (νxk). Previously in the case of multiple locations we were able to express Q as $\sum_{i=1}^{k} Q_i$ where Q_i was the contribution from the i^{th} location and thus $\frac{\partial Q}{\partial A} = 0$ $\Rightarrow \sum_{i=1}^{k} \frac{\partial Q_i}{\partial A} = 0$. However, this is no longer possible due to the presence of ψ in equation (2.2.1) and we are forced to consider (2.2.1) as a whole. The existence of a closed form solution to equation (2.2.1) has not been established at this time. As a practical alternative we can solve equation (2.2.1) numerically using a Newton-Raphson technique.

The estimate of A at the $(m+1)^{th}$ step is given as :

$$\mathbf{A}^{(m+1)} = \mathbf{A}^{(m)} - [\nabla^2 Q(\mathbf{A})]^{-1} \nabla\, Q(\mathbf{A}) \qquad (2.2.2)$$

where $\nabla^2 Q(\mathbf{A})$ is the Hessian of Q evaluated at the current vech(A) and $\nabla\, Q(\mathbf{A})$ is the gradient function of Q evaluated at vech(A).

The iterative procedure described by equation (2.2.2) has been programmed using the matrix-based language GAUSS. A copy of the source code is given in the Appendix. It should be pointed out that the process for determining the least squares solution for A using the EGLS approach is very computationally intensive since the covariance matrix $\hat{\psi}$ is updated at the end of the Newton-Raphson iterations and the cycle repeated using this revised $\hat{\psi}$. The process terminates when the norm of the difference between two successive approximations to A differ by less than some prescribed ammount.

2.3 Two stage regression approach.

With k locations and n observations per location in a ν-dimensional field we can derive an estimate for A which parallels the so-called two-stage regression procedure previously developed. This estimate is given in closed form as :

$$\hat{\mathbf{T}} = [Z^{\top}(\hat{\Sigma}^{-1} \otimes I)Z]^{-1} Z^{\top}(\hat{\Sigma}^{-1} \otimes I)V\, X^{\top}(XX^{\top})^{-1} \qquad (2.3.1)$$

provided $\nu \leq k$.

We now illustrate the computations associated with both approaches with the use of an example.

3. Example

The data used for the purpose of illustration of the techniques discussed in this paper were generated according to the following scheme.
At selected locations in a two-dimensional field compute values of the independent variables W_1 and W_2 as follows :

$$w_1 = 3\exp\{-\tfrac{1}{2}[\tfrac{x_1^2}{4} + \tfrac{x_2^2}{5}]\}$$

and

$$w_2 = 2\exp\{-[x_1 + x_2]/10\}$$

The parameter values are computed as :

$$\beta_0(X) = -2 + 3x_1 + 4x_2$$
$$\beta_1(X) = 5 - x_1 + 2x_2$$
$$\beta_2(X) = 2 + x_1 - 3x_2$$

and thus

$$A = \begin{bmatrix} -2 & 3 & 4 \\ 5 & -1 & 2 \\ 2 & 1 & -3 \end{bmatrix}$$

In the following table we have used 5 locations with 6 observations at each location. The values of W_1 and W_2 at a given location were randomly generated about the mean value given by the expressions above. Two sets of V data are also given. The first set (V_1) are those values of V that satisfy $V = W\beta$ exactly at each location while the second set (V_2) were obtained by adding to the V_1 values a normally distributed (0,1) random error.

X_1	X_2	W_1	W_2	V_1	V_2
1.0	1.0	2.439	1.701	19.6340	20.5147
1.0	1.0	2.419	1.733	19.5140	19.3308
1.0	1.0	2.343	1.595	19.0580	18.5321
1.0	1.0	2.409	1.667	19.4540	18.9472
1.0	1.0	2.297	1.638	18.7820	19.3467
1.0	1.0	2.456	1.807	19.7360	19.0564
1.5	-0.7	2.140	1.904	14.8564	16.4618
1.5	-0.7	2.156	1.884	14.7780	14.4764
1.5	-0.7	2.202	1.833	14.5890	13.9134
1.5	-0.7	2.195	1.717	13.9247	14.3847
1.5	-0.7	2.145	1.705	13.7525	13.8574
1.5	-0.7	2.194	1.867	14.7626	16.0857
-2.3	1.8	1.779	2.024	6.1543	4.9149
-2.3	1.8	1.067	2.096	-2.0169	-2.5244
-2.3	1.8	1.092	2.105	-1.7957	-1.9037
-2.3	1.8	1.103	2.129	-1.8126	-2.8437
-2.3	1.8	1.138	2.050	-0.9808	-1.5414
-2.3	1.8	1.136	2.056	-1.0368	0.0647
-1.4	-1.2	2.042	2.713	8.5626	8.8700
-1.4	-1.2	2.005	2.651	8.1542	7.7663
-1.4	-1.2	2.003	2.514	7.5708	9.1219
-1.4	-1.2	2.080	2.628	8.3576	8.0765
-1.4	-1.2	2.014	2.654	8.2028	9.9138
-1.4	-1.2	2.052	2.609	8.1658	6.6364
0.9	2.1	1.768	1.488	18.7152	17.4071
0.9	2.1	1.775	1.479	18.8039	18.3699
0.9	2.1	1.743	1.463	18.5927	17.4336
0.9	2.1	1.790	1.478	18.9318	20.7829
0.9	2.1	1.781	1.478	18.8571	17.6514
0.9	2.1	1.751	1.588	18.2341	19.8053

3.1 Analysis of V_1 data.

Using the V_1 data in the table as dependent we estimate the **A** matrix using the two approaches as :

2-stage regression (equation 2.3.1)

$$\mathbf{A} = \begin{bmatrix} -2 & 3 & 4 \\ 5 & -1 & 2 \\ 2 & 1 & -3 \end{bmatrix}$$

which is in perfect agreement with the actual A matrix used to generate the V_1 data.

Least squares with respect to A.

$$\mathbf{A} = \begin{bmatrix} -1.9996 & 3.0001 & 3.9997 \\ 4.9998 & -1.0000 & 2.0001 \\ 2.0000 & 1.0000 & -3.0000 \end{bmatrix}$$

Again, the estimate of **A** is in almost perfect agreement with the actual **A**.

3.2 Analysis of V_2 data.

3.2.1 OLS Analysis

The output below is from the GUASS program appearing in the Appendix.

```
                    DATA ENTRY
                    ==========

     Enter drive and (optionally) a path for stored matrices : e:
     Do you want OLS estimation or GLS estimation (Type O or G) : o
```

Two-stage regression estimate of matrix A :

11.3359	-17.3812	-20.0568
-1.0255	4.6563	13.4781
2.8953	6.6474	-1.1142

(matrix A_1)

Strike a key when ready . . .

Enter tolerance for determining stopping criterion : ? 1e-6

Iteration 1.0000

16.7866
-1.3619
-6.4514
-4.6814
-0.2161
7.5284
3.1391
2.8161
-3.5875

Iteration 2.0000

19.6008
-0.5280
-9.4370
-6.6670
-0.2688
8.6732
3.8456
2.5658
-3.1661

Convergence established at iteration 5.0000

Least squares estimate

Matrix A =

19.6000	-0.5280	-9.4365
-6.6666	-0.2689	8.6730
3.8456	2.5658	-3.1661

(matrix A_2)

Will now compute the predicted values of dependent variable using this A matrix . . .

Actual matrix of V data :

20.5147	16.4618	4.9149	8.8700	17.4071
19.3308	14.4764	-2.5244	7.7663	18.3699
18.5321	13.9134	-1.9037	9.1219	17.4336
18.9472	14.3847	-2.8437	8.0765	20.7829
19.3467	13.8574	-1.5414	9.9138	17.6514
19.0564	16.0857	0.0647	6.6364	19.8053

Matrix of predicted V values using A_2

19.3935	16.1617	5.1456	8.5615	18.5595
19.4626	15.7533	-2.2217	8.9281	18.6431
18.8827	14.6433	-2.0524	8.4062	18.2893
19.2310	13.5857	-2.1333	7.5825	18.8132
18.9423	14.1238	-1.1860	8.7900	18.7114
19.7670	15.0854	-1.2516	7.9731	18.3179

3.2.2 EGLS procedure.

DATA ENTRY
==========

Enter drive and (optionally) a path for stored matrices : e:

Do you want OLS estimation or GLS estimation (Type O or G) : g
Are observations WITHIN a location independent (y/n) : y

Two-stage regression estimate of matrix A :

5.2807	4.0204	-12.0021
4.4922	0.6046	4.6958
-0.0344	-0.0036	2.0038

(matrix A_3)

Strike a key when ready . . .

Enter tolerance for determining stopping criterion : ? 1e-6

Convergence established at iteration 5.0000

Matrix A =

7.3691	1.4302	2.2187
1.7679	3.0079	3.0224
-0.2485	-1.8380	-3.6782

(matrix A_4)

Will now compute the predicted values of dependent variable
using this A matrix . . .

Strike a key when ready . . .

Actual matrix of V data :

20.5147	16.4618	4.9149	8.8700	17.4071
19.3308	14.4764	-2.5244	7.7663	18.3699
18.5321	13.9134	-1.9037	9.1219	17.4336
18.9472	14.3847	-2.8437	8.0765	20.7829
19.3467	13.8574	-1.5414	9.9138	17.6514
19.0564	16.0857	0.0647	6.6364	19.8053

Estimated V values using matrix A_4 :

20.2321	16.0520	3.2427	8.5912	18.1242
19.8917	16.1273	2.8459	8.3979	18.2866
20.0945	16.3408	2.8293	7.4869	18.0943
20.1941	16.3616	2.7691	7.7877	18.4586
19.4879	16.1586	2.9880	8.3635	18.3612
19.7536	16.2928	2.9716	7.8296	16.9775

```
Do you want to continue iterating on psi matrix (y/n) : n
Execution stopped in line 356
```

3.3 Calibrating at a future location.

We now examine how well the various estimated A matrices from the previous section "predict" the actual V data at a location which was not used in the model fitting.

Position		$\hat{V}$ using matrix :				V
X_1	X_2	A_1	A_2	A_3	A_4	actual
2.0	-1.4	91.7878	28.0765	23.4553	14.7445	14.148
1.6	-2.3	88.2433	38.4083	22.4887	15.3834	14.582
1.8	1.4	76.9956	19.5034	19.7228	19.1832	18.289
	SSE :	14900.4	763.17	151.20	1.80	

Based on the performance at the chosen three test locations we see that $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ performed particularly poorly. The estimate of **A** obtained using the EGLS approach performed substantially better than the other three.

APPENDIX - LISTING OF GAUSS PROGRAMS

MAIN PROGRAM

```
cls;
count=1;
mseold=9999999999;
note="";
loadp gradp=c:\gauss\cp\gradp;
loadp hessp=c:\gauss\cp\hessp;
format /rd 3,0;
"                    Calibration in a correlated field";
"                    ================================";
"                   (Generalized Least-Squares Version)";
print;
"                             David R. Fox";
print;
"                          University of Wyoming";
"                             November 1988";
print;
print;
"  Input requirements :   It is assumed there are k locations for which the";
"  ==================     following data is available at each location : ";
print;
"         Xi : (v x 1)  position vector for ith. location.";
"         Wi : (n x p)  matrix of n observations on p independent variables.";
"         Vi : (n x 1)  vector of observations on dependent variable.";
print;
"  Given the linear model :  Vi=Xi*Bi + Ei where Bi is a (p x 1) vector of";
"  (unknown) parameters, the program will estimate using a least-squares";
"  procedure, the matrix A (p x v) where it is further assumed that :";
print;
"                          Bi = A*Xi";
print;
dos pause;
cls;
print;
print;
print;
print;
"                      HAVE YOU LOADED THE GLSCALC PROGRAM  (y/n) ? : ";;
reply=cons;
if reply $/="Y" and reply $/="y";
stop;
endif;
start:;
cls;
"                             DATA ENTRY";
"                             ==========";
print;
print;
loop1:;
"         Is data to come from keyboard or file (k/f) : ";;
device=cons;
```

```
if device $/="K" and device $/="k" and device $/="F" and device $/="f";
print;
"              * * * ERROR - MUST BE EITHER K OR T !";
"                       Please reenter";
"                     ";;
dos pause;
goto start;
elseif device $=="f" or device $=="F";
loop5:;
"         Enter drive and (optionally) a path for stored matrices : ";;
source=cons;
load path=^source;
loadm w;
loadm v;
loadm x;
nu=rows(x);
k=cols(x);
n=rows(w)/k;
p=cols(w);
goto flag1;
endif;
"         How many locations are there ? : ";;
k=con(1,1);
k=floor(k);
if k<=0;
"         < NOT A VALID SELECTION - REENTER >";
goto loop1;
endif;
print;
loop2:;
"         How many observations at each of these locations ? : ";;
n=con(1,1);
n=floor(n);
if n<=0;
"         < NOT A VALID SELECTION - REENTER >";
goto loop2;
endif;
print;
loop3:;
"         How many INDEPENDENT variables are there (Wi's) ? :";;
p=con(1,1);
p=floor(p);
if p<=0;
"         < NOT A VALID SELECTION - REENTER >";
goto loop3;
endif;
"         Is a constant (B0) term to be included in the model (y/n) : ";;
reply1=cons;
if reply1 $=="Y" or reply1 $=="y";
p=p+1;
note="(First value corresponds to B0 term)";
endif;
if p>n;
print;
"                 * * * ERROR IN INPUT DATA * * *";
PRINT;
```

```
"                There are more parameters than observations";
print;
"                     Please reenter from the beginning";
dos pause;
goto start;
endif;
print;
loop4:;
print;
"         What is the DIMENSION of the field (1,2,or 3) : ";;
nu=con(1,1);
if nu/=1 and nu/=2 and nu/=3;
print;
"               * * * ERROR - MUST BE 1,2, OR 3 ! * * *";
goto loop4;
endif;
nu1=nu;
cls;
if reply1$=="Y" or reply1$=="y";
nu=nu+1;
endif;
"                        SUMMARY OF DESIGN PARAMETERS";
"                        ============================";
print;
print;
format /ld 3,0;
"                       You have specified the following : ";
print;
print;
"     At each of ";;k;;" locations in ";;nu1;;"dimensional space";
"     there are ";;p-1;;" independent variables.";
print;
"     The number of observations on each variable at each location is ";;n;
print;
print;
print;
"             Is this information correct (y/n) : ";;
reply=cons;
if reply $/= "Y" and  reply $/="y";
print;
print;
dos pause;
goto start;
endif;
i=1;
do while i<=k;
cls;
j=1;
"                            DATA ENTRY";
"                            ==========";
print;
print;
"LOCATION ";;i;
print;
"Enter the ";;nu;;"POSITION coordinates : "$+note;
x1=con(nu,1);
```

```
if i==1;
x=x1;
else;
x=x~x1;
endif;
do while j<=n;
print;
"INDEPENDENT variable information  (LOCATION ";;i;;")";
"=========================================================";
"Enter the ";;p;;"values for observation ";;j;;note;
w1=con(1,p);
print;
"Enter the value of the DEPENDENT variable : ";;
v1=con(1,1);
if j==1 and i==1;
w=w1;
v=v1;
goto jump1;
endif;
w=w|w1;
v=v|v1;
jump1:;
j=j+1;
endo;
i=i+1;
endo;
print;
print;
"      Do you want to save these matrices to a file (y/n) : ";;
reply=cons;
if reply $=="Y" or reply $== "y";
print;
"      Enter drive (and optionally a path ) : ";;
destn=cons;
save path=^destn;
save w;
"     File ";;destn;;"W.FMT successfully saved";
save x;
"     File ";;destn;;"X.FMT successfully saved";
save v;
"     File ";;destn;;"V.FMT successfully saved";
endif;
/* loadp gradp=c:\gauss\cp\gradp; */
/*"  Matrix W";
w;
"  Matrix V";
v;
"  Matrix X";
x;  */
/* loadp fcalc=c:\gauss\cp\fcalc; */;
flag1:;
aold=zeros(p*nu,1);
print;
print;
"      Do you want OLS estimation or GLS estimation (Type O or G) : ";;
olsgls=cons;
```

```
if olsgls $=="G" or olsgls $=="g";
vv=reshape(v,k,n);vv=vv';
kk=1;
do while kk<=k;
vv[.,kk]=vv[.,kk]-meanc(vv[.,kk]);
kk=kk+1;
endo;
cov=vv'*vv/n;
print;
"       Are observations WITHIN a location independent (y/n) : ";;
reply=cons;
jump8:;
if reply $=="y" or reply$=="Y";
psi=cov.*.eye(n);
else;
psi=cov.*.ones(n,n);
loadexe path=c:\gauss\gxe;
loadp pinv=c:\gauss\cp\pinv;
psi=pinv(psi);
goto jump9;
endif;
if n<k and reply $=="y" or  reply $=="Y";
print;
print;
"      NOTE : Since the number of observations WITHIN each location" ;
"             is less than the number of locations, will have to "  ;
"             compute the Moore-Penrose inverse of the psi matrix.";
loadexe path=c:\gauss\gxe;
loadp pinv=c:\gauss\cp\pinv;
psi=pinv(psi);
else;
psi=inv(psi);
endif;
else;
psi=eye(k*n);
endif;
jump9:;
if count>1;
goto jump7;
endif;
z=zeros(k*n,k*p);
kk=1;
do while kk<=k;
kk1=(kk-1)*n+1;
kk2=kk*n;
kk3=(kk-1)*p+1;
kk4=kk*p;
w1=w[kk1:kk2,1:p];
z[kk1:kk2,kk3:kk4]=w1;
kk=kk+1;
endo;
/* z; */
jump7:;
itern=1;
kk=1;
inc=9999;
```

```
/* a0=ones(p*nu,1); */
gamhat=inv(z'*psi*z)*z'*psi*v;
gamhat=reshape(gamhat,k,p);gamhat=gamhat';
a0=gamhat*x'*inv(x*x');
format /rd 9,4;
print;
print;
"Two-stage regression estimate of matrix A : ";;a0;
dos pause;
a0=ones(p*nu,1);
cls;
if count>1;
goto jump6;
endif;
"      Enter tolerance for determining stopping criterion : ";;
epsilon=con(1,1);
jump6:;
do while inc>epsilon;
"Iteration ";;kk;
pp=gradp(&glscalc,a0);
ph=hessp(&glscalc,a0);
delta=inv(ph)*pp';
inc=delta'delta;
a0=a0-delta;
kk=kk+1;
a0;
endo;
cls;
print;
print;
a0=reshape(a0,p,nu);
"                    Convergence established at iteration ";;kk;
print;
print;
format /rd 9,4;
"                    Matrix A = ";
a0;
print;
print;
"      Will now compute the predicted values of dependent variable";
"      using this A matrix . . .";
print;
"                       ";;dos pause;
i=1;
do while i<=k;
i1=(i-1)*n;
x1=x[.,i];
i2=1;
do while i2<=n;
if i2==1;
w1=w[i1+i2,.];
else;
w1=w1|w[i1+i2,.];
endif;
i2=i2+1;
endo;
```

```
if i==1;
vhat=w1*a0*x1;
else;
vhat=vhat|w1*a0*x1;
endif;
i=i+1;
endo;
print;
"    V and Vhat . . .";
v~vhat;
resid=v-vhat;
mse=resid'resid;
mse=mse/k/n;
print;
format /re 15,9;
"           Mean Square Error is ";;mse;
print;
format /rd 9,4;
if olsgls $=="o" or olsgls $=="O";
stop;
else;
print;
"           Do you want to continue iterating on psi matrix (y/n) : ";;
cont=cons;
if cont $=="n" or cont $=="N";
stop;
endif;
if abs(mseold-mse)>=epsilon;
aold=a0;
mseold=mse;
resid=(reshape(resid,k,n))';
cov=(resid'resid)/n;
count=count+1;
goto jump8;
endif;
endif;
```

AUXILIARY PROCEDURE FOR COMPUTATION OF EQUATION (2.2.1)

```
proc glscalc(arg);
local q,i,i1,i2,w1,v1,x1,f,f1,gn,a1,b1;
a1=reshape(arg,p,nu);
/* i=1;
do while i<=k;
i1=(i-1)*n;
i2=1;
do while i2<=n;
if i2==1;
w1=w[i1+i2,.];
v1=v[i1+i2,1];
else;
w1=w1|w[i1+i2,.];
v1=v1|v[i1+i2,1];
endif;
i2=i2+1;
endo;
x1=x[.,i];
f1=(v1-w1*a1*x1)'psi*(v1-w1*a1*x1);
if i==1;
f=f1;
else;
f=f+f1;
endif;
i=i+1;
endo; */
b1=vec(a1*x);
f=v-z*b1;
q=f'*psi*f;
retp(q);
endp;
```