### Purdue University Purdue e-Pubs

Department of Electrical and Computer Engineering Technical Reports Department of Electrical and Computer Engineering

8-1-1988

# Spectral Feature Design In High Dimensional Multispectral Data

Chih-Chien Thomas Chen Purdue University

David A. Landgrebe *Purdue University* 

Follow this and additional works at: https://docs.lib.purdue.edu/ecetr

Chen, Chih-Chien Thomas and Landgrebe, David A., "Spectral Feature Design In High Dimensional Multispectral Data" (1988). Department of Electrical and Computer Engineering Technical Reports. Paper 616. https://docs.lib.purdue.edu/ecetr/616

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.



# Spectral Feature Design In High Dimensional Multispectral Data

Chih-Chien Thomas Chen David A. Landgrebe

TR-EE 88-35 August 1988

School of Electrical Engineering Purdue University West Lafayette, Indiana 47907

# TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	vili
ABSTRACT	<b>X</b>
CHAPTER I. INTRODUCTION	1
<ol> <li>1.1 Research Objective</li></ol>	1 
CHAPTER II. KARHUNEN-LOEVE TRANSFORM	13
<ul> <li>2.1 Minimum Mean Square Truncation Error</li> <li>2.2 Uncorrelated Transformed Coefficients</li> </ul>	14 20
CHAPTER III. SPECTRAL FEATURE DESIGN	22
<ul> <li>3.1 Non-Overlapping Band Feature Selection Algorith</li> <li>3.2 Walsh Function Approach</li></ul>	nm24 28 32 34
<ul> <li>3.5 Experimental System.</li> <li>3.6 Preliminary Results.</li> <li>3.7 Selection of the Best On-Board Preprocessing So</li> </ul>	
3.0 Canonical Analysis and Ground Station Data Pro	Juessing

# Page

CHAPTER	IV RESULTS AND DISCUSSIONS	60
4.1	Vegetation Data	.61
4.2 4.3 4.4	Soil Data Hughes Phenomenon Signal to Noise Ratio Considerations	.84 .91
CHAPTER	V CONCLUSIONS AND RECOMMENDATIONS	.95
5.1 5.2	Conclusions Recommendations	.95 97
LIST OF	REFERENCES	100
APPENDI	CES	era Tara T

Appendix A	-	IBM 3083 Macro File105
Appendix B	-	Spectral Feature Design System-Program Listing106

# LIST OF TABLES

Table		age
1.1	Data Set Designation for Preliminary Test	9
3.1	Band Edges Obtained by Infinite Clipping of the Average of the First 12 Eigenvectors for Data Set K2	26
3.2	Absolute Values of Inner Products Between Optimal Functions and Walsh Functions	30
3.3	Similarity Relation Between Optimal Functions and Walsh Functions	31
3.4	Linearly Dependent Bands Found by Overlapping Band Feature Selection Algorithm for Data Set K2	35
3.5	Linearly Independent Bands Found by Overlapping Band Feature Selection Algorithm for Data Set K2	37
3.6	Bands Found by Non-Overlapping Band Feature Selection Algorithm for Data Set K2	40
3.7	Bands Found by Overlapping Band Feature Selection Algorithm for Data Set K2	41
3.8	Band Edges of Landsat MSS, TM, PA and PB Sensors	42
3.9	Probability of Correct Classification for 6 Data Sets	42
4.1	Vegetation Data Sets	63
4.2	Soil Data Sets	77
4.3	Data Sets Used to Test the Occurrence of the Hughes Phenomenon .	86
4.4	Mean Square Representation Error for Data Set K2	93

# LIST OF FIGURES

Figur		Page
1.1	Realization of Spectral Feature Design	6
1.2	First 12 Eigenvectors of Data Set K2	10
3.1	Average of the First 12 Eigenvectors of Data Set K2	27
3.2	Thresholded Version of Figure 3.1	27
3.3	First 10 Walsh Functions Indexed by Number of Axis Crossings	29
3.4	Infinite Clipping Versions of the First 6 Eigenfunctions for K2	33
3.5	Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set K1	43
3.6	Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set K2	44
3.7	Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set K3	45
3.8	Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set N1	46
3.9	Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set N2	47
3.10	Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set N3	48
3.11	Performance Comparison for Data Set K1	49
3.12	Performance Comparison for Data Set K2	50
3.13	Performance Comparison for Data Set K3	51

1	١.	,	i	
	٧	Γ.	I	

Figure		Page
3.14	Performance Comparison for Data Set N1	52
3.15	Performance Comparison for Data Set N2	53
3.16	Performance Comparison for Data Set N3	54
4.1	Classification Performance for Data Set K1	64
4.2	Classification Performance for Data Set K2	65
4.3	Classification Performance for Data Set K3	66
4.4	Classification Performance for Data Set N1	67
4.5	Classification Performance for Data Set N2	68
4.6	Classification Performance for Data Set N3	69
4.7	Classification Performance for Kansas Multitemporal Data Set	70
4.8	Classification Performance for N. Dakota Multitemporal Data Set	71
4.9	Classification Performance for KS/ND Combined Data Set	72
4.10	Classification Performance for Iowa Multitemporal Data Set	73
4.11	Classification Performance for S. Dakota Multitemporal Data Set	74
4.12	Classification Performance for Soil Data Grouped by Soil Order	79
4.13	Classification Performance for Soil Data Grouped by Organic # 1	80
4.14	Classification Performance for Soil Data Grouped by Organic # 2	81
4.15	Classification Performance for Soil Data Grouped by Iron Oxide	82
4.16	Classification Performance for Soil Data Grouped by Soil Texture.	83
4.17	First Experiment of the Hughes Phenomenon : Data Set K1H	87
4.18	Second Experiment of the Hughes Phenomenon : Data Set K1Q	88
4.19	Third Experiment of the Hughes Phenomenon : Data Set N2H	89
4.20	Fourth Experiment of the Hughes Phenomenon : Data Set N2Q	90
4.21	Mean Square Representation Error for Data Set K2	94

#### ABSTRACT

Chen, Chih-Chien Thomas Ph.D., Purdue University, August 1988. Spectral Feature Design in High Dimensional Multispectral Data. Major Professor : David A. Landgrebe. School of Electrical Engineering.

The High resolution Imaging Spectrometer (HIRIS) is designed to acquire images simultaneously in 192 spectral bands in the 0.4-2.5  $\mu$ m wavelength region. It will make possible the collection of essentially continuous reflectance spectra at a spectral resolution sufficient to extract significantly enhanced amounts of information from return signals as compared to existing systems. By effectively utilizing these signals, direct identification of the parameters of species can be achieved and their subtle changes can also be observed and measured.

The advantages of such high dimensional data come at a cost of increased system and data complexity. For example, since the finer the spectral resolution, the higher the data rate, it becomes impractical to design the sensor to be operated continuously. Even operating HIRIS in a request only mode, its 512 Mbps raw data rate still constitutes a serious communication challenge. In order to solve this problem, it is essential to find new ways to preprocess the data which reduce the data rate while at the same time maintaining the information content of the high dimensional signal produced.

In this thesis, four spectral feature design techniques are developed from the Weighted Karhunen-Loeve Transforms. They are : non-overlapping band feature selection algorithm, overlapping band feature selection algorithm, Walsh function approach, and infinite clipped optimal function approach. From a simplicity and effectiveness point of view, the infinite clipped optimal function approach is chosen since the features are easiest to find and their classification performance is the best. This technique approximates the spectral structure of the optimal features via infinite clipping and results in transform coefficients which are either +1, -1 or 0. Therefore the necessary processing can be easily implemented on-board the spacecraft by using a set of programmable adders that operate on the grouping instructions received from the ground station.

After the preprocessed data has been received at the ground station, canonical analysis is further used to find the best set of features under the criterion that maximal class separability is achieved.

In this research, both 100 dimensional vegetation data and 200 dimensional soil data are used to test the spectral feature design system. It will be shown that the infinite clipped versions of the first 16 optimal features derived from the Weighted Karhunen-Loeve Transform have excellent classification performance. Further signal processing by canonical analysis increases the compression ratio and retains the classification accuracy. The overall probability of correct classification is over 90% while providing for a reduced downlink data rate by a factor of 10.

统法法自守

viii

# CHAPTER I INTRODUCTION

1

#### 1.1 Research Objective

Due to the recent advance in optics and solid state technology, it is now possible to build sensors with much finer spectral resolution. This will provide the opportunity for collecting data for a much enriched information source. For example, the future High resolution Imaging Spectrometer (HIRIS) is planned to have as many as 192 spectral bands [1]. Since the signal dimensionality is tremendously increased, current techniques for analyzing multispectral data would not be adequate. In order to effectively utilize the information collected and achieve these benefits from the high dimensional measurements, it is essential to find new ways to process the data which reduce the data rate while at the same time maintaining the information content of the signals produced.

The fundamental objective of this research is to develop an objective and practical spectral feature design technique for high dimensional multispectral data.

One possible approach that might be used to accomplish the design objective is to tailor the spectral features to the particular analysis problem at hand. Features might be made up by grouping (i.e. summing) the narrow band response functions in particular spectral regions on board the spacecraft, based upon the particular classes of ground cover parameters that are to be identified. The main advantage of this approach is the possibility of local optimality. Instead of finding optimal features with respect to all possible scenes (global optimal), a more practical and adaptive approach is introduced for each individual situation. The maximal attainable performance of local optimal features is indeed better and at least not worse, than that of global optimal ones. The problem then reduces to finding a means for deciding how to choose these band groupings effectively for each different analysis situation such that the data rate is greatly reduced while the classification performance is preserved or increased.

#### **1.2** Previous Approaches

There have been basically four approaches to this feature design problem. They are (1) in-depth studies of physical considerations, (2) empirical methods, (3) simulation methods, and (4) analytical approaches.

Important physical considerations which have been investigated are atmospheric effects and the interaction of light with various cover types. By evaluating the transmittance of the atmosphere over the spectral interval of interest [2,3], one can eliminate certain portions of the interval, since little or no information content is contained in those regions.

The interaction of electromagnetic radiation with plant leaves [4], soils [5] and waters [6] has been studied in the past to find the most effective spectral features for discrimination. A typical procedure for these studies is to take

measurements with a spectroradiometer on restricted information classes over the entire spectrum. Then the average of the spectral responses is found and the subsequent conclusion is drawn from the average. The basic disadvantage of this approach is that only the mean value is considered. The potential information in the variance and covariance is neglected and lost.

The second method is empirical in that a scanner with many spectral bands is constructed, and the selection of the bands is done experimentally. The studies have been done with agriculture cover types [7], forest covers [8], and geological applications [9]. The main advantage of the empirical method is the retaining of the information in the variations about the mean. The correlation is considered in the feature design process. However, the spectral sampling is crude and incomplete for representing the whole spectrum.

Simulation methods have been developed [10] to generate typical spectra according to a scene model. These artificial spectral response functions are then used to choose the best set of features. However, due to the complexity of the scene and the interrelations of various parameters [11], an accurate enough model of the scene is not available yet up to present.

The recent advances in optical and solid state technologies make it possible to build high dimensional multispectral sensors such as HIRIS, with a spectral resolution of 10 nm and a spatial resolution of 30m [1]. In order to effectively utilize, including acquire, archive, retrieve, transmit and analyze the data collected, analytical feature design approaches are sought because of their objective and machine-oriented natures. Early works of this approach are found in Wiswell's and Wiersma's Ph.D dissertations. Wiswell [12] studied the

feasibility of using the maximal average mutual information [13] as a criterion to evaluate the spectral features. The best set of features are chosen so as to obtain the minimal reduction in uncertainty about the scene after the observation is made. The research showed that average mutual information is a useful concept to construct the feature sets. However the relationship between average mutual information and global performance criterion such as classification accuracy was not demonstrated. Moreover, the technique was only applied to much lower dimensional signals (about 10); the feasibility for high dimensional signals in the range of one or two hundred spectral bands was not shown.

Wiersma and Landgrebe [14,15] proposed the use of minimum mean square representation error criterion for feature design. It was shown that an analytical feature design procedure can be established by applying a weighted Karhunen-Loeve Transform [16,17,18] to the observation space in which the eigenvectors of the transform are the optimal (though impractically complex) spectral features. The dimensionality in this research was 100 which was much higher than that in Wiswell's work. A manual band feature selection was suggested according to the relative importance of spectral regions as indicated by the eigenfunctions. The concept of spectral dominancy was introduced although the final stages of the feature design process were manually implemented. This appears to be tedious and impractical when the number of cover types is greatly increased. Another drawback in Wiersma's work lies basically in the subjective nature of the manual feature design process.

#### **1.3 Current Investigation**

The research results presented here will adopt some procedures to extend Wiersma's work in such a way that objective, machine implemented spectral feature design schemes become feasible. The idea of local optimality is introduced in this thesis. Instead of finding the features that are optimal with respect to all possible scenes (global optimal), it is now proposed to tailor the spectral features to the specific user problem at hand. The maximally attainable performance can then be increased. The new concept of structure similarity and its realization are discussed in this dissertation. This makes the feature design problem more general in the sense that overlapping features become practical and easily implemented.

5

In this research four methods are developed which in effect lead to suboptimal but now practical versions of the optimal features. These derived spectral features were obtained by combining groups of adjacent spectral samples into bands, usually one or more hundred nanometers wide, that are specially tailored to the analysis task at hand. These features could be implemented by utilizing simple programmable adders at the sensor output as shown in Figure 1.1



N = no. of Spectral Samples collected N = no. of Spectral Features desired

#### Figure 1.1 Realization of Spectral Feature Design

In Figure 1.1, N is the signal dimensionality from the sensor output, and  $N_f$  is the number of spectral features used. The programmable adders on board the spacecraft act according to the received grouping instruction from the ground station, either adding (+1), subtracting (-1) or omitting (0) bands for each spectral function. The resulting features are then transmitted down to the ground station for further processing.

The first method is based on the dominancy property of the spectral bands. A manually subjective selection process was used previously in Wiersma's work [14,15]. In this research, an objective and machine oriented process is developed. The spectral band edges are found by applying infinite clipping [21] to the average of the first few eigenvectors associated with the largest eigenvalues. This technique is referred to as a non-overlapping (N.O.L.) band feature selection algorithm due to the fact that designed features are not overlapping.

The second approach utilizes a transformation from the optimal feature space to a new space based upon Walsh Functions (W.F.) [19,20]. These functions have the attractive features of being everywhere equal to either +1 or -1, and being ordered by the number of axis crossings. Thus the transformation can be implemented by either adding or subtracting bands, and the various functions will correspond to spectral ranges of a variety of widths.

The third scheme applies infinite clipping (I.C.) [21] to the original optimal functions derived from the weighted K-L transform. The resulting features are the infinite clipped optimal functions. In this thesis, the experiment concludes

that this scheme is the most promising technique in the sense of best classification performance under the same compression requirement.

The fourth approach extracts the zero crossing information from each optimal function and chooses those spectrum intervals that are in between two zero crossings as band features. Since the band features derived from each optimal function in this way might be linearly dependent [22], special precaution must be taken to get rid of linearly dependent bands. This method is called overlapping (O.L.) band feature selection algorithm because the bands derived by this scheme are overlapping.

# 1.4 Preliminary Test of the On-Board Preprocessing System

From a simplicity and effectiveness point of view, not all the four developed approaches are ideal for data preprocessing. Six preliminary test data sets are used to select the best technique. The goal is to find the most effective scheme under the simplicity requirement. Of the six sets of high spectral resolution field measurement data, three were taken over Williams County, North Dakota, each with 3 information classes: spring wheat, summer fallow and natural pasture. The other three were taken over Finney County, Kansas, again with 3 information classes each: winter wheat, summer fallow, and grain sorghum or other crops. For convenience, these data sets are referred to with a letter/number designator as shown in Table 1.1.

These data were taken by the Field Spectrometer System (FSS) [23] mounted in a helicopter. The spectral resolution was 0.02  $\mu$ m for the interval from 0.4  $\mu$ m to 2.4  $\mu$ m.

Location	Date	Designation	#of Observ.
Kansas	9/28/76	K1	832
Kansas	5/03/77	K2	1551
Kansas	6/06/77	K3	1477
N. Dakota	5/08/77	N1	1265
N. Dakota	6/29/77	N2	1239
N. Dakota	8/04/77	N3	1444

 Table 1.1
 Data Set Designation for Preliminary Test

For each of the six data sets, the collection of the spectral sample functions forms the ensemble of a random process. The mean vector and the covariance matrix of this ensemble are first estimated. The estimate of the covariance matrix is used to solve the generalized Karhunen-Loeve equation which results in the eigenvalues and the eigenvectors of the transform. Figure 1.2 shows the magnitude of the first 12 eigenvectors associated with the largest eigenvalues for the data set K2 [15]. They will be used to explain the feature design schemes in chapter III. The spectral interval is 0.02  $\mu$ m as stated previously. Therefore the dimensionality used in these preliminary tests is 100.

From this preliminary test, it is concluded that the infinite clipped optimal transform is the simplest and most effective method for on-board data preprocessing.



Figure 1.2 First 12 Eigenvectors of Data Set K2





#### 1.5 Outline of the Thesis

In chapter 2, a theoretical review of the weighted K-L transform is given. Two important properties, minimum mean square truncation error and uncorrelated transformed coefficients are proved for this generalized transform.

Chapter 3 discusses in detail the four schemes developed to design the spectral features in high dimensional multispectral data. Two of them, non-overlapping band feature selection algorithm and overlapping band feature selection algorithm, are developed from the dominancy concept in eigenfunctions; and the other two, Walsh function approach and infinite clipped optimal function approach are derived from the idea of structure similarity between two sets of functions. Furthermore, a comparison among these data preprocessing schemes is included in this chapter. From the simplicity and effectiveness point of view, it is found that the infinite clipped optimal function approach is the best technique. After the preprocessed data would be received at the ground station, canonical analysis would be applied to the infinite clipped optimal transformed data to obtain maximal class separability.

Chapter 4 shows the final results of this research. Both the vegetation data and the soil data are included in this chapter. The Hughes phenomenon is also discussed.

Chapter 5 summaries the final conclusions and gives recommendations for the future work.

An IBM 3083 Macro file used to run the spectral feature design system and the source code of the system are given in the appendices.

# CHAPTER II KARHUNEN-LOEVE TRANSFORM

The Karhunen-Loeve (KL) expansion [44] was developed to represent random processes. It maps the continuous parameter random process into a sequence of random variables [24]. The expansion generates a set of deterministic orthonormal basis functions. This set has a unique errorminimizing property and uncorrelated transformed coefficients. These properties make it the optimal coordinate system for many feature design problems.

This transform can be generalized [25,26] to include a weighting function to account for certain types of a priori knowledge of the parameter set, and its proper use may have an important impact on the extraction of useful information [15]. Thus using the weighted form of K-L transform may result in more practical and realizable feature design.

In the following we will show that minimum mean square truncation error (MMSE) and uncorrelated coefficients properties, which are directly related to this research, also hold for the generalized K-L transform. The MMSE property ensures that the eigenfunctions associated with the largest eigenvalues derived from the weighted K-L transform are the optimal basis functions in the sense of signal representation. Uncorrelated coefficients property guarantees that the transformed coordinates are independent under Gaussian assumption.

### 2.1 Minimum Mean Square Truncation Error

Let  $X(\lambda)$  be a sample function of a random process. Assume that the random process is continuous in probability and almost every sample function of the random process has finite norm in L<sub>2</sub>( $\Lambda$ ) space [27]. Then X( $\lambda$ ) can be represented by an expansion of the form [24]

$$X(\lambda) = \sum_{i=1}^{\infty} y_i \Phi_i(\lambda)$$
 (2.1)

where the functions  $\{\Phi_i (\lambda)\}$  are deterministic and the expansion coefficients  $\{y_i\}$  are random variables.

Define a weighting function  $W(\lambda)$  with real finite positive values. Without loss of generality, the set { $\Phi_i$  ( $\lambda$ )} will be taken to be orthonormal with respect to  $W(\lambda)$ . From the generalized inner product [27] which defines the metric in L<sub>2</sub>( $\Lambda$ ) space, we have

$$(\Phi_{j}, \Phi_{j})_{W} = \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) \Phi_{j}(\lambda) d\lambda = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases}$$
(2.2)

and

$$y_{i} = (\Phi_{i}, X)_{W} = \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) X(\lambda) d\lambda \qquad (2.3)$$

If the set  $\{\Phi_i(\lambda)\}$  is not orthonormal to begin with, it can be orthonormalized by the Gram-Schmidt procedure [57]. That such sets exists in  $L_2(\Lambda)$  space has been demonstrated by the construction of sets such as complex sinusoids, Legendre polymonials, Chebyshev polymonials, Laguerre functions, Walsh functions and others.

Therefore  $\mathbf{Y} = \{ y_1, y_2, \dots, \}$  is simply an orthonormal transformation of the random function  $X(\lambda)$ , and is itself a random vector. Each component of  $\mathbf{Y}$  is a feature which contributes to representing the observed sample function  $X(\lambda)$ .

Furthermore, we are going to choose a set  $\{\Phi_i(\lambda)\}$  which is complete in  $L_2(\Lambda)$  space. That is, if we define the sequence

$$c_{n}(\lambda) = \sum_{i=1}^{n} y_{i} \Phi_{i}(\lambda) \qquad (2.4)$$

then,

$$\lim_{n \to \infty} \left\{ \int_{i=1}^{n} [X(\lambda) - \sum_{i=1}^{n} y_i \Phi_i(\lambda)]^2 W(\lambda) d\lambda \right\} = 0$$
(2.5)

That the sequence  $c_n(\lambda)$  converges to  $X(\lambda)$  in the mean square sense, is denoted by

$$X(\lambda) = \lim_{n \to \infty} c_n(\lambda)$$
(2.6)

This convergence guarantees that the series can be made arbitrarily close to  $X(\lambda)$  by increasing n in the expansion.

The problem of designing the optimal sensor then becomes that of selecting the set of complete orthonormal (CON) basis functions {  $\Phi_i(\lambda)$  } such that the series representation will be optimal with respect to the minimum mean square error criterion. In the stochastic environment, this representation error is taken over the ensemble of the random process. Hence, we need :

$$E\left\{\int_{\Lambda} \left[X(\lambda) - \sum_{i=1}^{\infty} y_i \Phi_i(\lambda)\right]^2 W(\lambda) d\lambda\right\} = 0$$
 (2.7)

Another desirable property is that the convergence be rapid in the first few terms, that is, each additional term used in the series expansion decreases the representation error by a maximum amount. This property is called energy packing.

In the real applications, however, it is impractical to transmit an infinite or even a very large number of channels to the ground. Therefore only a finite number of terms in the expansion would be used. Let n be a finite number such that the representation error by using the first n terms in the expansion is less than T, the maximal acceptable error. Then we require the selected orthonormal basis functions {  $\Phi_i(\lambda)$  } to be complete in a finite n dimensional subspace of L<sub>2</sub>( $\Lambda$ ). That is, for any T > 0, there is an n<sub>0</sub> such that

$$E\left\{\int_{\Lambda} [X(\lambda) - \sum_{i=1}^{n} y_{i} \Phi_{i}(\lambda)]^{2} W(\lambda) d\lambda\right\} < T ; n > n_{0}$$
(2.8)

for any  $X(\lambda)$  defined in the  $L_2(\Lambda)$  space.

This completeness property in finite dimensional space can guarantee that if we use the n dimensional subspace of  $L_2(\Lambda)$ , spanned by the first n elements of a complete orthonormal set { $\Phi_i(\lambda)$ }, for the representation of an arbitrary signal, then the norm of the error can be made arbitrarily small by choosing n sufficiently large.

The objective then is to find the a finite set of orthonormal basis functions that have the above minimum representation error and energy packing properties. In the following, we are going to show that the eigenfunctions derived from the Weighted Karhunen-Loeve transform are just the desired optimal basis functions.

In the above finite n dimensional subspace of  $L_2(\Lambda)$ , suppose only m terms in the expansion will be used to estimate the observed X( $\lambda$ ), then the estimate  $\hat{\mathbf{X}}(\lambda)$  can be expressed in the following form

$$\hat{X}(\lambda) = \sum_{i=1}^{m} y_i \Phi_i(\lambda) + \sum_{i=m+1}^{n} b_i \Phi_i(\lambda)$$
 (2.9)

The constants {  $b_i$  } are preselected. The objective is to find the basis functions and the constants {  $b_i$  } in such a way that the minimum mean square error can be obtained.

Since we do not use all of the basis functions, the representation error due to truncation is then equal to

$$\Delta X (\lambda) = X (\lambda) - \hat{X} (\lambda) = \sum_{i=m+1}^{n} (y_i - b_i) \Phi_i (\lambda)$$
(2.10)

We define the weighted mean square error to be

$$\mathbf{WMSE} = \mathbf{E}((\Delta X, \Delta X)_{\mathbf{W}}) = \mathbf{E}(\sum_{i=m+1}^{n} (y_i - b_i) \sum_{j=m+1}^{n} (y_j - b_j) \int_{\Lambda} \Phi_i(\lambda) \mathbf{W}(\lambda) \Phi_j(\lambda) d\lambda) \quad (2.11)$$

Since the basis functions are orthonormal, Eq (2.11) reduces to

WMSE = 
$$\sum_{i=m+1}^{n} E(y_i - b_i)^2$$
 (2.12)

The mean square error is minimized when

$$\frac{\partial E(y_i - b_i)^2}{\partial b_i} = -2E(y_i - b_i) = 0$$
(2.13)

That is, the preselected constant  $b_i$  must be equal to the expected value of the transform component  $E(y_i)$ .

We are left to show that when  $\Phi_i$  ( $\lambda$ ) is a weighted K-L basis, then the weighted mean square error is minimized. We need to minimize

$$\mathbf{WMSE} = \sum_{i=m+1}^{n} E(\mathbf{y}_{i} - E(\mathbf{y}_{i}))^{2} = \sum_{i=m+1}^{n} \iint_{\Lambda\Lambda} \Phi_{i}(\lambda) \mathbf{W}(\lambda) \mathbf{K}_{\mathbf{X}}(\lambda, \mathbf{u}) \mathbf{W}(\mathbf{u}) \Phi_{i}(\mathbf{u}) d\mathbf{u} d\lambda \quad (2.14)$$

where  $K_{\chi}(\lambda, u)$  is the covariance function of the random process.

Using the orthonormality constraint, we can write the mean square error as the quadratic functional [19] of  $\Phi_i$  ( $\lambda$ )

WMSE = 
$$\sum_{i=m+1}^{n} \int_{\Lambda} \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) K_{\chi}(\lambda, u) W(u) \Phi_{i}(u) du d\lambda$$
$$- \sum_{i=m+1}^{n} \lambda_{i} \{ \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) \Phi_{i}(\lambda) d\lambda - 1 \}$$
(2.15)

Minimizing with respect to  $\Phi_j$  yields [19]

$$\nabla_{\Phi_{i}}(\mathbf{WMSE}) = 2 \int_{\Lambda} \mathbf{W}(\lambda) \mathbf{K}_{\mathbf{x}}(\lambda, \mathbf{u}) \mathbf{W}(\mathbf{u}) \Phi_{i}(\mathbf{u}) d\mathbf{u} - 2\lambda_{i} \mathbf{W}(\lambda) \Phi_{i}(\lambda) = 0 \quad (2.16)$$

The set {  $\lambda_i$  } thus turns out to be the eigenvalues of the covariance function of the observed X( $\lambda$ ), and the basis functions satisfy the weighted K-L equation

$$\int_{\Lambda} \mathbf{K}_{\mathbf{x}}(\lambda, \mathbf{u}) \mathbf{W}(\mathbf{u}) \Phi_{\mathbf{i}}(\mathbf{u}) d\mathbf{u} = \lambda_{\mathbf{i}} \Phi_{\mathbf{i}}(\lambda) \qquad \mathbf{i} = 1, 2, ..., \mathbf{n}$$
(2.17)

From equations 2.14 and 2.17, we have

or.

**WMSE** = 
$$\sum_{i=m+1}^{n} \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) [\lambda_{i} \Phi_{i}(\lambda)] d\lambda$$
 (2.18)

$$WMSE = \sum_{i=m+1}^{n} \lambda_{i}$$
 (2.19)

If we rank the optimal functions according to the magnitudes of their associated eigenvalues from the largest to the smallest, then using the first few optimal functions in the series representation will results in the desired weighted minimum mean square error. Furthermore, the energy packing property will also be satisfied since the mean square error reduction for using each additional term in the expansion will be maximized.

#### 2.2 Uncorrelated Transformed Coefficients

The generalized K-L transform results in uncorrelated coefficients. This property can be derived as follows. Since

$$\mathbf{Y} = \{ y_1, y_2, \dots, y_n \}$$
 (2.20)

where

$$y_{i} = \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) X(\lambda) d\lambda$$
(2.21)

and the covariance between yi and yi is defined as

$$\sigma_{i,j} = E(y_i - E(y_i))(y_j - E(y_j))$$
(2.22)

Using Eq.(2.21), Eq.(2.22) becomes

$$\sigma_{i,j} = \iint_{\Lambda\Lambda} \Phi_{i}(\lambda) W(\lambda) K_{x}(\lambda, u) W(u) \Phi_{j}(u) du d\lambda$$
(2.23)

From the Weighted Karhunen-Loeve Equation derived in Eq.(2.17), we get

$$\sigma_{i,j} = \int_{\Lambda} \Phi_{i}(\lambda) W(\lambda) [\lambda_{j} \Phi_{i}(\lambda)] d\lambda = \begin{cases} \lambda_{i} & \text{if } i = j \\ 0^{i} & \text{if } i \neq j \end{cases}$$
(2.24)

Therefore the transformed coefficients are uncorrelated. If the underlying distribution of the random process is Gaussian, the coefficients are then independent.

# CHAPTER III SPECTRAL FEATURE DESIGN

From the discussion in chapter 2, we know the weighted K-L transform preserves the minimum weighted mean square error (MWMSE) and ordered uncorrelated coefficients properties. In fact, the K-L transform is a special case of its generalized form with unity weight matrix. The fundamentals in remote sensing indicate [14,15] that the eigenfunctions derived in the K-L transform with unity weight matrix can not be used satisfactorily for feature design. The reason for this is basically the fact that the reflectance around the two water absorption bands has high variance and thus tends to dominate the formation of the basis functions. Therefore the spectral response in these two regions is not information-bearing. Indeed, the spectral radiance emanates mostly from the atmosphere and must be considered as noise. Understanding this important a priori knowledge about the scene, we can incorporate a weighting function into the calculation process to eliminate the effect of noise. The generalized K-L transform is then the solution. The resulting optimal functions can be used to transform the original observation space into a new feature space.

In this chapter, four spectral feature design techniques will be presented first. Using simplicity and effectiveness as criteria, the most promising technique is then selected from these four schemes for our final feature design system. The four techniques developed in the course of this research are

- Non-overlapping band feature selection algorithm,
- 2. Walsh function approach,

1.

- 3. Infinite clipped optimal function approach, and
- 4. Overlapping band feature selection algorithm.

The non-overlapping and overlapping band feature selection algorithms are derived from the shape of the optimal features. The Walsh function approach and the infinite clipped optimal function approach are developed from the structure of the optimal features.

After performing the generalized K-L transformation to the data [15], where a weight function is incorporated into the transform to avoid portions of the spectrum where the atmosphere is known to be opaque, the eigenfunctions can be found. These eigenfunctions serve as optimal features that linearly transform the original measurement space to the new space in a minimum mean square error sense [18]. However, because of the inherently complex nature of the optimal functions, an easy and fast implementation directly using them to process the tremendous amount of information collected must be found. Therefore, more realistic features are sought in order to achieve this requirement. More realistic features can be found by carefully studying the shapes of the first few eigenfunctions. The importance of a wavelength region for the purpose of accurately representing the ensemble of functions is indicated by the magnitude of the eigenfunctions in that region. It is hypothesized that the importance of a region in an ensemble-representational sense is positively correlated with (though not identical to) its importance with respect to classification accuracy. Referring to Figure 1.2, it is observed that each eigenfunction thus points to the more important regions.

For instance, the magnitude of the first eigenfunction indicates that there are 3 important regions over the entire spectrum: 0.4-1.28  $\mu$ m, 1.48-1.78  $\mu$ m and 1.98-2.4  $\mu$ m, the magnitude of the second eigenfunction indicates that important regions are approximately 0.4-0.66  $\mu$ m, 0.66-1.28  $\mu$ m, 1.48-1.78  $\mu$ m and 1.98-2.4  $\mu$ m, etc. From the fact that the magnitude of the first eigenfunction is very similar to the soil response function, and the magnitude of the second eigenfunction is similar to the vegetation curve, it is observed that the dominant portion of the ensemble, i.e. summer fallow , winter wheat and unknown crops for this data set K2, can be shown in the first few eigenfunctions derived from the weighted K-L transform. Therefore, it is desired to choose the regions with larger magnitude in the eigenfunctions, especially from those with largest eigenvalues, as sensor bands since these regions contribute most to reduction of representation error as well as increasing of classification performance.

However, such a subjective process is difficult to carry out objectively due to the spectral detail in the eigenfunctions and the number of eigenfunctions to be examined. A machine implemented band selection algorithm based on this dominancy concept in the eigenfunctions is thus sought.

#### 3.1 Non-Overlapping Band Feature Selection Algorithm

Infinite clipping is a procedure used to transform the signal into its signed form [21]. There is evidence in various circumstances that the axis crossings of a signal carry a substantial portion of the information that the signal carries. For example, in the field of speech recognition [28-33], the infinite clipping procedure can been used to extract zero crossing information and perform speech recovery very successfully. For example, Ewing and Taylor [29] showed that zero-crossing information from a speech signal is a feasible way for computer speech recognition; and Niederjohn, et al [30] showed that the set of zero-crossings of a speech waveform represents a nearly minimal set of informational attributes in the sense that any reordering or averaging of the zero-crossing intervals has a detrimental effect upon speech intelligibility.

Some other examples of using zero-crossing information of a signal can also be found in the fields of radar target detection [51-52], biomedical engineering [53], communications [54-55] and image processing [56]. Rainal [52] described a zero-crossing principle for detecting weak narrow-band signals immersed in Gaussian noise. An application of the zero-crossing principle to the detection problem of a stationary radar target in clutter was discussed. Masuda, et al [53] demonstrated in a biomedical context that the muscle fiber conduction velocity, which is known to be an index of the degree of muscle fatigue or muscle disease, can be accurately measured by using zero-crossing information from a surface electromyogram signal. In conventional communications, Voelcker [54] showed that an angle-modulated signal can be demodulated given only its zero-crossings; Wiley, et al [55] proposed an iterative demodulation procedure for very wide-band FM by use of a zerocrossing discriminator. Haralick [56] showed that the zero-crossing of second directional derivatives within the pixel's area can be used to detect the step edges in the image.

Thus, one possible approach to finding the desired procedure would be to apply infinite clipping to extract the zero crossing information. The input to this algorithm will be the average of the first few eigenfunctions. The output of this algorithm is to be the band edges showing how the bands should be chosen. We will refer to this procedure as the non-overlapping (N.O.L.) band feature selection algorithm. Figure 3.1 shows the average of the first 12 eigenfunctions. After thresholding, the data of Figure 3.1 become as in Figure 3.2 where +1 represents the positive portions of Figure 3.1, -1 represents the negative portions, and 0 represents the water absorption bands centered at 1.4 and 1.9  $\mu$ m respectively. It should be noted that there is no response over the above water absorption bands due to the use of the weight function in the K-L transform, which has been set 1.0 over the entire spectrum and a very small positive value in the water bands.

The band edges are found as follows: whenever a transition in sign or magnitude occurs in Figure 3.2, the wavelength of the associated band is recorded. Table 3.1 shows the results after transition operation. The band edges in Table 3.1 can be used to set up the suboptimal basis functions for data compression [ refer to the 2nd column in Table 3.6 ].

Table	3.1.	Band	Edges	Obtained	by Infini	te Clippin	g of the	Average
	i e t <sub>a</sub> si	of th	he First	12 Eige	nvectors	for Data	Šet K2	

Band	wavelength (µm)
1	0.40 - 0.68
2	0.68 - 0.90
3	0.90 - 0.92
4	0.92 - 0.94
5	0.94 - 1.00
6	1.00 - 1.06
7	1.06 - 1.12
8	1.12 - 1.26
9	1.26 - 1.28
10	1.48 - 1.78
11	1.98 - 2.40



Figure 3.1 Average of the First 12 Eigenvectors of Data Set K2


#### 3.2 Walsh Function Approach

By carefully viewing the structure of the eigenfunctions in Figures 1.2, one may also observed that the eigenfunctions corresponding to the larger eigenvalues tend to have coarser structure than those with smaller eigenvalues. A similar effect exists in the Walsh functions indexed by the number of zero-crossings. The higher the index of the Walsh function, the finer the structure of the function [19,20]. The first 10 Walsh functions indexed by the number of axis crossings are shown in Figure 3.3, where curve 0 is the first Walsh function with no axis crossing, curve 1 is the second Walsh function with one axis crossing, etc.

The inner product of the two functions may be thought of as a mathematical measure of similarity of the two functions. The absolute values of the inner products of the first 16 eigenfunctions with the first 64 Walsh functions are calculated. Table 3.2 shows part of the results. Absolute values of the inner product are used since the polarity is not significant here. Table 3.3 shows the similarity relation between these two sets of functions. For example, the number "1" in the (1,1) matrix position indicates that the first eigenfunction is more similar to the first Walsh function than to any other 63 Walsh functions since the value 0.84 in Table 3.2 is the largest in the "first " column. The numbers "2", "3" and "4" in the (1,2), (1,3) and (1,4) matrix positions indicate that the 2nd, 3rd and 4th eigenfunctions mostly look like the 2nd, 3rd and 4th Walsh functions respectively in the sense of signal structure similarity. Therefore, the structure of the first 4 eigenfunctions can be approximated by that of the first 4 Walsh functions. By observing the first 16 Walsh functions have approximately

the same structure. The structure in the eigenfunctions is related to the axis crossings in the signals. The coarser the structure, the less the number of axis crossings; and vice versa. These axis crossings are hypothesized to contain important information that can be used for classification. Therefore, it is feasible to use the first few Walsh functions as spectral features in high dimensional multispectral data.



Figure 3.3 First 10 Walsh Functions Indexed By Number of Axis Crossings

Optimal#	1	2	3	4	5	6	7	8
Walsh#	Γ					a di 1999 da secolo di S		
1	0.84	0.21	0.21	0.09	0.01	0.01	0.00	0.00
2	0.21	0.68	0.42	0.12	0.24	0.01	0.13	0.03
3	0.04	0.23	0.66	0.05	0.43	0.02	0.17	0.17
4	0.09	0.03	0.09	0.78	0.12	0.01	0.03	0.09
5	0.04	0.39	0.13	0.05	0.40	0.03	0.13	0.17
6	0.11	0.32	0.09	0.01	0.28	0.14	0.33	0.25
7	0.06	0.11	0.13	0.09	0.20	0.35	0.23	0.10
8	0.03	0.10	0.15	0.06	0.03	0.03	0.52	0.36
9	0.25	0.07	0.03	0.05	0.29	0.14	0.16	0.28
10	0.12	0.05	0.26	0.24	0.27	0.02	0.20	0.14
11	0.13	0.15	0.21	0.06	0.15	0.08	0.18	0.15
12	0.03	0.15	0.05	0.32	0.09	0.07	0.21	0.02
13	0.02	0.18	0.00	0.04	0.08	0.00	0.09	0.03
14	0.15	0.10	0.04	0.15	0.00	0.07	0.09	0.10
15	0.08	0.03	0.09	0.16	0.09	0.15	0.01	0.14
16	0.03	0.04	0.03	0.04	0.20	0.18	0.05	0.10

 Table 3.2
 Absolute Values of Inner Products Between

 Optimal Functions and Walsh Functions

12 9 10 11 13 14 15 16 Optimal# Walsh# 0.01 0.00 0.01 0.01 0.03 0.04 0.01 0.05 1 2 0.00 0.08 0.08 0.07 0.04 0.02 0.06 0.03 3 4 5 6 7 0.00 0.04 0.06 0.00 0.07 0.02 0.06 0.09 0.06 0.04 0.02 0.00 0.12 0.18 0.18 0.14 0.13 0.21 0.02 0.09 0.04 0.14 0.14 0.19 0.09 0.24 0.03 0.07 0.29 0.02 0.16 0.16 0.07 0.09 0.05 0.10 0.02 0.39 0.23 0.03 8 0.03 0.06 0.15 0.15 0.03 0.05 0.10 0.13 9 0.22 0.06 0.08 0.29 0.21 0.19 0.13 0.07 10 0.07 0.10 0.32 0.00 0.06 0.27 0.12 0.17 0.14 11 0.14 0.16 0.08 0.10 0.07 0.01 0.40 12 0.21 0.00 0.05 0.23 0.08 0.14 0.11 0.16 13 0.11 0.33 0.19 0.13 0.08 0.00 0.01 0.09 14 0.11 0.00 0.24 0.12 0.08 0.10 0.07 0.08 0.06 0.18 0.05 15 0.27 0.06 0.05 0.07 0.04 16 0.12 0.24 0.19 0.01 0.01 0.07 0.02 0.05

Optimal#	1	2	3	4	5	6	7	8
Rank	Wa	alsh#		میں میں ایک میں ایک میں ایک میں ایک میں ایک میں کی میں میں میں میں میں میں میں کی م				
	1	2	3	4	<u>3</u>	7	8	8
2	57	5	2	12	5	36	Ŭ,	9
3	9	6	59	60	9	16		6
4	2	3	10	10	6	40	28	22
5	14	1	11	15	10	35	12	18
6	11	58	_1	14	2	19	10	24
7.	10	13	58	2	1	23	25 4 4	04
8	33	11	<u>8</u>	52	16	15		3
9	6	12	2 <u>7</u>	29	21	63	3	5
10	58	59	7	7	59	9	24	00
11	47	7	_5	35	58	32	9	11
12	4	42	50	1	11	6	64	19
13	25	8	35	50	26	20	30	25
14	15	14	4	17	28	54	10	30
15	42	25	15	43	45	4/ 57	19	10
16	18	18	26	57	49	5/	<u> </u>	17
			ng thời thế		and the second			
								1944
			4 4	10	12	11	15	16
Optimal#	9	10	11	12	13	14	15	16
Optimal# Rank	9 W	10 /alsh#	11	12	13	14	15	16
Optimal# Rank 1	9 15	10 /alsh# 17	11 13	12 9	<u>13</u> 6	<u>14</u> 7	<u>15</u> 20	<u>16</u> 11
Optimal# Rank 1 2	9 W 15 22	10 /alsh# 17 16	11 13 10	12 9 22	13 6 23	14 7 10	15 20 24	16 11 19
Optimal# Rank 1 2 3	9 W 15 22 9	10 /alsh# 17 16 6	11 13 10 14	12 9 22 12	13 6 23 19	14 7 10 21	15 20 24 18 7	16 11 19 20
Optimal# Rank 1 2 3 4	9 W 15 22 9 12	10 /alsh# 17 16 6 21	11 13 10 14 26	12 9 22 12 62	13 6 23 19 9	14 7 10 21 9	15 20 24 18 7	16 11 19 20 5
Optimal# Rank 1 2 3 4 5	9 <b>15</b> 22 9 12 18	10 /alsh# 17 16 6 21 5	11 13 10 14 26 58	12 9 22 12 62 54 54	13 6 23 19 9 28 15	14 7 10 21 9 33	15 20 24 18 7 19 52	16 11 19 20 5 36 27
Optimal# Rank 1 2 3 4 5 6	9 <b>X</b> <b>15</b> <b>22</b> 9 12 18 50	10 /alsh# 17 16 6 21 5 19	11 13 10 14 26 58 16 40	12 9 22 12 62 54 52 12	13 6 23 19 9 28 15 24	14 7 10 21 9 33 8 5	15 20 24 18 7 19 52 6	16 11 19 20 5 36 27 50
Optimal# Rank 1 2 3 4 5 6 7	9 <b>15</b> <b>22</b> 9 12 18 50 17	10 /alsh# 17 16 6 21 5 19 53	11 13 10 14 26 58 16 49 20	12 9 22 12 62 54 52 13 47	13 6 23 19 9 28 15 34 20	14 7 10 21 9 33 8 5	15 20 24 18 7 19 52 6	16 11 19 20 5 36 27 50 51
Optimal# Rank 1 2 3 4 5 6 7 8	9 <b>15</b> <b>22</b> 9 12 18 50 17 49 51	10 /alsh# 17 16 6 21 5 19 53 20	11 13 10 14 26 58 16 49 38	12 9 22 12 62 54 52 13 47 42	13 6 23 19 9 28 15 34 29 28	14 7 10 21 9 33 8 5 12 40	15 20 24 18 7 19 52 6 8 32	16 11 19 20 5 36 27 50 51 10
Optimal# Rank 1 2 3 4 5 6 7 8 9	9 <b>W</b> <b>15</b> <b>22</b> 9 12 18 50 17 49 54 54	10 /alsh# 17 16 6 21 5 19 53 20 4	11 13 10 14 26 58 16 49 38 4 <i>E</i>	12 9 22 12 62 54 52 13 47 43 20	13 6 23 19 9 28 15 34 29 38 11	14 7 10 21 9 33 8 5 12 49 42	15 20 24 18 7 19 52 6 8 33 5	16 11 19 20 5 36 27 50 51 10 6
Optimal# Rank 1 2 3 4 5 6 7 8 9 10	9 <b>W</b> <b>15</b> <b>22</b> 9 12 18 50 17 49 54 26 11	10 'alsh# 17 16 6 21 5 19 53 20 4 51 42	11 13 10 14 26 58 16 49 38 4 55 22	12 9 22 12 62 54 52 13 47 43 20 34	13 6 23 19 9 28 15 34 29 38 11 12	14 7 10 21 9 33 8 5 12 49 42 31	15 20 24 18 7 19 52 6 8 33 5 0	16 11 19 20 5 36 27 50 51 10 6 12
Optimal# Rank 1 2 3 4 5 6 7 8 9 10 11	9 15 22 9 12 18 50 17 49 54 26 11 20	10 'alsh# 17 16 6 21 5 19 53 20 4 51 43 42	11 1 3 1 0 14 26 58 16 49 38 4 55 23 44	12 9 22 12 62 54 52 13 47 43 20 34 19	13 6 23 19 9 28 15 34 29 38 11 13 25	14 7 10 21 9 33 8 5 12 49 42 31 29	15 20 24 18 7 19 52 6 8 33 5 9 20	16 11 19 20 5 36 27 50 51 10 6 12 22
Optimal# Rank 1 2 3 4 5 6 7 8 9 10 11 11 12	9 W 15 22 9 12 18 50 17 49 54 26 11 36 5	10 'alsh# 17 16 6 21 5 19 53 20 4 51 43 42 11	11 13 10 14 26 58 16 49 38 4 55 23 44 17	12 9 22 12 62 54 52 13 47 43 20 34 18 50	13 6 23 19 9 28 15 34 29 38 11 13 25 55	14 7 10 21 9 33 8 5 12 49 42 31 29 28	15 20 24 18 7 19 52 6 8 33 5 9 30 10	16 11 19 20 5 36 27 50 51 10 6 12 22 34
Optimal# Rank 1 2 3 4 5 6 7 8 9 10 11 12 13	9 <b>W</b> <b>15</b> <b>22</b> 9 12 18 50 17 49 54 26 11 36 5 20	10 valsh# 17 16 6 21 5 19 53 20 4 51 43 42 11 12	11 13 10 14 26 58 16 49 38 4 55 23 44 17 21	12 9 22 12 62 54 52 13 47 43 20 34 18 50 61	13 6 23 19 9 28 15 34 29 38 11 13 25 55 55 55	14 7 10 21 9 33 8 5 12 49 42 31 29 28 19	15 20 24 18 7 19 52 6 8 33 5 9 30 10 50	16 11 19 20 5 36 27 50 51 10 6 12 22 34 52
Optimal# Rank 1 2 3 4 5 6 7 8 9 10 11 12 13 14	9 15 22 9 12 18 50 17 49 54 26 11 36 5 29 10	10 'alsh# 17 16 6 21 5 19 53 20 4 51 43 42 11 18 61	11 13 10 14 26 58 16 49 38 4 55 23 44 17 31 27	12 9 22 12 62 54 52 13 47 43 20 34 18 50 61 20	13 6 23 19 9 28 15 34 29 38 11 13 25 55 55 52 62	14 7 10 21 9 33 8 5 12 49 42 31 29 28 18 14	15 20 24 18 7 19 52 6 8 33 5 9 30 10 50 12	16 11 19 20 5 36 27 50 51 10 6 12 22 34 52 9
Optimal# Rank 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	9 <b>W</b> <b>15</b> <b>22</b> 9 12 18 50 17 49 54 26 11 36 5 29 16 55	10 'alsh# 17 16 6 21 5 19 53 20 4 51 43 42 11 18 61 42	11 1 3 1 0 14 26 58 16 49 38 4 55 23 44 17 31 37 62	12 9 22 12 62 54 52 13 47 43 20 34 18 50 61 30 22	13 6 23 19 9 28 15 34 29 38 11 13 25 55 52 62 44	14 7 10 21 9 33 8 5 12 49 42 31 29 28 18 18 14 46	15 20 24 18 7 19 52 6 8 33 5 9 30 10 50 12 44	16 11 19 20 5 36 27 50 51 10 6 12 22 34 52 8 19

Table 3.3Similarity Relation Between OptimalFunctions and Walsh Functions

#### 3.3 Infinite Clipped Optimal Function Approach

If one studies the Walsh functions more carefully, it is found that although the Walsh functions approximate the optimal functions in the sense of structure similarity, they do distort some of the spectral spacing information in the optimal functions. The axis crossing separation in the optimal functions is a relatively irregular pattern, while it is quite regular in the Walsh functions.

One way that can be applied to avoid this information loss is to use the infinite clipped optimal functions as spectral features. The infinite clipped optimal function approach preserves the zero-crossing information in the optimal functions which is hypothesized to contain important spectral information that can be used for classification.

Furthermore, the Walsh function approach is less flexible than the infinite clipped optimal function approach since the spectral features using the Walsh functions tend to be fixed for all analysis situations; while, on the other hand, the infinite clipped optimal function approach does give some degree of adaptability. Figure 3.4 shows the infinite clipping versions of the first 6 eigenfunctions for data set K2.

The infinite clipped optimal functions, derived from the signs of the optimal functions, are then used as spectral features (i.e., basis functions) to linearly transform the high dimensional multispectral data to the ground station for further processing.

32





### 3.4 Overlapping Band Feature Selection Algorithm

The overlapping band feature selection algorithm originates from the inherent overlapping property of the optimal functions. This property suggests that overlapping bands might be even more powerful for spectral feature design. The idea of this algorithm is to find the locations of the important spectral bands without imposing the additional restriction that the bands be non-overlapping. The basic procedures used are very similar to those in the non-overlapping band feature selection algorithm. In the non-overlapping band feature selection algorithm. In the non-overlapping band feature selection algorithm in order to extract the information of the important spectral bands; while in this overlapping case, the infinite clipping procedure is applied to each individual eigenfunction.

The first step is to find the band edges of each individual eigenfunction. Table 3.4 shows part of the results for data set K2. In Table 3.4, comparing to Figure 1.2, it is found that there are 3 important bands for the first eigenfunction, 4 for the 2nd one, 8 for the 3rd one, etc.

It should be noted that the band features derived in this way are not all linearly independent. For example, the first and second band feature from the second eigenfunction, that is, 0.40-0.66  $\mu$ m and 0.66-1.28  $\mu$ m, are linearly dependent on the first band feature from the first eigenfunction (0.40-1.28  $\mu$ m). Another example is the identical band features (1.48-1.78 and 1.98-2.40  $\mu$ m) derived from the first 5 eigenfunctions. Indeed, these repeated bands and the bands which are linearly dependent on the previously selected bands can not

be used as spectral features since linearly dependent features will result in singular class covariance matrix.

EigenVector#		2	3
BAND 1 2 3 4 5 6 7	0.40 - 1.28 1.48 - 1.78 1.98 - 2.40	0.40 - 0.66 0.66 - 1.28 1.48 - 1.78 1.98 - 2.40	0.40 - 0.94 0.94 - 1.00 1.00 - 1.02 1.02 - 1.12 1.12 - 1.16 1.16 - 1.28 1.48 - 1.78 1.98 - 2.40
<u> </u>			
EigenVector#	4	5	6
BAND 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	0.40 - 0.92 0.92 - 1.26 1.26 - 1.28 1.48 - 1.78 1.98 - 2.40	0.40 - 0.70 0.70 - 0.92 0.92 - 0.96 0.96 - 1.06 1.06 - 1.28 1.48 - 1.78 1.98 - 2.40	0.40 - 0.44 0.44 - 0.50 0.50 - 0.52 0.52 - 0.66 0.66 - 0.84 0.92 - 0.94 0.94 - 1.00 1.00 - 1.04 1.04 - 1.12 1.12 - 1.28 1.48 - 1.64 1.64 - 1.78 1.98 - 2.20 2.20 - 2.40

Table 3.4	Linearly	Dependent	Bands	Found	by	Overlapping
	Band Fea	ature Select	ion Alg	orithm	for	Data Set K2

An algorithm is developed to automatically choose the linearly independent bands from the first 6 eigenfunctions. Table 3.5 shows the result. Basically, this algorithm checks the rank of the matrix consisting of the bands

derived in Table 3.4. First, the linearly dependent bands in Table 3.4 are ranked from the widest to the narrowest. Then, starting from the widest band, this algorithm checks the matrix rank. If the rank is less than the total number of the band features, the band features in the matrix are linearly dependent, the widest linearly dependent band in the matrix is then eliminated from the set. On the other hand, if the rank is equal to the total number of the band features, increase the matrix rank by one and test the next widest band.

The procedure used in the above overlapping band feature selection algorithm can find the largest set of smallest bands that are linearly independent. This procedure can be summarized as follows :

- (1) Find the band edges of each individual eigenfunction
- (2) Rank these linearly dependent bands from the widest to the narrowest, then set rank n = 1
- (3) Starting from the widest band, check the rank of the feature matrix
- (4) If the rank is less than the total number of the bands, eliminate the widest linearly dependent band in the matrix, then go to step (3) to test the next widest band;
- (5) If the rank is equal to the total number of the bands, increase nby 1, then go to step (3) to test the next widest band
- (6) Set up the final feature set

Band	wavelength (µm)
1	0.70 - 0.92
2	1.98 - 2.20
3	2.20 - 2.40
4	0.66 - 0.84
5	1.48 - 1.64
6	0.52 - 0.66
7	1.64 - 1.78
8	1.16 - 1.28
9	0.96 - 1.06
10	1.04 - 1.12
11	0.94 - 1.00
12	0.44 - 0.50
13	1.12 - 1.16
14	0.92 - 0.96
15	0.40 - 0.44
16	1.00 - 1.04
17	1.00 - 1.02
18	1.26 - 1.28
19	0.50 - 0.52
20	0.92 - 0.94

# Table 3.5 Linearly Independent Bands Found by Overlapping Band Feature Selection Algorithm for Data Set K2

### 3.5 Experimental System

In order to process the data in a digital computer, the spectral reflectance function  $X(\lambda)$ , the weight function  $W(\lambda)$ , the optimal basis function  $\Phi_i(\lambda)$  and the sequence of the optimal basis functions  $\Phi(\lambda)$  are represented by their discrete approximations, vector **X**, diagonal matrix **W**, basis vector  $\Phi_i$  and the matrix  $\Phi$  respectively.

An experimental software system has been set up to test the four approaches developed in the previous sections. This system has been implemented on IBM 3083 computer. A collection of field data consisting of spectral sample functions on three dates from Williams County, ND, and three dates from Finney County, KS, was available from the field measurement library at Purdue/LARS. The spectral functions were sampled at 0.02  $\mu$ m over the range 0.4 to 2.4  $\mu$ m, therefore, the dimensionality is 100.

The optimal features are found numerically by estimating the covariance matrix from the sample functions. Maximum likelihood estimates of the mean and covariance matrix are given [34] by

$$\mathbf{M}_{\mathbf{X}} = \mathbf{E}(\mathbf{X}) \approx \overline{\mathbf{X}} = \frac{1}{N_{s}} \sum_{i=1}^{N_{s}} \mathbf{X}_{i}$$
(3.1)

and

$$\mathbf{K}_{\mathbf{x}} = \frac{1}{\mathbf{N}_{\mathbf{s}} - 1} \sum_{i=1}^{\mathbf{N}_{\mathbf{s}}} (\mathbf{X}_{i} - \overline{\mathbf{X}}) (\mathbf{X}_{i} - \overline{\mathbf{X}})^{\mathsf{T}}$$
(3.2)

where  $N_s$  is the number of the sample functions and  $X_i$  is the i<sup>th</sup> sample vector. The covariance matrix is then used to solve the discrete form of the generalized Karhunen Loeve Equation [14,15] :

$$\mathbf{K} \mathbf{W} \Phi = \Phi \Gamma$$
(3.3)

where the  $\Phi$ ,  $\Gamma$  and W are the eigenvectors, eigenvalues and the weight matrix, respectively. The solutions of the equation are the optimal features.

In order to find appropriate non-overlapping bands used in feature design, the non-overlapping band feature selection algorithm is applied to the average of the first few eigenvectors. Three cases were studied, tests using the first 6, 12 or 24 eigenvectors in the algorithm. For the illustrative example shown in section 3.1, the second case is considered.

For overlapping band features, the infinite clipping procedure is applied to each individual eigenfunction. In this preliminary test the first 6 eigenfunctions from each of the 6 data sets are used. The locations of the important spectral bands are then extracted. After applying the overlapping band feature selection algorithm to the spectral bands derived above, the desired linearly independent (L.I.) band features are found.

The bands found by the above two algorithms, the Walsh functions or the infinite clipped optimal features developed from the structure similarity property are then used as spectral features to perform the linear transformation on the data sets.

$$\mathbf{y}_{\mathbf{i}} = \boldsymbol{\Phi}_{\mathbf{i}}^{\mathsf{T}} \mathbf{W} \mathbf{X}$$
(3.4)

In order to test the spectral features thus determined, the probability of correct classification is estimated using them. To do so, the class-conditional statistics are first computed using the transformed data. An algorithm based on the maximum likelihood estimator [34] is then applied, where the class conditional statistics are assumed to be multivariate Gaussian.

#### 3.6 Preliminary Results

After applying the N.O.L. band feature selection algorithm to the average of the first 6, 12 or 24 eigenvectors of the six test data sets, the band edges are found. Table 3.6 shows the results for the data set K2 for three different number of eigenvectors. These three feature sets are named as proposed sensor C1, C2 and C3 respectively. For brevity, they are denoted PC1, PC2 and PC3. On the other hand, the O.L. band feature selection algorithm is applied to the first 6 eigenfunctions, the result of the first 16 linearly independent bands is shown in Table 3.7 for data set K2.

Furthermore, the probabilities of correct classification using Landsat (LS) MSS bands, Thematic Mapper (TM) bands and the two sensors proposed in Wiersma's work (PA and PB) [14,15] are also computed here. Table 3.8 shows the band edges associated with each sensor [15].

Band	PC1	PC2	PC3
1	0.40 - 0.68	0.40 - 0.68	0.40 - 0.66
2	0.68 - 0.84	0.68 - 0.90	0.66 - 0.80
3	0.84 - 0.90	0.90 - 0.92	0.80 - 0.88
4	0.90 - 0.96	0.92 - 0.94	0.88 - 0.94
5	0.96 - 1.00	0.94 - 1.00	0.94 - 1.00
6	1.00 - 1.06	1.00 - 1.06	1.00 - 1.04
7	1.06 - 1.12	1.06 - 1.12	1.04 - 1.16
8	1.12 - 1.28	1.12 - 1.26	1.16 - 1.26
9	1.48 - 1.74	1.26 - 1.28	1.26 - 1.28
10	1.74 - 1.78	1.48 - 1.78	1.48 - 1.54
11	1.98 - 2.40	1.98 - 2.40	1.54 - 1.64
12			1.64 - 1.74
13			1.74 - 1.78
14			1.98 - 2.20
15			2.20 - 2.26
16			2.26 - 2.40

## Table 3.6 Bands Found by Non-Overlapping BandFeature Selection Algorithm for Data Set K2

	(in the second based of the second
Band	wavelength (µm)
1	0.70 - 0.92
2	1.98 - 2.20
3	2.20 - 2.40
4	0.66 - 0.84
5	1.48 - 1.64
6	0.52 - 0.66
7	1.64 - 1.78
8	1.16 - 1.28
9	0.96 - 1.06
10	1.04 - 1.12
11	0.94 - 1.00
12	0.44 - 0.50
13	1.12 - 1.16
14	0.92 - 0.96
15	0.40 - 0.44
16	1.00 - 1.04

Table 3.7Bands Found by Overlapping Band FeatureSelection Algorithm for Data Set K2

Figures 3.5 to 3.10 are the classification performance comparisons of the optimal functions (Optimal), Walsh functions (Walsh) and the infinite clipped optimal functions (Clipped) for the 6 data sets. Figure 3.11 to 16 are the comparisons of the LS, TM, Wiersma's proposed sensor PA, non-overlapping band features (NOL) derived from the first 24 eigenfunctions (i.e., PC3), overlapping band features (OL), Walsh functions, infinite clipped optimal functions and optimal functions for the 6 preliminary test data sets. From the implementation point of view, since there are only two values (+1, -1) for the Walsh functions and three values (+1, -1, 0) for the infinite clipped optimal functions, it can be concluded from Figures 3.5 to 3.16 that representing the optimal features using their infinite clipping versions or using the first 16 Walsh functions produces the more practical features used for classification which

provide a classification accuracy quite near that of optimal features. The classification performances estimated for the above sensors are shown in Table 3.9, where PC1, PC2 and PC3 represent the sensors derived from N.O.L. band feature selection algorithm using the first 6, 12 and 24 eigenvectors as their input respectively; Optimal, Walsh and Clipped stand for the sensors using the first 16 optimal functions, the first 16 Walsh functions and the first 16 infinite clipped optimal functions as spectral features respectively.

Band	LS	ТМ	PA	PB
1	0.50-0.60	0.45-0.52	0.42-0.54	0.42-0.66
2	0.60-0.70	0.52060	0.56-0.66	0.68-0.70
3	0.70-0.80	0.63-0.69	0.68-0.70	0.72-0.92
4	0.80-1.10	0.76-0.90	0.72-0.90	0.94-1.04
5		1.55-1.75	0.92-1.00	1.06-1.10
6		2.08-2.35	1.02-1.30	1.12-1.30
7			1.52-1.74	1.52-1.74
8			1.96-2.40	1.96-2.40

Table 3.8 Band Edges of Landsat MSS, TM, PA and PB Sensors

Table 3.9 Probability of Correct Classification for 6 Data Sets

SENSOR	K1	K2	К3	N 1	N 2	N 3
LS	0.90	0.78	0.85	0.77	0.83	0.96
ТМ	0.92	0.79	0.93	0.89	0.95	0.99
PA	0.94	0.86	0.95	0.92	0.96	0.99
PB	0.94	0.85	0.94	0.89	0.96	0.96
PC1	0.94	0.87	0.96	0.92	0.97	0.99
PC2	0.96	0.88	0.97	0.94	0.97	0.99
PC3 (NOL)	0.96	0.94	0.98	0.96	0.98	0.99
ÔL /	0.97	0.94	0.98	0.97	0.99	0.99
Walsh	0.98	0.95	0.98	0.95	0.98	0.99
Clipped	0.98	0.97	0.99	0.97	0.99	0.99
Optimal	0.98	0.97	0.98	0.97	0.99	0.99



Figure 3.5 Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set K1



Figure 3.6 Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set K2



Figure 3.7 Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set K3



Figure 3.8 Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set N1



Figure 3.9 Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set N2



Figure 3.10 Performance Comparison of Optimal, Infinite Clipped Optimal and Walsh Functions for Data Set N3

w. . .

4.5



Figure 3.11 Performance Comparison for Data Set K1



Figure 3.12 Performance Comparison for Data Set K2



Figure 3.13 Performance Comparison for Data Set K3







Figure 3.15 Performance Comparison for Data Set N2



Figure 3.16 Performance Comparison for Data Set N3

#### 3.7 Selection of the Best On-Board Preprocessing Scheme

From Table 3.9 and Figures 3.5 to 3.16, it is seen that the four approaches developed in this research, two based on the " shape " of the optimal features and the other two from their "structure" similarity with the optimal functions, are feasible ways for feature design.

The fundamental objective of this research is to develop an objective and practical spectral feature design technique for high dimensional multispectral data. There are two important factors, simplicity and effectiveness, which must be considered in this respect.

First of all, from simplicity point of view, the overlapping band feature selection algorithm is harder to perform than the other three because of the existence of linear dependence problem. In order to find appropriate overlapping band features, we have to check the rank of the matrix for each newly selected band. This procedure needs more time than the other three approaches. However, its classification performance [ referring to Figure 3.11 to 3.16 ] does not indicate much advantage over the other three, especially the infinite clipped optimal function approach.

For example, Figure 3.11 and 3.12 show that for Kansas September and Kansas May data the performances of the overlapping band feature selection algorithm are the 3rd best among the four techniques. The infinite clipped optimal function approach and the Walsh function approach have better performances than that of the overlapping band feature selection algorithm. Figure 3.13 to 3.16 indicate that the performances of the overlapping band feature selection algorithm are never better than those of the infinite clipped optimal function approach. Therefore, from simplicity point of view, the overlapping band feature selection algorithm would not be used in this thesis as the best technique for the final data preprocessing system.

On the other hand, from effectiveness point of view, referring to Table 3.9 and Figure 3.5 to 3.16 again, it is shown that the infinite clipped optimal transform has better performance than the Walsh transform and the non-overlapping band feature selection algorithm.

For instance, Figure 3.5 to 3.10 indicate that the infinite clipped optimal features have better classification accuracy than the Walsh features for all the six preliminary test data sets in Kansas and North Dakota. Figure 3.11 to 3.16 show that the infinite clipped optimal features perform better than the non-overlapping band features for all the 6 test data sets except for North Dakota August data (Figure 3.16) where these two techniques have the same performance.

Therefore, from simplicity and effectiveness point of view, the infinite clipped optimal transform is chosen to be the best scheme in the data preprocessing stage of the spectral feature design system.

The processing up to this point, consisting of the optimal features calculation, the infinite clipping, and the data transform is based solely upon the ensemble statistics of the field data. Additional a priori knowledge that might be used to improve the performance is the class statistics of the scene. The objective is then to find the best features under the criterion of maximal class separability.

### 3.8 Canonical Analysis and Ground Station Data Processing.

Canonical Analysis is a technique that can be used to find the optimal features under a maximal separability criterion [36-41]. Unlike principal component analysis, which is based on the global covariance matrix of the full data set, canonical analysis utilizes the class structure of the data. The advantage of canonical analysis is its ordering property on the separability measure. By using the features derived from canonical analysis to further process the received transformed data, the classification performance should, therefore, be improved.

Let  $M_i$  and  $S_i$  be the i<sup>th</sup> class mean vector and covariance matrix of a data set with L classes. In canonical analysis one first finds the within-class scatter and the among-class scatter matrices  $S_w$  and  $S_a$  respectively :

$$S_{w} = \sum_{i=1}^{L} \frac{(N_{i} - 1)}{N_{s}} * S_{i}$$
 (3.5)

where  $N_i$  is the number of samples of the i<sup>th</sup> class data and  $N_s$  is the total number of samples of the ensemble. And,

$$S_a = \frac{1}{L} \sum_{i=1}^{L} (M_i - M_o) (M_i - M_o)^T$$
 (3.6)

where Mo is the global mean, given by

$$M_o = \sum_{i=1}^{L} \frac{N_i}{N_s} * M_i$$
 (3.7)

The within class scatter matrix,  $S_w$ , is an average quantity that describes how closely the samples are distributed around their class means while the among class scatter matrix,  $S_a$ , is a quantity measuring the average degree of closeness between the ensemble mean and each class mean. The optimally separable feature is a feature such that  $S_w$  is minimized and  $S_a$  is maximized after the transformation. Define a quantity r and let the desired feature be vector **d**. Then the objective is to find the r and **d** that result in maximal class separability. That is,

$$\mathbf{\dot{T}} = \frac{\mathbf{d}^{\mathsf{T}} \mathbf{S}_{\mathsf{a}} \mathbf{d}}{\mathbf{d}^{\mathsf{T}} \mathbf{S}_{\mathsf{w}} \mathbf{d}}$$
(3.8)

must be maximized. The ratio of variances in the new space is maximized by the selection of feature **d** if,

$$\frac{\partial \mathbf{r}}{\partial \mathbf{d}} = \mathbf{0} \tag{3.9}$$

The above equation can be reduced to

$$(S_a - r * S_w) * d = 0$$
 (3.10)

which is called a generalized eigenvalue equation and must be solved now for the unknown r and **d**. The first canonical axis will be in the direction of **d**, and r will give the associated ratio of among-class to within-class variance for that axis. The development to this stage is usually referred to as discriminant analysis. One more step is included in the case of canonical analysis where the derived canonical features are normalized with respect to the within class scatter matrix. That is,

$$\mathbf{D}^{\mathsf{T}} \star \mathbf{S}_{\mathsf{W}} \star \mathbf{D} = \mathbf{I} \tag{3.11}$$

where **D** is the matrix of canonical features **d**. This says that the within class scatter matrix after the transformation must be the identity matrix. In other words, after transformation, the classes should appear spherical.

## CHAPTER IV RESULTS AND DISCUSSIONS

In the previous chapter, we have introduced the four spectral feature design techniques developed in the course of this research. Six preliminary test data sets in Kansas and North Dakota were used to test the schemes. From a simplicity and effectiveness point of view, the infinite clipped optimal transform is chosen as the better means for data preprocessing. Furthermore, canonical analysis is applied to the above received transformed data on the ground station to achieve the maximal class separability. In this chapter, both the vegetation and the soil data will be used to find the classification performance for the final spectral feature design system. The spectral range for the vegetation data is from 0.4 µm to 2.4 µm with resolution 0.02 µm while the range for the soil data is from 0.45 µm to 2.45 µm with resolution 0.01 µm. Therefore the dimensionality for the vegetation data and the soil data is 100 and 200 respectively. The final results of these data will be presented in section 4.1 and 4.2. Moreover, due to the limited sample size of the data set to estimate the covariance matrix, different degree of Hughes phenomenon occurs in some of the one-day Kansas and North Dakota vegetation data sets as well as in all soil data sets. This effect will be discussed in section 4.3.

#### 4.1 Vegetation Data

Four sets of multitemporal multispectral data collected in Kansas, North Dakota, Iowa and South Dakota are acquired to test the proposed spectral feature design system. Table 4.1 show the species, the dates on which the data were collected, and the total numbers of sample functions for each information class. In Table 4.1, the numbers appearing in the parentheses are the total numbers of sample functions collected for that class. Furthermore, W.Wheat and S.Wheat stand for winter wheat and spring wheat respectively.

Figure 4.1 to 4.6 show the probability of correct classification, Pc, using the optimal features, infinite clipped optimal features and features that are derived from infinite clipped optimal transform and canonical analysis for the six preliminary test data sets. These 6 data sets are part of the multitemporal data in Kansas and North Dakota (referring to Table 1.1 and Table 4.1). Each one of them consists of the sample functions collected on one single date and has 3 informational classes. The results indicate that using the first 16 infinite clipped versions of the optimal functions, 95% classification accuracy can be achieved.

Another important point is the occurrence of Hughes phenomenon [42,43] shown in Figure 4.1 to 4.4. It says that for data set K1, K2, K3 and N1, increasing the computational complexity [11] does not always increase the classification performance. For example, Figure 4.1 shows that canonical analysis improves the accuracy for the first 3 features, but it does not help beyond this complexity for data set K1. Figure 4.2 to 4.4 show that canonical analysis can only have better performance for the first 4 features for data sets K2, K3 and N1 respectively.

For data set N2 and N3, it is found in Figure 4.5 and 4.6 that Hughes phenomenon does not occur, and the classification performance using the features derived from infinite clipped optimal transform and canonical analysis is always better than those of the optimal features and the infinite clipped optimal features. It is also shown that only 2 features are needed to have about 94% and 99% classification accuracy for these 2 data sets respectively.

Figure 4.7 and 4.8 show the results for Kansas and North Dakota multitemporal data. Each one has 9 information classes collected on 3 different dates from 1976 to 1977. The results indicate that canonical analysis improves the accuracy by about 15% to 25% for the first feature and about 1% for the first 16 features. Figure 4.9 is the results of Kansas and North Dakota combined data with 18 information classes. It is used to show the robustness property of this spectral feature design system. The results show that the technique is not overly sensitive for spatially and temporally combined data.

Figure 4.10 and 4.11 are the results for 25-class lowa and 42-class South Dakota multi-temporal data. They are used to show the capability of this spectral feature design system for complex data sets. It can be seen that the system is very successful in this respect.

# Table 4.1: Vegetation Data Sets.Numbers in the parenthesis are the total numbers of samples.

#### Kansas Vegetation Data Set : 9 classes

9/28/76	5/3/77	6/26/77
W.Wheat (141)	W.Wheat ( 658 )	W.Wheat (677)
Summer Fallow (414)	Summer Fallow (211)	Summer Fallow (643)
Sorghum (277)	Unknown Class (682)	Sorghum (157)

#### North Dakota Vegetation Data Set : 9 classes

5/8/77	6/29/77	8/4/77
S.Wheat (664)	S.Wheat (787)	S.Wheat (931)
Summer Fallow (437)	Summer Fallow (291)	Summer Fallow (330)
Pasture (164)	Pasture (161)	Pasture(183)

Iowa Vegetation Data Set : 25 classes collected on 9 different dates of 1979;

5/15/79	5/23/79	6/11/79	6/29/79	7/16/79	7/17/79	8/30/79	10/25/79	11/2/79
Corn	Corn	Corn	Corn	Corn	Corn	Corn	Corn	Corn
(514)	(517)	(621)	(610)	( 437 )	(190)	(650)	( 435 )	( 393 )
	Soybeans							
	(36)	(517)	(485)	(377)	(172)	(568)	(417)	(267)
Oats	Oats	Oats	Oats	Oats	Oats	Oats	Oats	
(41)	(32)	(45)	(21)	(22)	(25)	(42)	(44)	

South Dakota Vegetation Data Set : 42 classes collected on 6 different dates of 1978 and 1979

9/21/78	10/26/78	6/1/79	6/21/79	7/25/79	8/11/79
Pasture (225)	Pasture (217)				
Alfalfa (61)	Alfalfa (51)			Alfalfa (45)	Alfalfa (42)
W.Wheat (292)	W.Wheat (393)				
S.Wheat (469)	S.Wheat(441)	S.Wheat(118)	S.Wheat(121)	S.Wheat (102)	S.Wheat (119)
Barley (82)	Barley (80)	Barley (43)	Barley (44)	Barley (66)	Barley (69)
Oats (182)	Oats (88)			Oats (89)	Oats (76)
IdleLand (63)					
Sorahum (103)	Sorghum (88)			Sorhgum (78)	Sorhgum (96)
Sunflower (39)	Sunflower (41)			Sunflower (53)	Sunflower (107)
Corn (39)	Corn (32)			Corn (147)	Corn (154)
	Millet (26)			Millet (39)	Millet (28)
				Safflower (24)	Safflower (19)


Figure 4.1 Classification Performance for Data Set K1



Figure 4.2 Classification Performance for Data Set K2



Figure 4.3 Classification Performance for Data Set K3



Figure 4.4 Classification Performance for Data Set N1



Figure 4.5 Classification Performance for Data Set N2



Figure 4.6 Classification Performance for Data Set N3



Figure 4.7

# Classification Performance for Kansas Multitemporal Data Set



Figure 4.8 Classification Performance for N. Dakota Multitemporal Data Set



Kansas and North Dakota Combined Data





Figure 4.10 Classification Performance for Iowa Multitemporal Data Set



Figure 4.11 Classification Performance for S. Dakota Multitemporal Data Set

### 4.2 Soil Data

In addition to the above FSS vegetation data, a soil data base with 571 soil samples collected by Eric Stoner [45] in 1979 was acquired to test the system. The soil reflectance functions were measured by an EXOTECH-C spectrometer in the laboratory. In this research, five data sets grouped by soil order, organic matter content #1, organic matter content #2, Iron oxide content and soil texture [46-50] were formed respectively to test the spectral feature design system. They are designated as data sets SO, OM1, OM2, IO and ST respectively. It should be noted that the same soil samples are used in the data sets, but they are only grouped differently into classes. The soil data set designated as organic matter content #1 is from the soil orders Mollisol and Alfisol [48] only, while the soil data set designated as organic matter content #2 is from all soil orders. These 5 soil data sets are shown in Table 4.2

Table 4.2(a) shows the 10 soil orders in American Soil Taxonomy [48]. Since the total numbers of sample functions for Spodosol, Vertisol, Histosol and Oxisol are very limited, in this research, these soils are not used to form the data set SO. Only the data in the first 6 soil orders are included in SO. Table 4.2(b), (c) and (d) indicate the ranges of organic matter content #1, organic matter content #2 and iron oxide content respectively. Six classes are chosen in these 3 data sets: OM1, OM2 and IO. Table 4.2(e) shows the 6 soil texture classes used in data set ST where some of the classes consist of more than one soil texture group. For example, class 1 in data set ST includes clay and silty clay; class 2 includes sandy clay loam, clay loam and silty clay loam; etc.

The results of these 5 soil data sets are shown in Figure 4.12 to 4.16. Taking a general view of these graphs, it is found that the cumulative performances of these soil data sets are less like a standard error function compared to those found in vegetation data sets (referring to Figure 4.1 to 4.11). The reason for this is that the total numbers of sample functions used to estimate the covariance matrices in the soil data sets are very limited, from a little more than the dimensionality in data set OM1, that is, 255 sample functions with dimensionality 200, to about 2.5 times the dimensionality in SO, OM2, IO and ST, that is about 500 sample functions for each data set; while on the other hand at least 8 times the dimensionality are available in the vegetation data sets. For example, the smallest data set K1 has 832 sample functions with dimensionality 100 and data sets other than K1 have more than 1000 sample functions to estimate the covariance matrix. Therefore, the estimates of the covariance matrices for the vegetation data sets are likely to be much more accurate than those for the soil data sets. The subsequent Gaussian model thus becomes more valid for the vegetation data and the cumulative classification curves are more like a standard error function.

Furthermore, Figure 4.12 to 4.16 show that the infinite clipped optimal functions are very effective to extract the information for soil classification. For instance, Figure 4.12 to 4.13 indicate that using the first 16 infinite clipped optimal functions, over 90% accuracy can be achieved while Figure 4.14 to 4.16 tell that over 85% accuracy is obtained. Due to the limited sample size for each of the soil data sets, different degrees of the Hughes phenomenon occur. Figure 4.12 to 4.14 show that canonical analysis improves the performance for the first 5 features while Figure 4.15 to 4.16 show that improvement is possible up to the first 7 features.

### Table 4.2 Soil Data Sets :

#### (a) SO by Soil Order Sample size for the first 6 classes : 479 # of Sample Functions Order Name class # 154 Mollisol 1 113 Alfisol 2 78 Entisol 3 52 Aridisol 4 45 5 Ultisol 37 Inceptisol 6 30 Spodosol 7 Vertisol 11 8 8 Histosol 9 11 Oxisol 10 Unclassified 32 11

(b) OM1 by Organic #1 Soil from Mollisol and Alfisol only. Sample size : 255

Class # Organic Matter Range %		# of Sample Functions	
	0.11 ~ 1.5	51	
2	1.5 ~ 2.0	54	
3	2.0 ~ 2.5	33	
4	2.5 ~ 3.5	45	
5	3.5 ~ 5.0	39	
6	5.0 ~ 10.12	33	

### (c) OM2 by Organic #2 Soil from all orders. Sample size : 514

Son nom an orders. Campie elect et				
Class #	Organic Matter Range %	# of Sample Functions		
1	0.08 ~ 1.0	82		
2	1.0 ~ 2.0	135		
3	2.0 ~ 3.0	120		
4	3.0 ~ 4.0	54		
5	4.0 ~ 6.0	59		
6	6.0 ~ 84.79	64		

### Table 4.2, continued

### (d) IO by Iron Oxide Content Sample size : 467

Class #	Iron Oxide Range %	# of Sample Functions
	0.02 ~ 0.4	102
2	0.4 ~ 0.6	73
3	0.6 ~ 0.8	69
4	0.8 ~ 1.2	105
5	1.2 ~ 1.6	52
6	1.6 ~ 25.6	66

### (e) ST by Soil Texture Total sample size : 483 excluding the unclassified

Class #	Soil Texture Group/Groups	# of Sample Function	Class Sample Size
	Clay Silty Clay	19 21	40
2	Sandy Clay Loam Clay Loam Silty Clay Loam	6 25 32	63
3	Coarse Sand Large Coarse Sand Sand Large Sand Large Fine Sand Fine Sand	3 6 13 16 18 20	76
4	Coarse Sandy Loam Very Fine Sandy Loam Sandy Loam Fine Sandy Loam	5 12 24 52	93
5	Loam	68	68
6	Silt Silt Loam	4 139	143
7	Unclassified	88	88



Soil Data Set Grouped by Soil Order

Figure 4.12 Classification Performance for Soil Data Grouped by Soil Order



Figure 4.13 Classification Performance for Soil Data Grouped by Organic #1



### Soil Data Set Grouped by Organic Matter#2





Figure 4.15 Classification Performance for Soil Data Grouped by Iron Oxide



Soil Data Set Grouped by Soil Texture

Figure 4.16 Classification Performance for Soil Data Grouped by Soil Texture

### 4.3 Hughes Phenomenon

In 1968, Hughes [42] showed theoretically that the mean recognition accuracy for the statistical pattern classifiers did not always increase as the measurement complexity increased so long as the number of training samples was fixed and finite. This result was experimentally demonstrated in a remote sensing context by Fu, Landgrebe and Phillips [43] in 1969. The conclusion of these investigations was that for a fixed number of training samples, there is an optimal measurement complexity. More complexity is undesirable from the standpoint of expected classification accuracy.

Kalayeh, Muasher and Landgrebe [51,52] developed a criterion to predict the occurrence of the Hughes phenomenon. They suggested that a number of sample functions equal to about 8 to 10 times the dimensionality must be available for the ensemble in order to avoid the Hughes phenomenon.

In this section, four experiments are described to show that the Hughes phenomenon did occur in the data sets with limited training samples. The data sets K1 and N2 were chosen for this purpose because K1 has the least training samples (referring to Table 1.1) among all vegetation data sets and N2 (referring to Figure 4.5) indicated some possibility for the occurrence of the Hughes phenomenon. Tables 4.3(a) to (d) show the data used for these 4 experiments and Figures 4.17 to 4.20 show the results. In the above tables and figures, K1H and N2H are the data sets with about one half of the original training samples while K1Q and N2Q represent those with approximately one quarter of the training samples.

Figure 4.17 and 4.18 show that for data set K1, the Hughes phenomenon has occurred (referring to Figure 4.1). If the size of the training samples is reduced to half or even to quarter, the effect of this phenomenon becomes more and more serious. On the other hand, for data set N2, there is no Hughes phenomenon (referring to Figure 4.5). If the size of the training samples becomes one half of the original N2, the Hughes phenomenon might or might not occur. Figure 4.19 indicates that for data set N2, reducing the size of the training samples to approximately one half, that is 630 samples with dimensionality 100, the estimate of covariance matrix is still accurate enough, and the Hughes phenomenon does not occur.

However, if the training size of the data set N2 is reduced to one quarter, the Hughes phenomenon does occur. Figure 4.20 says that the optimal number of features in this data set N2Q with 315 training samples is only 2. The maximal classification accuracy that can be achieved is about 85%. Furthermore, more than 2 features used for classification would not help the performance and in some cases even reduce the accuracy.

The four experiments in this section indicate that for data set K1, more than 832 samples are needed in order to avoid the effect of Hughes phenomenon; on the other hand, for data set N2, 1239 samples are enough to accurately estimate the covariance matrix. From the classification performances of data sets K1, K2, K3 and N1, shown in Figure 4.1 to 4.4, it is suggested that more than 15 times dimensionality sample functions may be required to avoid the effect of the Hughes phenomenon.

## Table 4.3Data Sets Used to Test the Occurrence of<br/>the Hughes Phenomenon :

### (a) Kansas September Data With Half Training Samples : Data Set K1H

K1H	Winter Wheat	Summer Fallow	Grain Sorghum	Total Samples
Training	70	200	140	410
Testing	71	214	137	412
Total	141	414	277	832
			the second s	

### (b) Kansas September Data With Quarter Training Samples : Data Set K1Q

Alex Col

K1Q	Winter Wheat	Summer Fallow	Grain Sorghum	Total Samples
Training	35	100	, 70	205
Testing	106	314	207	627
Total	141	414	277	832

### (c) North Dakota June Data With Half Training Samples : Data Set N2H

N2H	Spring Wheat	Summer Fallow	Natural Pasture	Total Samples
Training	400	150	80	630
Testing	387	141	81	609
Total	787	291	161	1239

### (d) North Dakota June Data With Quarter Training Samples : Data Set N2Q

N2Q	Spring Wheat	Summer Fallow	Natural Pasture	Total Samples
Training	200	75	40	315
Testing	587	216	121	924
Total	787	291	161	1239



Figure 4.17 First Experiment of the Hughes Phenomenon : Data Set K1H



Figure 4.18 Second Experiment of the Hughes Phenomenon : Data Set K1Q



Figure 4.19 Third Experiment of the Hughes Phenomenon : Data Set N2H

8 ( A 2 )



Figure 4.20 Fourth Experiment of the Hughes Phenomenon : Data Set N2Q

Electronic de P

### 4.4 Signal to Noise Ratio Considerations

In the previous sections, the classification results obtained by using the spectral features developed in this research are presented for 100 dimensional FSS vegetation data and 200 dimensional Exotech-C soil data. It is found (referring to Figure 4.1 to 4.16) that about 10 to 1 compression ratio can be achieved while maintaining satisfactory classification accuracy. One question an Earth scientist user of the algorithm may have is that the 10 to 1 downlink data rate reduction is not at a severe cost to the usefulness of the data. Thus, in this section, we will discuss the data volume reduction issue from the Earth scientist point of view, that is, from signal-to-noise ratio considerations.

Weighted Karhunen-Loeve transform rotates the original N-dimensional signal space to a more favorable orientation. This orientation is one in which the source energy is redistributed such that a larger percentage of the energy is distributed over fewer coordinates. Table 4.4 and Figure 4.21 show how the source energy is redistributed over the first 25 transformed coordinates for 100 dimensional vegetation data set K2.

In Table 4.4, the first row shows that the magnitude of the total source energy is 3497, which is the sum of all eigenvalues; Further, the mean square representation error (MSE) and percent mean square representation error (%MSE) are 3497 and 100% respectively if 'none' of the optimal feature is used to transform the data. The second row indicates that the magnitude of the first eigenvalue is 2779.8; If the first optimal feature is used to transform the data, the representation error and percent representation error will be 717 and 20.5% respectively, that is, the first transformed coordinate contains about 79.5% source energy in it. Similarly, it can be found that using the first 2 optimal features, about 97.5% of the total source energy can be preserved, and using the first 10 optimal features to transform the data in the measurement space, the percent mean square representation error, that is 0.17%, is indeed negligible. Figure 4.21 shows graphically how fast the representation error can be reduced by using the first few optimal features. It should be noticed that the representation error (MSE) is plotted in logarithmic scale.

The practical values of the signal to noise ratio in a typical remote sensing system are from 50 to 200 in most of the 0.4 to 2.5 µm spectrum range [1]. This indicates that the maximal noise level in the system is only 1/50, that is, 2%. Since using the first 10 optimal features derived from the Weighted K-L transform preserves almost all the signal energy in the original measurement space; Further, the representation error level is 0.17% which is much lower than the noise level in the system. Hence, the effect on the signal to noise ratio due to compression is quite limited even as the signal to noise ratio is down to 20. Therefore, a data volume reduction by a factor of 10 is achieved with essentially no loss of information.

Eigenvalue	Magnitude of Eigenvalue	Mean Square Error	% Mean Square Error
0	3497.0691	3497.0691	100.0000
1	2779.8821	717.1870	20.5082
2	627.0327	90.1543	2.5780
3	39.0218	51.1325	1.4622
4	18.4108	32.7217	0.9357
5	14.0425	18.6792	0.5341
6	4.9193	13.7599	0.3935
7	2.5450	11.2149	0.3207
8	1.8422	9.3727	0.2680
9	1.7561	7.6166	0.2178
10	1.3731	6.2435	0.1785
11	0.8927	5.3508	0.1530
12	0.8225	4.5283	0.1295
13	0.6291	3.8993	0.1115
14	0.4818	3.4175	0.0977
15	0.4498	2.9676	0.0849
16	0.3778	2.5898	0.0741
17	0.3469	2.2429	0.0641
18	0.3266	1.9163	0.0548
19	0.2328	1.6835	0.0481
20	0.2192	1.4643	0.0419
21	0.1696	1.2947	0.0370
22	0.1499	1.1448	0.0327
23	0.1268	1.0181	0.0291
24	0.1174	0.9006	0.0258
25	0.0904	0.8103	0.0232

Table 4.4 Mean Square Representation Error for Data Set K2



Figure 4.21 Mean Square Representation Error for Data Set K2

### CHAPTER V

### CONCLUSIONS AND RECOMMENDATIONS

### 5.1 Conclusions

The fundamental objective of this research is to develop an objective and practical spectral feature design technique for high dimensional multispectral data. In this thesis, four spectral feature design techniques have been developed. Two of them, non-overlapping band feature selection algorithm and overlapping band feature selection algorithm, are derived from the spectral dominancy concept of the optimal functions; the other two, Walsh function approach and infinite clipped optimal function approach, are derived from the spectral similarity concept of the optimal functions. These four approaches have been proved effective for data compression and classification purposes in high dimensional multispectral data.

A comparison among these four techniques indicates that the infinite clipped optimal function approach is the best scheme since the features are easiest to find and their classification performance is the best under the same compression requirement. This technique approximates the spectral structure of the optimal features via infinite clipping and results in transform coefficients which are either +1, -1 or 0. Therefore the necessary processing can be easily implemented on-board the spacecraft by using a set of programmable adders that operate on the grouping instructions received from the ground station. After the preprocessed data has been received, canonical analysis is further used to find the best set of features under the criterion that maximal class separability is achieved

Both vegetation and soil data have been tested in this research. For vegetation data, four sets of multitemporal multispectral vegetation data collected in Kansas, North Dakota, Iowa and South Dakota respectively with 9 to 42 information classes in 1976 to 1979 are used to test the spectral feature design system. One spatially and temporally combined data set is also formed by combining the Kansas and North Dakota Data sets to test the robustness property of the scheme. The results indicate that the system is not overly sensitive to spatial and temporal variation.

Furthermore, a soil data base collected by Eric Stoner in 1979 was also acquired and used to test the system. In this research, five different soil data sets grouped by the soil order, organic content #1, organic content #2, iron oxide content and soil texture are formed. The classification performances are then found. It is shown that soil order, organic content percentage, iron oxide content percentage and soil texture can be delineated and predicted by the proposed technique.

It is concluded that the infinite clipped versions of the first 16 optimal functions derived from the Weighted Karhunen-Loeve Transform have excellent classification performance. Further signal processing by canonical analysis increases the compression ratio while retains the classification accuracy. The overall probability of correct classification of the proposed system is over 90% while providing for a reduced downlink data rate by a factor of 10.

#### 5.2 Recommendations

The spectral feature design system developed in this research has been demonstrated for the FSS vegetation data and the Exotech-C soil data. In the future, it is proposed to test AVIRIS and HIRIS data. The following procedure is recommended :

- (A) Pre-Flight Stage :
  - Collect enough representable samples from all reference sources available, for example, the field data base collected in the past, to form the ensemble of a specific problem (Ground Truth Gathering)
  - (2) Calculate the mean vector and covariance matrix of this ensemble
  - (3) Find the eigenvectors of the covariance matrix
  - (4) Run the spectral feature design system on the ground to find the grouping coefficients ( either +1, -1, or 0 )
- (B) On-Board Preprocessing Stage :
  - (5) Send up these grouping coefficients (instructions) to the spacecraft for on-board data preprocessing
- (C) Post-Flight Stage :
  - (6) Receive the preprocessed low dimensional data
  - (7) Run the spectral feature design system on the ground to find the canonical features
  - (8) Use these canonical features to further transform the received data into the final signal space where the data classification is performed

In this procedure, there are basically 3 processing stages involved : pre-flight stage, on-board preprocessing stage and post-flight stage. The preflight stage, which consists of step 1 to step 4, is used to gather ground truth information, estimate ensemble statistics and find appropriate grouping coefficients from one of the four developed schemes. This stage would be done before the data take by the aids of aerial photography, topographical maps, historical information, field data base collected in the past or other reference sources available. One more comment about this stage is the problem of the sample size, it is suggested from the experience in this research that the total number of samples used to estimate the ensemble statistics needs to be at least 15 times their signal dimensionality in order to accurately estimate the covariance matrix.

The second stage, on-board preprocessing stage, which contains step 5, performs band groupings on board the spacecraft, either summing (+1), subtracting (-1) or omitting (0) bands for each spectral function according to the grouping instructions sent by the ground user. Since this data preprocessing stage would be done on board the spacecraft, from implementation point of view, the algorithm simplicity is then required and important. The spectral feature design system developed in this research makes this simplicity possible. Figure 1.1 shows how the data preprocessing can be implemented on board the spacecraft by a set of programmable adders.

Finally, the post-flight stage, which includes step 6 to step 8, is applied to further process the received transformed data such that the maximal class

separability is achieved. Since this stage and the pre-flight stage would be done at the ground station, the algorithm simplicity is therefore less important than that in the on-board preprocessing stage. Hence, it might be more effective to use the overlapping band feature selection algorithm to design the features in some future situations although it's the most complex among the four techniques developed in this research.
## LIST OF REFERENCES

- [1] A. F. H. Goetz and M. Herring, "The High Resolution Imaging Spectrometer (HIRIS) for EOS," Proceedings of IGARSS '87 Symposium, pp 367-372, Ann Arbor, 18-21 May 1987.
- [2] C. L. Korb, "Physical Considerations in the Channel Selection and Design Specification of an Airborne Multispectral Scanner," Purdue Centennial Year Symposium on Information Processing, Purdue University, West Lafayette, Indiana, pp 646-657, 1969
- [3] R. L. Hulstrom, "Spectral Measurements and Analyses of Atmospheric Effects on Remote Sensor Data," Proceedings of the Society of Photo-Optical Instrument Engineers, Vol. 51, Scanners and Imagery Systems for Earth Observation, San Diego, California, pp 90-100, 1974
- [4] J. L. Harnage and D. A. Landgrebe, "Landsat-D Thematic Mapper Technical Working Group - Final Report," NASA/Johnson Space Center Report No. JSC-90797, Houston, Texas, 1975
- [5] G. A. May and G. W. Petersen, "Spectral Signature Selection for Mapping Unvegetated Soils," Remote Sensing of Environment, Vol. 4, pp 211-220, 1975.
- [6] L. A. Bartolucci, B. F. Robinson, and L. F. Silva, "Field Measurements of the Spectral Response of Natural waters, "Photogrammetric Engineering and Remote Sensing, Vol. 43, pp 595-598, 1977.
- [7] D. A. Landgrebe, L. L. Biehl and W. R. Simmons, "An Empirical Study of Scanner System Parameters, " IEEE Transaction on Geoscience Electronics, Vol. GE-15, pp 120-130, 1977
- [8] M. E. Coggeshall and R. M. Hoffer, "Basic Forest Cover Mapping Using Digitized Remote Sensor Data and Automatic Data Processing Techniques," LARS Information Note 030573, Laboratory for Application of Remote Sensing, Purdue University, West Lafayette, Indiana, 1973
- [9] R. K. Vincent and F. Thompson, "Spectral Compositional Imaging of Silicate Rocks, " J. of Geophysics, Res., Vol. 77, pp 2465-2472, 1972.

100

- [10] W. A. Malila, J. M. Gleason, and R. C. Cicone, "Multispectral System Analysis through Modeling and Simulation," Proceedings of 11th International Symposium on Remote Sensing, Ann arbor, Michigan, pp 1319-1328, 1977.
- [11] P. H. Swain and S. M. Davis, *Remote Sensing : the Quantitative Approach*, McGraw-Hill, New York, 1978.
- [12] E. R. Wiswell, "Analytical Techniques for the Study of Some Parameters of Multispectral Scanner Systems for Remote Sensing," Ph.D thesis, Purdue University, West Lafayette, Indiana, 1978.
- [13] R. Ash, Information Theory, Wiley Interscience, New York, 1967
- [14] D. J. Wiersma and D. A. Landgrebe, "Analytical Design of Multispectral Sensors," IEEE Transactions on Geoscience and Remote Sensing, Vol. GE-18, No. 2, pp 180-189,1980.
- [15] D. J. Wiersma, "The Analytical Design of Spectral Measurements for Multispectral Remote Sensor System," Ph.D. dissertation, Purdue University, 1979
- [16] K. Karhunen, "Uber Lineare Methoden in der Wahrscheinlichkeitsrechnung," American Acad. Sci. Fennicade, Ser. A, I, Vol. 37, pp 3-79, (Translated by RAND Corp., Santa Monica, Calif., Rept. T-131, August 1960).
- [17] M. Loeve, "Probability Theory," Princeton, N. J., Van Nostrand, 1963.
- [18] H. L. VanTree, *Detection, Estimation, and Modulation Theory-Part I,* New York: Wiley, 1968
- [19] L.E. Franks, Signal Theory, Dowden & Culver, 1981.
- [20] K. G. Beauchamp, *Walsh Functions and their Applications*, Academic Press, New York, 1975
- [21] B. Pyron and F. Williams, Jr., "Signal Processing by Infinite Clipping and Related Techniques," Georgia Institute of technology, Atlanta, Georgia, Final Report, Project A-727, U.S. Government Contract DA 49-092-ARO-21, April 1964
- [22] B. Noble and J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, N. J., 1977.

- [23] L. L. Biehl, "Listing of LARSPEC Formatted Field Research Spectroradiometer-Multiband Radiometer, "Internal Paper, Laboratory of Application of Remote Sensing, Purdue University, West Lafayette, Indiana, Jan. 1983.
- [24] B. H. Bharucha and T. T. Kadota, "On the representation of Continuous Parameter Process by a Sequence of Random Variables," IEEE Transaction on Information Theory, Vol. IT-16, pp 139-141, 1970.
- [25] S. Watanabe, "Karhunen-Loeve Expansion and Factor Analysis, Theoretical Remarks and Applications," Proceedings of the Four Prague Conference on Information Theory, Statistical Decision Functions, and Random Processes," Prague, pp 635-660, 1965.
- [26] Y. T. Chien and K. S. Fu, "On the Generalized Karhunen-Loeve Expansion," IEEE Transaction on Information Theory, Vol. IT-13, pp 518-520, July 1967.
- [27] J. T. Oden, Applied Functional Analysis, Prentice-Hall, N. J., 1979.
- [28] R. W. A. Scarr, "Zero Crossing as a Means of Obtaining Spectral Information in Speech Analysis," IEEE Transaction on Audio and Electroacoustics, Vol. AU-16, No. 2, pp 247-255, June 1968.
- [29] G. D. Ewing and J. F. Taylor, "Computer Recognition of Speech Using Zero-Crossing Information, "IEEE Transactions on Audio Electroacoustics, Vol. AU-17, pp37-40, Mar. 1969
- [30] R. J. Niederjohn, M. W. Krutz and B. M. Brown, "An Experimental Investigation on the Perceptual Effects of Altering the Zero-Crossing of a Speech Signal ", IEEE Transaction on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 5, pp618-625, May 1987
- [31] F. E. Bond and C. R. Cahn, "On Sampling the Zeros of Bandwidth Limited Signals," IRE Transaction on Information Theory, Vol. IT-4, pp 110-113, September 1958.
- [32] M. S. Scivier and M. A. Fiddy, "Phase Ambiguities and the Zeros of Multidimensional Band-limited Functions," Optical Society of America, Vol. 2, No. 5, pp 693-697, May 1985.
- [33] R. J. Niederjohn, " A Mathematical Formulation and Comparison of Zero-Crossing Analysis Techniques Which Have Been Applied to Automatic Speech Recognition," IEEE Transaction on Acoustics, Speech and Signal Processing, Vol. ASSP-23, No. 4, pp 373-380, August 1975.

- [34] K. Fukunaga, Introduction to Statistical Pattern Recognition, New York: Academic, 1972
- [35] C-C Thomas Chen and D. A. Landgrebe, "Spectral Feature Design for Data Compression in High Dimensional Multispectral Data ", 1987 International Geoscience and Remote Sensing Symposium ('87 IGARSS), pp 685-690, May 1987
- [36] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, New York, Wiley 1978
- [37] John A. Richards, Remote Sensing Digital Image Analysis, An Introduction, Springer Verlag, pp 206 - 224.
- [38] H.M. Lachowski, "Canonical Analysis Applied to Interpretation of Multispectral Scanner Data ", MS Thesis, Pennsylvania State University, University Park, PA, 1973
- [39] B.F. Merembeck and F.Y. Borden, "Principal Component and Canonical Analysis for Reduction of Dimensionality of the Large Data Sets", ORSER Technical Report 5-78, Office for Remote Sensing of Earth Resources, Pennsylvania State University, University Park, PA, 1978
- [40] B.F. Merembeck, F.Y. Borden, M.H. Podwysocki and D.N. Applegate, "Application of Canonical Analysis to Multispectral Scanner Data ", Proceedings, the 14th Annual Symposium on Applications of Computer Methods in the Mineral Industries, Oct. 4-8, University Park, PA, pp867-879, 1976
- [41] B.F. Merembeck & B.J. Turner, "Directed Canonical Analysis and the Performance of the Classifiers under its associated Linear Transformation," IEEE Transaction on Geoscience and Remote Sensing, GE-18, 190-196, 1980.
- [42] G.F. Hughes, " On the Mean Accuracy of Statistical Pattern Recognizers ", IEEE Transaction on Information Theory, vol. IT-14, no. 1, pp55-63, 1968
- [43] K.S. Fu, D.A. Landgrebe and T.L. Phillips, "Information Processing of Remotely Sensed Agriculture Data ", Proceedings of IEEE, vol. 57, no. 4, pp639-653, 1969
- [44] Davenport and Root, *Random Signals and Noise*, McGraw-Hill Book Co., 1958
- [45] Eric Stoner, "Physicochemical, Site, and Bidirectional Reflectance Factor Characteristics of Uniformly-Moist Soils ", Ph.D. Dissertation, Department of of Agronomy, Purdue University, 1979

- [46] Soil Survey Staff, "Soil Taxonomy A basic system of soil classification for making and interpreting soil survey ", Soil Conservative Service, US Department of Agriculture, Agriculture Handbook No. 436, Washington, DC, 1975
- [47] Eric Stoner, "Atlas of Soil Reflectance Properties ", Agriculture Experiment Station, Research Bulletin, Purdue University, West Lafayette, Indiana, 1979
- [49] Marion F. Baumgardner, LeRoy F. Silva, Larry L. Biehl, and Eric R. Stoner, "Reflectance Properties of Soils ", Purdue University, West Lafayette, Indiana, Advances in Agronomy, vol. 38, 1985
- [50] R. M. Hoffer and C. J. Johannsen, Remote Sensing Ecology, P.L. Johnson, ed., pp1-29, University of Georgia Press, Athens, 1969
- [51] A.J. Rainal, "Zero-Crossing Principle for Detecting Narrowband Signals," IEEE Transaction on Instrumentation and Measurement, Vol. IM-15, pp38-43, March/June 1966
- [52] A.J. Rainal, "Another Zero-Crossing Principle for Detecting Narrowband Signals," IEEE Transaction on Instrumentation and Measurement, Vol. IM-16, pp135-138, June 1967
- [53] T. Masuda, H. Miyano, and T. Sadoyama, "The Measurement of Muscle Fiber Conduction Velocity Using a Gradient Threshold Zero-Crossing Method, "IEEE Transaction on Biomedical Engineering, Vol. BME-29, pp673-678, Oct. 1982
- [54] H. B. Voelcker, "Zero-Crossing Properties of Angle Modulated Signals," IEEE Transaction on Communication, Vol. COM-20, pp307-315, June 1972
- [55] R. G. Wiley, H. Schwarzlander, and D. D. Weiner, "Demodulation Procedure for Very Wide-band FM, "IEEE Transaction on Communication, Vol. COM-25, pp 318-327, March 1977
- [56] R. M. Haralick, "Digital Step Edges from Zero-Crossing of Second Directional Derivatives, "IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, pp 58-68, Jan. 1984
- [57] Courant and Hilbert, *Mehods of Mathematical Physics*, International Publisher Ltd, London, 1955

## Appendix A IBM 3083 Macro File

/\* RUN A FORTRAN PROGRAM USING IMSLSP OR IMSLDP SUBROUTINES \*/ ARG FN FN1 FN2 FN3 FN4 FN5 FN6 FN7 FN8 FN9 FN10 FN11 LINKTO IMSL GLOBAL TXTLIB IMSLSP IMSLDP PFORTLIB VSF2FORT CMSLIB GLOBAL LOADLIB VSF2LOAD FORTVS2 FN LOAD FN FILEDEF 11 DISK FN1 DATA C1 FILEDEF 12 DISK FN2 DATA C1 FILEDEF 13 DISK FN3 DATA C1 FILEDEF 14 DISK FN4 DATA C1 FILEDEF 15 DISK FN5 DATA C1 FILEDEF 16 DISK FN6 DATA C1 FILEDEF 17 DISK FN7 DATA C1 FILEDEF 18 DISK FN8 DATA C1 FILEDEF 19 DISK FN9 DATA C1 FILEDEF 20 DISK FN10 DATA C1 FILEDEF 21 DISK FN11 DATA C1 START

Appendix B Spectral Feature Design System - Program Listing

C

C.

С

С

С

Ċ

С

С

C

Ċ

C

С

С

С

C

С

MV 00010 PROGRAM MV PARAMETER (NP2=1551, NP1=100, NP3=NP1\*(NP1+1)/2, NF2=10, NF3=5) MV 00020 MV 00030 REAL X (NP2, NP1), XM (NP1), VCV (NP3) MV 00040 DATA IFLAG1, XM, VCV/0, NP1\*0.0, NP3\*0.0/ MV 00050 MV 00060 DIMENSIONALITY OF SAMPLE FUNCTIONS NP1 : MV 00070 TOTAL NUMBER OF SAMPLE FUNCTIONS NP2 : MV 00080 TOTAL NUMBER OF ELEMENTS IN COVARIANCE MATRIX VCV NP3 : MV 00090 RAW DATA INPUT FILE STORED IN FORMAT 10F8.3 NF2 : XM & VCV OUTPUT DATA FILE STORED IN FORMAT 5E15.7 MV 00100 NF3 : MV 00110 MV 00120 x : RAW DATA ( INPUT ) MV 00130 XM : MEAN VECTOR ( OUTPUT ) VCV : COVARIANCE MATRIX STORED IN SYMMETRIC MODE ( OUTPUT ) MV 00140 MV 00150 IFLAG1 ----- INTERNAL CHECKING PARAMETER MV 00160 MV 00170 11 = DATA FILE; 12 = MV FILEMV 00180 MV 00190 OPEN(11) MV 00200 OPEN(12) MV 00210 REWIND 11 MV 00220 REWIND 12 MV 00230 READ IN RAW DATA AND PRINT THE PROGRESS FOR EVERY 100 SAMPLES MV 00240 MV 00250 С MV 00260 DO 20 ISAMP=1,NP2 MV 00270 K=MOD(ISAMP,100) IF (K.EQ.0) PRINT\*, ' NP2 = ', NP2, '; ISAMP = ', ISAMP MV 00280 MV 00290 DO 20 I=1, NP1/NF2 MV 00300 20 READ(11,1)(X(ISAMP,J), J=1+(I-1)\*NF2, I\*NF2) MV 00310 PRINT\*, ' DATA READ IN FINISHED ' MV 00320 1 FORMAT (10F8.3) MV 00330 Ċ MV 00340 C FIND THE ENSEMBLE MEAN VECTOR MV 00350 С MV 00360 DO 30 J=1,NP1 MV 00370 DO 30 I=1,NP2 MV 00380 30 XM(J) = XM(J) + X(I, J)MV 00390 DO 40 I=1,NP1 MV 00400 40 XM(I) = XM(I) / FLOAT(NP2)MV 00410 PRINT\*, MEAN VECTOR FOUND ' С MV 00420 FIND THE ENSEMBLE COVARIANCE MATRIX AND PRINT THE PROGRESS FOR MV 00430 С MV 00440 C EVERY 10 DIMENSIONS MV 00450 C MV 00460 DO 50 I=1,NP1 MV 00470 KX = MOD(I, 10)MV 00480 IF (KX.EQ.0) PRINT\*, I MV 00490 DO 50 J=1,I MV 00500 IND = (I-1) + I/2 + J

			3.67	
	•	DO 50 K=1, NP2	MV	00210
	50	VCV(IND) = VCV(IND) + (X(K, I) * X(K, J) - XM(I) * XM(J))	MV	00520
:		DO $60$ I=1 NP3	MV	00530
	60	$V_{CV}(T) - V_{CV}(T)$ /FI OAT (ND2-1)	MT.7	00540
	00		-1-1 V	OODTO
· * .		PRINT*, COV. MATRIX FOUND	MV	00550
С	51 S	그는 것 같은 것 같아요. 말 좋은 것 같은 것 같은 것 같은 것 같아요. 말 같아요. 가지 않는 것 않는 것 같아요. 가지 않는 것 않는	MV	00560
C		INTERNAL CHECKING FOR ALGORITHM ACCURACY	MV	00570
~			3.677	00500
C.	$\sum_{i=1}^{n}   f_i  ^2$	에는 해도 이 사회의 사과에서 한 것이다. 이 것을 것 같아요. 사가 가지 않는 것이라. 이 가지 않는 가	140	00360
۰.	•	DO 80 I=1,NP1	MV	00590
		TND = T * (T+1) / 2	MV	00600
			MX7	00610
÷			2.14	00010
÷.,		$z ~{ m GO}$ $~{ m TO}$ $z 80$ (here even the contract of the	MV	00620
	70	WRITE (*, 2) I, VCV (IND)	MV	00630
	2	FORMAT (JACCURACY OF ALGORITHM IS POOR AT I ='. 15.	MV	00640
	. 2	I WITH THE FILE AND F	N/S 7	00650
	-	+ WHERE VCV $(1,1) = .7E12.7$	TATA	00050
		VCV(I) = -VCV(I)	MV	00660
	÷.,	TFLAG1=TFLAG1+1	MV	00670
	00	CONTINUE	MX7	00680
	00	CONTINUE	1.10	00000
Ç		그는 것 같은 것 같은 것 같은 것 같은 것 같은 것 같은 것 같아. 전화 문화가 있는 것 같은 것 같은 것 같이 같이 없다.	MV	00690
С		PRINT THE COMMENTS FOR ACCURACY	MV	00700
č		그렇게 잘 가지 않는 것이 있는 것이 같아요. 아이는 것이 가지 않는 것이 가지 않는 것이 가지 않는 것이 나라.	MV7	00710
C			1.07	00720
		IF (IFLAGI.GT.U) GO TO 90	MA	00720
	•	PRINT*, ' POSITIVE VARIANCES CHECK DONE '	MV	00730
			MV	00740
			1.07	00750
÷			MA	00750
	90	WRITE (*, 3) IFLAG1	MV	00760
	3	FORMAT (' THERE ARE ' 15. ' VARIANCES LESS THAN 0.0 ')	MV	00770
	· .	= ODMAT(1) ATT WARTANCES ARE UN= 0.00 ACCURACY TS (OOD!)	MUZ	00780
-	4	FORMAT (* ALL VARIANCES ARE >= 0.0 , ACCORACT IS GOOD )	1.1.4	00700
С		이 가지 않는 것 같아요. 이 것 같아요. 이 것 같아요. 이 가지 않는 것 같아요. 아니는 것 같아요. 이 것은 것 같아요. 이 것 같아요. 이 것 같아요. 이 것 같아요. 이 것	MV	00790
С	·	SEND THE RESULTS TO OUTPUT DATA FILE	MV	00800
ĉ		이 것 같아요. 이 가지 않는 것 이 가지 않는 것 같아요. 이 가지 않는 것 같아요. 이 가지 않는 것 않는		00010
0			MV	00810
	1		MV	00810
	100	DO 110 I=1,NP1/NF3	MV MV	00810
	100 110	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3)	MV MV MV	00810 00820 00830
•	100 110 5	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15,7)	MV MV MV MV	00810 00820 00830 00840
•	100 110 5	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7)	MV MV MV MV	00810 00820 00830 00840 00850
	100 110 5	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3	MV MV MV MV	00810 00820 00830 00840 00850
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3)	MV MV MV MV MV	00810 00820 00830 00840 00850 00850
· ·	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP	MV MV MV MV MV MV	00810 00820 00830 00840 00850 00850 00860 00870
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3 WRITE(12,5)(VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END	MV MV MV MV MV MV	00810 00820 00830 00840 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3 WRITE(12,5)(VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END	MV MV MV MV MV MV MV	00810 00820 00830 00840 00850 00860 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3 WRITE(12,5)(VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END	MV MV MV MV MV MV	00810 00820 00830 00840 00850 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3 WRITE(12,5)(VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END	MV MV MV MV MV MV MV	00810 00820 00830 00840 00850 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3 WRITE(12,5)(VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END	MV MV MV MV MV MV MV	00810 00820 00830 00840 00850 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE(12,5)(XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT(5E15.7) DO 120 I=1,NP3/NF3 WRITE(12,5)(VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END	MV MV MV MV MV MV MV	00810 00820 00830 00840 00850 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV	MV MV MV MV MV MV MV EV	00810 00820 00830 00840 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1,	MV MV MV MV MV MV MV EV	00810 00820 00830 00840 00850 00850 00860 00870 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5)	MV MV MV MV MV MV MV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1) VCV(NP3) VCVF(NP1,NP1),D(NP1)	MV MV MV MV MV MV MV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00880
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1),	MV MV MV MV MV MV MV MV EV EV EV	00810 00820 00830 00840 00850 00850 00860 00870 00880 00010 00020 00010 00020 00030 00040
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5)	MV MV MV MV MV MV MV MV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00880 00010 00020 00020 00030 00040 00050
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM	MV MV MV MV MV MV MV MV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00880 00010 00020 00030 00040 00050 00060
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z (NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2, IFLAG1,SUM, TRACE/2,0,2*0.0/	MV MV MV MV MV MV MV MV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00880 00010 00020 00030 00040 00050 00060 00070
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/	MV MV MV MV MV MV MV MV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00060 00070 00080
C	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM (J), J=1+(I-1)*NF3, I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV (J), J=1+(I-1)*NF3, I*NF3) STOP END PARAMETER (NP1=100, NP3=NP1* (NP1+1)/2, NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM (NP1), VCV (NP3), VCVF (NP1, NP1), D (NP1), +Z (NP1,NP1), WK2 (NP5) REAL TRACE, SUM DATA JOB2, IFLAG1, SUM, TRACE/2, 0, 2*0.0/	MV MV MV MV MV MV MV MV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00020 00030 00040 00050 00060 00070 00080
Сс	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z (NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00070 00080 00090
CCC	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z (NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00070 00080 00090 00100
0000	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00070 00060 00070 00080 00090 00100 00110
0000	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00050 00060 00070 00080 00090 00100 00110 00110
CCCCC	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00050 00060 00070 00080 00090 00100 00110 00120
00000	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2 XM : MEAN VECTOR	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV EV EV	00810 00820 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00020 00030 00040 00050 00060 00070 00080 00090 00100 00110 00120 00130
000000C	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM (J), J=1+(I-1)*NF3, I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV (J), J=1+(I-1)*NF3, I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100, NP3=NP1*(NP1+1)/2, NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM (NP1), VCV (NP3), VCVF (NP1, NP1), D (NP1), +Z (NP1,NP1), WK2 (NP5) REAL TRACE, SUM DATA JOB2, IFLAG1, SUM, TRACE/2, 0, 2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2 XM : MEAN VECTOR VCV : COVARIANCE MATRIX ( SYMMETRIC STORAGE MODE )	MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00880 00020 00020 00020 00030 00040 00050 00060 00070 00080 00090 00100 00100 00110 00120 00130 00140
0000000	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM (J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV (J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM (NP1),VCV (NP3),VCVF (NP1,NP1),D (NP1), +Z (NP1,NP1),WK2 (NP5) REAL XM (NP1),VCV (NP3),VCVF (NP1,NP1),D (NP1), +Z (NP1,NP1),WK2 (NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2 XM : MEAN VECTOR VCV : COVARIANCE MATRIX ( SYMMETRIC STORAGE MODE )	MV MV MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00070 00060 00070 00080 00090 00100 00100 00110 00120 00130 00140 00150
	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM(J),J=1+(I-1)*NF3,I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV(J),J=1+(I-1)*NF3,I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100,NP3=NP1*(NP1+1)/2,NP5=NP3+NP1, +NF2=10,NF3=5) REAL XM(NP1),VCV(NP3),VCVF(NP1,NP1),D(NP1), +Z(NP1,NP1),WK2(NP5) REAL TRACE,SUM DATA JOB2,IFLAG1,SUM,TRACE/2,0,2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2 XM : MEAN VECTOR VCV : COVARIANCE MATRIX ( SYMMETRIC STORAGE MODE ) VCVF : COVARIANCE MATRIX ( FULL STORAGE MODE )	MV MV MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00070 00060 00070 00060 00070 00080 00090 00100 00110 00120 00110 00120 00130 00140 00150
00000000	100 110 5 120	DO 110 I=1,NP1/NF3 WRITE (12,5) (XM (J), J=1+(I-1)*NF3, I*NF3) FORMAT (5E15.7) DO 120 I=1,NP3/NF3 WRITE (12,5) (VCV (J), J=1+(I-1)*NF3, I*NF3) STOP END PROGRAM EV PARAMETER (NP1=100, NP3=NP1*(NP1+1)/2, NP5=NP3+NP1, +NF2=10, NF3=5) REAL XM (NP1), VCV (NP3), VCVF (NP1, NP1), D (NP1), +Z (NP1,NP1), WK2 (NP5) REAL TRACE, SUM DATA JOB2, IFLAG1, SUM, TRACE/2, 0, 2*0.0/ NP1 : RAW DATA DIMENSIONALITY NP3 : TOTAL NUMBER OF ELEMENTS FOR VCV NP5 : DIMENSION FOR PERFORMANCE INDEX MATRIX WK2 XM : MEAN VECTOR VCV : COVARIANCE MATRIX ( SYMMETRIC STORAGE MODE ) VCVF : COVARIANCE MATRIX ( FULL STORAGE MODE ) D : EIGENVALUE	MV MV MV MV MV MV MV MV MV MV MV MV EV EV EV EV EV EV EV EV EV EV EV	00810 00820 00830 00840 00850 00860 00870 00880 00010 00020 00030 00040 00050 00060 00070 00060 00070 00060 00070 00080 00090 00100 00110 00120 00130 00140 00150 00160

EV 00170 : EIGENVECTOR C Z EV 00180 Ċ WK2 : PERFORMANCE INDEX MATRIX EV 00190 С EV 00200 Ċ 11 : INPUT MV FILE ; 12 : OUTPUT EV FILE EV 00210 C EV 00220 OPEN(11) EV 00230 OPEN(12)EV 00240 REWIND 11 EV 00250 **REWIND 12** EV 00260 С READ IN MEAN VECTOR AND COVARIANCE MATRIX EV 00270 С EV 00280 C EV 00290 DO 10 I=1.NP1/NF3 EV 00300 10 READ (11, \*) (XM(J), J=1+(I-1)\*NF3, I\*NF3) EV 00310 1 FORMAT (5E15.7) EV 00320 DO 20 I=1,NP3/NF3 EV 00330 20 READ(11,\*)(VCV(J), J=1+(I-1)\*NF3, I\*NF3) EV 00340 CALL VCVTSF (VCV, NP1, VCVF, NP1) EV 00350 C FIND TRACE, EIGENVALUES AND EIGENVECTORS OF THE COVARIANCE MATRIX EV 00360 С EV 00370 C EV 00380 DO 30 I=1,NP1 EV 00390 30 TRACE=TRACE+VCVF(I,I) EV 00400 CALL EIGRS (VCV, NP1, JOB2, D, Z, NP1, WK2, IER) EV 00410 С EV 00420 PRINT THE PERFORMANCE INDEX AND ACCURACY COMMENTS Ċ EV 00430 Ċ IF (IER.NE.0.OR.WK2(1).GE.1.0) GO TO 40 EV 00440 EV 00450 WRITE (\*, 3) IER, WK2(1) EV 00460 GO TO 50 EV 00470 40 WRITE (\*, 2) IER, WK2(1) 2 FORMAT (' PERFORMANCE OF "EIGRS" IS POOR, IER =', I5, EV 00480 + WK2(1) = ', E15.7) EV 00490 3 FORMAT (' PERFORMANCE OF "EIGRS" IS GOOD, IER =', I5, EV 00500 EV 00510 + WK2(1) = ', E15.7) EV 00520 С EV 00530 C INTERNAL CHECKING FOR ACCURACY EV 00540 C EV 00550 50 DO 70 I=1,NP1 EV 00560 IF (D (I).LE.0.0) GO TO 60 EV 00570 GO TO 70 EV 00580 60 WRITE (\*, 4) I, D(I) 4 FORMAT (! EIGEN VALUE IS "< = 0.0" AT I =', I5, EV 00590 EV 00600 +' WHERE D(I) =', E15.7) EV 00610 IFLAG1=IFLAG1+1 EV 00620 70 CONTINUE EV 00630 IF (IFLAG1.GT.0) GO TO 80 EV 00640 WRITE (\*, 6) EV 00650 GO TO 90 80 WRITE (\*, 5) IFLAG1 EV 00660 5 FORMAT (' THERE ARE', 15, ' NEGATIVE OR ZERO EIGEN VALUES ') EV 00670 EV 00680 6 FORMAT (' ALL EIGEN VALUES ARE GREATER THAN ZERO ') EV 00690 С FIND THE SUM OF THE EIGENVALUES AND PRINT THE ACCURACY COMMENTS EV 00700 C : EV 00710 С EV 00720 90 CALL VABSMF (D, NP1, 1, SUM) EV 00730 IF (ABS (TRACE-SUM) .GT.1.0E-1) GO TO 100

医马马克氏 医马马克氏 医白色 医白色 医白色 医白色 法法律 化合同分子 医白色 化乙基乙基 化乙基乙基 机合成 化乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基乙基		1997 - La 1
WRITE (*,8) TRACE, SUM	EV	00740
inter GO TO 110 a sector is a la side de la side de la sector de la s	EV	00750
100 WRITE (*, 7) TRACE, SUM	EV	00760
7 FORMAT (' ACCURACY OF "EIGRS" IS POOR, TRACE = ', E15.7.	ĒV	00770
+' SUM =' E15.7)	EV	00780
8 FORMAT (' ACCURACY OF "EIGRS" IS GOOD, TRACE =', E15.7,	ĒV	00790
+' SUM =',E15.7)	ΈV	00800
	EV	00810
SEND THE RESULTS TO THE OUTPUT DATA FILE	ΕV	00820
이 이 같은 사람들은 것은 것을 하는 것을 하는 것을 통하는 것을 가지 않는 것을 가지 않는 것을 하는 것을 하는 것을 수 있다.	EV	00020
110 WRITE (12, 9) TRACE, SUM	TV.	000000
9 FORMAT (2E15.7)	<u>ت</u> تر7	00040
$DO_{120}$ T=1 NP1/NP3	EV	00000
$\frac{1}{120} = \frac{1}{12} + \frac{1}{12} + \frac{1}{12} + \frac{1}{12} = \frac{1}{12} + \frac{1}{12}$		00000
D = 120 T-1 ND1	E۷	00870
	ΈV	00880
DO 130 1=1, NP1/NF3	EV	00890
130 WRITE (12, 1) (Z (K, NP1+1-J), K=1+(I-1)*NF3, I*NF3)	EV	00900
STOP	EV	00910
$\mathbf{END}$ . The second	EV	00920

	PROGRAM NOLBS PARAMETER (NP1=100.NTERM=6.NV=50.NZ1=NP1*NV.N1=1.N2=100)	BS BS	00010
		BS	00020
•	FOR FSS VEGETATION DATA : $N1 = 1$ ; $N2 = 100$	BS	00040
	FOR SOIL DATA : $N1 = 4$ ; $N2 = 192$	BS	00050
	이 같은 방법에 가지 않는 것이 같은 것이 같은 것이 같이 있는 것이 같은 동물을 통했다. 방법	BS	00060
	FOR SOIL DATA ( FROM EFFECTIVE WAVELENGTH 0.52 TO 2.32UM:180 DIM )	BS	00070
	N1=1, N2=180	BS	00080
	- <u>수업 방법 수</u> 집에 가지 않는 것이 것 같은 것이 다. 한 것이 가지 않는 것이 있는 것이 있다.	BS	00090
	REAL X (NP1, NTERM), AVE (NP1), S1 (NP1), Z (NP1, NV)	BS	00100
	DATA Z/NZ1*0.0/	BS	00110
•		BS	00120
	NET : RAW DATA DIMENSIONALITY	BS	00130
	NIERA: IOTAL NUMBER OF OPTIMAL FUNCTIONS USED IN THE ALGORITHM	BS	00140
	NI • THE STARTING WAVELENCTH DOINT	BO	00100
	N2 : THE ENDING WAVELENGTH POINT	DO BS	00160
1		BS	00180
•	X : EIGENVECTOR ( INPUT )	BS	00190
	AVE : AVERAGE OF THE FIRST 'NTERM' EIGENVECTORS	BS	00200
•	S1 : SIGNED VERSION OF AVE (NP1)	BS	00210
	Z : DESIRED N.O.L. BAND FEATURES ( OUTPUT )	BS	00220
		BS	00230
	11 : INPUT EIGENVECTOR FILE; 12 : OUTPUT N.O.L. BAND FILE	BS	00240
		BS	00250
	OPEN (11)	BS	00260
÷	OPEN(12) DEMITIND 11	BS	00270
	REWIND 11	BS	00280
2	**************************************	DC DC	00290
1	DO 10 I=1.NP1/5	20	00300
10	READ (11.*) X1. X2. X3. X4. X5	BS	00350
		BS	00330

C C C

0000000

0000000000000000

Ç

		教授 医乳液 化乙酰氨基化 化分子子 法通知 化乙酰胺乙酰胺 医鼻腔 网络小麦花 化分子 化合金 建筑		
		110 · · · · · · · · · · · · · · · · · ·		
· .			, isis	· · · · ·
-			DC	00340
С.		READ IN EIGENVECTORS	50	00340
С.			BS	00350
		DO 20 ITERM=1, NTERM	BS	00360
		DO 20 $J=1, NP1/5$	BS	00370
	20	<b>READ</b> $(11, *)$ (X (T. TTERM), T=1+ (J-1)*5, J*5)	BS	00380
~	20		BS	00390
<u> </u>		FIND THE AVEDACE OF THE FIDST INTERM! FIGENVECTORS AND	BS	00400
ل م		FIND THE AVERAGE OF THE FIRST NIERT EIGENVECTORS THE	BC	00410
C		ITS SIGNED VERSION	50	00410
С	1.1	지수 같은 것 같아요. 이 것 같아? 동물에 집에 가지 않는 것 않는 것 같은 것 같아. 것 같아.	53	00420
		DO 40 J=1,NP1	BS	00430
	1. A.	AVE(J) = 0.0	BS	00440
		DO 30 ITERM=1, NTERM	BS	00450
	30	AVE $(J) = AVE (J) + X (J, ITERM) / FLOAT (NTERM)$	BS	00460
		TE(NP1 NE 100) GO TO 35	BS	00470
		T = (M + 1) M + 1 M + 100 / 10 = 0 M + 10 M + 10 M + 100 /	BS	00480
		1F(0, GE, 43, AND, 0, LE, 34) AVE(0) = 0.0	BC	00100
		IF (J.GE. / U.AND. J. LE. / 9) AVE (J) = 0.0	00	00400
	35	IF (AVE (J) $.LT.0.0$ ) S1 (J) = -1.0	<u>Б</u> Э	00500
		IF(AVE(J).GT.0.0)S1(J)=1.0	BS	00510
		IF (AVE (J) $EQ.0.0$ ) S1 (J) = 0.0	BS	00520
	40	CONTINUE	BS	00530
Ċ			BS	00540
č		THE NEXT 3 LINES CAN BE USED TO PLOT AVE (I) AND S1(I)	BS	00550
č		· 경험 · 전화 · 전	BS	00560
č		DO 50 T=1 NP1	BS	00570
Č.	50	$\operatorname{MDTTF}_{12} (12, 51) \operatorname{AVF}_{11} (T) = S1(T)$	BS	00580
	50	$ \begin{array}{c} \text{WATTE} (12, 31) \text{AVE} (1), 1, 31 (1) \\ \text{PODMAR} (p) [ 5, 7, 15, 55, 0) \\ \end{array} $	BS	00590
U.	эт	EORMAI (E13.7,13,13.0)	BC	006000
		IVEC=I	00	00000
		Z(NI, IVEC) = ABS(SI(NI))	50	00010
C <sub>.</sub>		이 같은 것 같은	BS	00620
С	1	FIND N.O.L. BAND FEATURES FROM S1	BS	00630
С		· 철말 같이 있는 것 같이 없는 것 같이 않 것 같이 없는 것 같이 않는 것 같이 않 않는 것 같이 않는 것 않는 것 같이 않는 것 않는 것 않 것 않는 것 같이 않는 것 않 않이 않는 것 같이 않는 것 않이 않이 않는 않이 않는 것 않이 않이 않는 않이 않는 않는 않이 않 않이 않	BS	00640
		DO 60 I=N1+1,N2	BS	00650
	а.	IF (NP1.NE.100) GO TO 55	BS	00660
		TF (T. GE. 45. AND. T. LE. 54) GO TO 60	BS	00670
		TE (T CE 70 AND T LE 79) GO TO 60	BS	00680
	55	$TF(S_1(T_{-1}))$ NE S1(T)) TVFC=TVFC+1	BS	00690
	55	$I = \{J = \{1, J, J,$	BS	00700
		$\frac{1}{12} \frac{1}{12} \frac$	RQ	00710
		IF (IVEC.GE.NV) GO IO IZO	BC	00720
	~~	Z(1, 1VEC) = ABS(S1(1))	- DO	00720
_	60	CONTINUE	50	00730
С.			BS	00740
С		NORMALIZE THE FEATURES AND SEND THEM TO THE OUTPUT FILE	BS	00750
С	1		BS	00760
		DO 100 J=1, IVEC	BS	00770
•	11.	XN1=0.0	BS	00780
		DO 70 $I=1.NP1$	BS	00790
	70	XN1 = XN1 + Z(T, J) * Z(T, J)	BS	00800
		DO 80 T=1.NP1	BS	00810
	٥ń	7/T T = $7/T$ T / SOBT (XN1)	BS	00820
	00	$\Delta(1,0) - \Delta(1,0) / \Delta(1,0) / \Delta(1,0) = 0$	RC	00830
	~ ~	$\frac{1}{2} \frac{1}{2} \frac{1}$	00 00	00000
	90	WKITE ( $12, 91$ ) ( $3(1, 0), 1=1+(11-1)^{3}, 11^{3}$ )	50	00040
	91	FORMAT (5E15.7)	85	00850
÷ .	100	CONTINUE	BS	00860
	120	PRINT*, ' TOTAL NUMBER OF N.O.L. BAND FEATURES =', IVEC	BS	00870
	.*	STOP	BS	00880
· .		. <b>END</b> share a state of the second state of the state of	BS	00890

WAL00010 PROGRAM WALSH WAL00020 THIS PROGRAM IS USED TO GENERATE THE FIRST 64 100-DIM. WALSH FUN. WAL00030 IN THIS PROGRAM WE SET W1=0,1 AND W2=-0.1 SUCH THAT NORM (W)=1.0 WAL00040 NP1 = 100, M = 6, NF4 = 5 USED FOR 64 100-DIM WALSH FUNCTIONS WAL00050 WAL00060 WAL00070 PARAMETER (NP1=100, M=6, NTVEC=2\*\*M, NMAX=2\*\* (M-1), +W1=0.1,W2=-0.1,NF4=5,NP5=NP1/2,NP6=NP1/4) WAL00080 WAL00090 REAL Z (NP1, NTVEC), ZW1 (NP1, NMAX), ZW2 (NP1, NMAX) WAT.00100 INTEGER NZERO (NTVEC) WAL00110 WAL00120 NP1 : DIMENSIONALITY OF WALSH FUNCTION WAL00130 M : TOTAL NUMBER OF WALSH FUNCTIONS IS 2\*\*M WAL00140 NTVEC : TOTAL NUMBER OF WALSH FUNCTIONS : THE NORMALIZED LENGTH OF 100-DIM. WALSH FUNCTION WAL00150 **W1** WAL00160 W2 : THE NEGATIVE OF W1 WAL00170 : OUTPUT FORMAT USE NF4 WAL00180 : RESULTS OF WALSH FUNCTIONS ( OUTPUT ) Z WAL00190 : INTERMEDIATE MATRIX FOR WALSH FUNCTION GENERATION ZW1 : INTERMEDIATE MATRIX FOR WALSH FUNCTION GENERATION WAL00200 ZW2 NZERO : CHECKING VECTOR FOR AXIS CROSSINGS OF WALSH FUNCTIONS WAL00210 WAL00220 SET UP THE FIRST 4 WALSH FUNCTIONS WAL00230 WAL00240 WAL00250 DATA ((Z(I, J), I=1, NP1), J=1, 4)/NP1\*W1, NP5\*W1, NP5\*W2, +NP6\*W1, NP5\*W2, NP6\*W1, NP6\*W1, NP6\*W2, NP6\*W1, NP6\*W2/ WAL00260 WAL00270 WAL00280 OPEN(11) WAL00290 REWIND 11 WAL00300 WAL00310 STORE THE THIRD AND FOURTH WALSH FUNCTIONS WAL00320 WAL00330 DO 10 J=1,2 WAL00340 DO 10 I=1, NP1 WAL00350 10 ZW1(I,J) = Z(I,2+J)WAL00360 PRINT\*, 'IM = 0,1,2, SEQ : Z(I,1), Z(I,2), ZW1(I,1), ZW1(I,2)' WAL00370 DO 20 I=1,NP1 WAL00380 20 WRITE (\*, \*) I, Z (I, 1), Z (I, 2), ZW1 (I, 1), ZW1 (I, 2) WAL00390 WAL00400 GENERATE THE FIRST 2\*\*M WALSH FUNCTIONS WAL00410 WAL00420 DO 70 IM=3.M WAL00430 K=2\*\*(IM-1)WAL00440 DO 30 IK=1,K-1,2 WAL00450 IKM = (IK+1)/2WAL00460 DO 30 I=1,NP5 WAL00470 ZW2(I, IK) = ZW1(2\*I, IKM)WAL00480 30 ZW2(NP5+I,IK)=((-1.)\*\*(IKM+1))\*ZW1(2\*I,IKM) WAL00490 DO 40 IK=2,K,2 WAL00500 IKM = IK/2WAL00510 DO 40 I=1,NP5 WAL00520 ZW2(I,IK)=ZW1(2\*I,IKM) WAL00530 40 ZW2 (NP5+I, IK) = ((-1.) \*\* (IKM)) \* ZW1 (2\*I, IKM) WAL00540 DO 50 IK=1,K WAL00550 DO 50 I=1,NP1 WAL00560 Z(I,K+IK) = ZW2(I,IK)WAL00570 50 ZW1(I, IK) = ZW2(I, IK)

С

C C

С

C

Ċ

С

Ċ

С

C

C

С

С

С

С

С

С

С

С

C

С

С

С

С

С

С

INF00010 PROGRAM INFCLIP INF00020 PARAMETER (NP1=100, NTERM=16, IEV=1) INF00030 REAL X(NP1) INF00040 Ċ INF00050 RAW DATA DIMENSIONALITY С NP1 : TOTAL NUMBER OF OPTIMAL FUNCTIONS USED IN THE ALGORITHMINF00060 С NTERMS INF00070 INPUT AND OUTPUT VARIABLE Ċ X INF00080 С INPUT FILE READING INDEX ( CHOOSE EITHER 1 OR 0 ) INF00090 С TEV IEV = 1 IF INPUT FILE CONTAINS TRACE, EIGENVALUES AND THEIR SUM INF00100 Ċ INF00110 IEV = 0 IF INPUT FILE CONTAINS ONLY EIGENVECTORS С INF00120 C INF00130 С INPUT EV FILE; 12 : OUTPUT INF. CLIPPED OPT. FEATURE FILE INF00140 Ċ 11 : INF00150 С

```
WAL00580
      IF (IM.GE.6) GO TO 70
                                                                               WAL00590
      WRITE (*,1) IM,K
    1 FORMAT (' IM = ', 12, ', THE SEQ IS ZW2(I, J), J=1, K=', I3)
                                                                               WAL00600
                                                                               WAL00610
      DO 60 I=1,NP1
                                                                               WAL00620
   60 WRITE (*, 3) I, (ZW2 (I, J), J=1, K)
                                                                               WAL00630
    3 FORMAT (14, 2X, 16F4.1)
                                                                               WAL00640
   70 CONTINUE
                                                                               WAL00650
C
                                                                               WAL00660
      CHECK TOTAL NUMBER OF AXIS CROSSINGS FOR EACH WALSH FUNCTIONS
С
                                                                               WAL00670
Ċ
                                                                               WAL00680
      DO 80 J=1,NTVEC
                                                                               WAL00690
      DO 80 I=1,NP1-1
                                                                               WAL00700
      IF (Z (I, J) .NE.Z (I+1, J) ) NZERO (J) = NZERO (J) +1
                                                                               WAL00710
   80 CONTINUE
                                                                               WAL00720
С
      THE FOLLOWING 2 STATEMENTS CAN BE USED FOR INTERNAL CHECKING
                                                                               WAL00730
С
                                                                                WAL00740
С
                                                                               WAL00750
C
      DO 85 I1=1,NTVEC/8
   85 WRITE (11,86) (NZERO (J), J=1+(I1-1)*8, I1*8)
                                                                               WAL00760
C
                                                                               WAL00770
   86 FORMAT (818)
                                                                                WAL00780
      WRITE (*, *) (NZERO (J), J=1, NTVEC)
C
                                                                                WAL00790
      DO 90 J=1, NTVEC
                                                                                WAL00800
       IF (NZERO (J).NE. (J-1)) GO TO 200
                                                                                WAL00810
   90 CONTINUE
                                                                                WAL00820
С
                                                                                WAL00830
       SEND THE RESULTS TO OUTPUT FILE
С
                                                                                WAL00840
C
                                                                                WAL00850
       DO 140 J=1,NTVEC
                                                                                WAL00860
       DO 140 K=1, NP1/NF4
                                                                                WAL00870
  140 WRITE (11, 4) (Z (I, J), I=1+ (K-1) *NF4, K*NF4)
                                                                                WAL00880
C
                                                                                WAL00890
       CHOOSE FORMAT (10F8.1) IF NF4=10 INSTEAD OF 5
С
                                                                                WAL00900
С
                                                                                WAL00910
    4 FORMAT (10F8.1)
C
                                                                                WAL00920
     4 FORMAT (5E15.7)
                                                                                WAL00930
  200 STOP
                                                                                WAL00940
       END
```

20	FIND INFINITE CLIPPED VERSION FOR EVERY OPTIMAL FUNCTION	INF00330
	물건물 가장 같은 것 같은 것 같아요. 것 같아요. 이 것 같아요. 이 것 같아요. 나는 것 같아요.	INF00340
15	DO 50 ITERM=1,NTERM	INF00350
·. ·	DO 20 J=1.NP1/5	INF00360
20	READ (11, *) (X(I), $I=1+(J-1)$ *5, J*5)	INF00370
- <b>-</b> -	XN1=1./SORT(XNP1)	INF00380
	$DO_{30}$ $J=1.NP1$	INF00390
	TF (NP1 EO 100 AND J. GE 45. AND J. LE 54) $X(J) = 0.0$	INF00400
· '* •	TE (NP1, EQ, 100, AND, J, GE, 70, AND, J, LE, 79) $X(J) = 0.0$	INF00410
	TF (NP1 EO 200 AND J. GE 1 AND J. LE 3) $X(J) = 0.0$	INF00420
	TF(NP1, EO, 200, AND, J, GE, 193, AND, J, LE, 200) X(J) = 0.0	INF00430
•	TF(X(J), GT, 0, 0)X(J) = XN1	INF00440
	TF(X(J), LT, 0, 0)X(J) = -XN1	INF00450
30	CONTINUE	INF00460
	이 같은 것같은 것 같아요. 그는 그 그는 요. 그는 것 같아요. 그는 그는 요. 그는 것 같아요. 그는 그는 요. 그는 그는 요. 그는 그는 그는 요. 그는 그는 요. 그는 그는 요. 그는 그는 그는 요. 그는 그는 요. 그는 그는 그는 요. 그는 그는 요. 그는 그는 요. 그는 요. 그는 요. 그는 그는 요. 그는 그는 요. 그는 그는 요. 그는 요. 그는 요. 그는 요. 그는 요. 그는 그는 요. 그는 요. 그는 요. 그는 요. 그는 요. 그는 요.	INF00470
	SEND THE RESULT TO THE OUTPUT FILE	INF00480
	그렇게 잘 들어야 한다. 그는 것은 것은 것은 것은 것이 같아요. 것이 같아요. 것은 것이 같아요.	INF00490
	DO 40 J=1,NP1/5	INF00500
40	WRITE (12, 41) (X (I), I=1+(J-1)*5, J*5)	INF00510
41	FORMAT (5E15.7)	INF00520
50	CONTINUE	INF00530
:	STOP	INF00540
	END	INF00550
· •	그는 것들에 있는 것이 것을 많이 못 것을 위해 집을 알려요. 영화 관람들이 있는 것을 수 있는 것을 했다.	
	이 같은 것이 같은 것이 가지 않는 것이 같은 것이 같은 것이 같은 것이 같을 것이 같을 것이다.	
	이번 승규는 것이 지수는 것이 집에 있는 것이 많이 많이 가지 않는 것이 없다.	
	PROGRAM OLBS	OLB00010
	PARAMETER (NP1=100, NTERM=6, NV=120, NZ1=2*NV, NZ2=NP1*NV,	OLB00020
	+N1=1, N2=100, W1=0.40, DW=0.02, NVX=40, NV2=NV*NV)	OLB00030
	REAL X (NP1, NTERM), S1 (NP1), Z (NP1, NV), T1 (NV),	OLB00040
-	+TEST (NP1, NV), A (NP1, NVX)	OLB00050
	INTEGER NX (NTERM), NEDGE (2, NV), NWID (NV), NRANK (NV), NREP (NV),	OLB00060
	+MREP (NV)	OLB00070
	DATA NX.Z.NEDGE, NWID/NTERM*0, NZ2*0.0, NZ1*0, NV*0/	OLB00080
	DATA NREP. TEST/NV*1.NZ2*0.0/	OLB00090
		OLB00100
	NP1 • RAW DATA DIMENSIONALITY	OLB00110

IF (NP1.EQ.100) XNP1=FLOAT (NP1-20) IF (NP1.EQ.200) XNP1=FLOAT (NP1) READ INPUT EIGENVECTORS FOR TWO POSSIBLE CASES IF (IEV.EQ.0)GO TO 15

FIND NORMALIZATION FACTOR

OPEN(11)

OPEN(12)

С

С

С

С

С

C

С

С

С

С C C REWIND 11

REWIND 12

READ (11, \*) X1, X2

DO 10 I=1,NP1/5

10 READ(11,\*)X1,X2,X3,X4,X5

113

INF00210 INF00220 INF00230 INF00240 INF00250

INF00160

INF00170

INF00180

INF00190

INF00200

INF00260

INF00270

INF00280

INF00290

INF00300

INF00310

INF00320

INF00330

NP1

С

С

RAW DATA DIMENSIONALITY

TOTAL NUMBER OF OPTIMAL FUNCTIONS USED IN THE ALGORITHM OLB00120 NTERM : PRESET TOTAL NUMBER OF L.D. BANDS, INCREASE IT IF NEEDEDOLB00130 NV OLB00140 N1 STARTING WAVELENGTH POINT : OLB00150 N2 ENDING WAVELENGTH POINT STARTING WAVELENGTH IN MICRO METER ( UM ) OLB00160 Wl OLB00170 SPECTRAL RESOLUTION ( UM ) DW • PRESET TOTAL NUMBER OF L.I. BANDS, INCREASE IT IF NEEDEDOLB00180 NVX . OLB00190 OLB00200 INPUT EIGENVECTOR MATRIX X SIGNED VERSION OF THE EIGENVECTOR OLB00210 S1 . OLB00220 Z L.D. BAND FEATURES -**T1** : TEMPORARY STORAGE VECTOR OLB00230 OUTPUT O.L. BAND FEATURES ( L.I. FEATURES ) OLB00240 TEST . INTERMEDIATE MATRIX FOR RANK TEST OLB00250 Α : TOTAL NO. OF L.D. BANDS FOR THE FIRST K EIGENVECTOR (S) OLB00260 NX (K) NEDGE : BAND EDGES FOR EACH L.D. BANDS OLB00270 NWID : BAND WIDTH FOR EACH L.D. BANDS OLB00280 NRANK : POSITIONS OF THE RANKED FEATURES BY THE WIDTHS OLB00290 INDEX SHOWS IF THE L.D. BANDS ARE REPEATED OLB00300 NREP . INDEX SHOWS IF THE BANDS ARE L.I. BANDS OLB00310 MREP : OLB00320 NREP = 1 IF NON-REPEATED BAND ; NREP = 0 IF REPEATED OLB00330 MREP = 0 IF L.D.OLB00340 MREP = 1 IF L.I. BAND ; OLB00350 11 : INPUT EIGENVECTOR FILE OLB00360 12 : FIRST OUTPUT FILE ---- L.D. AND L.I. BAND INFORMATION С OLB00370 13 : SECOND OUTPUT FILE ---- DESIRED O.L. BAND FEATURE OLB00380 OLB00390 OLB00400 OPEN (11) **OPEN (12)** OLB00410 OLB00420 OPEN (13) OLB00430 REWIND 11 OLB00440 REWIND 12 OLB00450 REWIND 13 OLB00460 OLB00470 READ IN EIGENVECTORS OLB00480 READ (11, \*) X1, X2 OLB00490 OLB00500 DO 10 I=1,NP1/5 10 READ (11, \*) X1, X2, X3, X4, X5 OLB00510 OLB00520 DO 20 J=1,NTERM OLB00530 DO 20 I=1, NP1/5 20 READ (11, \*) (X (K, J), K=1+(I-1)\*5, I\*5) OLB00540 OLB00550 FIND THE L.D. BAND FEATURES FROM FIRST 'NTERM' OPTIMAL FUNCTIONS OLB00560 OLB00570 IVEC=1 OLB00580 DO 70 J=1,NTERM OLB00590 OLB00600 DO 40 I=1,NP1 IF(X(I,J).LT.0.0)SI(I) = -1.0OLB00610 IF(X(I, J).GT.0.0)SI(I) = +1.0OLB00620 IF (X(I,J).EQ.0.0) S1(I)=0.0 OLB00630 IF (NP1.NE.100) GO TO 40 OLB00640 IF (I.GE.45.AND.I.LE.54) S1 (I) =0.0 OLB00650 IF (I.GE. 70.AND.I.LE. 79) S1 (I) =0.0 OLB00660 OLB00670 **40 CONTINUE** Z (N1, IVEC) = ABS (S1 (N1)) OLB00680

Ċ

C

C

Ċ

С

C

C

C

C

С

С

C

C

Ċ

C

С

С

С

C:

С

С

Ċ

Ć

Ċ С

C

C

Ċ

C

С

С C

С

DO 60 I=N1+1,N2 OLB00690 IF (NP1.NE.100) GO TO 50 OLB00700 IF (I.GE. 45. AND. I. LE. 54) GO TO 60 OLB00710 IF (I.GE. 70. AND. I.LE. 79) GO TO 60 OLB00720 50 IF (S1 (I-1).NE.S1 (I)) IVEC=IVEC+1 OLB00730 IF (IVEC.GT.NV) GO TO 350 OLB00740 Z(I, IVEC) = ABS(S1(I))OLB00750 60 CONTINUE OLB00760 NX(J) = IVECOLB00770 IVEC=IVEC+1 OLB00780 70 CONTINUE OLB00790 OLB00800 FIND THE BAND EDGES AND BAND WIDTH FOR EACH L.D. BAND FEATURES OLB00810 1.25 OLB00820 NVTOT=NX (NTERM) OLB00830 DO 90 J=1, NVTOT OLB00840 I1=0 OLB00850 I2=0 OLB00860 DO 80 I=1,NP1 OLB00870 CK1=Z(I,J)OLB00880 IF (CK1.EQ.0.0) GO TO 80 OLB00890 IF (CK1.NE.0.0.AND.I1.EQ.0) I1=I OLB00900 IF (CK1.NE.0.0.AND.I1.NE.0) 12=1 OLB00910 **80 CONTINUE** OLB00920 IF (I2.EQ.0) I2=I1 OLB00930 NEDGE (1, J) = I1OLB00940 NEDGE (2, J) = I2OLB00950 NWID (J) = 12 - 11 + 1OLB00960 90 CONTINUE OLB00970 OLB00980 FIND THE WAVELENGTH EDGES AND SEND THEM TO THE FIRST OUTPUT FILE OLB00990 OLB01000 DO 100 J=1, NTERM OLB01010 WRITE (12, \*) J OLB01020 IF(J.EQ.1)NS1=NX(J)OLB01030 IF (J.NE.1) NS1=NX(J) -NX(J-1)OLB01040 DO 100 I=1,NS1 OLB01050 IF(J.EQ.1)NS2=IOLB01060 IF(J.NE.1)NS2=I+NX(J-1)OLB01070 I1=NEDGE(1,NS2) OLB01080 12 = NEDGE(2, NS2)OLB01090 XW1=W1+FLOAT (I1-1) \*DW OLB01100 XW2=W1+FLOAT(I2)\*DW OLB01110 WRITE (12,101) NS2, I, NEDGE (1, NS2), NEDGE (2, NS2), XW1, XW2, NWID (NS2) OLB01120 100 CONTINUE OLB01130 101 FORMAT (215, 2X, 13, 1X, '-', 13, 2X, '; ', F5.2, 1X, '-', F5.2, 15) OLB01140 PRINT\*, 'TOTAL NUMBER OF BANDS IS = ', NVTOT OLB01150 OLB01160 RANK THE L.D. BAND ACCORDING TO THEIR WIDTHS IN DESCENDING ORDER OLB01170 AND SEND THE RESULTS TO THE FIRST OUTPUT FILE OLB01180 OLB01190 DO 110 I=1,NV OLB01200 110 T1 (I) =FLOAT (NWID(I))OLB01210 DO 120 I=1, NVTOT OLB01220 CALL VABMXF (T1(1), NV, 1, IMAX, BIG) OLB01230 NRANK (I) = IMAXOLB01240 WRITE (12, \*) I, NRANK (I), NEDGE (1, IMAX), NEDGE (2, IMAX), NWID (IMAX) OLB01250

C

Ċ

С

С

C.

С

С

C

С

С

9. 90 i -

OLB01260 120 T1(IMAX) = 0.0OLB01270 С CHECK IF THE L.D. BAND IS REPEATED. IF IT IS, SET NREP(I) = 0 OLB01280 С OLB01290 Ċ OLB01300 DO 140 I=1, NVTOT OLB01310 DO 130 J=1, NVTOT OLB01320 IF (I.EQ.J) GO TO 130 OLB01330 I1=NRANK(I) OLB01340 12=NRANK(J) OLB01350 I3=NWID(I1) OLB01360 I4=NWID(I2) OLB01370 IF (13.NE.14) GO TO 130 OLB01380 ISTART=NEDGE (1, I1) OLB01390 JSTART=NEDGE(1, 12) OLB01400 IEND=NEDGE (2, I1) OLB01410 JEND=NEDGE (2, I2) IF (ISTART.EQ.JSTART.AND.IEND.EQ.JEND.AND.I.GT.J) NREP (I)=0 OLB01420 OLB01430 **130 CONTINUE** OLB01440 IF (NREP (I) . EQ. 0) GO TO 140 OLB01450 IX=NRANK(I) OLB01460 C THE FOLLOWING WRITE STATEMENT CAN BE USED FOR INTERNAL CHECKING OLB01470 C **OLB01480** С WRITE (12,131) I, NREP (I), NRANK (I), NEDGE (1, IX), NEDGE (2, IX), NWID (IX) OLB01490 С OLB01500 131 FORMAT (314, 5X, 14, '-', 14, 5X, 14) OLB01510 140 CONTINUE OLB01520 Ċ FIND TOTAL NUMBER OF NON-REPEATED L.D. BAND OLB01530 C. OLB01540 С OLB01550 NDIFF=0 OLB01560 DO 150 I=1, NVTOT OLB01570 IF (NREP (I).EQ.1) NDIFF=NDIFF+1 OLB01580 150 MREP(I) = NREP(I) PRINT\*, 'TOTAL NUMBER OF NON-IDENTICAL BANDS IS =', NDIFF OLB01590 OLB01600 C OLB01610 FIND L.I. BAND BY CHECKING THE MATRIX RANK С OLB01620 С OLB01630 ILI=1 OLB01640 JWID=1 OLB01650 DO 300 J=1, NVTOT OLB01660 IF (NREP (J) . EQ. 0) GO TO 300 OLB01670 JR=NRANK (J) OLB01680 DO 160 I=1,NP1 OLB01690 160 TEST(I, ILI) = Z(I, JR)OLB01700 DO 170 KI=1,NP1 OLB01710 DO 170 KJ=1, ILI OLB01720 170 A(KI, KJ) = TEST(KI, KJ)OLB01730 С **OLB01740** REDUCE THE MATRIX A TO ITS ECHELON FORM С OLB01750 С OLB01760 CALL ECHEL (A, NP1, NVX, NP1, ILI) OLB01770 IEV=0 OLB01780 DO 190 KI=1,NP1 OLB01790 DO 180 KJ=1, ILI OLB01800 IF(A(KI,KJ).NE.0.0)IEV=IEV+1OLB01810 IF (A (KI, KJ) .NE.0.0) GO TO 190 OLB01820 180 CONTINUE

Call S

OLB01830 **190 CONTINUE** OLB01840 C С SEND THE RANK INFORMATION TO THE FIRST OUTPUT FILE OLB01850 WHERE 'IEV' IS THE RANK AND 'ILI' IS TOTAL NUMBER OF BANDS TESTED OLB01860 С OLB01870 С WRITE (12, \*) 'IEV=', IEV, '; ILI=', ILI, 'AT J=', J OLB01880 OLB01890 IF (IEV.LT.ILI) WRITE (12, \*) 'IEV.LT.ILI AT J=', J IF (IEV.LT.ILI) GO TO 200 OLB01900 OLB01910 Ċ IF RANK IS EQUAL TO TOTAL NO. OF BANDS, TEST THE NEXT WIDEST BAND OLB01920 С OLB01930 С IF (IEV.EQ.ILI) ILI=ILI+1 OLB01940 OLB01950 GO TO 300 OLB01960 С OLB01970 IF RANK IS LESS THEN TOTAL NO. OF BANDS, C OLB01980 С ELIMINATE THE WIDEST L.D. BAND OLB01990 С OLB02000 200 DO 250 JXLD=1,ILI OLB02010 DO 210 KJ=1,ILI OLB02020 DO 210 KI=1,NP1 210 A (KI, KJ) = TEST (KI, KJ) OLB02030 OLB02040 DO 220 KI=1, NP1 OLB02050 220 A(KI, JXLD) = TEST(KI, ILI) OLB02060 JLI=ILI-1 OLB02070 CALL ECHEL (A, NP1, NVX, NP1, JLI) OLB02080 IEV=0 OLB02090 DO 240 KI=1,NP1 OLB02100 DO 230 KJ=1,JLI OLB02110 IF (A (KI, KJ) .NE.0.0) IEV=IEV+1 OLB02120 IF (A (KI, KJ) .NE.0.0) GO TO 240 OLB02130 230 CONTINUE OLB02140 240 CONTINUE PRINT\*, 'IEV=', IEV, '; ILI=', ILI, 'AT J=', J OLB02150 С OLB02160 IF (IEV.LT.ILI) PRINT\*, 'IEV.LT.ILI AT J=', J C OLB02170 IF (IEV.EQ.JLI) J2LD=JXLD OLB02180 IF (IEV.EQ.JLI) GO TO 260 OLB02190 250 CONTINUE OLB02200 260 I1=0 OLB02210 I2=0 OLB02220 DO 270 KI=1,NP1 OLB02230 CK1=TEST (KI, J2LD) OLB02240 IF (CK1.EQ.0.0) GO TO 270 OLB02250 IF (CK1.NE.0.0.AND.I1.EQ.0) I1=KI OLB02260 IF (CK1.NE.0.0.AND.I1.NE.0) I2=KI OLB02270 270 CONTINUE OLB02280 IF (I2.EQ.0) I2=I1 OLB02290 DO 275 KI=1, NVTOT OLB02300 IF (MREP (KI).EQ.0) GO TO 275 OLB02310 MAX=NRANK(KI) OLB02320 MEDGE1=NEDGE (1, MAX) OLB02330 MEDGE2=NEDGE (2, MAX) OLB02340 IF (I1.EO.MEDGE1.AND.I2.EQ.MEDGE2) J1LD=KI IF (I1.EQ.MEDGE1.AND.I2.EQ.MEDGE2) GO TO 280 OLB02350 OLB02360 275 CONTINUE OLB02370 280 MREP (J1LD)=0 OLB02380 С SEND THE POSITION OF THE WIDEST L.D. BAND FEATURE OLB02390 Ċ

```
OLB02400
      TO THE FIRST OUTPUT FILE WHERE :
С
                                                                              OLB02410
Ċ
      JILD IS THE POSITION ON THE VARIABLES NREP AND MREP
                                                                              OLB02420
С
                                                                              OLB02430
      J2LD IS THE POSITION ON THE RANK CHECKING MATRIX
С
                                                                              OLB02440
С
      WRITE (12, *) 'J =', J, '; J1LD =', J1LD, '; J2LD =', J2LD
                                                                              OLB02450
                                                                              OLB02460
      DO 290 J1=J2LD, ILI-1
                                                                              OLB02470
      DO 290 I1=1,NP1
                                                                              OLB02480
  290 TEST (I1, J1) = TEST (I1, J1+1)
                                                                              OLB02490
  300 CONTINUE
                                                                              OLB02500
С
                                                                              OLB02510
      SEND THE L.I. INDEX TO THE FIRST OUTPUT FILE
С
                                                                              OLB02520
C
                                                                              OLB02530
      PRINT*, 'TOTAL NUMBER OF L.I. BANDS IS =', IEV
                                                                              OLB02540
      DO 310 I=1, NVTOT
                                                                              OLB02550
  310 WRITE (12, *) I, NREP (I), MREP (I)
                                                                              OLB02560
С
      NORMALIZE THE O.L. BANDS AND SEND THEM TO THE SECOND OUTPUT FILE
                                                                              OLB02570
C
                                                                              OLB02580
Ĉ
                                                                              OLB02590
      DO 330 J=1, IEV
                                                                              OLB02600
      XN1=0.0
                                                                              OLB02610
      DO 320 I=1,NP1
                                                                              OLB02620
      IF (TEST (I, J) .EQ.1) XN1=XN1+1
                                                                              OLB02630
  320 CONTINUE
      DO 330 I=1,NP1
                                                                              OLB02640
  330 TEST (I, J) = TEST (I, J) / SQRT (XN1)
                                                                              OLB02650
                                                                              OLB02660
      DO 340 J=1, IEV
                                                                              OLB02670
       DO 340 K=1.NP1/5
                                                                              OLB02680
      J1=IEV-J+1
C
  340 WRITE (13, 341) (TEST (I, J), I=1+(K-1)*5, K*5)
                                                                              OLB02690
                                                                              OLB02700
  341 FORMAT (5E15.7)
                                                                              OLB02710
       GO TO 360
  350 PRINT*, 'TOTAL NUMBER OF BANDS IS OUT OF PRESET LIMIT'
                                                                              OLB02720
                                                                              OLB02730
  360 STOP
                                                                              OLB02740
       END
                                                                              OLB02750
       SUBROUTINE ECHEL (A, NP1, NVX, NROW, NCOL)
                                                                              OLB02760
       REAL A (NP1, NVX)
                                                                              OLB02770
Ċ
       THIS SUBROUTINE REDUCES MATRIX A INTO ITS ECHELON FORM
                                                                              OLB02780
С
                                                                              OLB02790
C
                                                                              OLB02800
       JCOL=1
                                                                              OLB02810
       IROW=1
                                                                              OLB02820
     5 DO 100 I=IROW, NROW
       IF (A (I, JCOL) . EQ. 0. 0) GO TO 100
                                                                              OLB02830
       INTERCHANGE I AND IROW TO GET NONZERO PIVOT
                                                                              OLB02840
С
                                                                              OLB02850
       IF (I.EQ.IROW) GO TO 20
                                                                              OLB02860
       DO 10 J=JCOL, NCOL
                                                                              OLB02870
       X1=A(I,J)
       A(I, J) = A(IROW, J)
                                                                              OLB02880
                                                                              OLB02890
   10 A (IROW, J) = X1
                                                                              OLB02900
       NORMALIZE ROW TO GET POSITIVE NUMBER FOR PIVOT
C
                                                                              OLB02910
   20 IF (A (IROW, JCOL).GT.0.0) GO TO 40
                                                                              OLB02920
       DO 30 J=JCOL, NCOL
                                                                              OLB02930
    30 A (IROW, J) = -A (IROW, J)
                                                                              OLB02940
    40 IF (IROW.GE.NROW) RETURN
                                                                              OLB02950
С
       ZERO COLUMN BELOW PIVOT
                                                                              OLB02960
       IROWX=IROW+1
```

118

	PARAMETER (NTERM=16, MTERM=NTERM* (NTERM+1) /2, NCLS=3, NP1=100,	CLS00020
	+NSET=1, MSET=1, NDSET=1, NTSET=1, NF2=10, NF3=5, NSMAX=1000,	CLS00030
	+NKLT=0, IEV=0, NLI=16, VLD=-0.0, NSAMP=10, NF=NF2)	CLS00040
		CLS00050
	NKLT = 1 : JUST FIND TRANSFORMED DATA XKLT	CLS00060
	NKLT = 0 : FIND XKLT AND CLASS STATISTICS	CLS00070
		CLS00080
	NF = NF2 = 10 USED TO READ (10F8.3) RAW DATA	CLS00090
	NF = NF3 = 5 USED TO READ (5E15.7) CANONICAL TRANSFORMED DATA	CLS00100
	WHEN : NF = NF3 = 5> NP1 MUST BE REDUCED TO LOWER DIM.	CLS00110
	이 가슴을 통해 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 있다. 이 것이 있는 가 있는 것이 없다. 이 있는 것이 있는 것이 있는 것이 있는 것이 없는 것이 있는 것이 없는 것이 있는 것이 없는 것이 없는 것이 없다. 것이 있는 것이 없는 것이 없다. 것이 없는 것이 없다. 것이 없는 것이 없는 같이 없는 것이 없 않는 것이 없는 것이 있 것이 없는 것이 없 않이 없는 것이 없는 것이 없는 것이 없는 것이 없는 것이 없는 것이 없 않는 것이 없는 것 않는 것이 없는 것이 없 않이	CLS00120
	NTERM = TOTAL NUMBER OF FEATURES (MAY NOT ALL BE NUMERICALLY L.I.)	CLS00130
	NCLS = TOTOAL NUMBER OF INFORMATION CLASSES	CLS00140
	NP1 = DIMENSIONALITY OF INPUT DATA	CLS00150
	NP1 = RAW DATA DIMENSIONALTY IF USED IN DATA PREPROCESSING	CLS00160
	NP1 = TRANSFORMED DATA DIMENSIONALTY IF USED IN CAN. ANAL.	.CLS00170
	IEV = INPUT FEATURE READING INDEX, EITHER 1 OR 0	CLS00180
	IEV = 0 IF FEATURE FILE DOES NOT CONTAIN TRACE & EVALUES	CLS00190
	IEV = 1 IF FEATURE FILE CONTAINS TRACE & EVALUES	CLS00200
	NLI = TOTAL NUMBER OF L.I. FEATURES DESIRED	CLS00210
	NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS	CLS00220
	NSAMP = TOTAL NUMBER OF TEST SAMPLES USED TO CHECK POS. DEF.	CLS00230
		CLS00240
		CLS00250
	REAL X (NSMAX, NTERM), Z (NP1, NTERM), RX (NP1),	CLS00260
	+T1 (NP1), T2 (NP1), T3 (NP1), XT (NP1), XM (NP1), D (NP1),	CLS00270
Ċ	+XMCT (NTERM, NCLS), XMC (NTERM), W (NP1), T (NP1),	CLS00280
	+VCT (MTERM, NCLS), VC (MTERM), CT (NCLS),	CLS00290
	+VCIT (MTERM, NCLS), VCI (MTERM), TEST (NTERM, NTERM),	CLS00300
	+VCIF (NTERM, NTERM), VCF (NTERM, NTERM),	CLS00310
	+VCTF (NTERM, NTERM, NCLS), XMCTF (NTERM, NCLS),	CLS00320
	+VCTLI (MTERM, NCLS), XMTLI (NTERM, NCLS),	CLS00330
	+WK (NTERM), VCV (MTERM), VEC (NSAMP, NTERM)	CLS00340
	INTEGER NBR(6), NST (NCLS, NTSET)	CLS00350
	DOUBLE PRECISION DSEED	CLS00360
•	DATA (NBR(I), I=4,6), W/1,0,0, NP1*1.0/	CLS00370
		CLS00380
		1.1.1

	DO 60 K=TROWX.NROW	OLB02970
	X1=A(K, JCOL)	OLB02980
5. s	IF (X1.EO.0.0) GO TO 60	OLB02990
	DO 50 J=JCOL, NCOL	OLB03000
50	A(K, J) = -X1 * A(IROW, J) + A(K, J)	OLB03010
60	CONTINUE	OLB03020
	IROW=IROW+1	OLB03030
· · .	JCOL=JCOL+1	OLB03040
	GO TO 5	OLB03050
100	CONTINUE	OLB03060
	IF (IROW.GT.NROW) RETURN	OLB03070
	JCOL=JCOL+1	OLB03080
	GO TO 5	OLB03090
100	END	OLB03100
	「海豚」から「子を放ける」が、「この」、「「」、「」、「」、「」、「」、「「」、「」、「」、「」、「」、「」、「」、	

CLS00010

С

PROGRAM CLST

	이 집 방법에 놓다 나는 이 것 집 같아요. 옷에 가지 않는 것 이 것 같아? 동안이 있는 것 것 같아? 말했는 것 같아?	
с	X = TRANSFORMED DATA	CLS00390
Ċ ·	$\mathbf{Z}$ = FEATURES	CLS00400
С	RX = TEMPORARY STORAGE FOR FEATURES	CLS00410
С	$\mathbf{XT}$ = INPUT DATA	CLS00420
C	XM = MEAN VECTOR	CLS00430
Ċ	D = EIGENVALUES	CLS00440
č	XMCT = MEAN VECTOR FOR ALL CLASSES	CLS00450
č	XMC = MEAN VECTOR FOR ONE CLASS	CLS00460
č	VCT = COVARIANCE MATRIX FOR ALL CLASSES	CLS00470
č	VC = COVARIANCE MATRIX FOR ONE CLASS	CLS00480
č	VCTT = INVERSE MATRIX OF ALL CLASS COVARIANCE MATRICES	CLS00490
č	VCT = INVERSE MATRIX OF ONE CLASS COVARIANCE MATRIX	CLS00500
č	TEST = INTERNAL MATRIX INVERSION CHECKING MATRIX	CLS00510
č	VCTLI = COV. MATRIX FOR ALL CLASSES BY USING ALL L.I. FEATURES	CLS00520
Ċ	XMTLT = MEAN VECTOR FOR ALL CLASSES BY USING L.I. FEATURES	CLS00530
č	WK = WORKING SPACE FOR IMSL ROUTINES	CLS00540
Ċ.	VCV = COV MATRIX USED TO TEST ITS POSITIVE DEFINITENESS	CLS00550
C .	VEC = GENERATED SAMPLES USED TO TEST POSITIVE DEFINITENESS	CLS00560
č		CLS00570
с	->>CHOOSE OF TYPE IN THE CORRECT NUMBERS OF SAMPLES IN THE DATA SETS	CLS00580
r C		CLS00590
		-CLS00600
C	NGET FI NP2 A B C DACO EXNU RUSE	CLS00610
	NSET 11 M2611 K1 832 $WW \cdot 141$ SF $\cdot 414$ GS $\cdot 277$ 760928 76102207 1-1622	CLS00620
Č	$\frac{1}{2} \frac{M2611K1}{M2611K2} \frac{1551}{1551} \frac{11}{100} \frac{515}{658} \frac{515}{211} \frac{1000}{1000} \frac{1000}$	CLS00630
	2 M2611R2 1051 WW.000 SF.211 00.002 770626 77102207 8097-9691	CLS00640
	$A = M261 AN1 = 1265 SW \cdot 664 SF \cdot 437 NP \cdot 164 770508 77102217 = 1-1396$	CLS00650
	5 M2614N2 1230 SW.787 SF.291 NP.161 770629 77102217 2777-4141	CLS00660
	6 M2614N2 1239 SW.707 SI.231 NE.101 770804 77102217 5426-6993	CLS00670
	0 MZ014NJ 1444 SW. 991 SF. 990 ML. 103 / 0004 //10221 0 0000	CLS00680
	DAMA NOW /1 41 414 277 658 211 682 677 643 157/	CLS00690
	DATA NSI/141,414,277,650,211,002,077,643,157	CLS00700
C	DATA NS1/141,414,217,000,211,002,017,043, $\pm 37$	CLS00710
C	+004,437,104,707,231,101,331,330,1037	CLS00720
C	DATA NS1/004, 431, 104, $707, 231, 101, 331, 330, 1037$	CLS00730
C	DATA NS1/141,414,277,000,211,002,077	CT.S00740
С	DATA NST/ 30 /, 210, 121/	CLS00750
0	DAIA NS1/030/211/002/	CLS00760
		-CLS00770
	THE FOR ONTHE DATA INST! ARE HISED FOR SOTI ORDER DATA SET. 'SO'	CLS00780
	THE FOLLOWING DATA NOT AND COLD LON DOLL CLOCK DIT	CLS00790
č	MD2-470, MOL ALE EN AR UL IN SP VE H OX UNCLASSIFIED	CLS00800
	NPZ-479, MOL ALF EN AR OL IN OL VI II ON UNOLIDOLI 120	CLS00810
	DATA NS1/134,113,70,52,43,37,30,11,0,11,70,11,72	CLS00820
	DATA NO1/104/110, $70, 52, 13, 57$	CLS00830
	DAIA N51/134/113,70/32,40/37/	CLS00840
		-CLS00850
	- 전 이용학을 다 수요한 것이라는 것이 같이 많은 것이라. 이상 가지 않는 것이 하나 있는 것이 것을 통을 통하는 것이 있는 것을 통을 수 있다. 이상 가지 않는 것이 있는 것이 있는 것이 없는 방	CT.SOO860
	THE TALL ONLY THE DATA INCOME TO LICED FOR COTT LONG DATA SET	CLS00870
C a	THE FOLLOWING DATA NOT IS USED FOR SOLL ONL DATA SET	CT/S00880
C C	T.P. (I) MORTID ODCANIC WALLDIAL & THELCAL	CTSUDED
C C	AUGORDING TO THEIR ORGANIC MAIERIAL: 8 WEIGHI	00000210
C	$\begin{array}{c} \text{ULASS 1} & \text{IU } 0 \\ \text{TO1} & \text{TO2} \\ \end{array} \begin{array}{c} \text{NP2} = 233 \\ \text{OP} & \text{OM} \\ \end{array} \begin{array}{c} \text{IP} & \text{I} \\ \text{F} \end{array} \begin{array}{c} \text{I} \\ \text{I} \\ \text{I} \\ \end{array} \begin{array}{c} \text{H} \\ \text{F} \end{array} \begin{array}{c} \text{I} \\ \text{I} \\ \end{array} $	
C	CLSI: .118 .0E. UM .LE. 1.36 : # 1 ~/ # J1	CT2003T0
C	CLSZ: 1.5% .GT. UM .LE. 2.0% : # 32 -7 # 104	CT200350
C	ULSJ : 2.U8 .GT. UM .LE. 2.38 : # 100 -> # 130	
C	CLS4 : ∠.5% .GT. UM .LE. J.5% : # 139 → # 183	
C	CLS5 : 3.5% .GT. OM .LE. 5.0% : # 184 ÷> # 222	00820

	CLS6 : 5.0% .GT. OM .LE. 10.12% : # 223 -> # 255	CLS00960
,		CLS00970
	DATA NST/51,54,33,45,39,33/	CLS00980
	ANOTHED TEST COMPANY BY THE SAME ON BANCES AS IOM2!	CLS00990
÷	OM PERCENTAGE : 0.1: 1.2: 2.3: 3.4: 4.6: 6 AND ABOVE	CLS01010
, in the second s		CLS01020
	DATA NST/26,78,64,32,55/	CLS01030
		CLS01040
		CLS01050
		CLS01060
	THE FOLLOWING DATA 'NSI' IS USED FOR 'OMZ' DATA SET ACCORDING TO THETE OPCINIC MATERIAL & WEIGHT	CLS01070
	CLASS 1 TO 6 $\cdot$ NP2 = 514	CLS01090
	CLS1 : .08% .GE. OM .LE. $1.0\%$ : # 1 -> # 82	CLS01100
e a	CLS2 : 1.0% .GT. OM .LE. 2.0% : # 83 -> # 217	CLS01110
	CLS3 : 2.0% .GT. OM .LE. 3.0% : # 218 -> # 337	CLS01120
	CLS4 : 3.0% .GT. OM .LE. 4.0% : # 338 -> # 391	CLS01130
	CLS5 : 4.0% .GT. OM .LE. 6.0% : # 392 -> # 450	CLS01140
	CLS6 : 6.0% .GT. OM .LE. 84./9% : # 451 -> # 514	CLS01150
	איז אופיין 125 120 54 50 64/	CLS01100
	DATA NS1/02,133,120,34,33,04/	CLS01180
	DATA NST/82.135.120.54.123/	CLS01190
	DATA NST/44, 31, 18, 23, 24, 51, 37, 27/	CLS01200
	DATA NST/83, 57, 94, 31, 37, 59, 103, 26, 24/	CLS01210
	DATA NST/103,26,24/	CLS01220
		CLS01230
	THE FOLLOWING DATA 'NST' IS USED FOR SOIL IRON OXIDE '10' DATA SET	ICLS01240
	ACCORDING TO THEIR FEZOS & WEIGHT CLASS 1 TO 6 $\cdot$ NP2 = 467	CLS01250
	CLASS 1 10 0 . M12 - $407$ CLS1 : .02% .GE. FE2O3 .LE. 0.4% : # 1 -> # 102	CLS01270
	CLS2 : 0.4% .GT. FE2O3 .LE. 0.6% : # 103 -> # 175	CLS01280
	CLS3 : 0.6% .GT. FE2O3 .LE. 0.8% : # 176 -> # 244	CLS01290
	CLS4 : 0.8% .GT. FE2O3 .LE. 1.2% : # 245 -> # 349	CLS01300
	CLS5 : 1.2% .GT. FE2O3 .LE. 1.6% : # 350 -> # 401	CLS01310
	CLS6 : 1.6% .GT. FE2O3 .LE. 25.60% : # 402 -> # 467	CLS01320
	DAMA NOM/102 72 60 105 52 66/	CLS01330
	DATA NST/102, 75, 69, 105, 52, 66/	CL301340
		CLS01360
	THE FOLLOWING DATA 'NST' IS USED FOR SOIL TEXTURE 'ST' DATA SET	CLS01370
	ACCORDING TO THEIR SAND-SILT-CLAY % CONTENT	CLS01380
	CLASS 1 TO 6 : NP2 = 483; DETAILS : SEE FILE ( S5L.DATA.C1)	CLS01390
		CLS01400
	DATA NST/40,63,76,93,68,143/	CLS01410
		CLS01420 CLS01430
	THE FOLLOWING DATA 'NST' IS USED FOR S.D. VEGETATION DATA	CLS01440
		CLS01450
	DATA NST/225,61,292,469, 82,182,63,103, 39,39,217,51,	CLS01460
	+393,441,80,88, 88,41,32,26, 118,43,121,44, 45,102,66,89,	CLS01470
	+78,53,147,39, 24,42,119,69, 76,96,107,154, 28,19/	CLS01480
	이 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 같은 것이 있는 것이 있 같은 것이 있는 것	CLS01490
		CLS01500
	THE FOLLOWING DATA 'NST' IS USED FOR IOWA VEGETATION DATA	CLS01510

DATA NST/514,41, 517,36,32, 621,517,45, 610,485,21, CLS01530 +437, 377, 22, 190, 172, 25, 650, 568, 42, 435, 417, 44, 393, 267/ CLS01540 CLS01550 CLS01560 CLS01570 CLS01580 11 = DATA; 12 = FEATURES; 13 = CLASS STATISTICS; 14 = TRANSFORMED DATA ; 15 = LDBAND ; 16 = RANDOM CLS01590 CLS01600 CLS01610 OPEN(11) CLS01620 **OPEN** (12) CLS01630 **OPEN** (13) CLS01640 OPEN(14) CLS01650 OPEN (15) CLS01660 REWIND 11 CLS01670 REWIND 12 CLS01680 **REWIND 13** CLS01690 REWIND 14 CLS01700 REWIND 15 CLS01710 SET UP DATA INPUT&OUTPUT DO LOOP PARAMETERS CLS01720 CLS01730 CLS01740 IK1=MOD (NCLS, 6) CLS01750 IM1=6\*(NCLS/6)+1CLS01760 ILP1=NCLS/6 CLS01770 IF (ILP1.EQ.0) ILP1=1 CLS01780 IK2=MOD (NTERM, 5) CLS01790 IM2=5\*(NTERM/5)+1CLS01800 ILP2=NTERM/5 CLS01810 IF (ILP2.EQ.0) ILP2=1 CLS01820 DO 650 ISET=NSET, MSET, NDSET CLS01830 C READ FEATURE FILE IN TWO CASES ( IEV = 0 OR 1 ) C CLS01840 CLS01850 С CLS01860 IF (IEV.EQ.0) GO TO 10 CLS01870 READ (12, \*) TRACE, SUM CLS01880 CALL SR1 (12, NP1, NF3, D) CLS01890 10 DO 30 JTERM=1, NTERM CLS01900 CALL SR1 (12, NP1, NF3, RX) CLS01910 DO 20 I=1,NP1 CLS01920 20 Z(I, JTERM) = RX(I) CLS01930 **30 CONTINUE** CLS01940 C FIND MEAN VECTOR AND COVARIANCE MATRIX FOR EACH CLASS CLS01950 С IN THE FEATURE TRANSFORMED DATA CLS01960 С CLS01970 C CLS01980 DO 150 LTERM=NTERM, NTERM CLS01990 KTERM=LTERM\* (LTERM+1)/2 CLS02000 DO 150 ICLS=1, NCLS NS=NST (ICLS, ISET) CLS02010 PRINT\*, ' ISET =', ISET, ';', LTERM, ICLS, NS CLS02020 CLS02030 DO 40 I=1,NSMAX CLS02040 DO 40 J=1, NTERM 40 X(I, J) = 0.0CLS02050 CLS02060 DO 100 ISAMP=1,NS CLS02070 CALL SR1 (11, NP1, NF, XT) CLS02080 DO 70 JTERM=1, LTERM CLS02090 DO 60 I=1,NP1

122

С

С

C

С

С

C

Ċ

С

С

С

C

T1(I) = XT(I)CLS02100 T2(I) = W(I) \* T1(I)CLS02110 60 T3(I)=Z(I, JTERM) CLS02120 CALL VIPRFF (T3, T2, NP1, 1, 1, XIP) CLS02130 70 X (ISAMP, JTERM) = XIP CLS02140 С CLS02150 С SEND THE RESULTS TO THE TRANSFORMED DATA FILE CLS02160 C CLS02170 IF (NTERM.LT.5) GO TO 90 CLS02180 DO 80 I1=1, ILP2 CLS02190 80 WRITE (14, 91) (X (ISAMP, J1), J1=1+(I1-1)\*5, I1\*5) CLS02200 IF (IK2.EQ.0) GO TO 100 CLS02210 90 WRITE (14, 91) (X (ISAMP, J1), J1=IM2, NTERM) CLS02220 91 FORMAT (5E15.7) CLS02230 100 CONTINUE CLS02240 С CLS02250 С FIND THE CLASS STATISTICS IF NKLT = 0CLS02260 С CLS02270 IF (NKLT.EQ.1) GO TO 150 CLS02280 NBR(1)=LTERM CLS02290 NBR(2) = NSCLS02300 NBR(3) = NSCLS02310 DO 110 I=1, NP1 CLS02320 110 T(I) = 0.0CLS02330 CALL BECOVM (X, NSMAX, NBR, T, XMC, VC, IER) CLS02340 С CLS02350 C STORE THE CLASS STATISTICS FOR POSITIVE DEFINITENESS CHECKING CLS02360 С CLS02370 DO 120 I=1, LTERM CLS02380 С WRITE (\*, \*) LTERM, I, XMC(I) CLS02390 120 XMCT(I, ICLS) = XMC(I)CLS02400 DO 130 I=1, KTERM CLS02410 С WRITE (\*, \*) LTERM, I, VC(I) CLS02420 130 VCT (I, ICLS) = VC(I)CLS02430 PRINT\*, ' THE IER MUST BE "0" FOR BECOVM ' CLS02440 PRINT\*, IER CLS02450 150 CONTINUE CLS02460 С CLS02470 STOP THE PROGRAM IF ONLY WANT TO FIND TRANSFORMED DATA (NKLT=1) C CLS02480 С CLS02490 IF (NKLT.EO.1) GO TO 650 CLS02500 С CLS02510 С STORE THE CLASS STATISTICS INTO FULL STORAGE MODE FOR CHECKING CLS02520 С CLS02530 DO 170 ICLS=1,NCLS CLS02540 DO 170 I=1,NTERM CLS02550 DO 160 J=1,I CLS02560 IND=I\*(I-1)/2+JCLS02570 VCTF(I, J, ICLS) = VCT(IND, ICLS) CLS02580 VCTF (J, I, ICLS) = VCTF (I, J, ICLS) CLS02590 WRITE (\*, \*) I, J, IND, VCTF (I, J, ICLS), VCTF (J, I, ICLS) CLS02600 C 160 CONTINUE CLS02610 XMCTF(I, ICLS) = XMCT(I, ICLS) CLS02620 170 CONTINUE CLS02630 С CLS02640 START CHECKING THE POSITIVE DEFINITENESS OF THE COV. MATRICES С CLS02650 С IF 'LTERM'TH FEATURE IS L.D. ON THE OTHER FEATURES, THE RELATED CLS02660

ELEMENTS IN THE MEAN VECTORS AND COVARIANCES WILL BE REMOVED CLS02670 CLS02680 CLS02690 ILI=1 CLS02700 JII=III\*(III+1)/2CLS02710 DO 600 LTERM=1, NTERM CLS02720 KTERM=LTERM\* (LTERM+1)/2 CLS02730 DO 400 ICLS=1, NCLS CLS02740 IX=0 CLS02750 DO 200 IROW=1, LTERM CLS02760 V1=0.0 CLS02770 DO 180 JCK=1, LTERM CLS02780 180 V1=V1+VCTF (IROW, JCK, ICLS) CLS02790 VCK=VLD\*LTERM CLS02800 IF (V1.EQ.VCK) GO TO 200 CLS02810 IX=IX+1 CLS02820 IY=0 CLS02830 DO 190 JCOL=1, LTERM CLS02840 V2=VCTF (IROW, JCOL, ICLS) CLS02850 IF (V2.EQ.VLD) GO TO 190 CLS02860 IY=IY+1 CLS02870 VCF(IX, IY) = V2CLS02880 190 CONTINUE CLS02890 200 CONTINUE CLS02900 WRITE (15, \*) IX, IY, ILI C PRINT\*, 'IX, IY, ILI MUST BE THE SAME', IX, IY, ILI CLS02910 Ċ CLS02920 CALL VCVTFS (VCF, ILI, NTERM, VC) CLS02930 WRITE (\*, \*) ICLS, VC (1) Ċ CLS02940 OPEN (16) CLS02950 REWIND 16 CLS02960 DO 210 I=1, JLI CLS02970 WRITE (16, 211) VC (I) CLS02980 VCV(I) = VC(I)C CLS02990 210 VCTLI (I, ICLS) = VC (I) CLS03000 211 FORMAT (E13.5) CLS03010 OPEN (16) CLS03020 REWIND 16 CLS03030 DO 220 I=1,JLI CLS03040 220 READ (16, 211) VCV (I) CLS03050 DO 230 I=1, NTERM CLS03060 230 WK(I) = 0.0CLS03070 DSEED=5.D0 CLS03080 С SECOND TEST ON NUMERICAL POSITIVE DEFINITENESS OF THE MATRICES CLS03090 С CLS03100 C CLS03110 CALL GGNSM (DSEED, NSAMP, ILI, VCV, NSAMP, VEC, WK, IER) CLS03120 IF (IER.NE.0) GO TO 440 CLS03130 WRITE (\*, \*) ICLS, VCTLI (1, ICLS), VC(1) С CLS03140 С CHECK IF ALL CLASS COVARIANCES HAVE INVERSE MATRICES CLS03150 C CLS03160 VC WILL BE CHANGED AFTER LINV1P C CLS03170 С CLS03180 CALL LINV1P (VC, ILI, VCI, IDGT, D1, D2, IER) CLS03190 С WRITE (\*, \*) ICLS, VCI (1) PRINT\*, ' THE FOLLOWING IER MUST BE 0 FOR LINV1P' CLS03200 C CLS03210 PRINT\*, ISET, LTERM, ICLS, '; IER =', IER C CLS03220 IF (IER.NE.0) GO TO 450 CLS03230 DO 240 I=1,JLI

С

C

240 VCIT(I,ICLS)=VCI(I) CLS03240 Ċ CLS03250 С STORE BACK THE VALUES OF VCVC FROM VCVCF CLS03260 С CLS03270 CALL VCVTFS (VCF, ILI, NTERM, VC) CLS03280 CALL VCVTSF (VCI, ILI, VCIF, NTERM) CLS03290 DET=D1\*2.\*\*D2 CLS03300 CX=(2.\*3.14159)\*\*(FLOAT(ILI)/2.) CLS03310 C=1./(CX\*SQRT(DET))CLS03320 CT(ICLS) = CCLS03330 IF (ICLS.NE.NCLS) GO TO 400 CLS03340 С CLS03350 С SEND THE FINAL RESULTS TO THE CLASS STATISTICS FILE CLS03360 С CLS03370 CLS03380 DO 250 KCLS=1, NCLS CLS03390 IX=0 CLS03400 DO 250 I=1, LTERM V3=XMCTF(I,KCLS) CLS03410 IF (V3.EQ.VLD) GO TO 250 CLS03420 CLS03430 IX=IX+1 CLS03440 XMTLI (IX, KCLS) = V3 CLS03450 250 CONTINUE DO 280 I=1,ILI CLS03460 IF (NCLS.LT.6) GO TO 270 CLS03470 CLS03480 DO 260 IL=1, ILP1 260 WRITE (13, 321) (XMTLI (I, LCLS), LCLS=1+(IL-1)\*6, IL\*6) CLS03490 IF (IK1.EQ.0) GO TO 280 CLS03500 270 WRITE (13, 321) (XMTLI (I, LCLS), LCLS=IM1, NCLS) CLS03510 CLS03520 280 CONTINUE CLS03530 DO 310 I=1,JLI IF (NCLS.LT.6) GO TO 300 CLS03540 CLS03550 DO 290 IL=1,ILP1 290 WRITE (13, 321) (VCTLI (I, LCLS), LCLS=1+(IL-1)\*6, IL\*6) CLS03560 IF (IK1.EQ.0) GO TO 310 CLS03570 300 WRITE (13, 321) (VCTLI (I, LCLS), LCLS=IM1, NCLS) CLS03580 310 CONTINUE CLS03590 IF (NCLS.LT.6) GO TO 330 CLS03600 DO 320 IL=1,ILP1 CLS03610 320 WRITE (13, 321) (CT (LCLS), LCLS=1+(IL-1)\*6, IL\*6) CLS03620 321 FORMAT (6E13.5) CLS03630 IF (IK1.EQ.0) GO TO 340 CLS03640 330 WRITE (13, 321) (CT (LCLS), LCLS=IM1, NCLS) CLS03650 CLS03660 340 DO 370 I=1,JLI CLS03670 IF (NCLS.LT.6) GO TO 360 CLS03680 DO 350 IL=1, ILP1 350 WRITE (13, 321) (VCIT (I, LCLS), LCLS=1+(IL-1)\*6, IL\*6) CLS03690 CLS03700 IF (IK1.EQ.0) GO TO 370 360 WRITE (13, 321) (VCIT (I, LCLS), LCLS=IM1, NCLS) CLS03710 CLS03720 370 CONTINUE 400 CONTINUE CLS03730 CLS03740 INTERNAL CHECKING FOR ACCURACY OF MATRIX INVERSION CLS03750 С CLS03760 CLS03770 DO 430 ICLS=1,NCLS DO 410 I=1,JLI CLS03780 VC(I)=VCTLI(I,ICLS) CLS03790 CLS03800 410 VCI(I)=VCIT(I,ICLS)

С

С

```
CLS03810
      DO 420 I=1, JLI
                                                                               CLS03820
 420 WRITE (*, *) VC (I), VCI (I)
C
                                                                              CLS03830
      CALL VMULSS (VC, VCI, ILI, TEST, NTERM)
                                                                               CLS03840
С
      THE FOLLOWING 3 STATEMENTS CAN BE USED FOR MATRIX INVERSION CHECK CLS03850
C
                                                                               CLS03860
C
      PRINT*, ' THE FOLLOWING MATRIX MUST BE AN IDENTITY MATRIX '
                                                                               CLS03870
С
                                                                               CLS03880
C
      DO 430 I=1.ILI
                                                                               CLS03890
      WRITE (*, 421) (TEST (I, J), J=1, ILI)
C
                                                                               CLS03900
  421 FORMAT (16F5.2)
                                                                               CLS03910
  430 CONTINUE
      PRINT*, ' ILI =', ILI
                                                                               CLS03920
                                                                               CLS03930
      TLT=ILI+1
                                                                               CLS03940
      JLI=ILI*(ILI+1)/2
                                                                               CLS03950
      IF (ILI.GT.NLI) GO TO 650
                                                                               CLS03960
      GO TO 600
                                                                               CLS03970
Ċ
      SEND THE INFORMATION OF L.D. FEATURES & REASONS FOR
Ċ
                                                                               CLS03980
      NON-POSITIVE-DEFINITENESS OF COV. MATRICES TO THE FILE 'LDBAND'
                                                                               CLS03990
Ċ
                                                                               CLS04000
C
                                                                               CLS04010
  440 WRITE (15, *) 'GGNSM HAS IER.NE.0'
  450 WRITE (15, *) 'ISET =', ISET, '; LTERM =', LTERM, '; ICLS =', ICLS
                                                                               CLS04020
      PRINT*, 'ISET =', ISET, '; LTERM =', LTERM, '; ICLS =', ICLS
                                                                               CLS04030
                                                                               CLS04040
      DO 500 JCLS=1, NCLS
                                                                               CLS04050
C
      THE FOLLOWING 5 STATEMENTS ARE USED FOR INTERNAL CHECKING
                                                                               CLS04060
С
                                                                               CLS04070
C
                                                                               CLS04080
      WRITE (15, *) 'JCLS =', JCLS
Ċ
                                                                               CLS04090
C
      DO 460 I=1, NTERM
                                                                               CLS04100
 460 WRITE (15, *) I, XMCTF (I, JCLS)
С
                                                                               CLS04110
      DO 470 I=1,NTERM
С
C 470 WRITE (15, 471) I, (VCTF (I, J, JCLS), J=1, NTERM)
                                                                               CLS04120
                                                                               CLS04130
  471 FORMAT (14, 8F9.2)
                                                                               CLS04140
Ĉ
      RESET THE VARIABLES TO '0.0' FOR FUTURE USE
                                                                               CLS04150
С
                                                                               CLS04160
С
      XMCTF (LTERM, JCLS) =VLD
                                                                               CLS04170
                                                                               CLS04180
      DO 480 I=1, NTERM
                                                                               CLS04190
  480 VCTF (I, LTERM, JCLS) = VLD
                                                                               CLS04200
      DO 490 J=1,NTERM
  490 VCTF (LTERM, J, JCLS) =VLD
                                                                               CLS04210
                                                                               CLS04220
  500 CONTINUE
                                                                               CLS04230
С
       THE FOLLOWING 7 STATEMENTS ARE USED FOR INTERNAL CHECKING
                                                                               CLS04240
С
                                                                               CLS04250
С
                                                                               CLS04260
С
      DO 550 JCLS=1,NCLS
                                                                               CLS04270
      WRITE (15, *) 'JCLS =', JCLS
С
                                                                               CLS04280
C
      DO 530 I=1, NTERM
  530 WRITE (15, *) I, XMCTF (I, JCLS)
                                                                               CLS04290
C
                                                                               CLS04300
      DO 540 I=1,NTERM
Ç
  540 WRITE (15, 421) I, (VCTF (I, J, JCLS), J=1, NTERM)
                                                                               CLS04310
C.
                                                                               CLS04320
C 550 CONTINUE
                                                                               CLS04330
  600 CONTINUE
  650 CONTINUE
                                                                               CLS04340
                                                                               CLS04350
       STOP
                                                                               CLS04360
      END
       SUBROUTINE SR1 (IFILE, NP1, NFX, RX)
                                                                               CLS04370
```

126

	그는 그 가장 예정 소설되는 것 같아. 이야기 가지 않는 것 같아요. 이야기 가지 않는 것 같아요.	CLS04380
	THIS SUBROUTINE IS USED TO READ THE INPUT DATA	CLS04390
	같은 것은 것은 것은 것은 것은 것이 같은 것이 같은 것은 것은 것은 것을 가지 않는 것이 없다.	CLS04400
	REAL RX (NP1)	CLS04410
j.	IKX=MOD (NP1, NFX)	CLS04420
:	IMX=NFX* (NP1/NFX) +1	CLS04430
	ILPX=NP1/NFX	CLS04440
ŀ	IF (ILPX.EQ.0) ILPR1=1	CLS04450
	IF (NP1.LT.NFX) GO TO 20	CLS04460
	DO 10 I=1, ILPX	CLS04470
LO	READ (IFILE, *) (RX (J), J=1+ (I-1) *NFX, I*NFX)	CLS04480
	IF (IKX.EQ.0) GO TO 30	CLS04490
20	READ (IFILE, *) (RX(J), J=IMX, NP1)	CLS04500
30	RETURN	CLS04510
	END	CLS04520

127

	PROGRAM CANONIC	CAN00010
	PARAMETER (NTERM=18, MTERM=NTERM* (NTERM+1) /2, NCLS=3,	CAN00020
	+NWK=NTERM* (NTERM+2))	CAN00030
	REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),	CAN00040
	+VCVIT (MTERM, NCLS), D (NTERM), Z (NTERM, NTERM), WK (NWK),	CAN00050
	+WCS (MTERM), ACS (MTERM), WCS1 (MTERM), TEST (NTERM, NTERM),	CAN00060
	+T (NTERM, NTERM), WCSI (MTERM), XMO (NTERM)	CAN00070
•	INTEGER NST (NCLS)	CAN00080
	DATA IJOB. IFLAG1. IOPT. NIN. NOUT/2.0.3.0.6/	CAN00090
		CAN00100
	NTERM = DIMENSIONALITY IN THE CLASS STATISTICS	CAN00110
-	NCLS = TOTAL NUMBER OF INFORMATION CLASSES	CAN00120
		CAN00130
1.1.1	XMT = MEAN VECTORS FOR ALL CLASSES	CAN00140
	VCVT = COV. MATRICES FOR ALL CLASSES	CAN00150
	CT = VARIABLE USED TO STORE M.L. THRESHOLD PARAMETER	CAN00160
	VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES	CAN00170
1.	D = ETGENVALUES	CAN00180
.'	Z = CANONICAL FEATURES	CAN00190
	WK = WORKING SPACE FOR IMSL ROUTINES	CAN00200
÷ .	WCS = WITHIN CLASS SCATTER MATRIX	CAN00210
	ACS = AMONG CLASS SCATTER MATRIX	CAN00220
	WCS1 = TEMPORARY STORAGE FOR WCS	CAN00230
	WCSI = INVERSE MATRIX OF WCS	CAN00240
	XMO = GLOBAL MEAN VECTOR	CAN00250
	TEST = INTERNAL CHECKING FOR MATRIX INVERSION ACCURACY	CAN00260
		CAN00270
		CAN00280
	그렇는 방법 문제에 가지 않는 것이 아이들이 가지 않는 것이 가지 않는 것이 가지 않는 것이 가지 않는 것이 나라.	CAN00290
>:	>CHOOSE OR TYPE IN THE CORRECT NUMBERS OF SAMPLES IN THE DATA SETS	CAN00300
		CAN00310
		-CAN00320
1	NSET F1 NP2 A B C DACO EXNU RUSE	CAN00330
	1 M2611K1 832 WW:141 SF:414 GS:277 760928 76102207 1-1622	CAN00340
	2 M2611K2 1551 WW:658 SF:211 UC:682 770503 77102207 6515-8096	CAN00350
	3 M2611K3 1477 WW:677 SF:643 GS:157 770626 77102207 8097-9691	CAN00360
•	4 M2614N1 1265 SW-664 SE-437 ND-164 770508 77102217 1-1396	CANO0370

C C C

5 M2614N2 1239 SW:787 SF:291 NP:161 770629 77102217 2777-4141 CAN00380 С 6 M2614N3 1444 SW:931 SF:330 NP:183 770804 77102217 5426-6993 CAN00390 С CAN00400 C DATA NST/141, 414, 277, 658, 211, 682, 677, 643, 157/ CAN00410 C DATA NST/141,414,277,658,211,682,677,643,157, CAN00420 С CAN00430 +664,437,164,787,291,161,931,330,183/ С DATA NST/664,437,164,787,291,161,931,330,183/ CAN00440 С DATA NST/141, 414, 277, 658, 211, 682, 677/ CAN00450 С CAN00460 DATA NST/587,216,121/ C CAN00470 DATA NST/658,211,682/ CAN00480 C CAN00490 С THE FOLLOWING DATA 'NST' ARE USED FOR SOIL ORDER DATA SET. 'SO' CAN00500 Ć CAN00510 С CAN00520 NP2=479; MOL ALF EN AR UL IN SP VE H OX UNCLASSIFIED C CAN00530 DATA NST/154, 113, 78, 52, 45, 37, 30, 11, 8, 11, 32/ С CAN00540 DATA NST/154,113,78,52,45,97/ С CAN00550 DATA NST/154,113,78,52,45,37/ С CAN00560 Ċ CAN00570 С CAN00580 С THE FOLLOWING DATA 'NST' IS USED FOR SOIL 'OM1' DATA SET CAN00590 С CAN00600 I.E. (1) MOLLISOL, OR (2) ALFISOL, AND GROUP SAMPLES C ACCORDING TO THEIR ORGANIC MATERIAL: % WEIGHT CAN00610 C CAN00620 CLASS 1 TO 6 : NP2 = 255Ċ CLS1 : .11% .GE. OM .LE. 1.5% : # 1 -> # 51 CAN00630 C: CLS2 : 1.5% .GT. OM .LE. 2.0% : # 52 -> # 104 CAN00640 С CLS3 : 2.0% ,GT. OM .LE. 2.5% : # 105 -> # 138 CAN00650 Ć CLS4 : 2.5% .GT. OM .LE. 3.5% : # 139 -> # 183 CAN00660 Ċ CLS5 : 3.5% .GT. OM .LE. 5.0% : # 184 -> # 222 CAN00670 C CLS6 : 5.0% .GT. OM .LE. 10.12% : # 223 -> # 255 CAN00680 C CAN00690 C CAN00700 DATA NST/51, 54, 33, 45, 39, 33/ С CAN00710 С DATA 'S2A' : ANOTHER TEST GROUPED BY THE SAME OM RANGES AS 'OM2' CAN00720 С OM PERCENTAGE : 0,1; 1,2; 2,3; 3,4; 4,6; 6 AND ABOVE CAN00730 C CAN00740 С CAN00750 C DATA NST/26, 78, 64, 32, 55/ CAN00760 С CAN00770 С CAN00780 С THE FOLLOWING DATA 'NST' IS USED FOR 'OM2' DATA SET CAN00790 С ACCORDING TO THEIR ORGANIC MATERIAL: % WEIGHT CAN00800 С CAN00810 CLASS 1 TO 6 : NP2 = 514С CLS1 : .08% .GE. OM .LE. 1.0% : # 1 -> # 82 CAN00820 С CLS2 : 1.0% ,GT. OM .LE, 2.0% : # 83 -> # 217 CAN00830 С : # 218 -> # 337 CLS3 : 2.0% .GT. OM .LE. 3.0% : # 218 -> # 337 CLS4 : 3.0% .GT. OM .LE. 4.0% : # 338 -> # 391 CLS5 : 4.0% .GT. OM .LE. 6.0% : # 392 -> # 450 CAN00840 С CAN00850 C. CAN00860 C. CLS6 : 6.0% .GT. OM .LE. 84.79% : # 451 -> # 514 CAN00870 C CAN00880 C CAN00890 DATA NST/82,135,120,54,59,64/ C CAN00900 C DATA NST/82,135,120,54,123/ CAN00910 С, CAN00920 DATA NST/44, 31, 18, 23, 24, 51, 37, 27/ C CAN00930 DATA NST/83, 57, 94, 31, 37, 59, 103, 26, 24/ C CAN00940 С DATA NST/103,26,24/

CAN00950 THE FOLLOWING DATA 'NST' IS USED FOR SOIL IRON OXIDE 'IO' DATA SETCAN00960 ACCORDING TO THEIR FE2O3 % WEIGHT CAN00970 CLASS 1 TO 6 : NP2 = 467CAN00980 CLS1 : .02% .GE. FE2O3 .LE. 0.4% : # 1 -> # 102 CAN00990 CLS2 : 0.4% .GT. FE2O3 .LÉ. 0.6% : # 103 -> # 175 CAN01000 CLS3 : 0.6% .GT. FE2O3 .LE. 0.8% : # 176 -> # 244 CAN01010 CLS4 : 0.8% .GT. FE2O3 .LE. 1.2% : # 245 -> # 349 CAN01020 CLS5 : 1.2% .GT. FE2O3 .LE. 1.6% : # 350 -> # 401 CAN01030 CLS6 : 1.6% .GT. FE2O3 .LE. 25.60% : # 402 -> # 467 CAN01040 CAN01050 CAN01060 DATA NST/102,73,69,105,52,66/ CAN01070 CAN01080 THE FOLLOWING DATA 'NST' IS USED FOR SOIL TEXTURE 'ST' DATA SET CAN01090 CAN01100 ACCORDING TO THEIR SAND-SILT-CLAY % CONTENT CAN01110 CIASS 1 TO 6 : NP2 = 483; DETAILS : SEE FILE (S5L.DATA.C1) CAN01120 DATA NST/40,63,76,93,68,143/ CAN01130 CAN01140 CAN01150 THE FOLLOWING DATA 'NST' IS USED FOR S.D. VEGETATION DATA CAN01160 CAN01170 DATA NST/225, 61, 292, 469, 82, 182, 63, 103, 39, 39, 217, 51, CAN01180 +393,441,80,88, 88,41,32,26, 118,43,121,44, 45,102,66,89, CAN01190 +78, 53, 147, 39, 24, 42, 119, 69, 76, 96, 107, 154, 28, 19/ CAN01200 CAN01210 CAN01220 THE FOLLOWING DATA 'NST' IS USED FOR IOWA VEGETATION DATA CAN01230 CAN01240 DATA NST/514,41, 517,36,32, 621,517,45, 610,485,21, CAN01250 +437, 377, 22, 190, 172, 25, 650, 568, 42, 435, 417, 44, 393, 267/ CAN01260 -CAN01270 C-CAN01280 CAN01290 11 = CLASS STATISTICS; 12 = CANONICAL FEATURES CAN01300 CAN01310 OPEN(11) CAN01320 OPEN (12) CAN01330 REWIND 11 CAN01340 REWIND 12 CAN01350 SET THE INPUT&OUTPUT DO LOOP PARAMETERS CAN01360 CAN01370 CAN01380 IK1=MOD (NCLS, 6) CAN01390 IM1=6\*(NCLS/6)+1CAN01400 IK2=MOD (NTERM, 5) IM2=5\* (NTERM/5)+1 CAN01410 CAN01420 IK3=MOD (NTERM, 16) CAN01430 IM3=16\*(NTERM/16)+1CAN01440 ILP1=NCLS/6 CAN01450 IF (ILP1.EQ.0) ILP1=1 CAN01460 ILP2=NTERM/5 CAN01470 IF (ILP2.EQ.0) ILP2=1 CAN01480 ILP3=NTERM/16 CAN01490 IF (ILP3.EQ.0) ILP3=1 CAN01500 С CAN01510 SET THE IMSL INPUT&OUTPUT TO THE FEATURE DESIGNER ( SCREEN ) С

С

С С

С

С

С

Ċ

С

С

C

C

C

Ĉ

С C

C

С

С

С

С С

С

С

С С

C

C

С

С

С

С

С

С

С

С

С

С

CAN01520 CAN01530 CALL UGETIO (IOPT, NIN, NOUT) CAN01540 DO 130 LTERM=1, NTERM CAN01550 KTERM=LTERM\* (LTERM+1)/2 CAN01560 CAN01570 READ IN CLASS STATISTICS CAN01580 CAN01590 DO 30 ITERM=1, LTERM CAN01600 IF (NCLS.LT.6) GO TO 20 CAN01610 DO 10 IL=1, ILP1 10 READ (11, \*) (XMT (ITERM, JCLS), JCLS=1+(IL-1)\*6, IL\*6) CAN01620 CAN01630 IF (IK1.EQ.0) GO TO 30 CAN01640 20 READ (11, \*) (XMT (ITERM, JCLS), JCLS=IM1, NCLS) **30 CONTINUE** CAN01650 CAN01660 DO 60 ITERM=1, KTERM CAN01670 IF (NCLS.LT.6) GO TO 50 CAN01680 DO 40 IL=1, ILP1 40 READ (11, \*) (VCVT (ITERM, JCLS), JCLS=1+(IL-1)\*6, IL\*6) CAN01690 IF (IK1.EQ.0) GO TO 60 CAN01700 50 READ (11, \*) (VCVT (ITERM, JCLS), JCLS=IM1, NCLS) CAN01710 CAN01720 60 CONTINUE IF (NCLS.LT.6) GO TO 80 CAN01730 DO 70 IL=1, ILP1 CAN01740 70 READ (11, \*) (CT (ICLS), ICLS=1+(IL-1)\*6, IL\*6) CAN01750 CAN01760 IF (IK1.EQ.0) GO TO 90 80 READ (11, \*) (CT (ICLS), ICLS=IM1, NCLS) CAN01770 90 DO 120 ITERM=1, KTERM CAN01780 IF (NCLS.LT.6) GO TO 110 CAN01790 CAN01800 DO 100 IL=1, ILP1 100 READ (11, \*) (VCVIT (ITERM, JCLS), JCLS=1+(IL-1)\*6, IL\*6) CAN01810 IF (IK1.EQ.0) GO TO 120 CAN01820 110 READ (11, \*) (VCVIT (ITERM, JCLS), JCLS=IM1, NCLS) CAN01830 CAN01840 120 CONTINUE CAN01850 **130 CONTINUE** CAN01860 CAN01870 FIND WITHIN CLASS SCATTER MATRIX CAN01880 CALL FWCS (VCVT, MTERM, NST, NCLS, WCS) CAN01890 CALL USWSM(' WCS MATRIX IS ',15, WCS, NTERM, 1) CAN01900 CAN01910 FIND AMONG CLASS SCATTER MATRIX CAN01920 CAN01930 CALL FACS (XMT, NTERM, MTERM, NST, NCLS, ACS, XMO) CAN01940 CALL USWSM (' ACS MATRIX IS ', 15, ACS, NTERM, 1) CAN01950 CAN01960 FIND CANONICAL FEATURES CAN01970 CAN01980 CALL EIGZS (ACS, WCS, NTERM, IJOB, D, Z, NTERM, WK, IER) CAN01990 CALL USWFV ('CANONIC EVALUES', 15, D, NTERM, 1, 1) CAN02000 CAN02010 CALL USWSM ('CANONIC EVECTOR', 15, Z, NTERM, 1) CAN02020 INTERNAL CHECKING FOR MATRIX INVERSION ACCURACY CAN02030 CAN02040 CALL SCOPY (MTERM, WCS, 1, WCS1, 1) CAN02050 CALL LINV1P (WCS1, NTERM, WCS1, IDGT, D1, D2, IER1) CAN02060 CALL VMULSS (WCSI, ACS, NTERM, TEST, NTERM) CAN02070 CALL FTRACE (TEST, NTERM, TRACE) CAN02080

130

C

C

C

С

C

С

С

С

C

C

С

C

С

C

С

C

C

C

С

CAN02090 C C SEND THE ACCURACY COMMENTS TO THE SCREEN CAN02100 CAN02110 С CAN02120 IF (IER.NE.0.OR.WK(1).GE.1.0) GO TO 140 CAN02130 WRITE (\*, 3) IER, WK(1) CAN02140 GO TO 150 CAN02150 140 WRITE (\*,2) IER, WK(1) CAN02160 1 FORMAT (5E15.7) 2 FORMAT (' PERFORMANCE OF "EIGZS" IS POOR, IER =', I5, CAN02170 CAN02180 +'; WK(1) =', E15.7) 3 FORMAT (' PERFORMANCE OF "EIGZS" IS GOOD, IER =', I5, CAN02190 CAN02200 +'; WK(1) =', E15.7CAN02210 150 DO 170 I=1,NTERM CAN02220 IF (D(I).LE.0.0) GO TO 160 CAN02230 GO TO 170 CAN02240 160 WRITE (\*, 4) I, D(I) CAN02250 4 FORMAT (' EIGEN VALUE IS "< = 0.0" AT I = ', I5, +' WHERE D(I) =',E15.7) CAN02260 CAN02270 IFLAG1=IFLAG1+1 CAN02280 **170 CONTINUE** CAN02290 IF (IFLAG1.GT.0) GO TO 180 CAN02300 WRITE (\*, 6) CAN02310 GO TO 190 CAN02320 180 WRITE (\*, 5) IFLAG1 5 FORMAT (' THERE ARE', 15, ' NEGATIVE OR ZERO EIGEN VALUES ') CAN02330 le de la 6 FORMAT (' ALL EIGEN VALUES ARE GREATER THAN ZERO ') CAN02340 CAN02350 190 CALL VABSMF (D, NTERM, 1, SUM) IF (ABS (TRACE-SUM) .GT.1.0E-1) GO TO 200 CAN02360 CAN02370 WRITE (\*, 8) TRACE, SUM CAN02380 GO TO 210 CAN02390 200 WRITE (\*, 7) TRACE, SUM 7 FORMAT (' ACCURACY OF "EIGZS" IS POOR, TRACE =', E15.7, CAN02400 CAN02410 +': SUM ='.E15.7) 8 FORMAT (" ACCURACY OF "EIGZS" IS GOOD, TRACE =', E15.7, CAN02420 CAN02430 +'; SUM =', E15.7) CAN02440 С С SEND THE FINAL RESULTS TO THE CANONICAL FEATURE FILE CAN02450 CAN02460 С CAN02470 210 WRITE (12, 9) TRACE, SUM CAN02480 9 FORMAT (2E15.7) IF (NTERM. LT. 5) GO TO 230 CAN02490 CAN02500 DO 220 I=1, ILP2 220 WRITE (12,1) (D (NTERM+1-J), J=1+(I-1)\*5, I\*5) CAN02510 CAN02520 IF (IK2.EQ.0)GO TO 240 CAN02530 230 WRITE (12,1) (D (NTERM+1-J), J=IM2, NTERM) CAN02540 240 DO 270 J=1,NTERM CAN02550 IF (NTERM.LT.5) GO TO 260 CAN02560 DO 250 I=1, ILP2 250 WRITE (12,1) (Z (K, NTERM+1-J), K=1+(I-1)\*5, I\*5) CAN02570 IF (IK2.EQ.0) GO TO 270 CAN02580 CAN02590 260 WRITE (12,1) (Z (K, NTERM+1-J), K=IM2, NTERM) CAN02600 270 CONTINUE CALL VMULSF (WCS, NTERM, Z, NTERM, NTERM, TEST, NTERM) CAN02610 CAN02620 NTM=NTERM CAN02630 CALL VMULFM (Z, TEST, NTM, NTM, NTM, NTM, NTM, T, NTM, IER) CAN02640 С CAN02650 SEND THE ACCURACY COMMENTS TO THE SCREEN С

CAN02660 С CAN02670 PRINT\*, ' THE FOLLOWING MATRIX MUST BE AN IDENTITY MATRIX' CAN02680 IF (NTERM.LT.16) GO TO 290 CAN02690 DO 280 IL=1, ILP3 CAN02700 DO 280 I=1, NTERM CAN02710 280 WRITE (\*, 281) (T(I, J), J=1+(IL-1)\*16, IL\*16) CAN02720 281 FORMAT (16F5.2) CAN02730 IF (IK3.EQ.0) GO TO 310 CAN02740 290 DO 300 I=1,NTERM CAN02750 300 WRITE (\*, 281) (T (I, J), J=IM3, NTERM) CAN02760 310 STOP CAN02770 END SUBROUTINE FWCS (VCVT, MTERM, NST, NCLS, WCS) CAN02780 CAN02790 C CAN02800 THIS SUBROUTINE FINDS WITHIN CLASS SCATTER MATRIX С CAN02810 С CAN02820 REAL VCVT (MTERM, NCLS), WCS (MTERM) CAN02830 INTEGER NST (NCLS) CAN02840 NX1=0CAN02850 DO 10 I=1,NCLS CAN02860 10 NX1=NX1+NST(I) CAN02870 DO 30 I=1,MTERM CAN02880 X1=0.0 CAN02890 DO 20 J=1, NCLS CAN02900 X2 = FLOAT (NST (J)) - 1.0CAN02910 20 X1=X1+X2\*VCVT(I, J)/FLOAT(NX1) CAN02920 30 WCS(I)=X1 CAN02930 RETURN CAN02940 END SUBROUTINE FACS (XMT, NTERM, MTERM, NST, NCLS, ACS, XMO) CAN02950 CAN02960 С THIS SUBROUTINE FINDS AMONG CLASS SCATTER MATRIX CAN02970 С CAN02980 С REAL XMT (NTERM, NCLS), ACS (MTERM), XMO (NTERM) CAN02990 CAN03000 INTEGER NST (NCLS) CAN03010 NX1=0 CAN03020 DO 10 I=1,NCLS CAN03030 10 NX1=NX1+NST(I)CAN03040 DO 30 I=1,NTERM X1=0.0 CAN03050 DO 20 J=1,NCLS CAN03060 CAN03070 X2=FLOAT (NST (J)) CAN03080 20 X1=X1+X2\*XMT(I,J)/FLOAT(NX1) $30 \times MO(I) = X1$ CAN03090 CAN03100 DO 50 I=1,NTERM DO 50 J=1,I CAN03110 IND = (I-1) + I/2 + JCAN03120 CAN03130 X1=0.0 CAN03140 DO 40 ICLS=1, NCLS X2=FLOAT (NST (ICLS)) /FLOAT (NX1) CAN03150 40 X1=X1+X2\* (XMT (I, ICLS) -XMO (I)) \* (XMT (J, ICLS) -XMO (J)) CAN03160 CAN03170 50 ACS(IND)=X1 RETURN CAN03180 CAN03190 END SUBROUTINE FTRACE (TEST, NTERM, TRACE) CAN03200 REAL TEST (NTERM, NTERM) CAN03210 CAN03220 TRACE=0.0

$ \begin{array}{llllllllllllllllllllllllllllllllllll$		PROGRAM PCFIND	PCF00010
<pre>+NSET-1, MSET-1, NSET-1, NSHAX-100, N21=NCLS*NCLS*NTERM, PCF0030 +IRES=0, IFIND=1, ICKMV=0, NDTRM=1, NZ2=NCLS*NTERM, NTERMC=15) PCF00060 C IFIND = 1&gt; NDTRM CONTROL : LTERM=1, NTERM, NDTRM PCF00060 C IFIND = 0&gt; NDTRM DISABLE : LTERM= NTERM ONLY PCF00070 C&gt;&gt; IRES = 1&gt; NSMAX MUST EXCEED MAX(NST(1)) &lt;&lt; NOTES!!! PCF00080 C IRES = 0&gt; NSMAX MUST EXCEED MAX(NST(1)) &lt;&lt; NOTES!!! PCF00100 C TRANSFORMED DATA; WHILE NTERM IS USED TO DECIDE PCF00110 C TRANSFORMED DATA; WHILE NTERM IS USED TO DECIDE PCF00120 C HOW MANY OF THEM WILL BE CONTRIBUTED TO PC PCF00130 C NTERM = TOTAL NUMBER OF FEATURES USED PCF00160 C NTERM = TOTAL NUMBER OF PEATURES USED PCF00170 C NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS PCF00170 C REAL XMT(INTERM, NCLS), VCVT(IMTERM, NCLS), CT(INCLS), PCF00180 +VCVIT(IMTERM, NCLS), TVEC(INSMAX, NTERM, NCLS), CT(INCLS), PCF00210 +VCVIT(INTERM, NCLS), TVEC(INSMAX, NTERM, NCLS), CT(INCLS), PCF00210 +VCVIT(INTERM, NCLS), TVEC(INSMAX, NTERM, NCLS), PCF00210 PCF00220 REAL XMCK(INTERM), VCVCK(IMTERM), TX (INTERM), PC NCLS), PCF00210 PC00230 REAL XMCK(INTERM), VCVCK(IMTERM), TX (INTERM), PC NCLS), PCF00230 REAL XMCK(INTERM), NC (INTERM), TX (INTERM), PC NCLS), PCF00230 REAL XMCK(INTERM), NC (INTERM), TX (INTERM), PC NCLS) REAL XMCK (INTERM), NC (INTERM), TX (INTERM), PC NC240 REAL XMCK (INTERM), NC (INTERM), TX (INTERM), PC NC250 C CHARACTER*2 XC1 PC ONSMAX, NTERM, NC NCLS), APE (INCLS) PC PC00230 DATA PC/NTERM*0.0/ DATA DSEED DATA PC/NTERM*0.0/ DATA PC NTERM*0.0/ DATA NECTORS FOR ALL CLASSES PC PC00330 C VCVT = COV. MATRICES FOR ALL CLASSES PC PC00330 C VCVT = COV. MATRICES FOR ALL CLASSES PC PC00330 C VCVT = COV. MATRICES FOR ALL CLASSES PC PC00330 C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PC PC00430 C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PC PC00430 C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PC PC00430 C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PC PC00430 C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00430 C VCVT = I</pre>		PARAMETER (NTSET=4, NTERM=16, MTERM=NTERM* (NTERM+1) /2, NCLS=3,	PCF00020
$\begin{aligned} + IRES=0, IFIND=1, ICKMV=0, NDTEM=1, NZ2=NCLS*NTERM, NTERMC=15) & PCF00050 \\ PCF00050 \\ C & IFIND = 1> NDTEM CONTROL : LTERM=1, NTERM ONLY & PCF00060 \\ PCF00050 \\ C & IFIND = 0> NDTEM DISABLE : LTERM = NTERM ONLY & PCF00070 \\ PCF00070 \\ C & IRES = 1> NSMAX MUST EXCEED MAX (NST (1)) <> NOTES!!! PCF00700 \\ PCF00100 \\ C & IRES = 1> NSMAX CONTROL : SUBROUTINE GGNSM & PCF00100 \\ PCF00100 \\ C & TRANSFORMED DATA; WHILE NTERM IS USED TO READ ENTIRE PCF00110 \\ TRANSFORMED DATA; WHILE NTERM IS USED TO DECIDE & PCF00120 \\ PCF00100 \\ C & HOW MANY OF THEM WILL BE CONTRIBUTED TO PC & PCF00130 \\ PCF00150 \\ C & NTERM = TOTAL NUMBER OF PEATURES USED & PCF00160 \\ PCF00150 \\ C & NTERM = TOTAL NUMBER OF INFORMATION CLASSES & PCF00160 \\ PCF00150 \\ C & NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS & PCF00160 \\ PCF00110 & PCF00150 \\ +VCVIF (INTERM, NCLS), VCVT (IMTERM, NCLS), CT (NCLS), & PCF00210 \\ +VCVIF (INTERM, NCLS), VCVT (IMTERM, NCLS), CT (NCLS), & PCF00210 \\ +VCVIF (INTERM, NCLS), VCVT (IMTERM, NCLS), CT (NCLS), & PCF00210 \\ PCF00210 \\ PCF00210 \\ PCF00220 \\ REAL XMC (INTERM), VCV (IMTERM), VCVI (IMTERM), XM (INTERM), PCF00220 \\ REAL XMC (INTERM), VCVC (IMTERM), VCVI (IMTERM), XM (INTERM), PCF00220 \\ REAL XMC (INTERM), VCVC (IMTERM), VCVI (IMTERM), XM (INTERM), PCF00220 \\ INTEGER NER (6), NPC (NCLS, NCLS, NTERM), YK (INTERM) & PCF00230 \\ REAL XMC (INTERM), VCVC (IMTERM), VCVI (IMTERM), YK (INTERM) & PCF00250 \\ INTEGER NER (6), NPC (NCLS, NCLS, NTERM), NST (NCLS) & PCF00250 \\ DATA XCL/' '/ & PCF00250 \\ DATA XCL/' '/ & PCF00250 \\ DATA AC// PC (NCLS, NCLS, NTERM), NST (NCLS) & PCF00300 \\ DATA CD, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/ & PCF00300 \\ DATA QP, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/ & PCF00300 \\ DATA QP, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/ & PCF00300 \\ DATA QP, PR, PX/N22*0.0, NATRIX IN SYMMETRIC STORAGE MODE & PCF00350 \\ C VCVT = GOV. MATRIX SFOR ALL CLASSES & PCF00360 \\ C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE & PCF00430 \\ C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE & PCF004$		+NSET=1,MSET=1,NDSET=1,NSMAX=100,NZ1=NCLS*NCLS*NTERM,	PCF00030
C INTERNATION OF THE ALVEST OF THE ALVEST ALVESTIGATION OF ALL STOCES AND ALVESTIGATION OF ALL CLASSES AND ALVESTIGATION		+TRES=0 IFIND=1 ICKMV=0 NDTRM=1 NZ2=NCLS*NTERM NTERMC=15)	PCF00040
C IFIND = 1> NDTEM CONTROL : LTERM=1, NTERM, NDTEM PCF00060 C IFIND = 0> NDTEM DISABLE : LTERM = NTERM ONLY PCF00700 C IFIND = 0> NSMAX MUST EXCEED MAX (NST (1) < NOTES!!! PCF0080 C PCF00100 C NTERMC > OR = NTERM , WHERE NTERMC IS USED TO READ ENTIRE PCF00100 C NTERMC > OR = NTERM , WHERE NTERMC IS USED TO READ ENTIRE PCF00120 C HOW MANY OF THEM WILL BE CONTRIBUTED TO PC PCF00120 C NTERM = TOTAL NUMBER OF PEATURES USED PCF00120 C NTERM = TOTAL NUMBER OF INFORMATION CLASSES PCF00160 C NTERM = TOTAL NUMBER OF SAMPLES FOR ONE CLASS PCF00160 C NTERM = TOTAL NUMBER OF SAMPLES FOR ONE CLASS PCF00160 C NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00200 +VCVIF (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00210 +VCVIF (NTERM, NCLS), VCVT (MTERM), VCVI (MTERM), XM (NTERM), PCF00210 +VCVIF (NTERM, VCVCK (MTERM), VCVI (MTERM), XM (NTERM), PCF00210 REAL XMT (NTERM, VCVCK (MTERM), VCVI (MTERM), YX (NTERM) PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), VCVI (MTERM), YX (NTERM) PCF00220 HVCVIF (NTERM, VCVCK (MTERM), YX (NTERM), YX (NTERM) PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), YX (NTERM) PC (NCLS), PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), YX (NTERM) PC (NCLS) PCF00220 CHARACTER*2 XC1 PC (NCLS, NCLS, NTERM), NST (NCLS) PCF00250 INTEGER NER (6), NPC (NCLS, NCLS, NTERM), NST (NCLS) PCF00250 DATA XC1/' '/ PC (NCLS, NCLS', NTERM), NST (NCLS) PCF00250 DATA XC1/' '/ PC (NCLS' 0.0, NCLS*0.0/ PCF00300 DATA DSEED, NPC/5.D0, NZ1*0/ DATA DSEED, NPC/5.D0, NZ1*0/ DATA DSEED, NPC/5.D0, NZ1*0/ C VCVT = COV. MATRIXES FOR ALL CLASSES PCF00360 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C VCVT = COV. MATRIXES FOR ALL CLASSES PCF00360 C VCVT = INVERSE COV. MATRIX IN SYMMETIC STORAGE MODE PCF00320 C VCVT = INVERSE COV. MATRIX IN SYMMETIC STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN SYMMETIC STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN SYMMETIC STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN SYMMETIC STORAGE MODE	<b>C</b> .		PCF00050
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	0		DCE00050
C IFIND = 0> NDTRM DISABLE : LTERM = NTERM ONLY PCF00080 C>> NERK MUST EXCEED MAX(NST (I) < NOTES!!! PCF00180 C IRES = 0> NSMAX CONTROL : SUBROUTINE GGNSM PCF00110 C NTERM > OR = NTERM , WHERE NTERMC IS USED TO READ ENTIRE PCF00110 C TRANSFORMED DATA; WHILE NTERM IS USED TO DECIDE PCF00120 C HOW MANY OF THEM WILL BE CONTRIBUTED TO PC PCF00140 C NTERM = TOTAL NUMBER OF FRATURES USED PCF00150 C NTERM = TOTAL NUMBER OF INFORMATION CLASSES PCF00160 C NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS PCF00170 C NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00190 +VCVIT (MTERM, NCLS) TVEC (NSMAX, NTERM, NCLS), CT (NCLS), PCF00190 +VCVIT (MTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00210 +VCVIT (MTERM, NCLS), VCVT (MTERM), XM (NTERM), PCF00210 +VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM), PC (NCLS), PCF00210 PC (NSMAX, NTERM), VCV (MTERM), XM (NTERM), PCF00220 REAL XMC (NSMAX, NTERM), PR (NCLS, NTERM), PX (NCLS), PCF00220 INTEGER NER (6), NPC (NCLS, NTERM), T1 (NTERM) REAL KMCC (NSMAX, NTERM, NCLS), AP (NCLS) INTEGER NER (6), NPC (NCLS, NCLS, NTERM), NST (NCLS) PCF00220 DATA PC (NTERM*0.0/ DATA QP, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/ PCF00320 DATA (DRE(1), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/ PCF00330 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00370 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00420 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN SYMETRIC STORAGE MODE PCF00430 C VCVT = INVERSE COV. MATRIX IN FULL S	C	IFIND = 1> NDIRM CONTROL : LIERM=1, NIERM, NDIRM	PCF00000
$\begin{array}{llllllllllllllllllllllllllllllllllll$	С	IFIND = 0> NDTRM DISABLE : LTERM = NTERM ONLY	PCF00070
CIRES= 0>NSMAX CONTROL : SUBROUTINE GGNSMPCF000100CTRANSFORMED DATA: WHLLE NTERM IS USED TO READ ENTIREPCF00110CTRANSFORMED DATA: WHLLE NTERM IS USED TO DECIDEPCF00120CHOW MANY OF THEM WILL BE CONTRIBUTED TO PCPCF00130CNTERM = TOTAL NUMBER OF FEATURES USEDPCF00160CNTERM = TOTAL NUMBER OF INFORMATION CLASSESPCF00160CNSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CREAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00180+VCVIT (MTERM, NCLS) TVEC (NSMAX, NTERM, NCLS),PCF00210+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM), PR (NCLS), TTERM), PR (NCLS),PCF00220+VCC (NSMAX, NTERM), PR (NCLS, NTERM), PX (NCLS),PCF00220REAL XMCK (NTERM), VCVK (MTERM), X (NTERM), PX (NTERM)PCF00220REAL XMCK (NTERM), VCVK (MTERM), T1 (NTERM), PX (NCLS),PCF00220REAL XMCC (NSMAX, NTERM, NCLS), AP (NCLS)PCF00220REAL XMCC (NSMAX, NTERM, NCLS), NTERM, NST (NCLS)PCF00220DATA CC (NSMAX, NTERM, CLS, NTERM), NST (NCLS)PCF00230DATA CC (NSMAX, NTERM, CLS, NTERM), NST (NCLS)PCF00300DATA CC (NTERM*0.0/PCF00310DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/PCF00320DATA PC/NTERM*0.0/	C>	> IRES = 1> NSMAX MUST EXCEED MAX(NST(I)) << NOTES!!!	PCF00080
CPCF00100CNTERMC > OR = NTERM, WHERE NTERMC IS USED TO READ ENTIREPCF00110CTRANSFORMED DATA; WHILE NTERM IS USED TO DECIDEPCF00120CHOW MANY OF THEM WILL BE CONTRIBUTED TO PCPCF00130CNTERM = TOTAL NUMBER OF FEATURES USEDPCF00160CNNSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CREAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00210+VCVIT (MTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00210+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM), NCLS),PCF00220+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM), NCLS),PCF00220+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM), VCV (IMTERM), X (NTERM), X (NTERM),PCF00220REAL XMCK (NTERM), VCV (IMTERM), NT (NTERM), Y (NTERM)PCF00220REAL XMCK (NTERM), VCV (IMTERM), T1 (NTERM)PCF00220REAL XMCK (NTERM), VCV (IMTERM), T1 (NTERM), Y (NTERM)PCF00220REAL XMCK (NTERM), VCV (IMTERM), T1 (NTERM)PCF00230REAL XMCK (NTERM), VCV (IMTERM), NST (NCLS)PCF00260DOUBLE PRECISION DSEEDPCF00270DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00330DATA (NBR (1), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CVCVIT = COV. MATRICES FOR ALL CLASSESPCF00330CVCVIT = IMVERSE COV. MATRIX IN FULL STORAGE MODEPCF00430CVCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00430CVCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00430CVCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MO	С	IRES = $0 >$ NSMAX CONTROL : SUBROUTINE GGNSM	PCF00090
CNTERMC > OR = NTERM, WHERE NTERMC IS USED TO READ ENTIREPCF00110CTRANSFORMED DATA; WHILE NTERM IS USED TO DECIDEPCF00120CHOW MANY OF THEM WILL BE CONTRIBUTED TO PCPCF00130CNTERM = TOTAL NUMBER OF FEATURES USEDPCF00160CNSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CREAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00180+VCVTT (MTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00200+VCVTT (MTERM, NTERM), VCV (MTERM), VCVI (MTERM), M(NTERM),PCF00210+VCVTT (MTERM, NTERM), VCV (MTERM), VCVI (MTERM), PC (NCLS),PCF00220+VCVTT (MTERM), VCV (MTERM), VCVI (MTERM), PX (NCLS),PCF00230REAL XMCK (NTERM), VCV (MTERM), TX (NTERM), PX (NCLS),PCF00230REAL XMCK (NTERM), WK (NTERM), TX (NTERM), TX (NTERM), PC (NO230PCF00250INTEGER NBR(6), NPC (NCLS, NTERM), TX (NTERM), Y (NTERM)PCF00260DATA YC/NTERM*0.0/DATA YC/NTERM*0, O/PCF00230DATA YC, PREM*0.0/DATA YC, PC (NO240, NZ2*0.0, NCLS*0.0/PCF00300DATA YC, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00306PCF00306CXMTMEAN VECTORS FOR ALL CLASSESPCF00306C VCVTCOV. MATRICES FOR ALL CLASSESPCF00306C VCVTINVERSE COV. MATRICES FOR ALL CLASSESPCF00306C VCVTINVERSE COV. MATRIX IN FULL STORAGE MODEPCF00420C VCVTINVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00430C VCVTINVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00430C VCVTINVERSE COV	Ċ	· 김승규는 가슴 약물입니다. 그는 것 같아요. 그는 것 같아요. 그는 것 같아요. 그는 것 같아요. 그는 것	PCF00100
CTRANSFORMED DATA; WHILE NTERM IS USED TO DECIDEPCF00120CHOW MANY OF THEM WILL BE CONTRIBUTED TO PCPCF00130CNTERM = TOTAL NUMBER OF FEATURES USEDPCF00160CNCLS = TOTAL NUMBER OF INFORMATION CLASSESPCF00170CNSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CREAL XMT (INTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00170+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS),PCF00210+VCVIT (INTERM, NCLS), TVEC (NSMAX, NTERM), VCV (INTERM), XM (NTERM),PCF00220+VCVIT (INTERM, NERM), VCV (INTERM, NCLS),PCF00220+VCVIT (INTERM, NERM), VCV (INTERM), VCV (INTERM), XM (NTERM),PCF00220+VCVIT (INTERM, NTERM), VCV (INTERM), VCU (INTERM), XM (NTERM),PCF00220+VCVIT (INTERM), VCV (INTERM), VCV (INTERM), XM (INTERM),PCF00220+VEC (NSMAX, NTERM), WK (NTERM), X (INTERM), YM (INTERM),PCF00220HOW REAL XMCK (INTERM), VCV (INTERM), TI (INTERM), YM (INTERM)PCF00220REAL XMCK (INTERM), VCV (INTERM), TI (INTERM), YM (INTERM)PCF00240REAL XMCK (INTERM), VCV (INTERM), NST (INCLS)PCF00260DATA CL/'/PCF00230DATA C/INTERM*0.0/PCF00310DATA PC/INTERM*0.0/PCF00320DATA CP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00330CVCVT = COV. MATRICES FOR ALL CLASSESPCF00300CCXMT = MEAN VECTORS FOR ALL CLASSESPCF00300CCCMATRICES FOR ALL CLASSESPCF00300CCVCVT = INVERSE COV. MATRIX IN FULL STORAGE MODEPC	č	NTERMC > OR = NTERM . WHERE NTERMC IS USED TO READ ENTIRE	PCF00110
CHAW NARLY OF THEM WILL BE CONTRIBUTED TO PCPCF00130CNTERM = TOTAL NUMBER OF FEATURES USEDPCF00150CNCLS = TOTAL NUMBER OF INFORMATION CLASSESPCF00160CNSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CREAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00180+VCVTIT (MTERM, NCLS), VCV (IMTERM, NCLS), CT (NCLS),PCF00210+VCVTIT (NTERM, NELS), VCV (IMTERM), VCLS, NTERM), PX (NCLS),PCF00220+VCVTIT (NTERM, NERM), VCV (IMTERM), XM (NTERM),PCF00220+VEC (NSMAX, NTERM), VCV (IMTERM), YX (INTERM), M (NTERM),PCF00220+VEC (NSMAX, NTERM), VCV (IMTERM), YX (INTERM), PX (NCLS),PCF00220REAL XMCK (INTERM), VCV (IMTERM), YX (INTERM), YX (INTERM)PCF00220REAL RVEC (NSMAX, NTERM, NCLS), AP (NCLS)PCF00240REAL RVEC (INSMAX, NTERM, NCLS), AP (NCLS)PCF00240REAL RVEC (INSMAX, NTERM, NCLS), NERM), NST (NCLS)PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00300DATA AC1/' '/PCF00320DATA DSEED, NPC/5.DO, NZ1*0.O, NCLS*0.0/PCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCF00330CVCVT = COV. MATRICES FOR ALL CLASSESPCF00330CVCVT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00330CVCVI = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00430CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00430CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00430CVCVI = INVERSE COV. MATR	č	TRANSFORMED DATA WHILE NTERM IS USED TO DECIDE	PCF00120
CNOW PART OF THEM WILL BE CONTRIBUTED TO FCPCF00140CNTERM = TOTAL NUMBER OF FEATURES USEDPCF00150CNCLS = TOTAL NUMBER OF FEATURES USEDPCF00160NCM SARX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CREAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00120+VCVTT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS), CT (NCLS),PCF00210+VCVTT (MTERM, NCLS), TVEC (NSMAX, NTERM), XM (NTERM), PCF00210PCF00210+VCVTT (MTERM, NCLS), TVEC (NSMAX, NTERM), PX (NCLS),PCF00210+VCVTT (NTERM, NCLS), TVEC (NSMAX, NTERM), PX (NCLS),PCF00220+VCC (NSMAX, NTERM), VCV (MTERM), TX (NTERM), PX (NCLS),PCF00220REAL XMCK (NTERM), VCVK (MTERM), TX (NTERM), Y (NTERM)PCF00230REAL XMCK (NTERM), VCVK (MTERM), TX (NTERM), Y (NTERM)PCF00240REAL RVEC (NSMAX, NTERMC, NCLS, AP (NCLS)PCF00260DATA XCL''/PCF00260DATA XCL''/PCF00270DOUBLE PRECISION DSEEDPCF00310DATA QP, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA NDRC(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CCMATMAEN VECTORS FOR ALL CLASSESCPCF00350PCF00350CVCVT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00360CTVEC = GENERATED SAMELE VECTORSPCF00430CVCVT = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00420CVCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MOD		HOW MANY OF THEM WILL BE CONTREDUCED TO DO	DCE00130
C TTERM = TOTAL NUMBER OF FEATURES USED PCF00150 PCF00150 C NCLS = TOTAL NUMBER OF INFORMATION CLASSES PCF00160 NMMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS PCF00170 C PCF00180 REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00180 +VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS), CT (NCLS), PCF00210 +VCVIT (NTERM, NTERM), VCV (MTERM), VCL (MTERM), XM (NTERM), PCF00210 +VCC (NSMAX, NTERM), VCV (MTERM), XM (NTERM), XM (NTERM), PCF00220 +VEC (NSMAX, NTERM), VCV (MTERM), YX (NTERM), YX (NCLS), PCF00220 REAL XMCK (NTERM), VCV (MTERM), TI (NTERM) PX (NCLS), PCF00230 REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), PCF00250 INTEGER NER (6), NPC (NCLS, NCLS, NTERM), NST (NCLS) PCF00270 DOUBLE PRECISION DSEED DATA AC1/' '/ PCF00310 DATA DEED, NPC/5.D0, N21*0/ DATA DSEED, NPC/5.D0, N21*0/ DATA DSEED, NPC/5.D0, N21*0/ DATA DSEED, NPC/5.D0, N21*0/ C XMT = MEAN VECTORS FOR ALL CLASSES PCF00350 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00350 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C TVEC = GENERATED SAMPLE VECTORS C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES PCF00380 C TVEC = GENERATED SAMPLE VECTORS C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES PCF00380 C VCVIT = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00410 C VCVI = COV. MATRIX PCF00410 C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00420 C MM = MEAN VECTOR C PC = PROBABILITY OF CORRECT CLASSIFICATION PCF00430 PCF00430 PCF00430 PCF00430 PCF00440 MMCK = CHECKING MATRIX FOR COVARIANCES PCF00450 C VCVK = CHECKING MATRIX FOR COVARIANCES PCF00460 NER = INSL ROUTINE-USED PARAMETER VECTOR PCF00470	.U	NOW MANI OF THEM WILL BE CONTRIBUTED TO FC	DCE00140
C MTERM = TOTAL NUMBER OF FEATURES USED PCF00150 C NCLS = TOTAL NUMBER OF INFORMATION CLASSES PCF00160 C NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS PCF00170 C PCF00180 REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00200 +VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS), CT (NCLS), PCF00210 +VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS), CT (NCLS), PCF00220 +VCVIT (MTERM, NCLS, NTERM), PC (NCLS, NTERM), XM (NTERM), PCF00220 +VCC (NSMAX, NTERM), VCV (MTERM), VCVI (MTERM), XM (NTERM), PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), XI (NTERM), YK (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), XX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), XX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), XX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), NST (NCLS) C CHARACTER*2 XC1 DOUBLE PRECISION DSEED DATA AC1/' '/ DOUBLE PRECISION DSEED DATA AC1/' '/ DATA OF, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/ DATA OF, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/ DATA (NBR(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/ C CT = M.L. DECISION RULE PARAMETER PCF00350 C VCVT = COV. MATRICES FOR ALL CLASSES C TVEC = GENERATED SAMPLE VECTORS C VCVT = INVERSE COV. MATRICES FOR ALL CLASSES C TVEC = GENERATED SAMPLE VECTORS C VCVT = INVERSE COV. MATRICES FOR ALL CLASSES C TVEC = GENERATED SAMPLE VECTORS C VCVT = INVERSE COV. MATRIX IN FULL STORAGE MODE C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00410 C VCVT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00410 C VCVT = CHECKING VECTOR FOR MEAN PCF00440 C MM = MEAN VECTOR POR MEAN PCF00440 C MM = MEAN VECTOR POR MEAN PCF00440 C MM = MEAN VECTOR PARAMETER VECTOR C PCF00440 C MM = MEAN VECTOR POR MEAN PCF00440 C MM = MEAN VECTOR PARAMETER VECTOR C PCF00440 C MM = MEAN VECTOR PAR	С		PCF00140
C NCLS = TOTAL NUMBER OF INFORMATION CLASSES PCF00160 NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS PCF00170 REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00180 +VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS), CT (NCLS), PCF00210 +VCVIT (NTERM, NTERM), VCV (MTERM), NCLS), CT (NCLS), PCF00220 +VCVIT (NTERM, NTERM), VCV (MTERM), VCV (MTERM), XM (NTERM), PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), PX (NCLS), PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), PX (NCLS), PCF00220 REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM) REAL NOCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM) REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM) REAL RVEC (NSMAX, NTERMC, NCLS), AP (NCLS) REAL RVEC (NSMAX, NTERMC, NCLS, NTERM), NST (NCLS) CHARCTERF*2 XC1 DOUBLE PRECISION DSEED DATA PC/NTERM*0.0/ DATA QP,PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/ DATA QP,PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/ DATA (NBR (I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/ C XMT = MEAN VECTORS FOR ALL CLASSES C VCVT = COV. MATRICES FOR ALL CLASSES C VCVT = COV. MATRICES FOR ALL CLASSES C VCVT = COV. MATRICES FOR ALL CLASSES C VCVT = INVERSE COV. MATRICES FOR ALL CLASSES C VCVT = INVERSE COV. MATRICES FOR ALL CLASSES C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES C VCVIT = INVERSE COV. MATRIX IN FULL STORAGE MODE C VCVIT = INVERSE COV. MATRIX IN FULL STORAGE MODE C VCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C PCF00430 C VCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C PCF00440 C MMCK = CHECKING WECTOR FOR MEAN PCF00450 C VCVK = CHECKING MATRIX FOR COVARIANCES C PCF00450 C VCVK = CHECKING MATRIX FOR COVARIANCES C PCF00450 C VCVK = CHECKING MATRIX FOR COVARIANCES C PCF00450 C VCVK = CLASSIFICATION RESULT MATRIX PCF00450	C	NTERM = TOTAL NUMBER OF FEATURES USED	PCF00150
CNSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASSPCF00170CPCF00180PCF00180REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00190+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS),PCF00200+VCVIT (NTERM, NTERM), VCVC (MTERM), VCUI (MTERM), XM (NTERM),PCF00210+VC (INTERM, NCLS), TTERM), VC (VCI (MTERM), XM (NTERM),PCF00220+VC (INTERM), WCLS, NTERM), VC (NTERM), VC (NTERM), TX (NTERM),PCF00230REAL XMCK (NTERM), WK (NTERM), X (NTERM), T1 (NTERM)PCF00240REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM)PCF00250INTTEGER NBR (6), NPC (NCLS, NCLS, NTERM), NST (NCLS)PCF00260CHARACTER*2 XC1PCF00280DOUBLE PRECISION DSEEDPCF00280DATA XCL/' '/PCF00280DATA QP, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA DSEED, NPC/5.D0, NZ2*0.0, NCLS*0.0/PCF00320DATA (IL, I, I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CXMTMATRICES FOR ALL CLASSESPCF00360CVCVTCOV. MATRICES FOR ALL CLASSESPCF00360CVCVTINVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCVIFINVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CWC = CHECKING VECTORPCF00420CYCWCKCHECKING VECTOR FOR MEANPCF00420CVCVIFINVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVIFINVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CYCWCKCHECKING VECT	С	NCLS = TOTAL NUMBER OF INFORMATION CLASSES	PCF00160
C PCF00180 REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS), PCF00190 +VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS), PCF00210 +VCVIF (NTERM, NTERM), VCV (MTERM), VCVI (MTERM), XM (NTERM), PCF00210 +PC (NTERM), QP (NCLS, NTERM), PR (NCLS, NTERM), XM (NTERM), PCF00220 REAL XMCK (NTERM), VCVK (MTERM), TX (NTERM), TX (NTERM) REAL XMCK (NTERM), VCVK (MTERM), TX (NTERM), Y (NTERM) REAL XMCK (NTERM, OLCLS, NCLS, NTERM), NST (NCLS) CHARACTER*2 XC1 DOUBLE PRECISION DSEED DATA YC/NTERM*0.0/ DATA QP, PR, PX/N22*0.0, NC2*0.0, NCLS*0.0/ DATA QP, PR, PX/N22*0.0, NZ2*0.0, NCLS*0.0/ DATA (NBR (I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/ C MT = MEAN VECTORS FOR ALL CLASSES PCF00350 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C TTEC = GENERATED SAMPLE VECTORS C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES PCF00370 C VCVIT = INVERSE COV. MATRIX IN FULL STORAGE MODE C VCVIT = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00410 C VCVI = COV. MATRIX PCF00410 C VCVI = COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00420 C MM = MEAN VECTOR FOR MEAN PCF00410 C VCVI = PROBABILITY OF CORRECT CLASSIFICATION PCF00440 C XMCK = CHECKING VECTOR FOR MEAN PCF00450 C VCVK = CHECKING VECTOR FOR MEAN PCF00450 PCF00450 PCF00450 PCF00450 PCF00450 P	C	NSMAX = PRESET MAX. NO. OF SAMPLES FOR ONE CLASS	PCF00170
REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),PCF00190+VCVIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS),PCCF00200+VCVIT (NTERM, NCLS), TVERM), VCVI (MTERM), XM (NTERM),PCCF00210+VCC (NTERM), QC (NCLS, NTERM), PC (NCLS, NTERM), PX (NCLS),PCCF00220+VEC (NSMAX, NTERM), WK (NTERM), YCVI (MTERM), PX (NCLS),PCCF00220REAL XMCK (NTERM), VCV (K (MTERM), T1 (NTERM)PCF00210REAL XMCK (NTERM), VCX (K (NTERM), T1 (NTERM)PCCF00220REAL XMCK (NTERM), VCX (STERM), TX (NTERM), Y (NTERM)PCCF00220REAL XMCK (NTERM), NERMC, NCLS)PCCF00220REAL XMCK (NTERM), NCX (SS AP (NCLS)PCCF00220DATA SCL /' '/PCCF00220DOUBLE PRECISION DSEEDPCCF00280DATA XCL /' '/PCCF00320DATA DSEED, NPC /S.DO, NZ2 *0.0, NCLS *0.0/PCCF00310DATA DSEED, NPC /S.DO, NZ2 *0.0, NCLS *0.0/PCCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCCF00360CXMT = MEAN VECTORS FOR ALL CLASSESPCCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCCF00380CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCCF00420C XM = MEAN VECTORPCCF00420PCCF00430CPC = PROBABILITY OF CORRECT CLASSIFICATIONPCCF00430C PC = PROBABILITY OF CORRECT CLASSIFICATIONPCCF00420C XMCK = CHECKING VECTOR FOR MEANPCCF00450C VCVK = CHECKING VECTOR FOR MEAN	Ċ		PCF00180
HUCUIT (MTERM, NCLS), TVEC (NSMAX, NTERM, NCLS),PCF00200+VCUIT (MTERM, NTERM), VCV (MTERM), VCVI (MTERM), XM (NTERM),PCF00210+VCUIT (MTERM), WICNTERM), PR (NCLS, NTERM), PX (NCLS),PCF00220+VEC (NSMAX, NTERM), WK (NTERM), YR (NTERM), PX (NCLS),PCF00230REAL XMCK (NTERM), WK (NTERM), X (NTERM), PX (NTERM)PCF00240REAL RVEC (NSMAX, NTERMC, NCLS), AP (NCLS)PCF00250INTEGER NER (6), NPC (NCLS, NCLS, NTERM), NST (NCLS)PCF00260CHARACTER*2 XC1PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00290DATA PC/NTERM*0.0/PCF00300DATA DSEED, NPC/5.D0, NZ1*0/PCF00310DATA DSEED, NPC/5.D0, NZ1*0/PCF00320DATA (NBR (I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00350CCT= ML. DECISION RULE PARAMETERCPCVIT= COV. MATRICES FOR ALL CLASSESCVCVT= COV. MATRICES FOR ALL CLASSESCVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODECVCVIF= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECVCVIF= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECVCVIF= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECPCF00420CXM= MEAN VECTORCPCF00430CVCVIF= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECPCF00400CVCVIF= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECPCF00410CVCVIF= INVERSE COV. MATRIX IN SYMMETR		REAL XMT (NTERM, NCLS), VCVT (MTERM, NCLS), CT (NCLS),	PCF00190
<ul> <li>+VCVIF (NTERM, NEEM), VCV (NTERM), VCVI (MTERM), XM (NTERM),</li> <li>+PC (NTERM, NTERM), VCV (NTERM), PC (NCLS, NTERM), PX (NCLS),</li> <li>+PC (NTERM), QP (NCLS, NTERM), PR (NCLS, NTERM), PX (NCLS),</li> <li>+PC (NTERM), VCVCK (NTERM), X (NTERM), T1 (NTERM)</li> <li>PCF00230</li> <li>REAL XMCK (NTERM), WK (NTERM), X (NTERM), Y (NTERM)</li> <li>PCF00240</li> <li>REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM)</li> <li>PCF00250</li> <li>INTEGER NER (6), NPC (NCLS, NCLS, NTERM), NST (NCLS)</li> <li>PCF00260</li> <li>CHARACTER*2 XC1</li> <li>PCF00270</li> <li>DOUBLE PRECISION DSEED</li> <li>PCF00280</li> <li>DATA XC1/' '/</li> <li>PCF00300</li> <li>DATA PC/NTERM*0.0/</li> <li>PCF00310</li> <li>DATA DC/NTERM*0.0/</li> <li>PCF00310</li> <li>DATA DSEED, NPC/5.D0, NZ1*0/</li> <li>PCF00320</li> <li>DATA (NBR (1), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/</li> <li>PCF00350</li> <li>C VCVT = COV. MATRICES FOR ALL CLASSES</li> <li>PCF00360</li> <li>C T = M.L. DECISION RULE PARAMETER</li> <li>PCF00370</li> <li>C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES</li> <li>PCF00380</li> <li>C VCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODE</li> <li>PCF00410</li> <li>C VCVI = COV. MATRIX</li> <li>PCF00410</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00410</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00420</li> <li>PC = PROBABILITY OF CORRECT CLASSIFICATION</li> <li>PCF00430</li> <li>PC = PC0450</li> <li>C VCVK = CHECKING VECTOR FOR MEAN</li> <li>PCF00450</li> <li>C VCVK = CHECKING VECTOR FOR MEAN</li> <li>PCF00450</li> <li>PCF00450</li> <li>PC = CLASSIFICATION RESULT MATRIX</li> </ul>		VICUTT (MTERM NCLS) TVEC (NSMAX NTERM NCLS)	PCF00200
<ul> <li>+VC (ITERM), OF (NIEKS), VCV (NIEKS), PCF00220</li> <li>+VEC (NSMAX, NTERM), WK (NTERM), X (NTERM), T1 (NTERM)</li> <li>PCF00230</li> <li>REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM)</li> <li>PCF00240</li> <li>REAL RVEC (NSMAX, NTERMC, NCLS), AP (NCLS)</li> <li>PCF00250</li> <li>INTEGER NBR (6), NPC (NCLS, NTERM), NST (NCLS)</li> <li>PCF00260</li> <li>CHARACTER*2 XC1</li> <li>PCCF00270</li> <li>DOUBLE PRECISION DSEED</li> <li>DATA XC1/''/</li> <li>PCF00290</li> <li>DATA CC/NTERM*0.0/</li> <li>PCF00300</li> <li>DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/</li> <li>PCF00310</li> <li>DATA DSEED, NPC/5.D0, NZ1*0/</li> <li>PCF00320</li> <li>DATA (NBR (I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/</li> <li>PCF00350</li> <li>C</li> <li>C</li> <li>XMT = MEAN VECTORS FOR ALL CLASSES</li> <li>PCF00360</li> <li>C CT = M.L. DECISION RULE PARAMETER</li> <li>PCF00370</li> <li>C VCVT = COV. MATRICES FOR ALL CLASSES</li> <li>PCF00380</li> <li>C TVEC = GENERATED SAMPLE VECTORS</li> <li>PCF00380</li> <li>C VCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODE</li> <li>PCF00410</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00420</li> <li>C MM = MEAN VECTOR</li> <li>PCF00420</li> <li>C MM = MEAN VECTOR</li> <li>PCF00420</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00420</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00420</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00420</li> <li>C M = MEAN VECTOR</li> <li>PCF00430</li> <li>PC = PROBABILITY OF CORRECT CLASSIFICATION</li> <li>PCF00440</li> <li>C XMCK = CHECKING VECTOR FOR MEAN</li> <li>PCF00450</li> <li>C VCVK = CHECKING MATRIX FOR COVARIANCES</li> <li>PCF00450</li> <li>C VCVK = CHECKING MATRIX FOR COVARIANCES</li> <li>PCF00450</li> <li>C VCVK = CLASSIFICATION RESULT MATRIX</li> </ul>		VCVIT (MTERN NTERN) VCV (MTERN) VCVI (MTERN) VM (NTERN)	PCF00210
<ul> <li>+PC (NIERM), CREAS, NIERM), FR (NCLS, NIERM), FA (NCLS), PCF00230</li> <li>+VEC (NSMAX, NTERM), WK (NTERM), TX (NTERM), T (NTERM)</li> <li>PCF00240</li> <li>REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM)</li> <li>PCF00250</li> <li>INTEGER NBR (6), NPC (NCLS, NCLS), AP (NCLS)</li> <li>PCF00260</li> <li>CHARACTER*2 XC1</li> <li>PCF00270</li> <li>DOUBLE PRECISION DSEED</li> <li>DATA XC1/' '/</li> <li>PCF00280</li> <li>DATA AC1/' '/</li> <li>PCF00290</li> <li>DATA PC/NTERM*0.0/</li> <li>PCF00310</li> <li>DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/</li> <li>PCF00320</li> <li>DATA (NBR (1), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/</li> <li>PCF00350</li> <li>C XMT = MEAN VECTORS FOR ALL CLASSES</li> <li>PCF00360</li> <li>C CT = M.I. DECISION RULE PARAMETER</li> <li>PCF00370</li> <li>C VCVT = COV. MATRICES FOR ALL CLASSES</li> <li>PCF00360</li> <li>C TVEC = GENERATED SAMPLE VECTORS</li> <li>PCF00380</li> <li>C VCVI = INVERSE COV. MATRIX IN FULL STORAGE MODE</li> <li>PCF00400</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00410</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00410</li> <li>C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE</li> <li>PCF00420</li> <li>C MM = MEAN VECTOR</li> <li>PCF00420</li> <li>C MM = MEAN VECTOR FOR MEAN</li> <li>PCF00420</li> <li>C MM = MEAN VECTOR FOR MEAN</li> <li>PCF00420</li> <li>C MM = MEAN VECTOR</li> <li>PCF00410</li> <li>PCVVK = CHECKING WATRIX TOR COVARIANCES</li> <li>PCF00450</li> <li>C MCK = CHECKING MATRIX FOR COVARIANCES</li> <li>PCF00450</li> <li>C NER = IMSL ROUTINE-USED PARAMETER VECTOR</li> <li>PCF00450</li> <li>NPC = CLASSIFICATION RESULT MATRIX</li> </ul>		TVCVI (NIEKM, NIEKM), VCV (MIERM), VCV (MIERM), AN (NIEKM),	PCF00220
+VEC (NSMAX, NTERM), X( (NTERM), X (NTERM), Y (NTERM))PCF00230REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM))PCF00250REAL RVEC (NSMAX, NTERMC, NCLS), AP (NCLS)PCF00250INTEGER NBR (6), NPC (NCLS, NCLS, NTERM), NST (NCLS)PCF00260CHARACTER*2 XC1PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00290DATA PC/NTERM*0.0/PCF00300DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00320DATA DSEED, NPC/5.D0, NZ1*0/PCF00320DATA NSR (I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CYCVT = COV. MATRICES FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00400CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CYCVIF = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CYCVIF = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CYCVIF = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CYM = MEAN VECTORPCF00450CYCVK = CHECKING WATRIX FOR COVARIANCESPCF00450C <td></td> <td>+PC (NIERM), QP (NCLS, NIERM), PR (NCLS, NIERM), PA (NCLS),</td> <td>PCF00220</td>		+PC (NIERM), QP (NCLS, NIERM), PR (NCLS, NIERM), PA (NCLS),	PCF00220
REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM)PCF00240REAL RVEC (NSMAX, NTERMC, NCLS) AP (NCLS)PCF00250INTEGER NBR (6), NPC (NCLS, NCLS, NTERM), NST (NCLS)PCF00260CHARACTER*2 XC1PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00290DATA PC/NTERM*0.0/PCF00300DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA DSEED, NPC/5.D0, NZ1*0/PCF00320DATA (NBR (I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00390CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CXM = MEAN VECTORPCF00420CXMCK = CHECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING MATRIX FOR COVARIANCESPCF00470CNBC = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNBC = CLASSIFICATION RESULT MATRIXPCF00480		+VEC (NSMAX, NTERM), WK (NTERM), X (NTERM), TI (NTERM)	PCF00230
REAL RVEC (NSMAX, NTERMC, NCLS), AP (NCLS)PCF00250INTEGER NBR(6), NPC (NCLS, NCLS, NTERM), NST (NCLS)PCF00260CHARACTER*2 XC1PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00300DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA DSEED, NPC/5.D0, NZ1*0/PCF00320DATA (NBR(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CPCF00340CXMTCPCF00350CVCVTC (XT) = MEAN VECTORS FOR ALL CLASSESPCF00350C (XT) = MEAN VECTORS FOR ALL CLASSESPCF00360C (XT) = INVERSE COV. MATRICES FOR ALL CLASSESPCF00370C (XVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00370C (XVIT = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400C (XCV) = COV. MATRIXPCF00450C (XV) = COV. MATRIXPCF00410C (XCV) = COV. MATRIXPCF00420C (XM) = MEAN VECTORPCF00420C (XM) = MEAN VECTORPCF00440C (XM) = MEAN VECTORPCF00450C (XM) = CHECKING MATRIX FOR COVARIANCESPCF00450C (XM) = CHECKING MATRIX FOR COVARIANCESPCF00460C (XMC) = CLASSIFICATION RESULT MATRIXPCF00480		REAL XMCK (NTERM), VCVCK (MTERM), TX (NTERM), Y (NTERM)	PCF00240
INTEGER NBR(6),NPC(NCLS,NCLS,NTERM),NST(NCLS)PCF00260CHARACTER*2 XC1PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00290DATA XC1/' '/PCF00300DATA QP,PR,PX/NZ2*0.0,NZ2*0.0,NCLS*0.0/PCF00310DATA DSEED,NPC/5.D0,NZ1*0/PCF00320DATA (NER(I),I=4,6),IOPT,NIN,NOUT/1,0,0,3,0,6/PCF00330CCCPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESCPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESCVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESCVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESCVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODECVCVI = COV. MATRIXCVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODECPCF00420CXM = MEAN VECTORCPCF00420CXM = MEAN VECTOR FOR MEANCPCF00420CXM = MEAN VECTORCPCF00450CPCPC00450PCF00450CVCVK = CHECKING WECTOR FOR MEANCPCF00450CNBR = IMSL ROUTINE-USED PARAMETER VECTORCNBC = CLASSIFICATION RESULT MATRIXCNPC = CLASSIFICATION RESULT MATRIX		REAL RVEC (NSMAX, NTERMC, NCLS), AP (NCLS)	PCF00250
CHARACTER*2 XC1PCF00270DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00290DATA PC/NTERM*0.0/PCF00300DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA QP, PR, PX/NZ2*0.0, NZ1*0/PCF00320DATA (NBR(I), I=4,6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00380CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV = COV. MATRIXPCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM = MEAN VECTORPCF00430CPC = PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CPC = TECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING MATRIX FOR COVARIANCESPCF00450CNBR = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC = CLASSIFICATION RESULT MATRIXPCF00480		INTEGER NBR(6), NPC(NCLS, NCLS, NTERM), NST(NCLS)	PCF00260
DOUBLE PRECISION DSEEDPCF00280DATA XC1/' '/PCF00290DATA PC/NTERM*0.0/PCF00300DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00310DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.0/PCF00320DATA DSEED, NPC/5.D0, NZ1*0/PCF00320DATA (NBR(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CCPCF00340CXMT = MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00390CVCVI = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00410CVCV = COV. MATRIXN SYMMETRIC STORAGE MODEPCF00420CXM = MEAN VECTORPCF00430CYCW = CHECKING VECTOR FOR MEANPCF00440CXMCK = CHECKING VECTOR FOR MEANPCF00440CXMCK = CHECKING MATRIX FOR COVARIANCESPCF00460CNBR = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC = CLASSIFICATION RESULT MATRIXPCF00480		CHARACTER*2 XC1	PCF00270
DATA XC1/' '/ PCF00290 DATA PC/NTERM*0.0/ PCF00300 DATA QP,PR,PX/NZ2*0.0,NZ2*0.0,NCLS*0.0/ PCF00310 DATA DSEED,NPC/5.D0,NZ1*0/ PCF00320 DATA (NBR(I),I=4,6),IOPT,NIN,NOUT/1,0,0,3,0,6/ PCF00320 C XMT = MEAN VECTORS FOR ALL CLASSES PCF00340 C XMT = MEAN VECTORS FOR ALL CLASSES PCF00350 C VCVT = COV. MATRICES FOR ALL CLASSES PCF00360 C T = M.L. DECISION RULE PARAMETER PCF00370 C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES PCF00380 C TVEC = GENERATED SAMPLE VECTORS PCF00390 C VCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODE PCF00390 C VCVIF = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00410 C VCVI = COV. MATRIX PCF00420 C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE PCF00420 C VCVI = ENCOR PCF00430 C PC = PROBABILITY OF CORRECT CLASSIFICATION PCF00440 C MM = MEAN VECTOR FOR MEAN PCF00450 C VCVCK = CHECKING VECTOR FOR MEAN PCF00450 C VCVCK = CHECKING MATRIX FOR COVARIANCES PCF00460 C NER = IMSL ROUTINE-USED PARAMETER VECTOR PCF00470 C NFC = CLASSIFICATION RESULT MATRIX PCF00480		DOUBLE PRECISION DSEED	PCF00280
DATA PC/NTERM*0.0/ DATA QP,PR,PX/NZ2*0.0,NZ2*0.0,NCLS*0.0/ DATA QP,PR,PX/NZ2*0.0,NZ2*0.0,NCLS*0.0/ DATA DSEED,NPC/5.D0,NZ1*0/ DATA (NBR(I),I=4,6),IOPT,NIN,NOUT/1,0,0,3,0,6/ C MATA (NBR(I),I=4,6),IOPT,NIN,NOUT/1,0,0,3,0,6/ C PCF00320 PCF00340 C XMT = MEAN VECTORS FOR ALL CLASSES C VCVT = COV. MATRICES FOR ALL CLASSES C VCVT = COV. MATRICES FOR ALL CLASSES C VCVT = INVERSE COV. MATRICES FOR ALL CLASSES C VCVIT = INVERSE COV. MATRICES FOR ALL CLASSES C VCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODE C VCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE C VCVCK = CHECKING VECTOR FOR MEAN C VCVCK = CHECKING VECTOR FOR MEAN C VCVCK = CHECKING MATRIX FOR COVARIANCES C VCVCK = CHECKING MATRIX FOR COVARIANCES C NBR = IMSL ROUTINE-USED PARAMETER VECTOR C NPC = CLASSIFICATION RESULT MATRIX PCF00480	•		PCF00290
DATA PC/NIERN*0.07PCF00310DATA QP, PR, PX/N22*0.0, N22*0.0, NCLS*0.0/PCF00310DATA DSEED, NPC/5.D0, N21*0/PCF00320DATA (NBR(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00390CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV = COV. MATRIXPCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVK = CHECKING VECTOR FOR MEANPCF00430CVCVCK = CHECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING MATRIX FOR COVARIANCESPCF00460CNBR = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC = CLASSIFICATION RESULT MATRIXPCF00480			PCF00300
DATA QP, PR, PX/NZ2*0.0, NZ2*0.0, NCLS*0.07PCF00310DATA DSEED, NPC/5.D0, NZ1*0/ DATA (NBR(I), I=4,6), IOPT, NIN, NOUT/1,0,0,3,0,6/PCF00320CCPCF00330CCPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00390CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV = COV. MATRIXPCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCV = COV. MATRIXPCF00420CYCW = MEAN VECTORPCF00430CPCC = PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CYCVCK = CHECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING MATRIX FOR COVARIANCESPCF00460CNBR = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC = CLASSIFICATION RESULT MATRIXPCF00480		DATA $PC/NIERM''''''''''''''''''''''''''''''''''''$	PCF00310
DATA DSEED, NPC/5.D0, NZ1*0/ DATA (NBR(I), I=4,6), IOPT, NIN, NOUT/1,0,0,3,0,6/PCF00320CDATA (NBR(I), I=4,6), IOPT, NIN, NOUT/1,0,0,3,0,6/PCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00390CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV = COV. MATRIXPCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CYM = MEAN VECTORPCF00430CPC = PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CYCVCK = CHECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING MATRIX FOR COVARIANCESPCF00450CNBR = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC = CLASSIFICATION RESULT MATRIXPCF00480		DATA $QP, PR, PX/NZ_{2} 0.0, NZ_{2} 0.0, NCLS_{0.0}$	PCF00310
DATA (NBR(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/PCF00330CXMT = MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT = COV. MATRICES FOR ALL CLASSESPCF00360CCT = M.L. DECISION RULE PARAMETERPCF00370CVCVIT = INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC = GENERATED SAMPLE VECTORSPCF00390CVCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV = COV. MATRIXPCF00410CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM = MEAN VECTORPCF00430CPC = PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CXMCK = CHECKING VECTOR FOR MEANPCF00450CVCVCK = CHECKING MATRIX FOR COVARIANCESPCF00460CNBR = IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC = CLASSIFICATION RESULT MATRIXPCF00480	· · · · ·	DATA DSEED, NPC/5.DU, N21*0/	PCF00320
CPCF00340CXMT= MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT= COV. MATRICES FOR ALL CLASSESPCF00360CCT= M.L. DECISION RULE PARAMETERPCF00370CVCVIT= INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC= GENERATED SAMPLE VECTORSPCF00390CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480		DATA $(NBR(I), I=4, 6), IOPT, NIN, NOUT/1, 0, 0, 3, 0, 6/$	PCF00330
CXMT= MEAN VECTORS FOR ALL CLASSESPCF00350CVCVT= COV. MATRICES FOR ALL CLASSESPCF00360CCT= M.L. DECISION RULE PARAMETERPCF00370CVCVIT= INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC= GENERATED SAMPLE VECTORSPCF00390CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXIN FULL STORAGE MODEPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	С		PCF00340
CVCVT= COV. MATRICES FOR ALL CLASSESPCF00360CCT= M.L. DECISION RULE PARAMETERPCF00370CVCVIT= INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC= GENERATED SAMPLE VECTORSPCF00390CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	C ·	XMT = MEAN VECTORS FOR ALL CLASSES	PCF00350
CCT= M.L. DECISION RULE PARAMETERPCF00370CVCVIT= INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC= GENERATED SAMPLE VECTORSPCF00390CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	C.	VCVT = COV. MATRICES FOR ALL CLASSES	PCF00360
CVCVIT= INVERSE COV. MATRICES FOR ALL CLASSESPCF00380CTVEC= GENERATED SAMPLE VECTORSPCF00390CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	č	CT = M I. DECISION BULE PARAMETER	PCF00370
CVCVII= INVERSE COV. MATRIX FOR TABLE CLASSIESPCF00390CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	č	VOVIT - INVERSE COV MATRICES FOR ALL CLASSES	PCF00380
CIVEC= GENERATED SAMPLE VECTORSICF00330CVCVIF= INVERSE COV. MATRIX IN FULL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480		THE - CENEDATED CAMPE VECTORS	DCE00300
CVCVIF= INVERSE COV. MATRIX IN FOLL STORAGE MODEPCF00400CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480		TVEC = GENERATED SATELLE VECTORS	DCE00400
CVCV= COV. MATRIXPCF00410CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	С	VCVIF = INVERSE COV. MATRIX IN FULL STORAGE MODE	PCF 00400
CVCVI= INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODEPCF00420CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	С	VCV = COV. MATRIX	PCF00410
CXM= MEAN VECTORPCF00430CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	С	VCVI = INVERSE COV. MATRIX IN SYMMETRIC STORAGE MODE	PCF00420
CPC= PROBABILITY OF CORRECT CLASSIFICATIONPCF00440CXMCK= CHECKING VECTOR FOR MEANPCF00450CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	С	XM = MEAN VECTOR	PCF00430
CXMCK=CHECKING VECTOR FOR MEANPCF00450CVCVCK=CHECKING MATRIX FOR COVARIANCESPCF00460CNBR=IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC=CLASSIFICATION RESULT MATRIXPCF00480	С	PC = PROBABILITY OF CORRECT CLASSIFICATION	PCF00440
CVCVCK= CHECKING MATRIX FOR COVARIANCESPCF00460CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	č	XMCK = CHECKING VECTOR FOR MEAN	PCF00450
CNBR= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480	č	VOVCK = CHECKING MATRIX FOR COVARIANCES	PCF00460
CNBK= IMSL ROUTINE-USED PARAMETER VECTORPCF00470CNPC= CLASSIFICATION RESULT MATRIXPCF00480			DCE00430
$C \qquad NPC = CLASSIFICATION RESULT MATRIX PCF00480$	C	NBK = IMSL KOUTINE-USED PARAMETER VECTOR	
	C	NPC = CLASSIFICATION RESULT MATRIX	PCF 00480

## DO 10 I=1,NTERM 10 TRACE=TRACE+TEST(I,I) RETURN END

CAN03230 CAN03240 CAN03250 CAN03260

2.2	THE THE THE TOTAL NO OF CAMPLES FOR FACH CLASS	PCF00490
C	NST = STORE THE TOTAL NO. OF SAMPLES FOR EACH CHROD	PCF00500
C .		PCF00510
с - с		PCF00520
C		PCF00530
č		PCF00540
č -		PCF00550
č	NSET F1 NP2 A B C DACO EXNU RUSE	PCF00560
č	1 M2611K1 832 WW:141 SF:414 GS:277 760928 76102207 1-1622	PCF00570
č	2 M2611K2 1551 WW:658 SF:211 UC:682 770503 77102207 6515-8096	PCF00580
C	3 M2611K3 1477 WW:677 SF:643 GS:157 770626 77102207 8097-9691	PCF00590
č	4 M2614N1 1265 SW:664 SF:437 NP:164 770508 77102217 1-1396	PCF00600
C	5 M2614N2 1239 SW:787 SF:291 NP:161 770629 77102217 2777-4141	PCF00610
C	6 M2614N3 1444 SW:931 SF:330 NP:183 770804 77102217 5426-6993	PCF00620
C	이는 것은 해외에서 이상에 가장 가장 있는 것은 것이 있는 것이 가장 같은 것을 했다. 것은 것은 것은 것은 것은 것은 것은 것이 가지 않는 것이 있는 것은 것을 가지 않는 것을 가지 않는 것은 것 같은 것은	PCF00630
Ċ	DATA NST/141,414,277,658,211,682,677,643,157/	PCF00640
C.	DATA NST/141,414,277,658,211,682,677,643,157,	PCF00650
C	+664,437,164,787,291,161,931,330,183/	PCF00660
С	DATA NST/664,437,164,787,291,161,931,330,183/	PCF00670
C	DATA NST/141,414,277,658,211,682,677/	PCF00680
С	DATA NST/587,216,121/	PCF00690
	DATA NST/658,211,682/	PCF00700
С		PCF00710
, C <sup>.</sup> ,		-PCF00720
C	THE FOLLOWING DATA 'NST' ARE USED FOR SOIL ORDER DATA SET. SO	PCF00730
С		PCF00740
С	NP2=479; MOL ALF EN AR UL IN SP VE H OX UNCLASSIFIED	PCF00750
C	DATA NST/154,113,/8,52,45,37,30,11,8,11,32/	PCF00780
C S	DATA NST/154,113,78,52,45,977	PCF00770
C	DATA NST/154,113, /8,52,45,3//	PCF00700
C		-PCF00800
С. С		PCF00810
	THE FOIL OWING DATA INSTITS LISED FOR SOIL 'OMI' DATA SET	PCF00820
C	TE (1) MOLLISOL OF (2) ALFISOL, AND GROUP SAMPLES	PCF00830
. C	ACCORDING TO THEIR ORGANIC MATERIAL: % WEIGHT	PCF00840
	$CLASS = 1 \text{ TO } 6  \cdot \text{ NP2} = 255$	PCF00850
č	CLS1 : 11% GE. OM .LE. 1.5% : $\#$ 1 -> $\#$ 51	PCF00860
č	CLS2 : 1.5% GT. OM .LE. 2.0% : # 52 -> # 104	PCF00870
C C	CLS3 : 2.0% .GT. OM .LE. 2.5% : # 105 -> # 138	PCF00880
č	CLS4 : 2.5% .GT. OM .LE. 3.5% : # 139 -> # 183	PCF00890
Ċ	CLS5 : 3.5% .GT. OM .LE. 5.0% : # 184 -> # 222	PCF00900
C	CLS6 : 5.0% .GT. OM .LE, 10.12% : # 223 -> # 255	PCF00910
С	방송 방송 전에 계획 것이 많이 많이 것이라. 방송 방송 말에 나온 것이는 것을 수	PCF00920
C	DATA NST/51, 54, 33, 45, 39, 33/	PCF00930
C	이는 것 같은 것에서 이상 것이 있는 것이다. 이는 것이 이 이는 것은 것은 것은 것은 것이다. 이는 것은 것은 것은 것이다. 가지 않는 것이 가지 않는 것이다. 가지 않는 것이다. 같은 것 같은 것은	PCF00940
С	DATA 'S2A' : ANOTHER TEST GROUPED BY THE SAME OM RANGES AS 'OM2'	PCF00950
.C .'	OM PERCENTAGE : 0,1; 1,2; 2,3; 3,4; 4,6; 6 AND ABOVE	PCF00960
Ċ	는 것 이렇는 것 같은 것이 없는 것을 알았는 것을 것을 것 같아. 아는 것 같이 많은 것을 수 있는 것을 못했다.	PCF00970
С	DATA NST/26,78,64,32,55/	PCF00980
C	의 가지는 것은 것은 것이 있는 것에서 가지 않는 것이 것을 가지 않는 것이다. 가지만 한 것은 것이 있는 것이 있는 것이 가지 않는 것이 가지 않는 것이다. 같은 것은 것은 것은 것은 것이 같은 것이 같은 것이 같은 것은 것이 같은 것이 같이 같이 같이 같이 같이 같이 같이 같이 있다.	PCF00990
С	이는 것 같은 것 같	PCF01000
С	에는 <u>이 같은 것은 것을 것을 수 있었다. 이 가지는 것</u> 에서 이 것을 가지 않는 것을 가지 않는 것이 있는 것이 가지 않는 것을 가지 않는 것이 가지 않는 것이 가지 않는 것이 가지 않는 것이다. 같은 것이 같은 것은 것은 것이 같은 것을 것 같은 것이다. 것이 같은 것이 같은 것이 같은 것이 같은 것이 같은 같은 것이 같은 것	PCF01010
С	THE FOLLOWING DATA 'NST' IS USED FOR 'OM2' DATA SET	PCF01020
C	ACCORDING TO THEIR ORGANIC MATERIAL: % WEIGHT	PCFUIU30
С	CLASS 1 TO 6 : NP2 = 514	PCFUIU40
С	CLS1 : .08% .GE. OM .LE. 1.0% : # 1 $\rightarrow$ # 82	LCLOT020

134

CLS2 : 1.0% .GT. OM .LE. 2.0% : # 83 -> # 217 CLS3 : 2.0% .GT. OM .LE. 3.0% : # 218 -> # 337 CLS4 : 3.0% .GT. OM .LE. 4.0% : # 338 -> # 391 CLS5 : 4.0% .GT. OM .LE. 6.0% : # 392 -> # 450 PCF01060 PCF01070 PCF01080 PCF01090 CLS6 : 6.0% .GT. OM .LE. 84.79% : # 451 -> # 514 PCF01100 PCF01110 DATA NST/82,135,120,54,59,64/ PCF01120 PCF01130 DATA NST/82,135,120,54,123/ PCF01140 DATA NST/44, 31, 18, 23, 24, 51, 37, 27/ PCF01150 DATA NST/83, 57, 94, 31, 37, 59, 103, 26, 24/ PCF01160 DATA NST/103,26,24/ PCF01170 PCF01180 \_\_\_\_ THE FOLLOWING DATA 'NST' IS USED FOR SOIL IRON OXIDE 'IO' DATA SETPCF01190 ACCORDING TO THEIR FE2O3 % WEIGHT PCF01200 PCF01210 CLASS 1 TO 6 : NP2 = 467CLS1 : .02% .GE. FE2O3 .LE. 0.4% : # 1 -> # 102 PCF01220 CLS2 : 0.4% .GT. FE2O3 .LE. 0.6% : # 103 -> # 175 PCF01230 CLS3 : 0.6% .GT. FE2O3 .LE. 0.8% : # 176 -> # 244 PCF01240 CLS4 : 0.8% .GT. FE2O3 .LE. 1.2% : # 245 -> # 349 CLS5 : 1.2% .GT. FE2O3 .LE. 1.6% : # 350 -> # 401 CLS6 : 1.6% .GT. FE2O3 .LE. 25.60% : # 402 -> # 467 PCF01250 PCF01260 PCF01270 PCF01280 PCF01290 DATA NST/102,73,69,105,52,66/ PCF01300 PCF01310 THE FOLLOWING DATA 'NST' IS USED FOR SOIL TEXTURE 'ST' DATA SET PCF01320 ACCORDING TO THEIR SAND-SILT-CLAY % CONTENT PCF01330 CLASS 1 TO 6 : NP2 = 483; DETAILS : SEE FILE ( S5L.DATA.C1) PCF01340 PCF01350 PCF01360 DATA NST/40,63,76,93,68,143/ PCF01370 \_\_\_\_\_ PCF01380 THE FOLLOWING DATA 'NST' IS USED FOR S.D. VEGETATION DATA PCF01390 PCF01400 DATA NST/225,61,292,469, 82,182,63,103, 39,39,217,51, PCF01410 PCF01420 +393,441,80,88, 88,41,32,26, 118,43,121,44, 45,102,66,89, +78, 53, 147, 39, 24, 42, 119, 69, 76, 96, 107, 154, 28, 19/ PCF01430 PCF01440 PCF01450 THE FOLLOWING DATA 'NST' IS USED FOR IOWA VEGETATION DATA PCF01460 PCF01470 PCF01480 DATA NST/514,41, 517,36,32, 621,517,45, 610,485,21, DATA NST/514,41, 517,36,32, 621,517,43, 610,403,21, +437,377,22, 190,172,25, 650,568,42, 435,417,44, 393,267/ PCF01490 PCF01500 \_\_\_\_\_ PCF01510 PCF01520 PCF01530 11 = TRANSFORMED DATA; 12 = CLASS STATISCTICS; 13 = PC PCF01540 PCF01550 OPEN(11) PCF01560 OPEN(12)PCF01570 OPEN (13) PCF01580 REWIND 11 PCF01590 REWIND 12 REWIND 13 PCF01600 PCF01610 NX2=0 DO 1 I=1,NCLS PCF01620

С

C

С

C C

C C

С

С

С

С

C

С

C C

С

С

С

С

С

Ċ

С

С

С

C

С

C

С

С

C

Ċ

C

C

C C

C

C

С

С

С

C

Ç

С

С

С

С

С

С

С
_			PCF01630
1	NX2=NX2+NST(1)		DCF01640
	IF(IRES.EQ.0)GO TO 3		
	DO 2 I=1,NCLS		PCF 01 650
	NX1=NST(I)		PCF01660
2	AP(I) = FLOAT(NX1) / FLOAT(NX2)		PCF01670
	GO TO 5		PCF01680
3	DO 4 I=1, NCLS		PCF01690
4	AP(I) = 1.0/FLOAT(NCLS)		PCF01700
. 5	IK=MOD (NCLS, 6)	A. A.	PCF01710
			PCF01720
	SET THE INPUT&OUTPUT DO LOOP PARAMETERS		PCF01730
			PCF01740
	TM = 6 * (NCT.S/6) + 1		PCF01750
	$\frac{11-0}{100} (\text{NCLS} 3)$		PCF01760
	TM1 - 2 + (NCT C / 2) + 1		PCF01770
	$\frac{1}{1} \frac{1}{2} \frac{1}{1} \frac{1}$		PCF01780
	IRZ=MOD(NCLO, IO) IRZ=MOD(NCLO, IO)		PCF01790
	$IMZ = 10^{\circ} (NCLO/10) + 1$		PCF01800
	$\frac{11}{11} = N = \frac{1}{10} = 0$	1997 - 1997 -	PCF01810
			PCF01820
. ÷			PCF01830
	$\frac{11}{11} \frac{11}{2} \frac{12}{10} \frac{11}{10} \frac{11}$		DCE01840
	ILP3=NCLS/15		DCE01050
	IF(ILP3.EQ.0) ILP3=1		DCE01050
	IF (IRES.EQ.U) NSAMP=NSMAX		PCF01000
	DO 550 ISET=NSET, MSET, NDSET	<b>T )</b>	PCF01070
	IF (IRES.EQ.1) CALL RDATA (ISET, RVEC, NSMAX, NTERMC, NCLS, NS	1)	PCF01880
			PCFU1890
	READ IN CLASS STATISTICS		PCF01900
			PCF01910
	DO 500 LTERM=1,NTERM		PCF01920
	KTERM=LTERM* (LTERM+1)/2		PCF01930
	DO 30 ITERM=1, LTERM		PCF01940
	IF (NCLS.LT.6) GO TO 20		PCF01950
	DO 10 IL=1,ILP1		PCF01960
10	READ (12, *) (XMT (ITERM, JCLS), JCLS=1+(IL-1)*6, IL*6)		PCF01970
	IF (IK.EQ.0) GO TO 30		PCF01980
20	READ (12, *) (XMT (ITERM, JCLS), JCLS=IM, NCLS)		PCF01990
30	CONTINUE		PCF02000
	DO 60 ITERM=1, KTERM		PCF02010
	IF (NCLS.LT.6) GO TO 50		PCF02020
	DO 40 IL=1, ILP1		PCF02030
40	READ (12, *) (VCVT (ITERM, JCLS), JCLS=1+(IL-1)*6, IL*6)		PCF02040
	IF (IK.EQ.0) GO TO 60	· · · · · ·	PCF02050
50	READ (12, *) (VCVT (ITERM, JCLS), JCLS=IM, NCLS)		PCF02060
60	CONTINUE	·	PCF02070
	IF (NCLS.LT.6) GO TO 80		PCF02080
	DO 70 IL=1, ILP1		PCF02090
70	READ(12,*)(CT(ICLS),ICLS=1+(IL-1)*6,IL*6)		PCF02100
	IF (IK.EO.0) GO TO 90		PCF02110
80	READ(12,*)(CT(ICLS), ICLS=IM, NCLS)		PCF02120
an	DO 120  TTFRM=1. KTERM	a di serie de la della	PCF02130
50	TR (NCLS LT $6$ ) CO TO 110		PCF02140
· .	T (100.101.000 TO TO TO 100		PCF02150
100	DU 100 $10^{+1}$ , $10^{-1}$ DEND (12 *) (UCUTT (TTEDM TCIC) TCIC=1+(TI-1)*6 TI*6)	de la cara de	PCF02160
	$\frac{1}{12} \frac{1}{12} \frac$		PCF02100
110	$\frac{12}{2} \frac{12}{2} \frac$		PCF02190
100			PCF02100
τZU			10102100

С С С

> C C C

IF (IFIND.EQ.1) GO TO 125 PCF02200 IF (LTERM. NE. NTERM) GO TO 500 PCF02210 IF (IFIND.EQ.0) GO TO 128 PCF02220 PCF02230 C FIND THE PC RESULTS FOR EVERY DTERM INCREMENT PCF02240 C PCF02250 125 NX1=LTERM+ (NDTRM-1) PCF02260 NX2=MOD (NX1, NDTRM) PCF02270 PCF02280 PRINT\*, NX1, NX2 С IF (NX2.NE.0) GO TO 500 PCF02290 128 DO 170 JCLS=1, NCLS PCF02300 DO 130 I=1, KTERM PCF02310 130 VCV (I) = VCVT (I, JCLS) PCF02320 CALL UGETIO (IOPT, NIN, NOUT) PCF02330 CALL USWSM(' THE MATRIX IS ',15,VCV,LTERM,1) Ċ PCF02340 NOTE : WK(1) MUST BE 0.0 EVERY TIME TO INITIALIZE ' GGNSM ' PCF02350 С NOTE : VCV WILL BE CHANGED AFTER ' GGNSM' PCF02360 С IF (IRES.EQ.1) NSAMP=NST (JCLS) PCF02370 IF (IRES.EO.1) GO TO 145 PCF02380 PCF02390 DO 140 I=1, NTERM 140 WK(I) = 0.0PCF02400 DSEED=5.D0 PCF02410 PCF02420 C GENERATE GAUSSIAN SAMPLES ACCORDING TO THE CLASS STATISTICS PCF02430 С PCF02440 С CALL GGNSM (DSEED, NSAMP, LTERM, VCV, NSMAX, VEC, WK, IER) PCF02450 145 DO 155 I=1, NSAMP PCF02460 DO 155 J=1, LTERM PCF02470 IF (IRES.EQ.1) GO TO 150 PCF02480 VEC(I, J) = VEC(I, J) + XMT(J, JCLS)PCF02490 С PCF02500 С STORE THE SAMPLES INTO ARRAY 'TVEC' PCF02510 С PCF02520 PCF02530 TVEC (I, J, JCLS) = VEC (I, J)PCF02540 GO TO 155 PCF02550 150 TVEC(I, J, JCLS) = RVEC(I, J, JCLS) VEC(I, J)=RVEC(I, J, JCLS) PCF02560 С PRINT\*, JCLS, I, J, TVEC (I, J, JCLS) PCF02570 PCF02580 155 CONTINUE PCF02590 IF (ICKMV.EQ.0)GO TO 170 PCF02600 C CHECK THE MEAN VECTOR AND COV. MATRIX OF THE GENERATED SAMPLES С PCF02610 THE MATRIX 'VEC' WILL BE CHANGED AFTER ' BECOVM ' PCF02620 C PCF02630 С DO 160 I=1, NTERM PCF02640 160 TX(I) = 0.0PCF02650 PCF02660 NBR(1)=LTERM PCF02670 NBR(2)=NSAMP PCF02680 NBR(3)=NSAMP IF (LTERM.GT.1) GO TO 600 PCF02690 Ċ CALL BECOVM (VEC, NSMAX, NBR, TX, XMCK, VCVCK, IER) PCF02700 Ċ PCF02710 SEND THE CHECKING RESULTS TO THE SCREEN IF NEEDED PCF02720 С Ċ PCF02730 С CALL USWFV (' THE VECTOR IS ', 15, XMCK, LTERM, 1, 1) PCF02740 CALL USWSM(' THE MATRIX IS ',15, VCVCK, LTERM, 1) PCF02750 PCF02760 **170 CONTINUE** 

PCF02770 C PCF02780 START CLASSIFICATION JOB FOR EACH CLASS SAMPLES С PCF02790 С PCF02800 DO 230 JCLS=1, NCLS PCF02810 IF (IRES.EO.1) NSAMP=NST (JCLS) PCF02820 PRINT\*, LTERM, JCLS, NSAMP PCF02830 DO 230 ISAMP=1, NSAMP PCF02840 DO 180 J=1, LTERM PCF02850 180 Y (J) = TVEC (ISAMP, J, JCLS) PCF02860 DO 220 KCLS=1, NCLS THE FOLLOWING IS NEEDED SINCE X HAS BEEN CHANGED FOR EVERY KCLS! PCF02870 C PCF02880 DO 190 I=1, LTERM PCF02890 190 X(I) = Y(I)PCF02900 DO 200 I=1, KTERM PCF02910 200 VCVI(I)=VCVIT(I,KCLS) PCF02920 CALL VCVTSF (VCVI, LTERM, VCVIF, NTERM) PCF02930 DO 210 I=1, LTERM PCF02940 210 XM(I) = XMT(I, KCLS)PCF02950 CALL SAXPY (LTERM, -1., XM, 1, X, 1) CALL VMULFM (X, VCVIF, LTERM, 1, LTERM, NTERM, NTERM, T1, 1, IER) PCF02960 PCF02970 CALL VMULFF (T1, X, 1, LTERM, 1, 1, NTERM, T2, 1, IER) PCF02980  $T3 = EXP(-0.5 \times T2)$ 220 PX (KCLS) = AP (KCLS) \* CT (KCLS) \* T3 PCF02990 PCF03000 C PCF03010 PERFORM M.L. DECISION RULE С PCF03020 С PCF03030 CALL VABMXF (PX (1), NCLS, 1, IMAX, BIG) NPC (JCLS, IMAX, LTERM) = NPC (JCLS, IMAX, LTERM) +1 PCF03040 CALL VABSMF (PX, NCLS, 1, DEN) PCF03050 С PCF03060 С O=BIG/DEN WRITE (13, \*) JCLS, ISAMP, IMAX, NPC (JCLS, IMAX, LTERM) PCF03070 C WRITE (13, \*) (PX(I), I=1, NCLS), IMAX, BIG PCF03080 С PCF03090 230 CONTINUE PCF03100 C FIND PROBABILITY OF CORRECT CLASSIFICATION PC FROM NPC PCF03110 С PCF03120 C PCF03130 NC1=0PCF03140 NC2=0PCF03150 DO 240 I=1, NCLS PCF03160 IF (IRES.EQ.0) NST (I) = NSMAX PR(I, LTERM) = (FLOAT (NPC(I, I, LTERM))) / FLOAT (NST(I)) PCF03170 PCF03180 NC1=NC1+NPC(I,I,LTERM) PCF03190 240 NC2=NC2+NST(I)PCF03200 IF (IRES.EQ.0) NC2=NSMAX\*NCLS PCF03210 PC (LTERM) = (FLOAT (NC1)) / FLOAT (NC2) PCF03220 IF (NCLS.LT.3) GO TO 260 PCF03230 C PCF03240 С SEND THE RESULTS TO THE SCREEN PCF03250 Ċ PCF03260 DO 250 IL=1, ILP2 PCF03270 250 WRITE (\*, \*) ISET, LTERM, (PR(I, LTERM), I=1+(IL-1)\*3, IL\*3) PCF03280 IF (IK1.EQ.0) GO TO 270 260 WRITE (\*,\*) ISET, LTERM, (PR(I, LTERM), I=IM1, NCLS) PCF03290 PCF03300 270 PRINT\*, ISET, LTERM, PC (LTERM) PCF03310 C SEND THE RESULTS TO THE PC FILE PCF03320 C PCF03330 С

PCF03340 WRITE (13, \*) ' LTERM = ', LTERM PCF03350 IF (NCLS.LT.6) GO TO 290 PCF03360 DO 280 IL=1, ILP1 280 WRITE (13, 301) (PR(I, LTERM), I=1+(IL-1)\*6, IL\*6) PCF03370 PCF03380 IF (IK.EQ.0) GO TO 300 PCF03390 290 WRITE (13, 301) (PR (I, LTERM), I=IM, NCLS) PCF03400 300 WRITE (13, 301) PC (LTERM) 301 FORMAT (6F13.5) PCF03410 PCF03420 С PCF03430 C---< RESET ALL RELATED VARIABLES >-----THE FOLLOWING ZEROING PROCEDURES ARE 'ABSOLUTELY' NEEDED!! PCF03440 C THIS IS DONE FOR EVERY " LTERM = 1, NTERM " PCF03450 С PCF03460 С PCF03470 DO 310 K=1.NCLS PCF03480 DO 310 I=1, NSMAX PCF03490 DO 310 J=1, NTERM PCF03500 310 TVEC(I, J, K) = 0.0PCF03510 DO 320 I=1, NCLS PCF03520 DO 320 J=1.NTERM PCF03530 QP(I, J) = 0.0PCF03540 320 PR(I, J) = 0.0PCF03550 DO 330 I=1,NTERM PCF03560 330 PC(I) = 0.0PCF03570 IF (NCLS.LT.15) GO TO 360 PCF03580 С SEND THE FINAL CLASSIFICATION MATRIX NPC TO THE PC FILE PCF03590 C PCF03600 С PCF03610 DO 350 J=1, ILP3 PCF03620 DO 340 I=1,NCLS 340 WRITE (13, 341) I, (NPC (I, K, LTERM), K=1+ (J+1)\*15, J\*15) PCF03630 PCF03640 341 FORMAT (13, 2X, 1515) PCF03650 WRITE (13, 342) XC1 PCF03660 342 FORMAT (A2) PCF03670 350 CONTINUE PCF03680 IF (IK2.EQ.0) GO TO 500 PCF03690 360 DO 370 I=1,NCLS PCF03700 370 WRITE (13, 341) I, (NPC (I, K, LTERM), K=IM2, NCLS) PCF03710 500 CONTINUE PCF03720 DO 510 I=1, NCLS PCF03730 DO 510 J=1,NCLS PCF03740 DO 510 K=1, NTERM PCF03750 510 NPC(I, J, K) = 0PCF03760 550 CONTINUE PCF03770 С PCF03780 THE FOLLOWING STATEMENT IS USED FOR INTERNAL CHECKING С PCF03790 С PCF03800 600 STOP С PCF03810 STOP PCF03820 END PCF03830 SUBROUTINE RDATA (LSET, RVEC, NSMAX, NTERMC, NCLS, NST) PCF03840 REAL RVEC (NSMAX, NTERMC, NCLS) PCF03850 INTEGER NST (NCLS) PCF03860 IKX=MOD (NTERMC, 5) PCF03870 IMX=5\*(NTERMC/5)+1PCF03880 ILPX=NTERMC/5 PCF03890 IF (ILPX.EQ.0) ILPX=1 PCF03900 IFILE1=11+(LSET-1)\*10

```
DO 40 K=1,NCLS
N1=NST(K)
PRINT*,'KCLS = ',K,'; NSAMP = ',N1
DO 30 I=1,N1
IF(NTERMC.LT.5)GO TO 20
DO 10 J1=1,ILPX
10 READ(IFILE1,*)(RVEC(I,J,K),J=1+(J1-1)*5,J1*5)
IF(IKX.EQ.0)GO TO 30
20 READ(IFILE1,*)(RVEC(I,J,K),J=IMX,NTERMC)
30 CONTINUE
40 CONTINUE
RETURN
END
```

 $\bigcirc$ 

Ŕ

PCF03910 PCF03920 PCF03930 PCF03940 PCF03950 PCF03960 PCF03970 PCF03980 PCF03990 PCF04000 PCF04010 PCF04020 PCF04030