

# Representation-Constrained Canonical Correlation Analysis

## A Hybridization of Canonical Correlation and Principal Component Analyses

SK Mishra  
 Department of Economics  
 North-Eastern Hill University  
 Shillong, Meghalaya (India)  
 Contact: [mishrasknehu@yahoo.com](mailto:mishrasknehu@yahoo.com)

**I. Introduction:** We begin this paper with reference to a dataset that, when subjected to the classical canonical correlation analysis, gives us the leading (first or largest) canonical correlation which is misleading. It is misleading in the sense that, in this example, the canonical correlation (which is the coefficient of correlation between the two canonical variates, each being a linear weighted combination of the variables in the associated dataset) is, indeed, not a measure of the true association of the variables in the two datasets, but, instead, the datasets have been hijacked by a lone couple of variables across the two datasets.

In Table-1.1 the dataset X is presented which is a pooled set of two datasets,  $X_1$  and  $X_2$ , such that  $X=[X_1|X_2]$ . The first dataset has  $m_1$  (=4) variables and the second dataset has  $m_2$  (=5) variables, each in  $n$  (=30) observations. These seemingly normal datasets, when subjected to the classical canonical correlation analysis, yield canonical correlation between the composite variables,  $z_1$  and  $z_2$  (the canonical variates),  $r(z_1, z_2) = 1.0$  :  $z_1 = \sum_{j=1}^4 w_j x_{1j}; x_{ij} \in X_1$ ;  $z_2 = \sum_{j=1}^5 w_j x_{2j}; x_{2j} \in X_2$ . The weight vectors are:  $w_1=(1, 0, 0, 0, 0)$  and  $w_2=(0, 0, 0, 0, 1)$ . This anomalous situation has arisen due to the fact that  $x_{25}$  is perfectly linearly dependent on  $x_{11}$  and the canonical correlation,  $r(z_1, z_2)$ , is in fact  $r(x_{11}, x_{25})$ . Other variables have no contribution to  $z_1$  or  $z_2$ . It follows, therefore, that  $z_1$  and  $z_2$  do not represent other variables in  $X_1$  and  $X_2$ . Nor is the canonical correlation,  $r(z_1, z_2)$ , a correlation between the two sets,  $X_1$  and  $X_2$ , in any relevant or significant sense. Thus, the leading canonical correlation may deceive us if we are only a little less careful to look into the correlation matrix encompassing all variables.

Sl No.	X <sub>1</sub> or Dataset-1				X <sub>2</sub> or Dataset-2				
	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>21</sub>	X <sub>22</sub>	X <sub>23</sub>	X <sub>24</sub>	X <sub>25</sub>
1	0.7	2.6	0.1	1.7	0.2	0.8	1.6	0.5	1.6
2	1.5	1.7	1.2	1.5	1.6	2.4	2.3	1.4	3.2
3	2.3	0.3	2.7	1.2	2.5	2.9	0.6	1.3	4.8
4	0.6	2.0	0.9	2.8	2.8	2.5	1.1	1.8	1.4
5	0.1	0.9	1.6	1.8	2.2	2.7	2.1	0.2	0.4
6	1.9	1.1	1.7	2.6	1.5	2.2	2.2	2.0	4.0
7	1.0	2.7	2.4	2.7	1.0	0.2	2.0	0.4	2.2
8	1.8	2.9	1.4	0.9	1.7	1.0	1.8	1.2	3.8
9	2.8	0.1	1.8	0.4	2.3	0.6	1.7	0.6	5.8
10	1.4	0.6	2.8	1.4	2.6	1.8	0.8	1.7	3.0
11	1.2	2.5	2.9	0.8	2.1	0.7	1.4	2.3	2.6
12	1.1	1.3	0.2	2.5	0.7	1.5	1.0	2.2	2.4
13	3.0	1.9	1.1	1.6	0.1	0.1	2.7	3.0	6.2
14	2.0	0.8	0.6	1.3	1.9	0.5	0.4	0.8	4.2
15	1.6	2.2	2.6	1.9	1.4	1.3	1.3	2.5	3.4
16	2.9	0.7	1.9	2.9	2.4	1.2	2.5	2.1	6.0
17	1.3	1.4	2.0	0.2	1.8	2.8	0.3	2.6	2.8
18	0.8	0.2	2.3	2.0	2.9	1.4	3.0	0.7	1.8
19	1.7	0.5	1.3	0.1	2.0	0.9	2.9	1.5	3.6
20	2.1	2.4	0.7	0.5	0.9	2.3	0.7	0.3	4.4
21	2.5	1.0	3.0	2.2	1.2	2.6	2.6	1.0	5.2
22	2.2	2.8	2.5	0.7	3.0	3.0	0.2	1.9	4.6
23	0.5	0.4	0.8	1.0	0.8	0.4	0.1	1.1	1.2
24	2.7	2.1	1.5	2.3	1.1	1.1	0.9	2.7	5.6
25	2.4	1.8	0.5	0.3	2.7	1.6	2.8	0.1	5.0
26	0.2	1.6	0.3	1.1	0.6	0.3	2.4	2.8	0.6
27	0.9	2.3	0.4	0.6	1.3	1.7	1.5	2.4	2.0
28	2.6	3.0	2.2	3.0	0.5	1.9	1.9	1.6	5.4
29	0.4	1.2	1.0	2.4	0.4	2.0	0.5	2.9	1.0
30	0.3	1.5	2.1	2.1	0.3	2.1	1.2	0.9	0.8

Such examples may be multiplied *ad infinitum*. If one is cautious, the anomalous cases can be detected. However, such cases, if not detected, make scientific analysis and

interpretation of empirical results rather hazardous. One may easily be misled to a conclusion that such two datasets are highly correlated while the truth may be quite far from it.

**II. Objectives of the Present Work:** We intend here to propose an alternative measure of association between two sets of variables that will not permit the greed of a select few variables in the datasets to prevail upon the fellow variables so much as to deprive the latter of contributing their say and share to the representative variables ( $\zeta_1$  and  $\zeta_2$ ), which they make by their participation in the linear combination. We may not call  $\zeta_1 = \sum_{j=1}^{m_1} \omega_{1j} x_{1j}$  and  $\zeta_2 = \sum_{j=1}^{m_2} \omega_{2j} x_{2j}$  the canonical variables (defined before as  $z_1 = \sum_{j=1}^4 w_j x_{1j}$ ;  $z_2 = \sum_{j=1}^5 w_j x_{2j}$  obtained from the classical canonical correlation analysis).

In the classical canonical correlation analysis the objective is to maximize  $r^2(z_1, z_2)$ :  $z_1 = \sum_{j=1}^{m_1} w_{1j} x_{1j}$ ;  $z_2 = \sum_{j=1}^{m_2} w_{2j} x_{2j}$  irrespective of  $r(z_1, x_{1j})$ :  $x_{1j} \in X_1$  and  $r(z_2, x_{2j})$ :  $x_{2j} \in X_2$ , and, therefore,  $r^2(z_1, z_2)$  is subject to an unconstrained maximization. However, in the method that we are proposing here, the objective will be to maximize  $r^2(\zeta_1, \zeta_2)$ :  $\zeta_1 = \sum_{j=1}^{m_1} \omega_{1j} x_{1j}$  and  $\zeta_2 = \sum_{j=1}^{m_2} \omega_{2j} x_{2j}$  with certain constraints in terms of  $r(\zeta_1, x_{1j})$ :  $x_{1j} \in X_1$  and  $r(\zeta_2, x_{2j})$ :  $x_{2j} \in X_2$ . These constraints would ensure the representativeness of  $\zeta_1$  to  $X_1$  and that of  $\zeta_2$  to  $X_2$ . Hence, the proposed method may be called the *Representation-Constrained Canonical Correlation Analysis*.

**III. The Nature and Implications of the Proposed Constraints:** There are a number of ways in which the canonical variates can be constrained insofar as their association and concordance with their fellow variables in their respective native datasets are concerned. In other words, their representativeness to their native datasets can be defined variously. We discuss here some of the alternatives in terms of correlation as a measure of representativeness.

- (i) Mean absolute correlation principle: A (constrained) canonical variate  $\zeta_a = \sum_{j=1}^{m_a} \omega_{aj} x_{aj}$ ;  $x_{aj} \in X_a$  is a better representative of  $X_a$  if the mean absolute correlation,  $\sum_{j=1}^{m_a} |r(\zeta_a, x_{aj})|$ , is larger. This approach is equalitarian in effect.
- (ii) Mean squared correlation principle: A (constrained) canonical variate  $\zeta_a = \sum_{j=1}^{m_a} \omega_{aj} x_{aj}$ ;  $x_{aj} \in X_a$  is a better representative of  $X_a$  if the mean squared correlation,  $\sum_{j=1}^{m_a} r^2(\zeta_a, x_{aj})$ , is larger. This approach is elitist in effect, favouring dominant members.
- (iii) Minimal absolute correlation principle: A (constrained) canonical variate  $\zeta_a = \sum_{j=1}^{m_a} \omega_{aj} x_{aj}$ ;  $x_{aj} \in X_a$  is a better representative of  $X_a$  if the minimal absolute correlation,  $\min_j |r(\zeta_a, x_{aj})|$ , is larger. A larger  $\min_j |r(\zeta_a, x_{aj})|$  implies that the minimal squared correlation,  $\min_j [r^2(\zeta_a, x_{aj})]$ , is larger. This approach is in favour of the weak.

These three approaches lead to three alternative objective functions:

- (i). Maximize  $r^2(\zeta_1, \zeta_2) + \lambda [\sum_{j=1}^{m_1} |r(\zeta_1, x_{1j})| / m_1 + \sum_{j=1}^{m_2} |r(\zeta_2, x_{2j})| / m_2]$ :  $\zeta_1 = \sum_{j=1}^{m_1} \omega_{1j} x_{1j}$ ;  $\zeta_2 = \sum_{j=1}^{m_2} \omega_{2j} x_{2j}$ .
- (ii). Maximize  $r^2(\zeta_1, \zeta_2) + \lambda [\sum_{j=1}^{m_1} r^2(\zeta_1, x_{1j}) / m_1 + \sum_{j=1}^{m_2} r^2(\zeta_2, x_{2j}) / m_2]$ :  $\zeta_1 = \sum_{j=1}^{m_1} \omega_{1j} x_{1j}$ ;  $\zeta_2 = \sum_{j=1}^{m_2} \omega_{2j} x_{2j}$ .
- (iii). Maximize  $r^2(\zeta_1, \zeta_2) + \lambda [\min_j |r(\zeta_1, x_{1j})| + \min_j |r(\zeta_2, x_{2j})|]$ :  $\zeta_1 = \sum_{j=1}^{m_1} \omega_{1j} x_{1j}$ ;  $\zeta_2 = \sum_{j=1}^{m_2} \omega_{2j} x_{2j}$ .

In these objective functions, the value of  $\lambda$  may be chosen subjectively. If  $\lambda = 0$ , the objective function would degenerate to the classical canonical correlation analysis, but  $\lambda$  has no upper bound. Also note that if the first term is  $|r(\zeta_1, \zeta_2)|$  rather than  $r^2(\zeta_1, \zeta_2)$  and  $\lambda \neq 0$ , its implied weight vis-à-vis the second term increases since  $|r(\zeta_1, \zeta_2)| > r^2(\zeta_1, \zeta_2)$  for  $|r| < 1$ .

**IV. The Method of Optimization:** The classical canonical correlation analysis (Hotelling, 1936) sets up the objective function to maximize  $r^2(\zeta_1, \zeta_2) : \zeta_1 = \sum_{j=1}^{m_1} \omega_{1j} x_{1j}; \zeta_2 = \sum_{j=1}^{m_2} \omega_{2j} x_{2j}$  and using the calculus methods of maximization resolves the problem to finding out the largest eigenvalue and the associated eigenvector of the matrix,  $[X_1'X_1]^{-1}X_1'X_2[X_2'X_2]^{-1}X_2'X_1$ . The largest eigenvalue turns out to be the leading  $r^2(z_1, z_2) : z_1 = \sum_{j=1}^{m_1} w_{1j} x_{1j}; z_2 = \sum_{j=1}^{m_2} w_{2j} x_{2j}$ , and the standardized eigenvector is used to obtain  $w_1$  and  $w_2$ . However, a general calculus-based method cannot be applied to maximize the (arbitrary) objective function set up for the constrained canonical correlation analysis. At any rate, the first and the third objective functions are not amenable to maximization by the calculus-based methods.

We choose, therefore, to use a relatively new and more versatile method of (global) optimization, namely, the Particle Swarm Optimization (PSO) proposed by Eberhart and Kennedy (1995). A lucid description of its foundations is available in Fleischer (2005). The PSO is a biologically inspired population-based stochastic search method modeled on the ornithological observations, simulating the behavior of members of the flocks of birds in searching food and communicating among themselves. It is in conformity with the principles of decentralized decision making (Hayek, 1948; 1952) leading to self-organization and macroscopic order. The effectiveness of PSO has been very encouraging in solving extremely difficult and varied types of nonlinear optimization problems (Mishra, 2006). We have used a particular variant of the PSO called the Repulsive Particle Swarm Optimization (Urfalioglu, 2004).

**V. Findings and Discussion:** We have subjected the data in Table-1.1 to the representation-constrained canonical correlation analysis with the three alternative objective functions elaborated in section-III. The first term, measuring the degree of association between the two datasets,  $X_1$  and  $X_2$ , is in the squared form, that is  $r^2(\zeta_1, \zeta_2)$ , although we have reported its positive square root ( $=|r(\zeta_1, \zeta_2)|$ ) in Table-1.2. The three objective functions have been optimized for the different values of  $\lambda$ , varying from zero to 50 with an increment of 0.5. For the first objective function, the values of  $|r(\zeta_1, \zeta_2)|$ , mean absolute  $r(\zeta_1, x_1)$  and mean absolute  $r(\zeta_2, x_2)$  at different values of  $\lambda$  have been plotted in Fig.-1.1. Similarly, for the second objective function, the values of  $|r(\zeta_1, \zeta_2)|$ , mean squared  $r(\zeta_1, x_1)$  and mean squared  $r(\zeta_2, x_2)$  at different values of  $\lambda$  have been plotted in Fig.-1.2. Fig.-1.3 presents  $|r(\zeta_1, \zeta_2)|$ , minimum absolute  $r(\zeta_1, x_1)$  and minimum absolute  $r(\zeta_2, x_2)$  relating to the 3<sup>rd</sup> objective maximized at different values of  $\lambda$ .

From Fig.-1.1 and Fig.-1.2 it is clear that for increasing values of  $\lambda$ , the value of  $|r(\zeta_1, \zeta_2)|$  decreases monotonically, while the values of mean absolute (or squared)  $r(\zeta_1, x_1)$  and mean absolute (or squared)  $r(\zeta_2, x_2)$  increase monotonically. All of them exhibit asymptotic tendencies. However, for the third objective function the monotonicity of all the correlation functions is lost (shown in Fig.-1.3). Of course, the trends in minimum absolute  $r(\zeta_1, x_1)$  and minimum absolute  $r(\zeta_2, x_2)$  are clearly observable. These observations may be useful to the choice of  $\lambda$ . For the case

that we are presently dealing with, the value of  $\lambda$  need not exceed 10 to assure a fairly satisfactory representation of the two datasets by the corresponding canonical variates.

In particular, optimization of the second objective function has shown that the values of mean squared  $r(\zeta_1, x_1)$  and mean squared  $r(\zeta_2, x_2)$  exhibit asymptotic tendencies. For  $\lambda=50$ , the mean squared  $r(\zeta_1, x_1)$  is 0.3176 while the mean squared  $r(\zeta_2, x_2)$  is 0.2870.

Now, let us digress for a while to compute the first principal components of  $X_1$  and  $X_2$  (from the data given in Table-1.1). We find that for  $X_1$  the sum of squared correlation (component loadings) of the component score ( $\xi_1$ ) with its constituent variables is 0.317757. In other words, the first eigenvalue of the inter-correlation matrix  $R_1$  obtained from  $X_1$  is 1.271029, which divided by 4 (order of  $R_1$ ) gives 0.317757. This is, in a way, a measure of representation of  $X_1$  by its first principal component. Similarly, for  $X_2$  the sum of squared correlation of the component score ( $\xi_2$ ) with its constituent variables is 0.287521.

We resume our discussion for comparing these results (obtained from the Principal Component Analysis) with the results of our proposed representation-constrained canonical correlation analysis. We observe that the asymptotic tendencies of mean squared  $r(\zeta_1, x_1)$  and mean squared  $r(\zeta_2, x_2)$  clearly point to the explanatory powers of the first principal components of  $X_1$  and  $X_2$  respectively.

However, if we compute the coefficient of correlation between the two component scores ( $r(\xi_1, \xi_2)=0.390767$ ) and compare it with the constrained canonical correlation ( $r(\zeta_1, \zeta_2)=0.4480$  for  $\lambda=50$ ) we find that the latter is larger. Then, is the constrained canonical correlation analysis a hybrid of the classical canonical correlation and principal component analyses which has better properties of representation of data than its parents?

We conduct another experiment with the dataset presented in Table-2.1. We find that  $\xi_1$  for  $X_1$  has the representation power 0.333261 (eigenvalue=1.333042) while  $\xi_2$  for  $X_2$  has the representation power 0.382825 (eigenvalue=1.914123). The  $r(\xi_1, \xi_2)$  is 0.466513. On the other hand, results of the constrained canonical correlation (for  $\lambda=49$ ) are: mean squared  $r(\zeta_1, x_1)=0.33317$ ; mean squared  $r(\zeta_2, x_2)=0.38270$  and the representation-constrained canonical correlation,  $r(\zeta_1, \zeta_2) = 0.48761$ . These findings are corroborative to our earlier results with regard to the dataset in Table-1.1.

We conduct yet another experiment with the dataset presented in Table-3.1. We find that  $\xi_1$  for  $X_1$  has the representation power 0.661265 (eigenvalue=2.645058) while  $\xi_2$  for  $X_2$  has the representation power 0.752979 (eigenvalue=3.764895). The  $r(\xi_1, \xi_2)$  is 0.922764. Against these, results of the constrained canonical correlation (for  $\lambda=49$ ) are: mean squared  $r(\zeta_1, x_1)=0.661261$ ; mean squared  $r(\zeta_2, x_2)=0.752966$  and the constrained canonical correlation,  $r(\zeta_1, \zeta_2) = 0.923647$ . These results are once again corroborative to our earlier findings.

**VI. A Computer Program for RCCCA:** We append here the computer program (in FORTRAN) that we have developed and used for solving the problems in this paper. Its main program (RCCCA) is assisted by 13 subroutines. The user needs setting the parameters in the main program as well

as in the subroutines CORD and DORANK. Parameter setting in RPS may seldom be required. This program can be used for obtaining Ordinal Canonical Correlation (Mishra, 2009) also. Different schemes of rank-ordering may be used (Wikipedia, 2008).

**VII. Concluding Remarks:** Our proposed Representation-Constrained Canonical correlation (RCCCA) Analysis has the classical canonical correlation analysis (CCCA) at its one end ( $\lambda=0$ ) and the Classical Principal Component Analysis (CPCA) at the other (as  $\lambda$  tends to be very large). In between it gives us a compromise solution. By a proper choice of  $\lambda$ , one can avoid hijacking of the representation issue of two datasets by a lone couple of highly correlated variables across those datasets. This advantage of the RCCCA over the CCCA deserves a serious attention by the researchers using statistical tools for data analysis.

## References

Eberhart R.C. and Kennedy J. (1995): "A New Optimizer using Particle Swarm Theory", *Proceedings Sixth Symposium on Micro Machine and Human Science*: 39–43. IEEE Service Center, Piscataway, NJ.

Fleischer, M. (2005): "Foundations of Swarm Intelligence: From Principles to Practice", *Swarming Network Enabled C4ISR*, arXiv:nlin.AO/0502003 v1.

Hayek, F. A. (1948) *Individualism and Economic Order*, The University of Chicago Press, Chicago.

Hayek, F. A. (1952) *The Sensory Order: An Inquiry into the Foundations of Theoretical Psychology*, University of Chicago Press, Chicago.

Hotelling, H. (1936) "Relations between Two Sets of Variates", *Biometrika*, 28: 321-377.

Mishra, S.K. (2006) "Global Optimization by Differential Evolution and Particle Swarm Methods: Evaluation on Some Benchmark Functions", available at SSRN: <http://ssrn.com/abstract=933827>

Mishra, S.K. (2009) "A Note on the Ordinal Canonical Correlation Analysis of Two Sets of Ranking Scores". Available at SSRN: <http://ssrn.com/abstract=1328319>

Urfalioglu, O. (2004) "Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates", *Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, IEEE Computer Society Washington, DC, USA: 464- 471.

Wikipedia (2008) "Ranking", available at *Wikipedia*: [http://en.wikipedia.org/wiki/Rank\\_order](http://en.wikipedia.org/wiki/Rank_order)

Sl No.	$\lambda$	Canonical $ r(\zeta_1, \zeta_2) $	Mean Absolute		Canonical $ r(\zeta_1, \zeta_2) $	Mean Squared		Canonical $ r(\zeta_1, \zeta_2) $	Minimum Absolute	
			$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$		$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$		$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$
1	0.0	1.0000	0.3342	0.2814	1.0000	0.2668	0.2121	1.0000	0.0234	0.0246
2	0.5	0.9831	0.3942	0.3254	0.9990	0.2717	0.2152	0.9755	0.1615	0.1361
3	1.0	0.9440	0.4434	0.3785	0.9961	0.2763	0.2183	0.8223	0.3921	0.2671
4	1.5	0.8942	0.4772	0.4188	0.9916	0.2805	0.2214	0.5072	0.5302	0.4618
5	2.0	0.8432	0.4992	0.4479	0.9855	0.2843	0.2244	0.4662	0.5319	0.4853
6	2.5	0.7975	0.5128	0.4679	0.9780	0.2878	0.2275	0.4556	0.5337	0.4889
7	3.0	0.7597	0.5210	0.4813	0.9691	0.2909	0.2306	0.4473	0.5338	0.4917
8	3.5	0.7298	0.5259	0.4902	0.9590	0.2938	0.2338	0.4296	0.5337	0.4968
9	4.0	0.7060	0.5290	0.4962	0.9477	0.2964	0.2369	0.4349	0.5334	0.4954
10	4.5	0.6870	0.5310	0.5005	0.9352	0.2987	0.2401	0.4230	0.5335	0.4978
11	5.0	0.6715	0.5323	0.5036	0.9217	0.3008	0.2433	0.4342	0.5337	0.4955
12	5.5	0.6590	0.5333	0.5058	0.9073	0.3027	0.2464	0.4359	0.5338	0.4950
13	6.0	0.6483	0.5339	0.5076	0.8921	0.3044	0.2495	0.4404	0.5338	0.4940
14	6.5	0.6394	0.5345	0.5089	0.8762	0.3059	0.2525	0.3743	0.5389	0.4963
15	7.0	0.6318	0.5348	0.5100	0.8599	0.3072	0.2554	0.4170	0.5337	0.4994
16	7.5	0.6251	0.5351	0.5108	0.8434	0.3083	0.2581	0.4175	0.5338	0.4992
17	8.0	0.6193	0.5354	0.5115	0.8270	0.3094	0.2607	0.4278	0.5338	0.4970
18	8.5	0.6142	0.5356	0.5121	0.8106	0.3102	0.2630	0.4167	0.5335	0.4990
19	9.0	0.6098	0.5357	0.5126	0.7945	0.3110	0.2652	0.4293	0.5337	0.4967
20	9.5	0.6056	0.5358	0.5130	0.7789	0.3117	0.2672	0.4206	0.5339	0.4986
21	10.0	0.6019	0.5360	0.5133	0.7641	0.3122	0.2690	0.3746	0.5389	0.4962
22	10.5	0.5988	0.5360	0.5136	0.7495	0.3127	0.2706	0.2904	0.5023	0.4748
23	11.0	0.5958	0.5361	0.5139	0.7359	0.3132	0.2721	0.4167	0.5338	0.4990
24	11.5	0.5931	0.5362	0.5141	0.7227	0.3136	0.2734	0.4201	0.4789	0.4281
25	12.0	0.5906	0.5362	0.5143	0.7103	0.3139	0.2746	0.4206	0.5338	0.4987
26	12.5	0.5884	0.5363	0.5144	0.6985	0.3142	0.2756	0.5150	0.4781	0.3664
27	13.0	0.5861	0.5363	0.5146	0.6872	0.3145	0.2766	0.4167	0.5337	0.4993
28	13.5	0.5842	0.5364	0.5147	0.6764	0.3148	0.2774	0.3745	0.5389	0.4964
29	14.0	0.5826	0.5364	0.5148	0.6665	0.3150	0.2782	0.3742	0.5390	0.4963
30	14.5	0.5807	0.5364	0.5150	0.6570	0.3152	0.2789	0.4022	0.4648	0.4532
31	15.0	0.5791	0.5365	0.5151	0.6478	0.3154	0.2795	0.4170	0.5338	0.4991
32	15.5	0.5778	0.5365	0.5151	0.6390	0.3155	0.2801	0.4179	0.5003	0.4860
33	16.0	0.5765	0.5365	0.5152	0.6310	0.3157	0.2806	0.2791	0.5387	0.4990
34	16.5	0.5751	0.5365	0.5153	0.6231	0.3158	0.2810	0.3992	0.4764	0.4347
35	17.0	0.5739	0.5365	0.5154	0.6158	0.3159	0.2815	0.3742	0.5388	0.4964
36	17.5	0.5728	0.5366	0.5154	0.6088	0.3160	0.2819	0.0285	0.4457	0.4501
37	18.0	0.5715	0.5366	0.5155	0.6021	0.3161	0.2822	0.2794	0.5389	0.4992
38	18.5	0.5706	0.5366	0.5155	0.5960	0.3162	0.2825	0.3811	0.4744	0.4599
39	19.0	0.5697	0.5366	0.5156	0.5898	0.3163	0.2828	0.3741	0.5389	0.4963
40	19.5	0.5688	0.5366	0.5156	0.5840	0.3164	0.2831	0.3743	0.5389	0.4962
41	20.0	0.5680	0.5366	0.5157	0.5783	0.3165	0.2834	0.3345	0.4838	0.3983
42	20.5	0.5671	0.5366	0.5157	0.5732	0.3166	0.2836	0.2795	0.5389	0.4994
43	21.0	0.5663	0.5366	0.5157	0.5682	0.3166	0.2838	0.4194	0.4718	0.4439
44	21.5	0.5655	0.5366	0.5158	0.5632	0.3167	0.2840	0.3746	0.5389	0.4963
45	22.0	0.5650	0.5367	0.5158	0.5587	0.3167	0.2842	0.5496	0.5103	0.3823
46	22.5	0.5643	0.5367	0.5158	0.5542	0.3168	0.2843	0.2539	0.5138	0.4743
47	23.0	0.5635	0.5367	0.5158	0.5499	0.3168	0.2845	0.2795	0.5390	0.4993
48	23.5	0.5630	0.5367	0.5159	0.5459	0.3169	0.2846	0.2865	0.4643	0.4394
49	24.0	0.5623	0.5367	0.5159	0.5419	0.3169	0.2848	0.3688	0.5389	0.4944
50	24.5	0.5618	0.5367	0.5159	0.5383	0.3170	0.2849	0.2490	0.5347	0.4720
51	25.0	0.5612	0.5367	0.5159	0.5347	0.3170	0.2850	0.2792	0.5387	0.4994

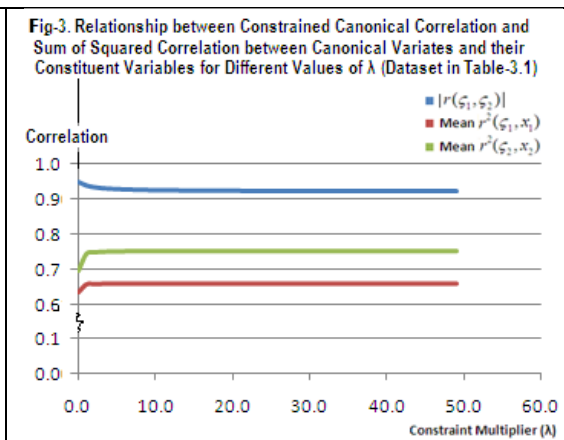
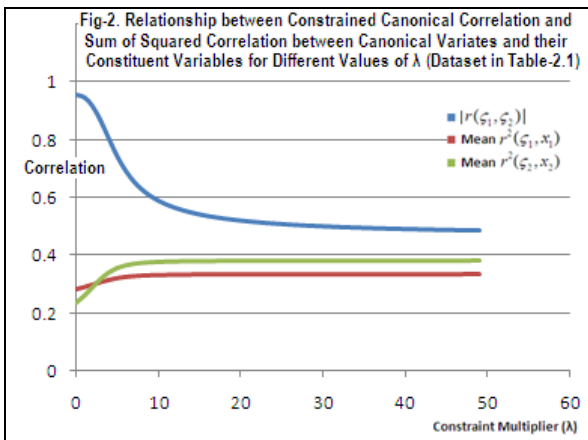
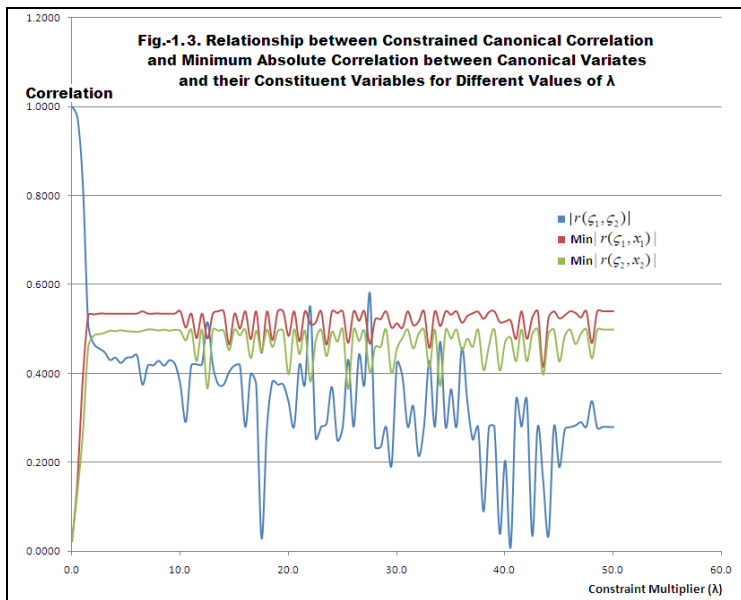
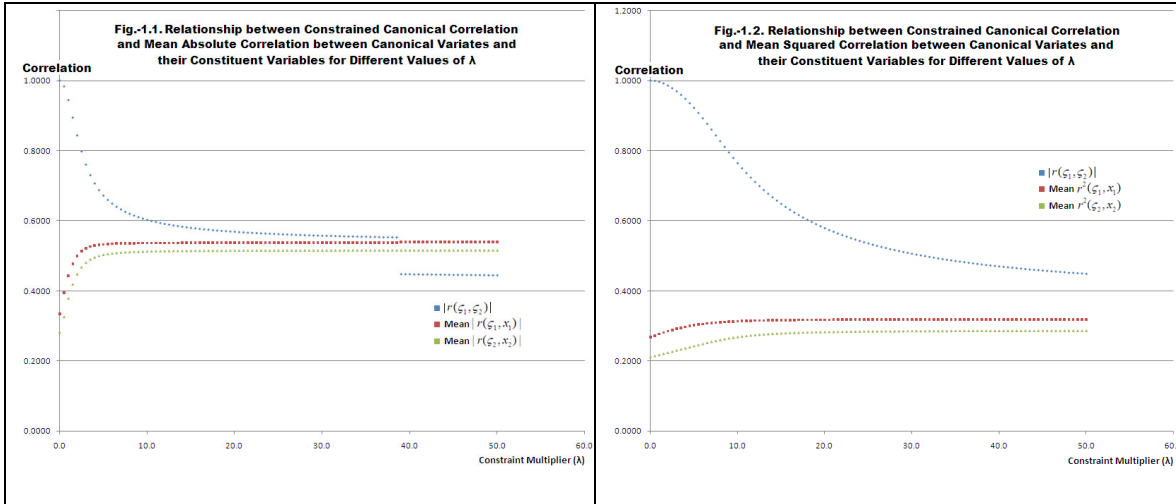
52	25.5	0.5607	0.5367	0.5159	0.5312	0.3170	0.2851	0.4305	0.4684	0.3653
53	26.0	0.5603	0.5367	0.5160	0.5280	0.3171	0.2852	0.2793	0.5387	0.4993
54	26.5	0.5597	0.5367	0.5160	0.5249	0.3171	0.2853	0.4418	0.5176	0.4731
55	27.0	0.5592	0.5367	0.5160	0.5219	0.3171	0.2854	0.3741	0.5388	0.4963
56	27.5	0.5589	0.5367	0.5160	0.5186	0.3171	0.2855	0.5795	0.4661	0.4031
57	28.0	0.5584	0.5367	0.5160	0.5160	0.3172	0.2856	0.2335	0.5213	0.4604
58	28.5	0.5581	0.5367	0.5160	0.5131	0.3172	0.2857	0.2335	0.5213	0.4604
59	29.0	0.5575	0.5367	0.5161	0.5103	0.3172	0.2858	0.2790	0.5388	0.4993
60	29.5	0.5572	0.5367	0.5161	0.5080	0.3172	0.2858	0.1922	0.5023	0.4015
61	30.0	0.5568	0.5367	0.5161	0.5054	0.3173	0.2859	0.4223	0.5119	0.4564
62	30.5	0.5564	0.5367	0.5161	0.5030	0.3173	0.2859	0.3929	0.5016	0.4801
63	31.0	0.5561	0.5367	0.5161	0.5008	0.3173	0.2860	0.2795	0.5390	0.4993
64	31.5	0.5558	0.5367	0.5161	0.4987	0.3173	0.2861	0.3260	0.5081	0.4567
65	32.0	0.5555	0.5367	0.5161	0.4964	0.3173	0.2861	0.2140	0.5156	0.4897
66	32.5	0.5549	0.5367	0.5161	0.4942	0.3173	0.2862	0.2793	0.5389	0.4992
67	33.0	0.5547	0.5367	0.5161	0.4921	0.3174	0.2862	0.4277	0.4566	0.4137
68	33.5	0.5545	0.5367	0.5161	0.4902	0.3174	0.2863	0.2794	0.5389	0.4993
69	34.0	0.5542	0.5367	0.5161	0.4883	0.3174	0.2863	0.4708	0.5056	0.3723
70	34.5	0.5539	0.5367	0.5162	0.4865	0.3174	0.2863	0.2787	0.5388	0.4988
71	35.0	0.5539	0.5367	0.5162	0.4846	0.3174	0.2864	0.3639	0.5312	0.4787
72	35.5	0.5534	0.5367	0.5162	0.4830	0.3174	0.2864	0.2793	0.5389	0.4992
73	36.0	0.5532	0.5367	0.5162	0.4814	0.3174	0.2864	0.4560	0.5133	0.4533
74	36.5	0.5528	0.5367	0.5162	0.4796	0.3174	0.2865	0.3375	0.5282	0.4788
75	37.0	0.5524	0.5368	0.5162	0.4780	0.3174	0.2865	0.2504	0.5345	0.4600
76	37.5	0.5524	0.5368	0.5162	0.4765	0.3175	0.2865	0.2784	0.5380	0.4988
77	38.0	0.5521	0.5368	0.5162	0.4749	0.3175	0.2866	0.0886	0.5222	0.4078
78	38.5	0.5520	0.5368	0.5162	0.4733	0.3175	0.2866	0.2791	0.5372	0.4631
79	39.0	0.4469	0.5394	0.5163	0.4721	0.3175	0.2866	0.2795	0.5389	0.4992
80	39.5	0.4468	0.5394	0.5163	0.4707	0.3175	0.2866	0.0385	0.5148	0.4071
81	40.0	0.4467	0.5394	0.5163	0.4693	0.3175	0.2867	0.2028	0.5160	0.4721
82	40.5	0.4463	0.5394	0.5163	0.4681	0.3175	0.2867	0.0080	0.5182	0.4812
83	41.0	0.4463	0.5394	0.5163	0.4666	0.3175	0.2867	0.3389	0.4771	0.4282
84	41.5	0.4461	0.5394	0.5163	0.4653	0.3175	0.2867	0.2795	0.5389	0.4994
85	42.0	0.4460	0.5394	0.5163	0.4644	0.3175	0.2868	0.3389	0.4771	0.4282
86	42.5	0.4458	0.5394	0.5163	0.4631	0.3175	0.2868	0.0338	0.5248	0.4897
87	43.0	0.4456	0.5394	0.5163	0.4617	0.3175	0.2868	0.2793	0.5389	0.4993
88	43.5	0.4454	0.5394	0.5163	0.4606	0.3175	0.2868	0.1597	0.4139	0.3977
89	44.0	0.4453	0.5394	0.5163	0.4593	0.3176	0.2868	0.0338	0.5248	0.4897
90	44.5	0.4452	0.5394	0.5163	0.4586	0.3176	0.2869	0.2794	0.5389	0.4994
91	45.0	0.4451	0.5394	0.5163	0.4576	0.3176	0.2869	0.1880	0.5229	0.4274
92	45.5	0.4450	0.5394	0.5163	0.4564	0.3176	0.2869	0.2733	0.5300	0.4848
93	46.0	0.4448	0.5394	0.5163	0.4555	0.3176	0.2869	0.2786	0.5389	0.4991
94	46.5	0.4447	0.5394	0.5163	0.4547	0.3176	0.2869	0.2822	0.5354	0.4665
95	47.0	0.4445	0.5394	0.5163	0.4535	0.3176	0.2869	0.2898	0.5252	0.4905
96	47.5	0.4444	0.5394	0.5163	0.4527	0.3176	0.2869	0.2796	0.5389	0.4993
97	48.0	0.4444	0.5394	0.5163	0.4510	0.3176	0.2870	0.3372	0.4676	0.4344
98	48.5	0.4442	0.5394	0.5163	0.4509	0.3176	0.2870	0.2768	0.5389	0.4985
99	49.0	0.4440	0.5394	0.5163	0.4500	0.3176	0.2870	0.2792	0.5388	0.4993
100	49.5	0.4439	0.5394	0.5163	0.4491	0.3176	0.2870	0.2790	0.5389	0.4993
101	50.0	0.4438	0.5394	0.5163	0.4480	0.3176	0.2870	0.2784	0.5390	0.4989

Sl No.	X <sub>1</sub> or Dataset-1					X <sub>2</sub> or Dataset-2					Sl No.	X <sub>1</sub> or Dataset-1					X <sub>2</sub> or Dataset-2				
	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>21</sub>	X <sub>22</sub>	X <sub>23</sub>	X <sub>24</sub>	X <sub>25</sub>	X <sub>11</sub>		X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>21</sub>	X <sub>22</sub>	X <sub>23</sub>	X <sub>24</sub>	X <sub>25</sub>		
1	2.7	1.9	2.4	1.2	2.6	2.3	1.5	0.1	6.6	16	1.4	1.4	0.2	0.4	1.1	2.2	2.2	2.6	2.4		
2	1.7	0.1	0.8	1.8	0.4	0.2	2.3	0.2	0.1	17	1.5	0.4	2.2	1.9	1.9	0.6	2.1	1.9	5.5		
3	0.2	2.8	2.6	0.9	1.3	2.0	2.0	0.3	7.3	18	0.6	1.3	2.5	2.8	2.8	2.5	2.7	2.2	4.9		
4	0.4	0.3	1.1	0.2	1.5	1.3	1.1	1.8	3.3	19	2.5	1.1	0.1	1.1	2.5	1.0	1.0	2.9	3.8		
5	0.9	1.8	1.6	1.4	0.8	1.2	2.4	2.4	5.7	20	1.0	2.3	1.8	1.5	2.9	1.8	1.6	2.0	5.8		
6	0.5	0.9	2.7	0.7	1.4	1.6	1.2	3.0	6.2	21	0.8	1.7	1.0	1.6	1.6	2.4	0.6	1.4	4.5		
7	2.0	1.0	2.9	1.7	0.3	0.1	0.4	1.1	5.4	22	0.3	1.2	2.1	0.3	2.0	1.9	0.7	0.9	4.5		
8	0.1	1.6	0.5	2.7	0.7	2.1	1.3	1.7	3.1	23	1.3	0.7	1.3	2.4	2.2	0.7	0.8	1.0	3.4		
9	1.2	0.6	2.8	1.0	0.1	0.9	0.1	0.8	3.7	24	2.6	1.5	2.3	0.6	1.7	2.9	2.9	2.5	7.3		
10	2.9	2.1	0.4	0.8	0.5	0.3	1.7	0.4	4.7	25	3.0	2.6	1.2	3.0	2.7	2.6	2.8	1.5	7.2		
11	0.7	0.5	0.6	1.3	2.1	0.5	0.3	0.7	0.7	26	1.1	2.2	0.7	2.5	2.4	0.8	2.6	1.2	3.8		
12	2.8	2.5	1.5	2.9	2.3	2.8	3.0	1.6	6.5	27	1.8	2.0	1.9	2.2	1.8	1.7	1.8	0.6	6.1		
13	2.2	0.2	1.7	2.3	3.0	1.1	0.5	2.7	3.9	28	1.9	2.7	3.0	2.0	1.0	1.4	1.4	0.5	9.5		
14	2.1	0.8	0.9	2.6	0.9	2.7	2.5	2.1	3.6	29	1.6	2.4	0.3	0.5	0.2	0.4	0.2	2.3	6.5		
15	2.3	3.0	1.4	0.1	0.6	3.0	0.9	2.8	8.4	30	2.4	2.9	2.0	2.1	1.2	1.5	1.9	1.3	7.6		

Sl No.	$\lambda$	Canonical $ r(\zeta_1, \zeta_2) $	Mean Squared		Sl No.	$\lambda$	Canonical $ r(\zeta_1, \zeta_2) $	Mean Squared	
			$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$				$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$
1	0.0	0.95772	0.28514	0.23519	26	25.0	0.50983	0.33290	0.38230
2	1.0	0.94904	0.29212	0.26011	27	26.0	0.50804	0.33293	0.38234
3	2.0	0.91881	0.29987	0.28983	28	27.0	0.50638	0.33296	0.38238
4	3.0	0.86701	0.30782	0.31901	29	28.0	0.50475	0.33298	0.38241
5	4.0	0.80425	0.31506	0.34197	30	29.0	0.50340	0.33300	0.38244
6	5.0	0.74448	0.32066	0.35709	31	30.0	0.50203	0.33302	0.38247
7	6.0	0.69541	0.32452	0.36618	32	31.0	0.50086	0.33304	0.38249
8	7.0	0.65777	0.32703	0.37155	33	32.0	0.49966	0.33305	0.38252
9	8.0	0.62930	0.32867	0.37482	34	33.0	0.49858	0.33306	0.38254
10	9.0	0.60764	0.32976	0.37690	35	34.0	0.49755	0.33308	0.38256
11	10.0	0.59071	0.33052	0.37828	36	35.0	0.49659	0.33309	0.38257
12	11.0	0.57730	0.33106	0.37924	37	36.0	0.49571	0.33310	0.38259
13	12.0	0.56634	0.33146	0.37993	38	37.0	0.49492	0.33311	0.38260
14	13.0	0.55733	0.33176	0.38044	39	38.0	0.49409	0.33311	0.38261
15	14.0	0.54983	0.33199	0.38083	40	39.0	0.49333	0.33312	0.38262
16	15.0	0.54338	0.33217	0.38113	41	40.0	0.49265	0.33313	0.38263
17	16.0	0.53786	0.33232	0.38137	42	41.0	0.49193	0.33314	0.38264
18	17.0	0.53310	0.33244	0.38156	43	42.0	0.49132	0.33314	0.38265
19	18.0	0.52896	0.33253	0.38171	44	43.0	0.49074	0.33315	0.38266
20	19.0	0.52523	0.33262	0.38185	45	44.0	0.49018	0.33315	0.38267
21	20.0	0.52196	0.33268	0.38195	46	45.0	0.48958	0.33316	0.38268
22	21.0	0.51904	0.33274	0.38205	47	46.0	0.48912	0.33316	0.38268
23	22.0	0.51638	0.33279	0.38212	48	47.0	0.48852	0.33317	0.38269
24	23.0	0.51395	0.33283	0.38219	49	48.0	0.48812	0.33317	0.38270
25	24.0	0.51184	0.33287	0.38225	50	49.0	0.48761	0.33317	0.38270

Sl No.	X <sub>1</sub> or Dataset-1				X <sub>2</sub> or Dataset-2					Sl No.	X <sub>1</sub> or Dataset-1				X <sub>2</sub> or Dataset-2				
	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>21</sub>	X <sub>22</sub>	X <sub>23</sub>	X <sub>24</sub>	X <sub>25</sub>		X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>21</sub>	X <sub>22</sub>	X <sub>23</sub>	X <sub>24</sub>	X <sub>25</sub>
1	0.7	1.1	1.6	1.3	1.1	0.1	2.7	1.5	1.8	16	2.1	2.7	2.7	1.8	1.8	0.9	6.4	1.6	2.3
2	1.3	1.2	0.8	1.0	0.3	1.2	1.1	0.4	0.3	17	2.2	2.9	2.6	2.3	3.0	3.0	6.9	1.9	2.8
3	1.7	2.5	1.8	1.1	1.9	2.3	5.5	2.6	2.1	18	2.6	1.9	2.8	2.2	2.5	2.8	5.0	2.4	3.0
4	2.9	2.4	1.9	2.8	2.7	2.2	4.9	2.3	1.9	19	1.6	1.3	2.4	3.0	1.7	2.1	4.3	2.0	2.9
5	0.4	1.0	0.2	0.9	0.5	2.5	1.7	1.0	0.1	20	1.9	0.9	2.9	1.9	1.5	2.0	3.6	2.1	1.4
6	0.6	0.4	2.2	0.4	1.0	0.8	2.4	1.7	0.4	21	1.5	0.2	0.4	0.7	1.3	1.6	0.8	0.2	0.9
7	0.8	0.1	0.7	0.8	0.1	0.2	0.4	0.1	0.5	22	1.8	0.5	1.1	0.5	1.2	1.4	2.1	0.8	1.0
8	2.3	2.8	3.0	2.6	2.6	2.9	6.7	2.5	2.7	23	1.4	0.7	0.5	1.6	0.4	1.9	2.6	2.2	1.5
9	1.2	2.0	0.9	1.7	2.4	0.7	3.2	1.8	2.0	24	1.0	1.6	0.3	0.1	0.7	1.1	1.2	0.3	0.7
10	2.4	2.1	2.5	2.5	2.1	1.5	3.9	2.7	1.7	25	0.5	1.8	1.4	2.7	0.2	1.8	3.3	1.3	1.3
11	0.2	0.6	0.1	1.5	1.4	0.4	0.6	0.5	0.2	26	3.0	2.6	2.3	2.4	2.0	1.7	5.1	3.0	2.2
12	2.0	1.5	0.6	0.3	0.8	0.6	1.5	0.6	1.6	27	2.5	1.4	1.3	2.1	2.3	2.6	3.8	2.9	2.4
13	0.9	0.3	1.7	2.0	1.6	0.5	2.5	0.9	0.8	28	0.3	2.2	1.2	0.2	0.9	1.3	2.3	1.1	0.6
14	2.7	3.0	2.1	2.9	2.8	2.7	7.0	2.8	2.6	29	2.8	2.3	2.0	1.4	2.9	2.4	5.8	1.4	2.5
15	1.1	0.8	1.0	1.2	0.6	0.3	2.0	1.2	1.2	30	0.1	1.7	1.5	0.6	2.2	1.0	4.2	0.7	1.1

Sl No.	$\lambda$	Canonical $ r(\zeta_1, \zeta_2) $	Mean Squared		Sl No.	$\lambda$	Canonical $ r(\zeta_1, \zeta_2) $	Mean Squared	
			$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$				$r(\zeta_1, x_1)$	$r(\zeta_2, x_2)$
1	0.0	0.94813	0.63711	0.69854	26	25.0	0.92443	0.66125	0.75293
2	1.0	0.93901	0.65895	0.74485	27	26.0	0.92437	0.66125	0.75294
3	2.0	0.93449	0.66036	0.74955	28	27.0	0.92431	0.66125	0.75294
4	3.0	0.93199	0.66076	0.75106	29	28.0	0.92426	0.66125	0.75294
5	4.0	0.93039	0.66094	0.75175	30	29.0	0.92421	0.66125	0.75294
6	5.0	0.92927	0.66104	0.75212	31	30.0	0.92417	0.66126	0.75295
7	6.0	0.92844	0.66110	0.75235	32	31.0	0.92412	0.66126	0.75295
8	7.0	0.92780	0.66114	0.75250	33	32.0	0.92409	0.66126	0.75295
9	8.0	0.92729	0.66116	0.75260	34	33.0	0.92405	0.66126	0.75295
10	9.0	0.92687	0.66118	0.75267	35	34.0	0.92401	0.66126	0.75295
11	10.0	0.92653	0.66119	0.75272	36	35.0	0.92398	0.66126	0.75295
12	11.0	0.92624	0.66121	0.75276	37	36.0	0.92394	0.66126	0.75296
13	12.0	0.92598	0.66121	0.75279	38	37.0	0.92392	0.66126	0.75296
14	13.0	0.92577	0.66122	0.75282	39	38.0	0.92389	0.66126	0.75296
15	14.0	0.92558	0.66123	0.75284	40	39.0	0.92386	0.66126	0.75296
16	15.0	0.92541	0.66123	0.75286	41	40.0	0.92383	0.66126	0.75296
17	16.0	0.92526	0.66123	0.75287	42	41.0	0.92381	0.66126	0.75296
18	17.0	0.92513	0.66124	0.75288	43	42.0	0.92378	0.66126	0.75296
19	18.0	0.92501	0.66124	0.75289	44	43.0	0.92376	0.66126	0.75296
20	19.0	0.92490	0.66124	0.75290	45	44.0	0.92374	0.66126	0.75296
21	20.0	0.92481	0.66124	0.75291	46	45.0	0.92372	0.66126	0.75296
22	21.0	0.92472	0.66125	0.75291	47	46.0	0.92370	0.66126	0.75296
23	22.0	0.92464	0.66125	0.75292	48	47.0	0.92368	0.66126	0.75297
24	23.0	0.92455	0.66125	0.75292	49	48.0	0.92366	0.66126	0.75297
25	24.0	0.92450	0.66125	0.75293	50	49.0	0.92365	0.66126	0.75297



```

1: C      !----- MAIN PROGRAM : RCCCA -----
2: C      PROVIDES TO USE REPULSIVE PARTICLE SWARM METHOD TO OBTAIN
3: C      THE REPRESENTATION-CONSTRAINED CANONICAL CORRELATION & VARIATES
4: C      PRODUCT MOMENT AS WELL AS ABSOLUTE CORRELATION (BRADLEY, 1985) MAY
5: C      BE USED.
6: C      PROGRAM BY SK MISHRA, DEPT. OF ECONOMICS,
7: C      NORTH-EASTERN HILL UNIVERSITY, SHILLONG (INDIA)
8: C      -----
9: C      ADJUST THE PARAMETERS SUITABLY
10: C     IN THIS MAIN PROGRAM AND IN THE SOBROUTINE CORD
11: C     WHEN THE PROGRAM ASKS FOR ANY OTHER PARAMETERS, FEED THEM SUITABLY
12: C     -----
13: C     PROGRAM RCCCA
14: C     PARAMETER (NOB=30,MVAR=9)!CHANGE THE PARAMETERS HERE AS NEEDED.
15: C     -----
16: C     NOB=NO. OF CASES AND MVAR=NO. OF VARIABLES IN ALL M= (M1+M2)
17: C     NOB AND MVAR TO BE ADJUSTED IN SUBROUTINE CORD(M,X,F) ALSO.
18: C     SET NRL TO DESIRED VALUE IN SUBROUTINE DORANK FOR RANKING SCHEME
19: C     -----
20: C     IMPLICIT DOUBLE PRECISION (A-H, O-Z)
21: C     COMMON /KFF/KF,NFCALL,FTIT ! FUNCTION CODE, NO. OF CALLS & TITLE
22: C     CHARACTER *30 METHOD(1)
23: C     CHARACTER *70 FTIT
24: C     CHARACTER *40 INFILE,OUTFILE
25: C     COMMON /CANON/MONE,MTWO
26: C     COMMON /CORDAT/CDAT (NOB,MVAR),QIND1 (NOB),QIND2 (NOB),R(1),NORM,NCOR
27: C     COMMON /XBASE/XBAS
28: C     COMMON /RNDM/IU,IV ! RANDOM NUMBER GENERATION (IU = 4-DIGIT SEED)
29: C     COMMON /GETRANK/MRNK
30: C     COMMON /MRCCA/OWNR(2,MVAR),FROH,SOWNR1,SOWNR2,MRCC
31: C     INTEGER IU,IV
32: C     DIMENSION XX(3,50),KKF(3),MM(3),FMINN(3),XBAS(1000,50)
33: C     DIMENSION ZDAT (NOB,MVAR+1),FRANK1 (NOB),FRANK2 (NOB),RMT(2,2)
34: C     DIMENSION X(50)! X IS THE DECISION VARIABLE X IN F(X) TO MINIMIZE
35: C     M = DIMENSION OF THE PROBLEM, KF(=1) = TEST FUNCTION CODE AND
36: C     FMIN IS THE MIN VALUE OF F(X) OBTAINED FROM RPS
37: C     WRITE (*,*) '===== WARNING ===== '
38: C     WRITE (*,*) 'ADJUST PARAMETERS IN SUBROUTINES RPS IF NEEDED '
39: C     ----- OPTIMIZATION BY RPS METHOD -----
40: C     NORM=2!WORKS WITH THE EUCLIDEAN NORM (IDENTICAL RESULTS IF NORM=1)
41: C     NOPT=1 ! ONLY ONE FUNCTION IS OPTIMIZED
42: C     WRITE (*,*) '===== '
43: C     METHOD(1)=' : REPULSIVE PARTICLE SWARM OPTIMIZATION'
44: C     INITIALIZE. THIS XBAS WILL BE USED TO INITIALIZE THE POPULATION.
45: C     WRITE (*,*) ' '
46: C     WRITE (*,*) '----- FEED RANDOM NUMBER SEED, AND NCOR -----'
47: C     WRITE (*,*) ' '
48: C     WRITE (*,*) 'FEED SEED [ANY 4-DIGIT NUMBER] AND NCOR[0,1]'
49: C     WRITE (*,*) 'NCOR(0)=PRODUCT MOMENT; NCOR(1)=ABSOLUTE CORRELATION'
50: C     WRITE (*,*) ' '
51: C     1 READ (*,*) IU,NCOR
52: C     IF (NCOR.LT.0.OR.NCOR.GT.1) THEN
53: C     WRITE (*,*) 'SORRY. NCOR TAKES ON[0,1] ONLY. FEED SEED & NCOR AGAIN'
54: C     GOTO 1
55: C     ENDIF
56: C     WRITE (*,*) 'WANT RANK SCORE OPTIMIZATION? YES(1); NO(OTHER THAN 1)'
57: C     READ (*,*) MRNK
58: C     WRITE (*,*) 'INPUT FILE TO READ DATA:YOUR DATA MUST BE IN THIS FILE'
59: C     WRITE (*,*) 'CASES (NOB) IN ROWS ; VARIABLES (MVAR) IN COLUMNS'
60: C     READ (*,*) INFILE
61: C     WRITE (*,*) 'SPECIFY THE OUTPUT FILE TO STORE THE RESULTS'
62: C     READ (*,*) OUTFILE
63: C     OPEN (9, FILE=OUTFILE)
64: C     OPEN (7, FILE=INFILE)
65: C     DO I=1,NOB
66: C     READ (7,*) ,CDA,(CDAT(I,J),J=1,MVAR)
67: C     ENDDO

```

```

68:      CLOSE (7)
69:      DO I=1,NOB
70:      DO J=1,MVAR
71:      ZDAT(I,J+1)=CDAT(I,J)
72:      ENDDO
73:      ENDDO
74:      WRITE(*,*) 'DATA HAS BEEN READ. WOULD YOU UNITIZE VARIABLES? [YES=1
75: & ELSE NO UNITIZATION]'
76:      WRITE(*,*) 'UNITIZE MEANS TRANSFORMATION FROM X(I,J) TO UNITIZED X'
77:      WRITE(*,*) '[X(I,J)-MIN(X(.,J))]/[MAX(X(.,J))-MIN(X(.,J))]'
78:      READ(*,*) NUN
79:      IF(NUN.EQ.1) THEN
80:      DO J=1,MVAR
81:      CMIN=CDAT(1,J)
82:      CMAX=CDAT(1,J)
83:      DO I=2,NOB
84:      IF(CMIN.GT.CDAT(I,J)) CMIN=CDAT(I,J)
85:      IF(CMAX.LT.CDAT(I,J)) CMAX=CDAT(I,J)
86:      ENDDO
87:      DO I=1,NOB
88:      CDAT(I,J)=(CDAT(I,J)-CMIN)/(CMAX-CMIN)
89:      ENDDO
90:      ENDDO
91:      ENDIF
92: C -----
93: C THIS XBAS WILL BE USED AS INITIAL X
94:      DO I=1,1000
95:      DO J=1,50
96:      CALL RANDOM(RAND)
97:      XBAS(I,J)=RAND ! RANDOM NUMBER BETWEEN (0, 1)
98:      ENDDO
99:      ENDDO
100: C -----
101:      WRITE(*,*) ' *****'
102: C -----
103:      K=1
104:      WRITE(*,*) 'PARTICLE SWARM PROGRAM TO OBTAIN CANONICAL CORRELATION'
105:      CALL RPS(M,X,FMINRPS,Q1) !CALLS RPS AND RETURNS OPTIMAL X AND FMIN
106:      WRITE(*,*) 'RPS BRINGS THE FOLLOWING VALUES TO THE MAIN PROGRAM'
107:      WRITE(*,*) (X(JOPT),JOPT=1,M), ' OPTIMUM FUNCTION=',FMINRPS
108:      IF(KF.EQ.1) THEN
109:      WRITE(9,*) 'REPULSIVE PARTICLE SWARM OPTIMIZATION RESULTS'
110:      WRITE(9,*) 'THE LARGEST CANONICAL R BETWEEN THE SETS OF VARIABLES'
111:      WRITE(9,*) ' ABS(R)=' ,DABS(R(1)), ' ; SQUARE(R)=' ,R(1)**2
112:      IF(NCOR.EQ.0) THEN
113:      WRITE(*,*) 'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
114:      WRITE(9,*) 'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
115:      ELSE
116:      WRITE(*,*) 'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
117:      WRITE(9,*) 'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
118:      ENDIF
119:      WRITE(*,*) ' _____'
120:      WRITE(9,*) ' _____'
121:
122:      DO II=1,NOB
123:      FRANK1(II)=QIND1(II)
124:      FRANK2(II)=QIND2(II)
125:      ENDDO
126:      ENDIF
127:      FMIN=FMINRPS
128: C -----
129:      DO J=1,M
130:      XX(K,J)=X(J)
131:      ENDDO
132:      KKF(K)=KF
133:      MM(K)=M
134:      FMINN(K)=FMIN

```

```

135:      WRITE (*,*) ' '
136:      WRITE (*,*) ' '
137:      WRITE (*,*) '----- FINAL RESULTS-----'
138:      WRITE (*,*) 'FUNCT CODE=',KKF(K), ' FMIN=',FMINN(K), ' : DIM=',MM(K)
139:      WRITE (*,*) 'OPTIMAL DECISION VARIABLES : ',METHOD(K)
140:      WRITE (*,*) 'FOR THE FIRST SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
141:      WRITE (9,*) 'FOR THE FIRST SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
142:      WRITE (9,*) (XX(K,J),J=1,MONE)
143:      WRITE (*,*) (XX(K,J),J=1,MONE)
144:      WRITE (*,*) 'FOR THE SECOND SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
145:      WRITE (9,*) 'FOR THE SECOND SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
146:      WRITE (9,*) (XX(K,J),J=MONE+1,M)
147:      WRITE (*,*) (XX(K,J),J=MONE+1,M)
148:      WRITE (9,*) 'CANONICAL R=',FROH, ' OWN CORRELATIONS=',SOWNR1,SOWNR2
149:      WRITE (*,*) 'CANONICAL R=',FROH, ' OWN CORRELATIONS=',SOWNR1,SOWNR2
150:      WRITE (*,*) '/////////////////////////////////////'
151:      WRITE (*,*) 'OPTIMIZATION PROGRAM ENDED'
152:      WRITE (*,*) '*****'
153:      WRITE (*,*) 'MEASURE OF EQUALITY/INEQUALITY'
154:      WRITE (*,*) 'RPS: BEFORE AND AFTER OPTIMIZATION = ',Q0,Q1
155:      WRITE (*,*) ' '
156:      WRITE (*,*) 'RESULTS STORED IN FILE= ',OUTFILE
157:      WRITE (*,*) 'OPEN BY MSWORD OR EDIT OR ANY OTHER EDITOR'
158:      WRITE (*,*) ' '
159:      WRITE (*,*) 'NOTE:VECTORS OF CORRELATIONS & INDEX(BOTH TOGETHER) ARE
160:      & IDETERMINATE FOR SIGN & MAY BE MULTIPLIED BY (-1) IF NEEDED'
161:      WRITE (*,*) 'THAT IS IF R(J) IS TRANSFORMED TO -R(J) FOR ALL J THEN
162:      &THE INDEX(I) TOO IS TRANSFORMED TO -INDEX(I) FOR ALL I'
163:      WRITE (9,*) ' '
164:      WRITE (9,*) 'NOTE: VECTORS OF CORRELATIONS AND INDEX (BOTH TOGETHER)
165:      & ARE IDETERMINATE FOR SIGN AND MAY BE MULTIPLIED BY (-1) IF NEEDED'
166:      WRITE (9,*) 'THAT IS IF R(J) IS TRANSFORMED TO -R(J) FOR ALL J THEN
167:      &THE INDEX(I) TOO IS TRANSFORMED TO -INDEX(I) FOR ALL I'
168:      CALL DORANK(FRANK1,NOB)
169:      CALL DORANK(FRANK2,NOB)
170:      DO I=1,NOB
171:      ZDAT(I,1)=FRANK1(I)
172:      ZDAT(I,2)=FRANK2(I)
173:      ENDDO
174:      IF(NCOR.EQ.0) THEN
175:      CALL CORREL(ZDAT,NOB,2,RMAT)
176:      ELSE
177:      CALL DOCORA(ZDAT,NOB,2,RMAT)
178:      ENDIF
179:      WRITE (9,*) '===== '
180:      WRITE (*,*) '===== '
181:      WRITE (9,*) '1ST 2 ARE CANONICAL SCORES AND LAST 2 ARE THEIR RANK'
182:      WRITE (*,*) '1ST 2 ARE CANONICAL SCORES AND LAST 2 ARE THEIR RANK'
183:      WRITE (9,*) '===== '
184:      WRITE (*,*) '===== '
185:      DO I=1,NOB
186:      IF(MRNK.EQ.1) THEN
187:      QIND1(I)=0.DO
188:      QIND2(I)=0.DO
189:      DO J=1,MONE
190:      QIND1(I)=QIND1(I)+ZDAT(I,J+1)*XX(NOPT,J)
191:      ENDDO
192:      DO J=MONE+1,MVAR
193:      QIND2(I)=QIND2(I)+ZDAT(I,J+1)*XX(NOPT,J)
194:      ENDDO
195:      ENDIF
196:      WRITE (9,2) I,QIND1(I),QIND2(I),(ZDAT(I,J),J=1,2)
197:      WRITE (*,2) I,QIND1(I),QIND2(I),(ZDAT(I,J),J=1,2)
198:      ENDDO
199:      2 FORMAT(I5,2F15.6,2F10.3)
200:      WRITE (9,*) 'SQUARE OF CANONICAL CORRELATION =',RMAT(1,2)**2
201:      WRITE (*,*) 'SQUARE OF CANONICAL CORRELATION =',RMAT(1,2)**2

```

```

202:      WRITE (9,*) 'ABSOLUTE OF CANONICAL CORRELATION =', DABS (RMAT (1,2))
203:      WRITE (*,*) 'ABSOLUTE OF CANONICAL CORRELATION =', DABS (RMAT (1,2))
204:      IF (NCOR.EQ.0) THEN
205:      WRITE (*,*) 'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
206:      WRITE (9,*) 'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
207:      ELSE
208:      WRITE (*,*) 'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
209:      WRITE (9,*) 'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
210:      ENDIF
211:      IF (MRCC.NE.0) THEN
212:      WRITE (9,*) 'THE REPRESENTATION CORRELATIONS OF THE TWO SETS ARE:'
213:      WRITE (9,*) (OWNR (1,J), J=1, MONE)
214:      WRITE (9,*) (OWNR (2,J), J=1, MTWO)
215:      WRITE (*,*) 'THE REPRESENTATION CORRELATIONS OF THE TWO SETS ARE:'
216:      WRITE (*,*) (OWNR (1,J), J=1, MONE)
217:      WRITE (*,*) (OWNR (2,J), J=1, MTWO)
218:      WRITE (9,*) 'CANONICAL R=', FROH, ' OWN CORRELATIONS=', SOWNR1, SOWNR2
219:      WRITE (*,*) 'CANONICAL R=', FROH, ' OWN CORRELATIONS=', SOWNR1, SOWNR2
220:      ENDIF
221:      CLOSE (9)
222:      WRITE (*,*) 'THE JOB IS OVER'
223:      END
224:  C
225:      SUBROUTINE RPS (M, ABEST, FBEST, G1)
226:  C      PROGRAM TO FIND GLOBAL MINIMUM BY REPULSIVE PARTICLE SWARM METHOD
227:  C      WRITTEN BY SK MISHRA, DEPT. OF ECONOMICS, NEHU, SHILLONG (INDIA)
228:  C
229:      PARAMETER (N=100, NN=20, MX=100, NSTEP=11, ITRN=10000, NSIGMA=1, ITOP=1)
230:      PARAMETER (NPRN=50) ! DISPLAYS RESULTS AT EVERY 500 TH ITERATION
231:  C      PARAMETER (N=50, NN=25, MX=100, NSTEP=9, ITRN=10000, NSIGMA=1, ITOP=3)
232:  C      PARAMETER (N=100, NN=15, MX=100, NSTEP=9, ITRN=10000, NSIGMA=1, ITOP=3)
233:  C      IN CERTAIN CASES THE ONE OR THE OTHER SPECIFICATION WORKS BETTER
234:  C      DIFFERENT SPECIFICATIONS OF PARAMETERS MAY SUIT DIFFERENT TYPES
235:  C      OF FUNCTIONS OR DIMENSIONS - ONE HAS TO DO SOME TRIAL AND ERROR
236:  C
237:  C      N = POPULATION SIZE. IN MOST OF THE CASES N=30 IS OK. ITS VALUE
238:  C      MAY BE INCREASED TO 50 OR 100 TOO. THE PARAMETER NN IS THE SIZE OF
239:  C      RANDOMLY CHOSEN NEIGHBOURS. 15 TO 25 (BUT SUFFICIENTLY LESS THAN
240:  C      N) IS A GOOD CHOICE. MX IS THE MAXIMAL SIZE OF DECISION VARIABLES.
241:  C      IN F(X1, X2, ..., XM) M SHOULD BE LESS THAN OR EQUAL TO MX. ITRN IS
242:  C      THE NO. OF ITERATIONS. IT MAY DEPEND ON THE PROBLEM. 200 (AT LEAST)
243:  C      TO 500 ITERATIONS MAY BE GOOD ENOUGH. BUT FOR FUNCTIONS LIKE
244:  C      ROSENBROCKOR GRIEWANK OF LARGE SIZE (SAY M=30) IT IS NEEDED THAT
245:  C      ITRN IS LARGE, SAY 5000 OR EVEN 10000.
246:  C      SIGMA INTRODUCES PERTURBATION & HELPS THE SEARCH JUMP OUT OF LOCAL
247:  C      OPTIMA. FOR EXAMPLE : RASTRIGIN FUNCTION OF DMENSION 30 OR LARGER
248:  C      NSTEP DOES LOCAL SEARCH BY TUNNELLING AND WORKS WELL BETWEEN 5 AND
249:  C      15, WHICH IS MUCH ON THE HIGHER SIDE.
250:  C      ITOP <=1 (RING); ITOP=2 (RING AND RANDOM); ITOP=>3 (RANDOM)
251:  C      NSIGMA=0 (NO CHAOTIC PERTURBATION); NSIGMA=1 (CHAOTIC PERTURBATION)
252:  C      NOTE THAT NSIGMA=1 NEED NOT ALWAYS WORK BETTER (OR WORSE)
253:  C      SUBROUTINE FUNC( ) DEFINES OR CALLS THE FUNCTION TO BE OPTIMIZED.
254:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
255:      COMMON /RNDM/IU, IV
256:      COMMON /KFF/KF, NFCALL, FTIT
257:      INTEGER IU, IV
258:      CHARACTER *70 FTIT
259:      DIMENSION X (N, MX), V (N, MX), A (MX), VI (MX), TIT (50), ABEST (*)
260:      DIMENSION XX (N, MX), F (N), V1 (MX), V2 (MX), V3 (MX), V4 (MX), BST (MX)
261:  C      A1 A2 AND A3 ARE CONSTANTS AND W IS THE INERTIA WEIGHT.
262:  C      OCCASIONALLY, TINKERING WITH THESE VALUES, ESPECIALLY A3, MAY BE
263:  C      NEEDED.
264:      DATA A1, A2, A3, W, SIGMA /.5D00, .5D00, .0005D00, .5D00, 1.D-03/
265:      EPSILON=1.D-12 ! ACCURACY NEEDED FOR TERMINATION
266:  C      -----CHOOSING THE TEST FUNCTION -----
267:      CALL FSELECT (KF, M, FTIT)
268:  C

```

```

269:      FFMIN=1.D30
270:      LCOUNT=0
271:      NFCALL=0
272:      WRITE(*,*) '4-DIGITS SEED FOR RANDOM NUMBER GENERATION'
273:      READ(*,*) IU
274:      DATA FMIN /1.0E30/
275: C     GENERATE N-SIZE POPULATION OF M-TUPLE PARAMETERS X(I,J) RANDOMLY
276:      DO I=1,N
277:          DO J=1,M
278:              CALL RANDOM(RAND)
279:              X(I,J)=RAND
280: C     WE GENERATE RANDOM(-5,5). HERE MULTIPLIER IS 10. TINKERING IN SOME
281: C     CASES MAY BE NEEDED
282:          ENDDO
283:          F(I)=1.0D30
284:      ENDDO
285: C     INITIALISE VELOCITIES V(I) FOR EACH INDIVIDUAL IN THE POPULATION
286:      DO I=1,N
287:          DO J=1,M
288:              CALL RANDOM(RAND)
289:              V(I,J)=(RAND-0.5D+00)
290: C     V(I,J)=RAND
291:          ENDDO
292:      ENDDO
293:      DO 100 ITER=1, ITRN
294: C     WRITE(*,*) 'ITERATION=', ITER
295: C     LET EACH INDIVIDUAL SEARCH FOR THE BEST IN ITS NEIGHBOURHOOD
296:          DO I=1,N
297:              DO J=1,M
298:                  A(J)=X(I,J)
299:                  VI(J)=V(I,J)
300:              ENDDO
301:              CALL LSRCH(A,M,VI,NSTEP,FI)
302:              IF(FI.LT.F(I)) THEN
303:                  F(I)=FI
304:                  DO IN=1,M
305:                      BST(IN)=A(IN)
306:                  ENDDO
307: C     F(I) CONTAINS THE LOCAL BEST VALUE OF FUNCTION FOR ITH INDIVIDUAL
308: C     XX(I,J) IS THE M-TUPLE VALUE OF X ASSOCIATED WITH LOCAL BEST F(I)
309:                  DO J=1,M
310:                      XX(I,J)=A(J)
311:                  ENDDO
312:              ENDIF
313:          ENDDO
314: C     NOW LET EVERY INDIVIDUAL RANDOMLY COSULT NN(<<N) COLLEAGUES AND
315: C     FIND THE BEST AMONG THEM
316:          DO I=1,N
317: C     -----
318:          IF(ITOP.GE.3) THEN
319: C     RANDOM TOPOLOGY *****
320: C     CHOOSE NN COLLEAGUES RANDOMLY AND FIND THE BEST AMONG THEM
321:              BEST=1.0D30
322:              DO II=1,NN
323:                  CALL RANDOM(RAND)
324:                  NF=INT(RAND*N)+1
325:                  IF(BEST.GT.F(NF)) THEN
326:                      BEST=F(NF)
327:                      NFBEST=NF
328:                  ENDIF
329:              ENDDO
330:          ENDIF
331: C     -----
332:          IF(ITOP.EQ.2) THEN
333: C     RING + RANDOM TOPOLOGY *****
334: C     REQUIRES THAT THE SUBROUTINE NEIGHBOR IS TURNED ALIVE
335:              BEST=1.0D30

```

```

336:      CALL NEIGHBOR(I,N,I1,I3)
337:      DO II=1,NN
338:          IF (II.EQ.1) NF=I1
339:          IF (II.EQ.2) NF=I
340:          IF (II.EQ.3) NF=I3
341:          IF (II.GT.3) THEN
342:              CALL RANDOM(RAND)
343:              NF=INT(RAND*N)+1
344:          ENDIF
345:          IF (BEST.GT.F(NF)) THEN
346:              BEST=F(NF)
347:              NFBEST=NF
348:          ENDIF
349:      ENDDO
350:  ENDIF
351: C-----
352:  IF (ITOP.LE.1) THEN
353: C   RING TOPOLOGY *****
354: C   REQUIRES THAT THE SUBROUTINE NEIGHBOR IS TURNED ALIVE
355:      BEST=1.0D30
356:      CALL NEIGHBOR(I,N,I1,I3)
357:      DO II=1,3
358:          IF (II.NE.I) THEN
359:              IF (II.EQ.1) NF=I1
360:              IF (II.EQ.3) NF=I3
361:              IF (BEST.GT.F(NF)) THEN
362:                  BEST=F(NF)
363:                  NFBEST=NF
364:              ENDIF
365:          ENDIF
366:      ENDDO
367:  ENDIF
368: C-----
369: C   IN THE LIGHT OF HIS OWN AND HIS BEST COLLEAGUES EXPERIENCE, THE
370: C   INDIVIDUAL I WILL MODIFY HIS MOVE AS PER THE FOLLOWING CRITERION
371: C   FIRST, ADJUSTMENT BASED ON ONES OWN EXPERIENCE
372: C   AND OWN BEST EXPERIENCE IN THE PAST (XX(I))
373:      DO J=1,M
374:          CALL RANDOM(RAND)
375:          V1(J)=A1*RAND*(XX(I,J)-X(I,J))
376:
377: C   THEN BASED ON THE OTHER COLLEAGUES BEST EXPERIENCE WITH WEIGHT W
378: C   HERE W IS CALLED AN INERTIA WEIGHT 0.01< W < 0.7
379: C   A2 IS THE CONSTANT NEAR BUT LESS THAN UNITY
380:          CALL RANDOM(RAND)
381:          V2(J)=V(I,J)
382:          IF (F(NFBEST).LT.F(I)) THEN
383:              V2(J)=A2*W*RAND*(XX(NFBEST,J)-X(I,J))
384:          ENDIF
385: C   THEN SOME RANDOMNESS AND A CONSTANT A3 CLOSE TO BUT LESS THAN UNITY
386:          CALL RANDOM(RAND)
387:          RND1=RAND
388:          CALL RANDOM(RAND)
389:          V3(J)=A3*RAND*W*RND1
390: C   V3(J)=A3*RAND*W
391: C   THEN ON PAST VELOCITY WITH INERTIA WEIGHT W
392:          V4(J)=W*V(I,J)
393: C   FINALLY A SUM OF THEM
394:          V(I,J)= V1(J)+V2(J)+V3(J)+V4(J)
395:      ENDDO
396:  ENDDO
397: C   CHANGE X
398:      DO I=1,N
399:          DO J=1,M
400:              RANDB=0.D00
401: C-----
402:  IF (NSIGMA.EQ.1) THEN

```

```

403:      CALL RANDOM(RAND) ! FOR CHAOTIC PERTURBATION
404:      IF (DABS(RAND-.5D00) .LT. SIGMA) RANDS=RAND-0.5D00
405: C      SIGMA CONDITIONED RANDS INTRODUCES CHAOTIC ELEMENT IN TO LOCATION
406: C      IN SOME CASES THIS PERTURBATION HAS WORKED VERY EFFECTIVELY WITH
407: C      PARAMETER (N=100,NN=15,MX=100,NSTEP=9,ITRN=100000,NSIGMA=1,ITOP=2)
408:      ENDIF
409: C      -----
410:      X(I,J)=X(I,J)+V(I,J)*(1.D00+RANDS)
411:      ENDDO
412:      ENDDO
413:      DO I=1,N
414:          IF (F(I) .LT. FMIN) THEN
415:              FMIN=F(I)
416:              II=I
417:              DO J=1,M
418:                  BST(J)=XX(II,J)
419:              ENDDO
420:          ENDIF
421:      ENDDO
422:
423:      IF (LCOUNT .EQ. NPRN) THEN
424:          LCOUNT=0
425:          WRITE(*,*) 'OPTIMAL SOLUTION UPTO THIS (FUNCTION CALLS=',NFCALL,')'
426:          WRITE(*,*) 'X = ',(BST(J),J=1,M), ' MIN F = ',FMIN
427: C      WRITE(*,*) 'NO. OF FUNCTION CALLS = ',NFCALL
428:          DO J=1,M
429:              ABEST(J)=BST(J)
430:          ENDDO
431:          IF (DABS(FFMIN-FMIN) .LT. EPSILON) GOTO 999
432:          FFMIN=FMIN
433:          ENDDO
434:          LCOUNT=LCOUNT+1
435:          100 CONTINUE
436:          999 WRITE(*,*) '-----'
437:          DO I=1,N
438:              IF (F(I) .LT. FBEST) THEN
439:                  FBEST=F(I)
440:                  DO J=1,M
441:                      ABEST(J)=XX(I,J)
442:                  ENDDO
443:              ENDIF
444:          ENDDO
445:          CALL FUNC(ABEST,M,FBEST)
446:          CALL GINI(F,N,G1)
447:          WRITE(*,*) 'FINAL X = ',(BST(J),J=1,M), ' FINAL MIN F = ',FMIN
448:          WRITE(*,*) 'COMPUTATION OVER:FOR ',FTIT
449:          WRITE(*,*) 'NO. OF VARIABLES=',M, ' END.'
450:          RETURN
451:      END
452: C      -----
453:      SUBROUTINE LSRCH(A,M,VI,NSTEP,FI)
454:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
455:      CHARACTER *70 FTIT
456:      COMMON /KFF/KF,NFCALL,FTIT
457:      COMMON /RNDM/IU,IV
458:      INTEGER IU,IV
459:      DIMENSION A(*),B(100),VI(*)
460:      AMN=1.0D30
461:      DO J=1,NSTEP
462:          DO JJ=1,M
463:              B(JJ)=A(JJ)+(J-(NSTEP/2)-1)*VI(JJ)
464:          ENDDO
465:          CALL FUNC(B,M,FI)
466:          IF (FI .LT. AMN) THEN
467:              AMN=FI
468:              DO JJ=1,M
469:                  A(JJ)=B(JJ)

```

```

470:         ENDDO
471:         ENDIF
472:     ENDDO
473:     FI=AMN
474:     RETURN
475:     END
476: C -----
477: C THIS SUBROUTINE IS NEEDED IF THE NEIGHBOURHOOD HAS RING TOPOLOGY
478: C EITHER PURE OR HYBRIDIZED
479:     SUBROUTINE NEIGHBOR(I,N,J,K)
480:     IF(I-1.GE.1 .AND. I.LT.N) THEN
481:         J=I-1
482:         K=I+1
483:     ELSE
484:     IF(I-1.LT.1) THEN
485:         J=N-I+1
486:         K=I+1
487:     ENDIF
488:     IF(I.EQ.N) THEN
489:         J=I-1
490:         K=1
491:     ENDIF
492:     ENDIF
493:     RETURN
494:     END
495: C -----
496: C RANDOM NUMBER GENERATOR (UNIFORM BETWEEN 0 AND 1 - BOTH EXCLUSIVE)
497:     SUBROUTINE RANDOM(RAND1)
498:     DOUBLE PRECISION RAND1
499:     COMMON /RNDM/IU,IV
500:     INTEGER IU,IV
501:     IV=IU*65539
502:     IF(IV.LT.0) THEN
503:         IV=IV+2147483647+1
504:     ENDIF
505:     RAND=IV
506:     IU=IV
507:     RAND=RAND*0.4656613E-09
508:     RAND1= DBLE (RAND)
509:     RETURN
510:     END
511: C -----
512:     SUBROUTINE GINI(F,N,G)
513:     PARAMETER (K=1) !K=1 GINI COEFFICIENT; K=2 COEFFICIENT OF VARIATION
514: C THIS PROGRAM COMPUTES MEASURE OF INEQUALITY
515: C IF K =1 GET THE GINI COEFFICIENT. IF K=2 GET COEFF OF VARIATIONE
516:     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
517:     DIMENSION F(*)
518:     S=0.D0
519:     DO I=1,N
520:         S=S+F(I)
521:     ENDDO
522:     S=S/N
523:     H=0.D00
524:     DO I=1,N-1
525:         DO J=I+1,N
526:             H=H+(DABS(F(I)-F(J)))**K
527:         ENDDO
528:     ENDDO
529:     H=(H/(N**2))**(1.D0/K) ! FOR K=1 H IS MEAN DEVIATION;
530: C FOR K=2 H IS STANDARD DEVIATION
531:     WRITE(*,*) 'MEASURES OF DISPERSION AND CENTRAL TENDENCY = ',G,S
532:     G=DEXP(-H) ! G IS THE MEASURE OF EQUALITY (NOT GINI OR CV)
533: C G=H/DABS(S) !IF S NOT ZERO, K=1 THEN G=GINI, K=2 G=COEFF VARIATION
534:     RETURN
535:     END
536: C -----

```

```

537:      SUBROUTINE FSELECT(KF,M,FTIT)
538:      COMMON /CANON/MONE,MTWO
539: C      THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING -----
540: C      (1) FUNCTION CODE (KF), (2) NO. OF VARIABLES IN THE FUNCTION (M);
541:      CHARACTER *70 TIT(100),FTIT
542:      NFN=1
543:      KF=1
544:      WRITE(*,*)'-----'
545:      DATA TIT(1) /'COMPUTE CANONICAL CORRELATION FROM 2 DATA SETS' /
546: C      -----
547:      DO I=1,NFN
548:      WRITE(*,*)TIT(I)
549:      ENDDO
550:      WRITE(*,*)'-----'
551:      WRITE(*,*)'SPECIFY NO. OF VARIABLES IN SET-1[=M1] AND SET-2[=M2]'
552:      READ(*,*) MONE, MTWO
553:      M=MONE+MTWO
554:      FTIT=TIT(KF) ! STORE THE NAME OF THE CHOSEN FUNCTION IN FTIT
555:      RETURN
556:      END
557: C      -----
558:      SUBROUTINE FUNC(X,M,F)
559: C      TEST FUNCTIONS FOR GLOBAL OPTIMIZATION PROGRAM
560:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
561:      COMMON /RNDM/IU,IV
562:      COMMON /KFF/KF,NFCALL,FTIT
563:      INTEGER IU,IV
564:      DIMENSION X(*)
565:      CHARACTER *70 FTIT
566:      NFCALL=NFCALL+1 ! INCREMENT TO NUMBER OF FUNCTION CALLS
567: C      KF IS THE CODE OF THE TEST FUNCTION
568:      IF (KF.EQ.1) THEN
569:      CALL CORD(M,X,F)
570:      RETURN
571:      ENDIF
572: C      =====
573:      WRITE(*,*)'FUNCTION NOT DEFINED. PROGRAM ABORTED'
574:      STOP
575:      END
576: C      -----
577:      SUBROUTINE CORD(M,X,F)
578:      PARAMETER (NOB=30,MVAR=9) ! CHANGE THE PARAMETERS HERE AS NEEDED.
579:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
580:      PARAMETER (MU=2,NF=1,ALAMBDA=5.0D0) !ALAMBDA BETWEEN[0, INDEFINITE)
581: C      MU=1 ABSOLUTE CANONICAL R ; MU=2 SQUARED CANONICAL R;
582: C      NF=1 SUM OF OWN FELLOW ABSOLUTE CORRELATION
583: C      NF=2 SUM OF OWN FELLOW SQUARED CORRELATION
584: C      NF=3 MINIMUM OF OWN FELLOW ABSOLUTE (OR SQUARED) CORRELATION
585: C      -----
586: C      NOB=NO. OF OBSERVATIONS (CASES) & MVAR= NO. OF VARIABLES
587:      COMMON /CANON/MONE,MTWO
588:      COMMON /RNDM/IU,IV
589:      COMMON /CORDAT/CDAT(NOB,MVAR),QIND1(NOB),QIND2(NOB),R(1),NORM,NCOR
590:      COMMON /GETRANK/MRNC
591:      COMMON /MRCCA/OWNR(2,MVAR),FROH,SOWNR1,SOWNR2,MRCC
592:      INTEGER IU,IV
593:      DIMENSION X(*),Z(NOB,2)
594:      MRCC=1 ! FOR CLASSICAL=0; FOR MOST REPRESENTATIVE=1
595:      DO I=1,M
596:      IF (X(I).LT.-1.0D0.OR.X(I).GT.1.0D0) THEN
597:      CALL RANDOM(RAND)
598:      X(I)=(RAND-0.5D0)*2
599:      ENDIF
600:      ENDDO
601: C      NORMALIZATION OF WEIGHTS
602:      XNORM=0.D0
603:      DO J=1,M

```

```

604:      XNORM=XNORM+X (J) **2
605:      ENDDO
606:      XNORM=DSQRT (XNORM)
607:      DO J=1,M
608:      X (J)=X (J) /XNORM
609:      ENDDO
610: C      CONSTRUCT INDEX
611:      DO I=1,NOB
612:      QIND1 (I)=0.D0
613:      QIND2 (I)=0.D0
614:      DO J=1,MONE
615:      QIND1 (I)=QIND1 (I)+CDAT (I,J) *X (J)
616:      ENDDO
617:      DO J=MONE+1,M
618:      QIND2 (I)=QIND2 (I)+CDAT (I,J) *X (J)
619:      ENDDO
620:      ENDDO
621:
622: C      -----
623: C      !FIND THE RANK OF QIND
624:      IF (MRNK.EQ.1) THEN
625:      CALL DORANK (QIND1,NOB)
626:      CALL DORANK (QIND2,NOB)
627:      ENDF
628: C      -----
629: C      FIND CORRELATION OF QIND1 WITH ITS OWN MEMBER VARIABLES
630:      SOWNR1=0.D0
631:      SOWNR2=0.D0
632:      IF (MRCC.NE.0) THEN
633:      DO J=1,MONE
634:      DO I=1,NOB
635:      Z (I,1)=QIND1 (I)
636:      Z (I,2)=CDAT (I,J)
637:      ENDDO
638:      CALL CORLN (Z,NOB,RHO)
639:      OWNR (1,J)=RHO
640:      ENDDO
641: C      FIND CORRELATION OF QIND2 WITH ITS OWN MEMBER VARIABLES
642:      DO J=MONE+1,M
643:      DO I=1,NOB
644:      Z (I,1)=QIND2 (I)
645:      Z (I,2)=CDAT (I,J)
646:      ENDDO
647:      CALL CORLN (Z,NOB,RHO)
648:      OWNR (2,J-MONE)=RHO
649:      ENDDO
650: C      FORMULATION OF CONSTRAINTS
651:      DO J=1,MONE
652:      IF (NF.EQ.1) SOWNR1=SOWNR1+DABS (OWNR (1,J) )
653:      IF (NF.EQ.2) SOWNR1=SOWNR1+OWNR (1,J) **2
654:      ENDDO
655:      SOWNR1=SOWNR1/MONE
656:      IF (NF.EQ.3) THEN
657:      SOWNR1=1.D0
658:      DO J=1,MONE
659:      IF (SOWNR1.GT.DABS (OWNR (1,J) ) ) SOWNR1=DABS (OWNR (1,J) )
660:      ENDDO
661:      ENDF
662:
663:      DO J=1,MTWO
664:      IF (NF.EQ.1) SOWNR2=SOWNR2+DABS (OWNR (2,J) )
665:      IF (NF.EQ.2) SOWNR2=SOWNR2+OWNR (2,J) **2
666:      ENDDO
667:      SOWNR2=SOWNR2/MTWO
668:      IF (NF.EQ.3) THEN
669:      SOWNR2=1.D0
670:      DO J=1,MTWO

```

```

671:         IF (SOWNR2 .GT. DABS (OWNR (2, J) ) ) SOWNR2=DABS (OWNR (2, J) )
672:         ENDDO
673:         ENDIF
674:     ENDIF
675: C -----
676: C COMPUTE CORRELATIONS
677:     DO I=1, NOB
678:         Z (I, 1)=QIND1 (I)
679:         Z (I, 2)=QIND2 (I)
680:     ENDDO
681:     IF (NCOR.EQ.0) THEN
682:         CALL CORLN (Z, NOB, RHO)
683:     ELSE
684:         CALL CORA (Z, NOB, RHO)
685:     ENDIF
686:     R (1)=RHO
687:     FROH=DABS (R (1) )
688:     F=FROH**MU+ALAMBDA* (SOWNR1+SOWNR2)
689: C -----
690:     F=-F
691:     RETURN
692:     END
693:     SUBROUTINE CORLN (Z, NOB, RHO)
694: C NOB = NO. OF CASES
695:     IMPLICIT DOUBLE PRECISION (A-H, O-Z)
696:     DIMENSION Z (NOB, 2), AV (2), SD (2)
697:     DO J=1, 2
698:         AV (J)=0.D0
699:         SD (J)=0.D0
700:     DO I=1, NOB
701:         AV (J)=AV (J)+Z (I, J)
702:         SD (J)=SD (J)+Z (I, J)**2
703:     ENDDO
704:     ENDDO
705:     DO J=1, 2
706:         AV (J)=AV (J)/NOB
707:         SD (J)=DSQRT (SD (J)/NOB-AV (J)**2)
708:     ENDDO
709: C WRITE (*, *) 'AV AND SD ', AV (1), AV (2), SD (1), SD (2)
710:     RHO=0.D0
711:     DO I=1, NOB
712:         RHO=RHO+ (Z (I, 1)-AV (1) ) * (Z (I, 2)-AV (2) )
713:     ENDDO
714:     RHO= (RHO/NOB) / (SD (1) *SD (2) )
715:     RETURN
716:     END
717: C -----
718:     SUBROUTINE CORA (Z, N, R)
719: C COMPUTING BRADLEY'S ABSOLUTE CORRELATION MATRIX
720: C BRADLEY, C. (1985) "THE ABSOLUTE CORRELATION", THE MATHEMATICAL
721: C GAZETTE, 69(447): 12-17.
722:     IMPLICIT DOUBLE PRECISION (A-H, O-Z)
723:     DIMENSION Z (N, 2), X (N), Y (N)
724: C -----
725: C PUT Z INTO X AND Y
726:     DO I=1, N
727:         X (I)=Z (I, 1)
728:         Y (I)=Z (I, 2)
729:     ENDDO
730: C ARRANGE X ANY IN AN ASCENDING ORDER
731:     DO I=1, N-1
732:     DO II=I+1, N
733:         IF (X (I) .GT. X (II) ) THEN
734:             TEMP=X (I)
735:             X (I)=X (II)
736:             X (II)=TEMP
737:         ENDIF

```

```

738:      IF (Y(I) .GT. Y(II)) THEN
739:      TEMP=Y(I)
740:      Y(I)=Y(II)
741:      Y(II)=TEMP
742:      ENDIF
743:      ENDDO
744:      ENDDO
745: C      FIND MEDIAN
746:      IF (INT(N/2) .EQ. N/2.D0) THEN
747:      XMED=(X(N/2)+X(N/2+1))/2.D0
748:      YMED=(Y(N/2)+Y(N/2+1))/2.D0
749:      ENDIF
750:      IF (INT(N/2) .NE. N/2.D0) THEN
751:      XMED=X(N/2+1)
752:      YMED=Y(N/2+1)
753:      ENDIF
754: C      SUBTRACT RESPECTIVE MEDIANS FROM X AND Y AND FIND ABS DEVIATIONS
755:      VX=0.D0
756:      VY=0.D0
757:      DO I=1,N
758:      X(I)=X(I)-XMED
759:      Y(I)=Y(I)-YMED
760:      VX=VX+DABS(X(I))
761:      VY=VY+DABS(Y(I))
762:      ENDDO
763: C      SCALE THE VARIABLES X AND Y SUCH THAT VX=VY
764:      IF (VX .EQ. 0.D0 .OR. VY .EQ. 0.D0) THEN
765:      R=0.D0
766:      RETURN
767:      ENDIF
768:      DO I=1,N
769:      X(I)=X(I)*VY/VX
770:      ENDDO
771: C      COMPUTE CORRELATION COEFFICIENT
772:      VZ=0.D0
773:      R=0.D0
774:      DO I=1,N
775:      VZ=VZ+DABS(X(I))+DABS(Y(I))
776:      R=R+DABS(X(I)+Y(I))-DABS(X(I)-Y(I))
777:      ENDDO
778:      R=R/VZ
779:      RETURN
780:      END
781: C      -----
782:      SUBROUTINE DORANK(X,N) ! N IS THE NUMBER OF OBSERVATIONS
783:      PARAMETER (NRL=0) ! THIS VALUE IS TO BE SET BY THE USER
784: C      !THE VALUE OF NRL DECIDES THE SCHEME OF RANKINGS
785: C      !THIS PROGRAM RETURNS RANK-ORDER OF A GIVEN VECTOR
786:      PARAMETER (MXD=1000) ! MXD IS MAX DIMENSION FOR TEMPORARY VARIABLES
787:      ! THAT ARE LOCAL AND DO NOT GO TO THE INVOKING PROGRAM
788:      ! X IS THE VARIABLE TO BE SUBSTITUTED BY ITS RANK VALUES
789: C      NRULE=0 FOR ORDINAL RANKING (1-2-3-4 RULE);
790: C      NRULE=1 FOR DENSE RANKING (1-2-2-3 RULE);
791: C      NRULE=2 FOR STANDARD COMPETITION RANKING (1-2-2-4 RULE);
792: C      NRULE=3 FOR MODIFIED COMPETITION RANKING (1-3-3-4 RULE);
793: C      NRULE=4 FOR FRACTIONAL RANKING (1-2.5-2.5-4 RULE);
794:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
795:      DIMENSION X(N),NF(MXD),NCF(MXD),RANK(MXD),ID(MXD),XX(MXD)
796: C      GENERATE ID(I),I=1,N
797:      DO I=1,N
798:      ID(I)=I
799:      NF(I)=0
800:      ENDDO
801: C      ARRANGE DATA (X) AND THE IDS IN ASCENDING ORDER
802:      DO I=1,N-1
803:      DO II=I,N
804:      IF (X(II) .LT. X(I)) THEN

```

```

805:      TEMP=X (I)
806:      X (I)=X (II)
807:      X (II)=TEMP
808:      ITEMP=ID (I)
809:      ID (I)=ID (II)
810:      ID (II)=ITEMP
811:      ENDIF
812:      ENDDO
813:      ENDDO
814:  C      MAKE DISCRETE UNGROUPED FREQUENCY TABLE
815:      K=0
816:      J=1
817:      1 K=K+1
818:      XX (K)=X (J)
819:      NF (K)=0
820:      DO I=J, N
821:      IF (XX (K) .EQ. X (I)) THEN
822:      NF (K)=NF (K)+1
823:      ELSE
824:      J=I
825:      IF (J .LE. N) THEN
826:      GOTO 1
827:      ELSE
828:      GOTO 2
829:      ENDIF
830:      ENDIF
831:      ENDDO
832:      2 KK=K
833:      DO K=1, KK
834:      IF (K .EQ. 1) THEN
835:      NCF (K)=NF (K)
836:      ELSE
837:      NCF (K)=NCF (K-1)+NF (K)
838:      ENDIF
839:      ENDDO
840:      DO I=1, N
841:      RANK (I)=1.D0
842:      ENDDO
843:
844:      IF (NRL .GT. 4) THEN
845:      WRITE (*, *) 'RANKING RULE CODE GREATER THAN 4 NOT PERMITTED', NRL
846:      STOP
847:      ENDIF
848:
849:      IF (NRL .LT. 0) THEN
850:      WRITE (*, *) 'RANKING RULE CODE LESS THAN 0 NOT PERMITTED', NRL
851:      STOP
852:      ENDIF
853:
854:      IF (NRL .EQ. 0) THEN
855:      DO I=1, N
856:      RANK (I)=I
857:      ENDDO
858:      ENDIF
859:  C      -----
860:      IF (NRL .GT. 0) THEN
861:      DO K=1, KK
862:      IF (K .EQ. 1) THEN
863:      K1=1
864:      ELSE
865:      K1=NCF (K-1)+1
866:      ENDIF
867:      K2=NCF (K)
868:      DO I=K1, K2
869:      SUM=0.D0
870:      DO II=K1, K2
871:      SUM=SUM+II

```

```

872:      ENDDO
873:      KX=(K2-K1+1)
874:      IF (NRL.EQ.1) RANK(I)=K ! DENSE RANKING (1-2-2-3 RULE)
875:      IF (NRL.EQ.2) RANK(I)=K1 ! STANDARD COMPETITION RANKING(1-2-2-4 RULE)
876:      IF (NRL.EQ.3) RANK(I)=K2 ! MODIFIED COMPETITION RANKING(1-3-3-4 RULE)
877:      IF (NRL.EQ.4) RANK(I)=SUM/KX ! FRACTIONAL RANKING (1-2.5-2.5-4 RULE)
878:      ENDDO
879:      ENDDO
880:      ENDIF
881: C -----
882:      DO I=1,N
883:      X(ID(I))=RANK(I) ! BRINGS THE DATA TO ORIGINAL SEQUENCE
884:      ENDDO
885:      RETURN
886:      END
887: C -----
888:      SUBROUTINE CORREL(X,N,M,RMAT)
889:      PARAMETER (NMX=30) ! DO NOT CHANGE UNLESS NO. OF VARIABLES EXCEED 30
890:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
891:      DIMENSION X(N,M),RMAT(2,2),AV(NMX),SD(NMX)
892:      DO J=1,2
893:      AV(J)=0.D0
894:      SD(J)=0.D0
895:      DO I=1,N
896:      AV(J)=AV(J)+X(I,J)
897:      SD(J)=SD(J)+X(I,J)**2
898:      ENDDO
899:      AV(J)=AV(J)/N
900:      SD(J)=DSQRT(SD(J)/N-AV(J)**2)
901:      ENDDO
902:      DO J=1,2
903:      DO JJ=1,2
904:      RMAT(J,JJ)=0.D0
905:      DO I=1,N
906:      RMAT(J,JJ)=RMAT(J,JJ)+X(I,J)*X(I,JJ)
907:      ENDDO
908:      ENDDO
909:      ENDDO
910:      DO J=1,2
911:      DO JJ=1,2
912:      RMAT(J,JJ)=RMAT(J,JJ)/N-AV(J)*AV(JJ)
913:      RMAT(J,JJ)=RMAT(J,JJ)/(SD(J)*SD(JJ))
914:      ENDDO
915:      ENDDO
916:      RETURN
917:      END
918: C -----
919:      SUBROUTINE DOCORA(ZDAT,N,M,RMAT)
920:      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
921:      DIMENSION ZDAT(N,M),RMAT(2,2),Z(N,2)
922:      DO I=1,N
923:      Z(I,1)=ZDAT(I,1)
924:      Z(I,2)=ZDAT(I,2)
925:      ENDDO
926:      CALL CORA(Z,N,R)
927:      RMAT(1,2)=R
928:      RMAT(2,1)=R
929:      DO J=1,2
930:      RMAT(J,J)=1.D0
931:      ENDDO
932:      RETURN
933:      END

```