SLAC Computation Research Group
Stanford, California

CGTM No. 156
April 1974

CHAN -- Computer Program for One-Dimensional Projection Pursuit

J. H. Friedman

Abstract

A computer program that implements the Projection Pursuit algorithm of Friedman and Tukey is described. This routine seeks to find one-dimensional linear projections of multivariate data that are relatively highly revealing.

INTRODUCTION

CHAN is a collection of subroutines for finding those linear one-dimensional projections of multivariate data that reveal interesting structure. The algorithm and its properties are detailed elsewhere[1] and are only discussed briefly here as they relate to its implementation.

The Projection Pursuit algorithm assigns to each projection direction, $\hat{k}$, in the multidimensional space, a numerical index, $I(\hat{k})$, that corresponds to the degree of structuring present in the data as projected onto $\hat{k}$. The more structure present in the projection, the larger $I(\hat{k})$ becomes. The essence of the algorithm is to find those projection directions, $\hat{k}$, that maximize $I(\hat{k})$. For complex data structures, several solutions may exist, and for each of these, the solution projections can be visually inspected by the researcher for interpretation and judgment as to their usefulness. This multiplicity is often important.

The input to the algorithm consists, principally, of the multivariate data point set and a starting direction, $\hat{k}_s$, in the multidimensional data space. Starting at $\hat{k}_s$, the algorithm finds the direction $\hat{k}^*$ that represents the first local maximum of $I(\hat{k})$ uphill from $\hat{k}_s$. Useful starting directions include the larger principal axes of the data set, the original coordinate axes, and even directions chosen at random. From these searches, several quite distinct solutions may result. Each of these projections can then be examined to determine their usefulness in data interpretation.

THE PROJECTION INDEX

The intent of the projection index is to assign to each direction, $\hat{k}$, in the multivariate space, a numerical value that closely corresponds to the degree of data structuring when projected onto $\hat{k}$. It was found that those projections that were most interesting to researchers were those that tended to

produce many very small interpoint distances while, at the same time, maintaining the overall spread of the data in the projection. Such projections will, for instance, tend to concentrate the points into clusters while, at the same time, separating the clusters.

The spread of the data as projected onto $\hat{k}$ is estimated by taking the trimmed standard deviation of the data from the mean:

$$s(\hat{k}) = \left[\sum_{i=pN}^{(1-p)N} (\vec{X}_i \cdot \hat{k} - \bar{X}_k)^2 \; / \; (1-2p)N \right]^{1/2}$$

where (1)

$$\bar{X}_k = \sum_{i=pN}^{(1-p)N} \vec{X}_i \cdot \hat{k} \; / \; (1-2p)N$$

Here N is the total number of data points, and $\vec{X}_i$ (i=1,N) are the multivariate vectors representing each of the data points, ordered according to their projections $\vec{X}_i \cdot \hat{k}$. A small fraction, p, of the points that lie at each of the extremes of the projections are omitted from both sums. Thus, extreme values of $\vec{X}_i \cdot \hat{k}$ do not contribute to $s(\hat{k})$, which is thus robust against extreme outliers.

For a "small interpoint distance" measure, an average nearness function is used which has the form

$$d(\hat{k}) = \sum_{i=1}^{N} \sum_{j=1}^{N} (R-r_{ij}) \; 1(R-r_{ij}) \qquad (2)$$

where

$$r_{ij} = | \vec{X}_i \cdot \hat{k} - \vec{X}_j \cdot \hat{k} |$$

and $1(\eta)$ is unity for positive valued arguments and zero for negative values. Thus, the double sum is confined to pairs with $0 \leq r_{ij} \leq R$.

CHAN offers two choices for projection indices. One of these simply takes the product

$$I(\hat{k}) = d(\hat{k})\, s(\hat{k}) \tag{3a}$$

keeping the local cutoff radius, R, in $d(\hat{k})$ (Eqn. 2) at a constant value, R_o. (This is the index described in Ref. 1). The other index uses

$$I(\hat{k}) = d(\hat{k}) \tag{3b}$$

but where $R = f_o\, s(\hat{k})$.

Here f_o is a constant fraction. The first index (Eqn. 3a) has the advantage that the cutoff radius, R, (which establishes the distance in the projected subspace over which the local density is averaged and thus establishes the scale of density variation to which the algorithm is sensitive) is a constant completely under user control. The advantage of the second index (Eqn. 3b) is that it is Affine invariant. That is, it is unchanged by scale transformations on the input data. Both projection indices have comparable performance but they do not always find the same solutions.

CONSTRAINT DIRECTIONS

In order to encourage the algorithm to find as many distinct solutions as possible, it is useful to be able to reduce the dimensionality of the space to be searched. This can be done by choosing an arbitrary set of directions, $\{\hat{o}_i\}_{i=1}^{m}$, $m < n$, which need not be mutually orthogonal, and applying the constraints

$$\hat{k}^* \cdot \hat{o}_i = 0 \qquad i = 1,m \tag{4}$$

on the solution direction $\hat{k}^*$. Possible choices for constraint directions might be solution directions found on previous searches, or directions that are known in advance to contain considerable, but well understood, structure. Also, when the choice of scales for the several coordinates is guided by considerations outside the data, one might wish to remove directions with

small variance about the mean since these directions often provide little information about the structure of the data. The introduction of each such constraint direction reduces by one the number of search variables, and thus increases the computational efficiency of the algorithm. CHAN allows for the introduction of an arbitrary number, m < NDIM of non-parallel constraint directions.

IMPLEMENTATION

One-dimensional projection pursuit is invoked from a FORTRAN program by a subroutine call

CALL CHAN(M,RADIUS,NDIM,NPTS,X,KEY,NF,NL,XB,KEYSCL,SCALE,D).

All quantities are input to the routine except XB and SCALE which serve a dual purpose as both input and output arrays. These quantities have the following meanings:

The quantities M and RADIUS refer to the projection index, $I(\hat{k})$.

$$M = \begin{cases} 1 & \text{use Affine invariant projection index (Eqn. 3b)} \\ 2 & \text{use constant radius projection index (Eqn. 3a)} \end{cases}$$

$$\text{RADIUS} = \begin{cases} \text{for M=1:} & \text{constant fraction, } f_o, \text{ (Eqn. 3b)} \\ \text{for M=2:} & \text{constant local cutoff radius, } R_o, \text{ (Eqn. 3a)} \end{cases}$$

RADIUS = 'DFLT': f_o and/or R are automatically calculated by the program.

$$f_o = \text{RADSCL}/\sqrt{\text{NPTS}} \qquad (\text{NPTS} \le \text{NPTSO})$$

$$f_o = \text{RADSCL}*(\sqrt{\text{NPTSO}}\,/\,\log(\text{NPTSO})) * (\log(\text{NPTS})/(\text{NPTS}) \qquad (\text{NPTS} > \text{NPTSO})$$

$$R = f_o * s(\hat{e}_1)$$

In CHAN, RADSCAL is set to 1.0 and NPTSO is set to 1000. (See below on how to change these values.) The quantity $s(\hat{e}_1)$ is the standard deviation about the mean of the data as projected onto the principal axis with the largest eigenvalue. (The trimming factor, p, (Eqn. 1) is zero for this calculation only; i.e., $s(\hat{e}_1)$ is the square root of the largest eigenvalue of the total

sample covariance matrix.) The values for f_o and R are calculated so that the average number of points within the local cutoff radius grows as the square root of the sample size for NPTS $\leq$ NPTSO, and grows as the logarithm for NPTS > NPTSO.

The quantities NDIM, NPTS and X refer to the input data.

NDIM = number of attributes per data point (dimensionality of data space)

NPTS = number of data points (sample size)

X = array, dimensioned X(NDIM,NPTS) in the calling program, that contains the coordinates of the data points.

The quantities KEY,NF,NL and XB refer to the starting directions, $\hat{k}_s$, for the search.

KEY = axis flag =
- 'ORIG': use original axes for starting directions
- 'EIGN': use principal axes as starting directions (ordered in decreasing value of corresponding eigenvalue)
- 'USER': use user supplied vectors as starting directions

NF = axis number of starting direction for first search

NL = axis number of starting direction for last search

There will be NL - NF + 1 different searches whose starting directions will either be the original ('KEY = 'ORIG'), principal (KEY = 'EIGN'), or user supplied (KEY = 'USER') axes NF, NF + 1, ... NL. If NF > NL, there will be no searches, only a principal axis analysis (see below under output).

XB = array, dimensioned XB(NDIM,NL) in calling program, that contains the user supplied starting directions for KEY = 'USER'. For KEY = 'ORIG' or 'EIGN', XB is ignored on input. Upon return from CHAN to the calling program, XB contains the NL - NF + 1 solution directions, $\hat{k}^*$, from the searches starting in XB(1,NF).

The quantities KEYSCL and SCALE refer to the scaling of the input data along the original axes before the projection pursuit. (For the Affine

invariant projection index (M = 1), the results are independent of the scaling of the input data.)

KEYSCL = scaling flag =
- 'NONE' do not scale data
- 'USER' use user provided scales
- 'ORIG' employ automatic scaling
- 'SMAX' automatic scaling with user provided lower limits for scales

SCALE = array, dimensioned SCALE(NDIM). For KEYSCL = 'NONE' or 'ORIG' this is a scratch array for CHAN. For KEYSCL = 'USER', it contains the user provided scales and for KEYSCL = 'SMAX', it contains the user provided lower limits for the axis scales.

The automatically calculated scales for each original axis are the standard deviations of the data about their means, as projected onto each axis. Under the KEYSCL = 'SMAX' option, the axis scale is set to the maximum of the calculated scale and the user provided minimum scale for that axis. (The standard deviation calculation for the scales is trimmed by the global trimming factor IPER, described below. IPER is set to zero in CHAN but may be changed by the user.) Upon return from CHAN to the calling program, SCALE contains the scales used for each axis.

D = Real*8 scratch array for CHAN, dimensioned

Real*8 D(NDIM,NDIM,3).

Labeled Common Scratch Arrays

Three labeled common scratch arrays must be declared in the calling program as working storage for CHAN. These are listed here along with their various dimensions:

```
COMMON // V(240) /POINTS/ Z(NPTS+2) /DATA/P(NDIM*NPTS+1)
```

Constraint Directions

As discussed above, CHAN allows for the introduction of an arbitrary member (<NDIM) of constraint directions. The solution direction, $\hat{k}^*$, will be constrained to be simultaneously perpendicular to each of these directions. The constraint directions need not themselves be mutually orthogonal, but no two may be exactly paralled (to machine accuracy). The constraint directions are entered into a labeled common whose label and dimensions are given below:

```
COMMON / NORMAL / NORM / NORMV / VEC(NDIM,NORM)
```

NORM = number of user supplied constraint directions

VEC = coordinates of constraint vectors

If not referenced by the user, the default value for NORM is zero.

Alternate CALLS

CHAN may be called repeatedly from the calling program, each call initiating NL-NF+1 projection pursuits. If subsequent calls do not change the input data or its scaling, an alternate shortened calling sequence may be used

```
CALL CHARLY(M,RADIUS,KEY,NF,NL,XB)
```

The arguments in the calling sequence have the same meanings as described above. This call may not be the first call to CHAN. All quantities that are absent from this shortened calling sequence have the values assigned to them on the last call to CHAN, using the full calling sequence.

Random Direction Generator

To facilitate starting the projection pursuit algorithm at random directions in the multidimensional space, CHAN provides a random direction generator

```
CALL RANDIR(XB,NDIM,NDIR)
```

Upon return to the calling program, XB [dimensioned XB(NDIM,NDIR) in the calling program] contains the coordinates of NDIR random directions in the NDIM-dimensional space. Repeated calls to RANDIR generates new sets of random directions for each call.

Optional User Control

There are several parameters internal to CHAN that the user may change at his discretion. Some of these parameters control the verboseness of the output. Others are parameters of the projection pursuit algorithm to which it is reasonably insensitive, and thus, they need not be changed frequently. These parameters are stored in the labeled common blocks listed below, along with their default values (set in a BLOCK DATA subprogram internal to CHAN).

```
COMMON /TERSE/ ITERSE
               (0)
```

ITERSE controls the number of pictorial histograms displayed in the output. For ITERSE=1, a pictorial histogram is displayed of the data as projected onto each solution direction, $\hat{k}^*$. If KEY = 'ORIG', then, in addition, the data as projected onto each user supplied starting direction, $\hat{k}_s$, is also displayed. For ITERSE=0, twice NDIM additional histograms are displayed of the data as projected onto each of the original and principal axes.

```
COMMON /COMST/ IPER,PEROUT
               (0)  (.01)
```

IPER is a global trimming parameter for the evaluation of both $d(\hat{k})$ and $s(\hat{k})$ (Eqn's 1 and 2). That is, IPER projected points are deleted from each extreme of the projection before both $d(\hat{k})$ and $s(\hat{k})$ are evaluated for the projection. PEROUT is the parameter p in Eqn. 1 and is a fractional trimming factor for the evaluation of $s(\hat{k})$ only.

```
COMMON /EIT/ EITMIN,RADSCL,NEVO
             (.01)  (1.0) (1000)
```

If the variance of the data as projected onto a principal axis is sufficiently small (as compared to the largest eigenvalue of the covariance matrix), then CHAN will automatically consider that axis to be a constraint direction. EITMIN is the minumum value of the square root of the ratio of the axis variance to the largest eigenvalue, for the principal axis not to be considered a constraint direction. The parameters RADSCL and NEVO are discussed above under the default option for the local cutoff radius.

```
COMMON /CNTRL/ NBINS,CONV1,CONV2,NSTPS,NPRNT
               (120) (.05) (-.01) (2)   (1)
```

NBINS is the number of bins (channels) in the pictorial histogram displays ($\leq$120). CONV1 and CONV2 control the convergence criteria of the maximization algorithm used to maximize $I(\hat{k})$. The search converges when the change in every parameter is less than .1*CONV1, or when $|[I(\hat{k}) - I(\hat{k}')]/I(\hat{k})|$ $\leq$ -CONV2, where $I(\hat{k})$ is the projection index value at the current iteration and $I(\hat{k}')$ is the value at the previous iteration. NSTPS pertains to the maximum number of iterations allowed for the maximizer. This maximum number of iterations is NSTPS, plus the number of search variables. NPRNT controls the verbosity of the printed output from the maximizer.

NPRNT
- $\leq$ 0: no maximizer printed output
- > 0: printed output for every iteration number that is an integral multiple of NPRNT

PRINTED OUTPUT

As indicated above, the nature of the printed output from CHAN depends upon the options chosen by the user. Figure 2 illustrates the printed output from the FORTRAN program listed in Figure 1.

Listed first (SCALE =) are the scale factors that were used to scale each of the NDIM original axes. For the ITERSE=0 option, this is followed by the 2*NDIM pictorial histograms (one per page) of the data as projected onto each of the original and principal axes of the data. For ITERSE=1 (the option used in this example), these histograms are skipped. Shown next

(EIGENSTDS =) are the square roots of the eigenvalues of the total sample (no trimming) covariance matrix, listed in decreasing order. The following line lists the value for f_o of Eqn. 3b (RADIUS SCALE =), R of Eqn. 2 (LOCAL RADIUS =), and EITMIN (see optional user control) times the first (largest) eigen square root value (CUTOFF EIGEN STD =). This latter value is the smallest allowable eigenvalue square root for the corresponding principal axis not to be considered to be a constraint direction. For KEY = 'USER' (as in this example), a pictorial histogram of the data as projected onto the user supplied starting direction, is shown. For KEY = 'ORIG' or 'EIGN', this histogram does not appear. The printed output from the maximizer follows next (if not surpressed by the user). For each iteration in the search for the maximum of the projection index, $I(\hat{k})$, the iteration number, the cumulative number of evaluations of $I(\hat{k})$, the value of $-I(\hat{k})$ (F =), and the values of the search parameters (X=), are listed. Iteration zero corresponds to the starting direction. The parameters of the search are those of the solid angle transform, $SAT(\hat{k})$, of the corresponding direction $\hat{k}$ in the NDIM-dimensional space (see Ref. 1, pg. 7 and Appendix), and have no direct interpretation.

After a solution, $\hat{k}^*$, is found, its components, expressed in terms of both the original coordinates and the principal axes, are listed. This is followed by a pictorial histogram of the data as projected onto this solution direction.

For each call to CHAN, the output sequence is repeated. If, however, subsequent calls are made to CHARLY using the shortened calling sequence, then the purely data dependent information is not repeated.

REFERENCE

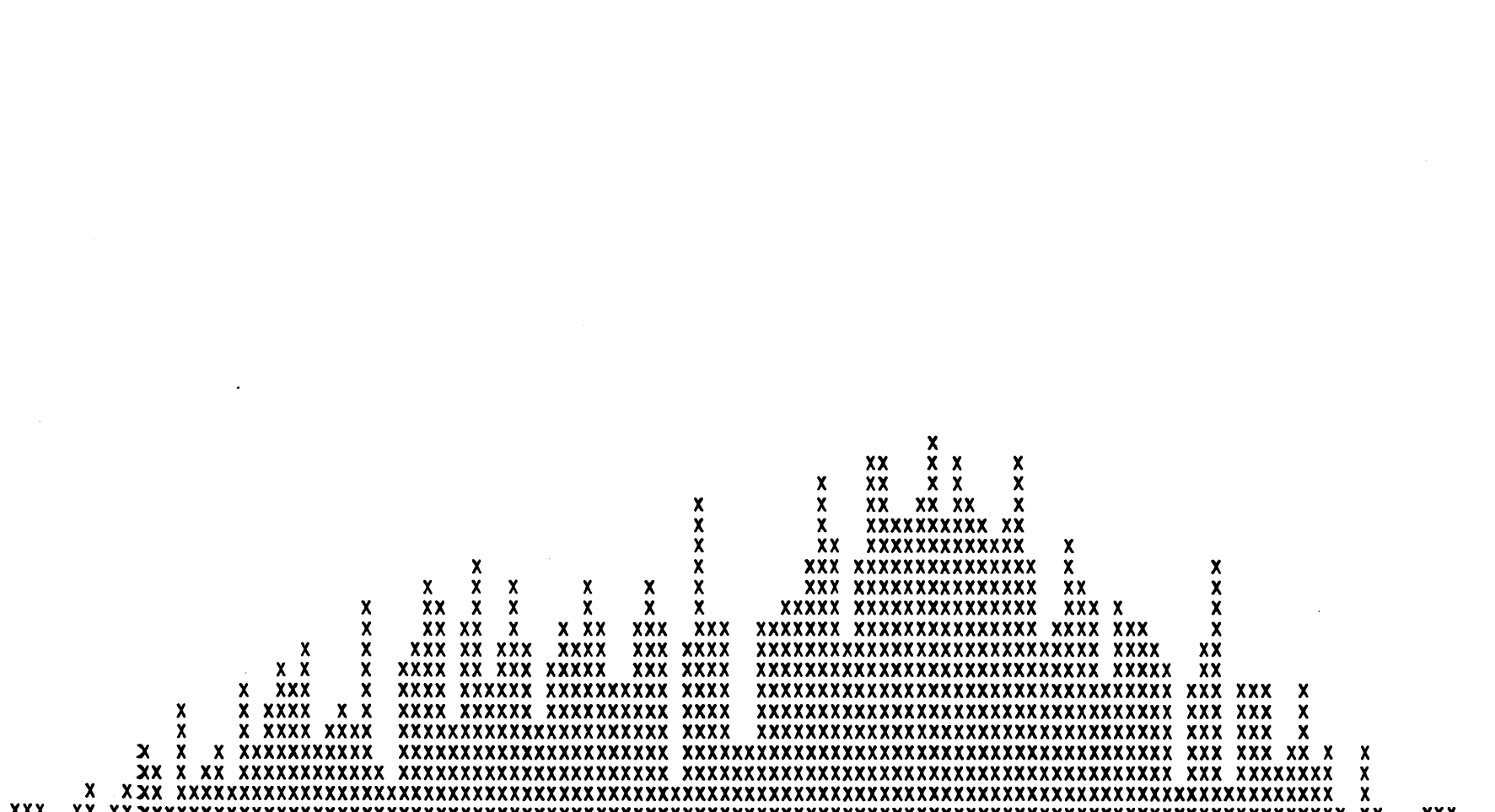
1. J. H. Friedman and J. W. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," Stanford Linear Accelerator Center, Stanford, California, Report, SLAC PUB-1312, September 1973. (To be published in IEEE Trans. Computers.)

```
C  ONE-DIMENSIONAL PROJECTION PURSUIT EXAMPLE. 900 POINTS, 8 DIMENSIONS.
C
C
      COMMON /POINTS/ MM(2),Z(900) /DATA/ NM,P(8,900) /TERSE/ ITERSE
      REAL SCALE(8),X(8,900),XB(8)
      REAL*8 D(8,8,3)
C                                          READ IN DATA
      CALL DATAPT(X,900,8)
C                                        GENERATE RANDOM STARTING DIRECTION
      CALL RANDIR(XB,8,1)
C                                         SET FLAG FOR TERSE OUTPUT
      ITERSE=1
C                                        INITIATE PROJECTION PURSUIT
C
      CALL CHAN(1,'DFLT',8,900,X,'USER',1,1,XB,'NONE',SCALE,D)
C
      STOP
      END
```

FIGURE 1

FIGURE 2a

```
SCALE =  )10000E 01  0.10000E 01  0.10000E 01  0.10000E 01  0.1  )0E 01  0.10000E 01  0.10000E 01  0.10000E 01
EIGENSTD.  0.26598E 01  0.26443E 01  0.26006E 01  0.25448E 01  0.25386E 01  0.25212E 01  0.25078E 01  0.24402E 01
RADIUS SCALE =   0.33333E-01 LOCAL RADIUS =  0.0          CUTOFF EIGEN STD =  0.26598E-01
---CHAN HISTOGRAM--- USER PROJECTION AXIS-P( 1) =
 -0.58970E-01  0.14263E-01  0.28455E 00 -0.59689E 00  0.18062E 00 -0.29691E 00 -0.65349E 00 -0.10605E 00
EIV DECOMP =  0.46365E 00  0.44279E 00 -0.41353E 00  0.63933E-01 -0.12311E 00  0.88291E-01  0.47841E 00 -0.40254E 00
```



```
      LOWER 7777777776666666665555555554444444443333333333222222222111111111100000000000000000011111111112222222223333333334444
        BIN ..........................................................................................................................
       EDGE 9876532109876543210987654321098765432109876543210987654321098765432101234567890123456789012356789012345678901234
            1110099988776665544333221100099887766655443322110099887766655444332211100001112233444556667788999001122233445556677 78
            9516284051738406173940628395162849517384061739406283951628495173840617394062260493715048271594826059372604937150482715 94
            6319742075308631964297520863196429752085318641974208531864197420753086319642035702479146813580257914681358025792469136 80
            *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
                                    1    11 11  1   1 11  111  111  1111111 111111111111111 1111 111     1
            1110012012431623427468794565132892150379295809207702029600440011374938855695865458390421710098279327773473410410001110 00
  CONTENTS  ..........................................................................................................................
            0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
ITER    0      1 CALLS  F=-0.5935874E 03  X= 0.1287E 00 0.7744E 00 0.7013E 00 0.8490E 00 0.9261E 00 0.3613E 00 0.2260E 00

ITER    1     27 CALLS  F=-0.1038127E 04  X= 0.4516E-01 0.4036E 00 0.9794E 00 0.7321E 00 0.9261E 00 0.3488E 00 0.1333E 00
```

FIGURE 2b

```
ITER     2     54 CALLS  F=-0.1063106E 04  X= 0.4671E-01 0.4030E 00 0.9807E 00 0.6242E 00 0.9355E 00 0.3486E 00 0.1316E 00

ITER     3     81 CALLS  F=-0.1079041E 04  X= 0.4780E-01 0.4137E 00 0.9782E 00 0.6236E 00 0.9325E 00 0.3475E 00 0.1299E 00

ITER     4    107 CALLS  F=-0.1081021E 04  X= 0.4780E-01 0.4183E 00 0.9779E 00 0.6236E 00 0.9339E 00 0.3474E 00 0.1308E 00

---CHAN HISTOGRAM--- MAXIMUM PROJECTION AXIS-P( 1) =
  0.92981E-02 -0.57118E-02  0.42183E-03 -0.75109E 00  0.92701E-01  0.16599E 00 -0.61447E 00  0.14841E 00
EIV DECOMP =  0.20489E 00  0.66160E 00  0.51166E-01 -0.90716E-01 -0.91601E-01 -0.93173E-01  0.57440E 00 -0.40305E 00
```