SLAC-PUB-1312
(M)
September 1973

A PROJECTION PURSUIT ALGORITHM FOR EXPLORATORY DATA ANALYSIS

Jerome H. Friedman
Stanford Linear Accelerator Center*
Stanford, California 94305

and

John W. Tukey
Princeton University**
Princeton, New Jersey 08540
and
Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

An algorithm for the analysis of multivariate data is presented, and discussed in terms of specific examples. The algorithm seeks to find one- and two-dimensional linear projections of multivariate data that are relatively highly revealing.

*Supported by the U.S. Atomic Energy Commission under Contract AT(043)515.

**Prepared in part in connection with research at Princeton University supported by the U.S. Atomic Energy Commission.

(Submitted to IEEE Trans. on Computers)

KEY PHRASES

Multivariate data analysis	Dimensionality reduction
Statistics	Clustering
Multidimensional scaling	Mappings

Non-parametric pattern recognition

Introduction

Mapping of multivariate data onto low-dimensional manifolds for visual inspection is a commonly used technique in data analysis. The discovery of mappings that reveal the salient features of the multidimensional point swarm is often far from trivial. Even when every adequate description of the data requires more variables than can be conveniently perceived (at one time) by humans, it is quite often still useful to map the data into a lower, humanly perceivable, dimensionality where the human gift for pattern recognition can be applied.

While the particular dimension-reducing mapping used may sometimes be influenced by the nature of the problem at hand, it seems usually to be dictated by the intuition of the researcher. Potentially useful techniques can be divided into three classes:

1) Linear dimension-reducers, which can usually be usefully thought of as projections.

2) Non-linear dimension-reducers that are defined over the whole high-dimensional space. (No examples seem as yet to have been seriously proposed for use in any generality.)

3) Non-linear mappings that are only defined for the given points -- most of these begin with the mutual interpoint distances as the basic ingredient. (Minimal spanning trees,[2] and iterative algorithms for non-linear mappings,[3,4] are examples. The literature of clustering techniques is extensive.*)

While the non-linear algorithms have the ability to provide a more faithful representation of the multidimensional point swarm than the linear methods, they

*See References 1 and 2 and their references for a reasonably extensive bibliography on clustering techniques.

can suffer from some serious shortcomings; namely, the resulting mapping is often difficult to interpret; it cannot be summarized by a few parameters; it only exists for the data set used in the analysis, so that additional data cannot be identically mapped; and, for moderate to large data bases, its use is extremely costly in computational resources (both CPU cycles and memory).

Our attention here is devoted to linear methods, more specifically to those expressable as projections (though the technique seems extendable to more general linear methods). Classical linear methods include principal components and linear factor analysis. Linear methods have the advantages of straight-forward interpretability and computational economy. Linear mappings provide parameters which are independently useful in the understanding of the data, as well as being defined throughout the space, thus allowing the same mapping to be performed on additional data that was not part of the original analysis. The disadvantage of many classical linear methods is that the only property of the point swarm that is used to determine the mapping is a global one, usually the swarm's variance along various directions in the multidimensional space. Techniques that, like projection pursuit, combine global and local properties of multivariate point swarms to obtain useful linear mappings have been proposed by Kruskal.[5,6] Since projection pursuit uses trimmed global measures, it has the additional advantage of robustness against outliers.

Projection Pursuit

This note describes a linear mapping algorithm that uses interpoint distances as well as the variance of the point swarm to pursue optimum projections. This projection pursuit algorithm associates with each direction in the multidimensional space, a continuous index that measures its "usefulness" as a projection axis, and then varies the projection direction so as to maximize this index. This projection index is sufficiently continuous to allow the use of sophisticated hill climbing algorithms for the maximization, thus increasing computational efficiency. (In particular, both Rosenbrock[7] and Powell principal axis[8] methods have proved very successful). For complex data structures, several solutions may exist, and for each of these the projections can be visually inspected by the researcher for interpretation and judgment as to their usefulness. This multiplicity is often important.

Computationally, the projection pursuit (PP) algorithm is considerably more economical than the non-linear mapping algorithms. Its memory requirements are simply proportional to the number of data points, N, while the number of CPU cycles required grows as NlogN for increasing N. This allows the PP algorithm to be applied to much larger data bases than is possible with nonlinear mapping algorithms, where both the memory and CPU requirements tend to grow as N^2. Since the mappings are linear, they have the advantages of straightforward interpretability and convenient summarization. Also, once the parameters that define a solution projection are obtained, additional data that did not participate in the search can be mapped onto it. For example, one may apply projection pursuit to a data subsample of size N_s. This requires computation proportional to $N_s \log N_s$. Once a subsample solution projection is found, the entire data set can be projected onto it (for inspection by the researcher) with CPU requirements simply proportional to N.

When combined with isolation,[9] projection pursuit has been found to be an effective tool for cluster detection and separation. As projections are found that separate the data into two or more apparent clusters, the data points in each cluster can be isolated. The PP-algorithm can then be applied to each cluster separately, finding new projections that may reveal further clustering within each isolated data set. These sub-clusters can each be isolated and the process repeated.

Because of its computational economy, projection pursuit can be repeated many times on the entire data base, or its isolated subsets, making it a feasible tool for exploratory data analysis. The algorithm has so far been implemented for projection onto one and two dimensions; however, there is no fundamental limitation on the dimensionality of the projection space.

The Projection Index

The choice of the intent of the projection index, and to a somewhat lesser degree the choice of its details, are crucial for the success of the algorithm. Our choice of intent was motivated by studying the interaction between human operators and the computer on the PRIM-9 interactive data display system.[10] This system provides the operator with the ability to rotate the data to any desired orientation while continuously viewing a two-dimensional projection of the multi-dimensional data. These rotations are performed in real time and in a continuous manner under operator control. This gives the operator the ability to perform manual projection pursuit. That is, by controlling the rotations and viewing the changing projections, the operator can try to discover those data orientations (or equivalently projection directions) that reveal to him interesting structure. It was found that the strategy most frequently employed by researchers operating the system was to seek out projections that tended to produce many very small interpoint distances while, at the same time, maintaining the overall spread of the data. Such strategies will, for instance, tend to concentrate the points into clusters while, at the same time, separating the clusters.

The P-indexes we use to express quantitatively such properties of a projection axis, $\hat{k}$, can be written as a product of two functions

$$I(\hat{k}) = s(\hat{k})\, d(\hat{k}) \tag{1}$$

where $s(\hat{k})$ measures the spread of the data, and $d(\hat{k})$ describes the "local density" of the points after projection onto $\hat{k}$. For $s(\hat{k})$, we take the trimmed standard deviation of the data from the mean as projected onto $\hat{k}$;

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

where (2)

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

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 projection 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 $d(\hat{k})$, we use an average nearness function of the form

$$d(\hat{k}) = \sum_{i=1}^{N} \sum_{j=1}^{N} f(r_{ij})\, 1(R - r_{ij}) \tag{3}$$

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

and $1(\eta)$ is unity for positive valued arguments and zero for negative values. (Thus, the double sum is confined to pairs with $0 \le r_{ij} < R$.) The function f(r) should be monotonically decreasing for increasing r in the range $r \le R$, reducing to zero at r=R. This continuity assures maximum smoothness of the objective function, $I(\hat{k})$.

For moderate to large sample size, N, the cutoff radius, R, is usually chosen so that the average number of points contained within the window, defined by the step function $1(R-r_{ij})$, is not only a small fraction of N but increases much more slowly than N, say as log N. After sorting the projected point values, $\vec{X}_i \cdot \hat{k}$, the number of operations required to evaluate $d(\hat{k})$ [as well as $s(\hat{k})$] is thus about a fixed multiple of N log N (probably no more than this for large N). Since sorting requires a number of operations proportional to N log N, the same is true of the entire evaluation of $d(\hat{k})$ and $s(\hat{k})$ combined.

Projections onto two dimensions are characterized by two directions $\hat{k}$ and $\hat{\ell}$ (conveniently taken to be orthogonal with respect to the initially given coordinates and their scales). For this case, equation (2) generalizes to

$$s(\hat{k},\hat{\ell}) = s(\hat{k})\ s(\hat{\ell}) \tag{2a}$$

and r_{ij} becomes $r_{ij} = [(\vec{X}_i \cdot \hat{k} - \vec{X}_j \cdot \hat{k})^2 + (\vec{X}_i \cdot \hat{\ell} - \vec{X}_j \cdot \hat{\ell})^2]^{1/2}$ (3a)

in equation (3).

Repeated application of the algorithm has shown that it is insensitive to the explicit functional form of f(r) and shows major dependence only on its characteristic width

$$\bar{r} = \int_0^R r f(r)\,dr \Big/ \int_0^R f(r)\,dr. \qquad \text{(one dimension)} \tag{4}$$

$$\bar{r} = \int_0^R r f(r)\,r\,dr \Big/ \int_0^R f(r)\,r\,dr \qquad \text{(two dimensions)}. \tag{4a}$$

It is this characteristic width that seems to define "local" in the search for maximum local density. Its value 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. Experimentation has also shown that when a preferred direction is available, the algorithm is remarkably stable against small to moderate changes in $\bar{r}$, but it does respond to large changes in its value (say a few factors of two).

The projection index $I(\hat{k})$ [or $I(\hat{k},\hat{\ell})$ for two-dimensional projections] measures the degree to which the data points in the projection are both concentrated locally ($d(\hat{k})$ large) while, at the same time, expanded globally ($s(\hat{k})$ large). Experience has shown that projections that have this property to a large degree tend to be those that are most interesting to researchers. Thus, it seems natural to pursue those projections that maximize this index.

One-Dimensional Projection Pursuit

The projection index for projection onto a one-dimensional line imbedded in an n-dimensional space is a function of n-1 independent variables that define the direction of the line, conveniently its direction cosines. These cosines are the n components of a vector parallel to the line subject to the constraint that the squares of these components sum to unity. Thus, we seek the maxima of the P-index, $I(\hat{k})$, on the (n-1)-dimensional surface of a sphere of unit radius, $S^n(1)$, in an n-dimensional Euclidean space.

One technique for accomplishing this is to apply a solid angle transform (SAT)[11] which reversibly maps such a sphere to an (n-1)-dimensional infinite Euclidean space $E^{n-1}(-\infty,\infty)$ (see appendix). This reduces the problem from finding the maxima of $I(\hat{k})$ on the unit sphere in n-dimensions, to finding the equivalent maxima of $I[SAT(\hat{k})]$ in $E^{n-1}(-\infty,\infty)$. This replacement of the constrained optimization problem with a totally unconstrained one, greatly increases the stability of the algorithm and simplifies its implementation. The variables of the search are the n-1 SAT parameters, and for any such set of parameters there exists a unique point on the n-dimensional sphere defined by the n components of $\hat{k}$.

The computational resources required by projection pursuit are greatly affected by the algorithm used in the search for the maxima of the projection index. Since the number of CPU cycles required to evaluate the P-index for N data points grows as NlogN, it is important for moderate to large data sets to employ a search algorithm that requires as few evaluations of the object function as possible. It is usually the case that the more sophisticated the search algorithm, the smoother the object function is required to be for stability. The P-index, $I(\hat{k})$, as defined above, is remarkably smooth, and both Rosenbrock[12] and Powell principal axis[13] search algorithms have been successfully applied without encountering any instability problems. For these algorithms, the number of objective function evaluations per varied parameter required to find a solution projection has been found to vary considerably from instance to instance and to be strongly influenced by the convergence criteria established by the user. Applying a rather demanding convergence criteria, approximately 15-25 evaluations per varied parameter were required to achieve a solution. (Stopping when the P-index changes by only a few percent seems reasonable. A convergence criteria of one percent was used in all of our applications.)

In order to be useful as a tool for exploratory data analysis on data sets with complex structure, it is important that the algorithm find several solutions that represent potentially informative projections for inspection by the researcher. This can be accomplished by applying the algorithm many times with different starting directions for the search. Useful starting directions include the larger principal axes of the data set, the original coordinate axes, and even directions chosen at random. From each starting direction, $\hat{k}_s$, the algorithm finds the solution projection axis, $\hat{k}_s^*$, corresponding to the first maximum of the P-index uphill from the starting point. From these searches, several quite distinct solutions often result. Each of these projections can then be examined to determine their usefulness in data interpretation.

In order to encourage the algorithm to find additional distinct solutions, it is useful to be able to reduce the dimensionality of the sphere 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

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

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.

The algorithm can allow for the introduction of an arbitrary number, $m < n$, of non-parallel constraint directions. This is accomplished by using Householder reductions[14] to form an orthogonal basis for the (n-m)-dimensional orthogonal subspace of the m-dimensional subspace spanned by the m constraint vectors $\{\hat{o}_i\}_{i=1}^m$. The n-m-1 search variables are then the solid angle transform parameters of the unit sphere in this (n-m)-dimensional space. The transformation from the original n-dimensional data space proceeds in two steps. First, a linear dimension reducing transformation to the (n-m)-dimensional complement subspace, and then the nonlinear SAT that maps the sphere, $S^{n-m}(1)$, to $E^{n-m-1}(-\infty,\infty)$.

The projection index, $I(\hat{k},\hat{\ell})$, for a two-dimensional plane imbedded in an n-dimensional space is defined by eqns 2a, 3 and 3a. This index is a function of the parameters that define such a plane. Proceeding in analogy with one-dimensional projection pursuit, one could seek the maximum of $I(\hat{k},\hat{\ell})$ with respect to these parameters. The data projected onto the plane represented by the solution vectors, $\hat{k}^*$ and $\hat{\ell}^*$, can then be inspected by the researcher.

Another useful strategy is to hold one of the directions (for example $\hat{k}$) constant along some interesting direction, and then seek the maximum of $I(\hat{k},\hat{\ell})$ with respect to $\hat{\ell}$ in $\tilde{E}^{n-1}(\hat{k})$, the (n-1)-dimensional subspace orthogonal to $\hat{k}$. This reduces the number of search parameters to n-2. The choice of the constant direction, $\hat{k}$, could be motivated by the problem at hand (like one of the original or principal axes), or it could be a solution direction found in a one dimensional projection pursuit.

A third, intermediate, strategy would be to first fix $\hat{k}$ and seek the maximum of $I(\hat{k},\hat{\ell})$ in $\tilde{E}^{n-1}(\hat{k})$, as described above. Then, holding $\hat{\ell}$ fixed at the solution value $\hat{\ell}^*$, vary $\hat{k}$ in $\tilde{E}^{n-1}(\hat{\ell}^*)$ seeking a further maximum of $I(\hat{k},\hat{\ell}^*)$. This process of alternately fixing one direction and varying the other in the orthogonal subspace of the first, can be repeated until the solution becomes stable. The final directions $\hat{k}^*$ and $\hat{\ell}^*$ are then regarded as defining the solution plane. This third strategy, while not as completely general as the first, is computationally much more efficient. This is due to the economies that can be achieved in computing $I(\hat{k},\hat{\ell})$, knowing that one of the directions is constant and that $\hat{k}\cdot\hat{\ell} = 0$. (Using similar criteria for choosing the cutoff radius as that used for one-dimensional projection pursuit, and sorting the projected values along the constant direction, allows $I(\hat{k},\hat{\ell})$ to be evaluated with a number of operations proportional to N log N.)

As for the one-dimensional case, the two-dimensional P-index, $I(\hat{k},\hat{\ell})$ is sufficiently smooth to allow the use of sophisticated optimization algorithms. Also, constraint directions can be introduced in the same manner as described above for one dimensional projection pursuit.

Some Experimental Results

To illustrate the application of the algorithm, we describe its effect upon several data sets. The first two are artifically generated so that the results can be compared with the known data structure. The third is the well known Iris data used by Fisher[15] and the fourth is a data set taken from a particle physics experiment. For these examples, $f(r) = R-r$, for one-dimensional projection pursuit (eqn 3), while for two-dimensional projection pursuit (eqn 3a), $f(r) = R^2-r^2$. In both cases R was set to ten percent of the square root of the data variance along the largest principal axis, and the trimming (eqn 2) was $P = .05$.

1) Uniformally Distributed Random Data

To test the effect of projection pursuit on artificial data having no preferred projection axes, we generated 975 data points, randomly, from a uniform distribution inside a 14-dimensional sphere, and repeatedly applied one and two-dimensional projection pursuit to the sample with different starting directions. Table 1 shows the results of 28 such trials with one-dimensional projection pursuit where the starting directions were the 14 original axes and the 14 principal axes of the data set. The results of the two-dimensional projection pursuit trials were very similar.

The results shown in Table 1 strongly reflect the uniform nature of the 14-dimensional data set. The standard deviation of the index values for the starting directions is less than one percent, while the increase achieved at the solutions averages four percent. Also, only two searches (runs 13 and 14) appeared to converge to the same solution. The angle between the two directions corresponding to the largest P-indices found (runs 19 and 21) was 67 degrees. The small increase in the P-index from the starting to the solution directions, indicates that the algorithm considers these solution directions at most only slightly better projection axes than the starting directions. Visual inspection of the data projections verifies this assessment.

2) Gaussian Data Distributed at the Vertices of a Simplex

The previous example shows that projection pursuit finds no seriously preferred projection axes when applied to spherically-uniform random data. Another interesting experiment is to test its effect on an artificial data set with considerable multidimensional structure. Following Sammon,[4] we applied one-dimensional projection pursuit to a data set consisting of 15 spherical Gaussian clusters of 65 points, each centered at the vertices of a 14-dimensional simplex. The variance of each cluster is one, while the distance between centers is ten. Thus, the clusters are well separated in the 14-dimensional space. Figure 1a shows this data projected onto the direction of its largest principal axis. (For this sample, the largest standard deviation was about 1.15 times the smallest.) As can be seen, this projection shows no hint of the multidimensional structure of the data. Inspection of the one and two-dimensional projections onto the other principal axes shows the same result.

Using the largest principal axis (Fig. 1a) as the starting direction, the one-dimensional PP algorithm yielded the solution shown in Figure 1b. The threefold increase in the P-index at the solution indicates that the algorithm considers it a much better projection axis than the starting direction. This is verified by visual inspection of the data as projected onto the solution axis, where the data set is seen to break up into two well separated clusters of 65 and 910 points.

In order to investigate possible additional structure, we isolated each of the clusters and applied projection pursuit to each one individually. The results are shown in Figures 1c and 1d. The solution projection for the 65 point isolate showed no evidence for additional clustering, while the 910 point sample clearly separated into two subclusters of 130 and 780 points. We further isolated these two subclusters and applied projection pursuit to each one individually. The results are illustrated in Figures 1e and 1f. The solution for the 130 point subcluster shows it divided into two clusters of 65 points each, while the 780 point cluster separates into a 65 point cluster and 715 point cluster.

Continuing with these repeated applications of isolation and projection pursuit, one finds, after using a sequence of linear projections, that the data set is composed of 15 clusters of 65 points each.

Two-dimensional projection pursuit could equally well be applied at each stage in the above analysis. This has the advantage that the solution at a given stage sometimes separates the data set into three apparent clusters. The disadvantage is the increased computational requirements of the two-dimensional projection pursuit algorithm.

3) Iris Data

This is a classical data set first used by Fisher[15] and subsequently by many other researchers for testing statistical procedures. The data consists of measurements made on 50 observations from each of three species (one quite different than the other two) of Iris flowers. Four measurements were made on each flower and there were 150 flowers in the entire data set. Taking the largest principal axes as starting directions, we applied projection pursuit to the entire four-dimensional data set. The result for two-dimensional projection pursuit is shown in Figure 2a. As can be seen, the data as projected on the solution plane shows clear separation into two well defined clusters of 50 (one species) and 100 (two unresolved species) points. The one-dimensional algorithm also clearly separates the data into these two clusters. However, this two-cluster separation is easy to achieve and is readily apparent from simple inspection of the original data.

Applying the procedure discussed above, we isolate the 100 point cluster (largest standard deviation was about six times the smallest) and re-apply projection pursuit, starting with the largest principal axes of the isolate. Figure 2b shows the data projected onto the plane defined by the two largest principal axes. Here the data seem to show no apparent clustering.

One dimensional projection pursuit, starting with the largest principal axis, was unable to separate this isolate into discernible clusters. Figure 2c shows the results of two-dimensional projection pursuit starting with the plane of Figure 2b. This solution plane seems to divide the projected data into two discernible clusters. One in the lower right hand quadrant with higher than average density seems slightly separated from another, which is somewhat sparser and occupies most of the rest of the projection. In order to see to what extent this apparent clustering corresponds to the different Iris species known to be contained in this isolate, Figure 2d tags these species. As can be seen, the two clusters very closely correspond to the two Iris species. Also shown in Figure 2d are some level lines of (the projection onto the same plane of) Fisher's linear discriminant function[15], $\hat{f}$, for this isolate calculated by using the known identities of the two species. In this example, the angle between the direction upon which this linear discriminant function is a projection, and this plane, is a little more than 45°.

The projection pursuit solution can be compared to a two-dimensional projection of this isolate that is chosen to provide maximum separation of the two species, given the a priori information as to which species each data point represents. Figure 2e shows the isolate projected onto such a plane whose horizontal coordinate is the value of Fisher's linear discriminant for the isolate in the full four-dimensional space, $\hat{f}$, while the vertical axis is the value of a similar Fisher linear discriminant in $\tilde{E}^3(\hat{f})$, the three-dimensional space orthogonal to $\hat{f}$.[16] (If the "within variance" were spherical in the initially given coordinate system, this vertical coordinate would not be well defined, since the centers of the two species groups would coincide in $\tilde{E}^3(\hat{f})$. While we may feel that the vertical coordinate adds little to the horizontal one, linear discrimination seems to offer no better choice of a second coordinate, especially since we would like this view also to be a projection of the original data -- a projection in terms of the original coordinates and scales -- as all two-dimensional projection pursuit views are required to be.) A comparison of Figures 2d and 2e shows that the unsupervised projection pursuit solution achieves separation of the two species equivalent to this discriminant plane. Since these two species are known to touch in the full four-dimensional space,[2,4] it is probably not possible to find a projection that completely separates them.

4) <u>Particle Physics Data</u>

For the final example, we apply projection pursuit to a data set taken from a high energy particle physics scattering experiment[17]. In this experiment, a beam of positively charged pi mesons, with an energy of 16 billion electron volts, was used to bombard a stationary target of protons contained in hydrogen nuclei. Five hundred examples were recorded of those nuclear reactions in which the final products were a proton, two positively charged pi-mesons, and a negatively charged pi-meson. Such a nuclear reaction with four reaction products can be completely described by 7 independent measurables*. This data can thus be regarded as 500 points in a seven-dimensional space.

The data projected onto its largest principal axis is shown in Figure 3a, while the projection onto the plane defined by the largest two principal axes is shown in Figure 3c. (The largest standard deviation was about eight times the smallest). One-dimensional projection pursuit was applied starting with the largest principal axis. Figure 3b shows the data projected onto the solution direction. The result of a two-dimensional projection pursuit starting with the plane of Figure 3c is shown in Figure 3d.

Although the principal axis projections indicate possible structure within the data set, the projection pursuit solutions are clearly more revealing. This is indicated by the substantial increase in the P-index, and is verified by visual inspection. In particular, the two-dimensional solution projection shows that there are at least three clusters, possibly connected, one of which reasonably separates from the other two. Proceeding as above, one could isolate this cluster from the others and apply projection pursuit to the two samples separately, continuing the analysis.

---

*For this reaction, $\pi_b^+ p_t \rightarrow p\pi_1^+ \pi_2^+ \pi^-$, the following measurables were used: $X_1 = \mu^2(\pi^-, \pi_1^+, \pi_2^+)$, $X_2 = \mu^2(\pi^-, \pi_1^+)$, $X_3 = \mu^2(p, \pi^-)$, $X_4 = \mu^2(\pi^-, \pi_2^+)$, $X_5 = \mu^2(p, \pi_1^+)$, $X_6 = \mu^2(p, \pi_1^+, -p_t)$, and $X_7 = \mu^2(p, \pi_2^+, -p_t)$. Here, $\mu^2(A, B, \pm C) = (E_A + E_B \pm E_C)^2 - (\vec{P}_A + \vec{P}_B \pm \vec{P}_C)^2$ and $\mu^2(A, \pm B) = (E_A \pm E_B)^2 - (\vec{P}_A \pm \vec{P}_B)^2$, where E and $\vec{P}$ represent the particle's energy and momentum respectively, as measured in billions of electron volts. The notation $(\vec{p})^2$ represents the inner product $\vec{P}\cdot\vec{P}$. The ordinal assignment of the two π^+'s was done randomly.

Discussion

The experimental results of the previous section indicate that the PP algorithm behaves reasonably. That is, it tends to find structure when it exists in the multidimensional data and it does not find structure when it is known not to exist. When combined with isolation, projection pursuit seems to be an effective tool for the detection of certain types of clustering.

Because projection pursuit is a linear mapping algorithm, it suffers from some of the well known limitations of linear mapping. The algorithm will have difficulty in detecting clustering about highly curved surfaces in the full dimensionality. In particular, it cannot detect nested spherical clustering. It can, however, detect nested cylindrical clustering where the cylinders have parallel generators.

Projection pursuit leaves to the researcher's discretion the choice of measurement variables and metric. The algorithm is, of course, sensitive to change of relative scale of the input measurement variables, as well as to highly nonlinear transformations of them. If there is no a priori motivation for a choice of scale for the measurement variables, then they can be independently scaled (standardized) so as to all have the same variance. In the spirit of exploratory data analysis, the researcher might employ projection pursuit to several carefully selected non-linear transformations of his measurement variables. For example, transformations to various spherical polar coordinate representations[11] would enable projection pursuit to detect nested spherical clustering.

Frequently with multidimensional data, only a few of the measurement variables contribute to the structure or clustering. The clusters may overlap in many of the dimensions and separate in only a few. As pointed out by both Sammon[4] and Kruskal[6], those variables that are irrelevant to the structure or clustering can dilute the effect of those that display it, especially for those mapping algorithms that depend solely on the multidimensional interpoint distances. It is easy to see that the projection pursuit algorithm does not suffer seriously from this effect. Projection pursuit will automatically tend to avoid projections involving those measurement variables that do not contribute to data structure, since the inclusion of these variables will tend to reduce $d(\hat{k})$ while not modifying $s(\hat{k})$ greatly.

In order to apply the PP algorithm, the researcher is not required to possess a great deal of a priori knowledge concerning the structure of his data, either for setting up the control parameters for the algorithm or for interpreting its results. The only control parameter requiring care is the characteristic radius $\bar{r}$ defined in eqn. 4. Its value establishes the minimum scale of density variation detectable by the algorithm. A choice for its value can be influenced by the global scale of the data as well as any information that may be known about the nature of the variations in the multivariate density of the points. The sample size is also an important consideration since the radius should be large enough to include, on the average, enough points (in each projection) to obtain a reasonable estimate of the local density. These considerations usually result in a compromise, between making $\bar{r}$ as small as possible, consistent with the sample size requirement. Because of the computational efficiency of the algorithm, however, it is possible to apply it several times with different values for $\bar{r}$. Interpretation of the results of projection pursuit is especially straightforward owing to the linear nature of the mapping.

The researcher also has the choice of the dimensionality of the projection subspace. That is, whether to employ one, two or higher dimensional projection pursuit. The two-dimensional projection pursuit algorithm is slower and slightly less stable than the one-dimensional algorithm; however, the resulting two-dimensional map contains much more information about the data. Experience has shown that a useful strategy is to first find several one-dimensional PP solutions, then use each of these directions as one of the starting axes for two-dimensional projection pursuits.

Acknowledgment

Helpful discussions with William H. Rogers and Gene H. Golub are gratefully acknowledged.

APPENDIX

This section presents the solid angle transform (SAT) that reversibly maps the surface of a unit sphere in an n-dimensional space, $S^n(1)$, to an (n-1)-dimensional infinite Euclidean space, $E^{n-1}(-\infty,\infty)$. This transformation is derived in Reference 11 and only the results are presented here.

Let $(X_1, X_2, \ldots, X_n)$ be the coordinates of a point lying on the surface of an n-dimensional unit sphere and $(\eta_1, \eta_2, \ldots \eta_{n-1})$ be the corresponding point in an (n-1)-dimensional unit hypercube $E^{n-1}(0,1)$. Then for n even, the transformation is

$$X_{2i} = \left[\prod_{j=1}^{i-1} \eta_{2j}^{1/(n-2j)}\right] \cos\left[\sin^{-1}\eta_{2i}^{1/(n-2i)}\right] \sin(2\pi\eta_{2i-1}) \qquad (1 \le i \le n/2-1)$$

$$X_n = \left[\prod_{j=1}^{n/2-1} \eta_{2j}^{1/(n-2j)}\right] \sin(2\pi\eta_{n-1}) .$$

$$X_{2i-1} = X_{2i} \cot(2\pi\eta_{2i-1}) \qquad (1 \le i \le n/2) ,$$

for n odd, we take

$$X_n = \left[\prod_{j=1}^{\frac{n-3}{2}} \eta_{2j}^{1/(n-2j)}\right] (2\eta_{n-2}-1)$$

$$X_{n-1} = \left[\prod_{j=1}^{\frac{n-3}{2}} \eta_{2j}^{1/(n-2j)}\right] (\eta_{n-2} - \eta_{n-2}^2)^{1/2} \sin(2\pi\eta_{n-1})$$

$$X_{2i} = \left[\prod_{j=1}^{i-1} \eta_{2j}^{1/(n-2j)}\right] \cos\left[\sin^{-1}\eta_{2i}^{1/(n-2i)}\right] \sin(2\pi\eta_{2i-1}) \qquad 1 \le i \le (n-3)/2$$

$$X_{2i-1} = X_{2i} \cot(2\pi\eta_{2i-1}) \qquad 1 \le i \le (n-1)/2 .$$

The Jacobian of this transformation,

$$J_n(1) = \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \quad ,$$

is a constant, namely the well known expression for the surface area of an n-dimensional sphere of unit radius. Adjusted by a factor of the (n-1)st root of J_n, the transformation is volume preserving, one to one, and onto. The inverse transformation can easily be obtained by solving the above equations for the η's in terms of the X-coordinates. The unit hypercube, $E^{n-1}(0,1)$, can be expanded to the infinite Euclidean space, $E^{n-1}(-\infty,\infty)$, by using standard techniques,[18] specifically multiple reflection.

REFERENCES

1. Bolshev, L.N. 1969. Bull. Internat. Stat. Inst., 43, pp 411-425.

2. Zahn, C.T., "Graph-theoretical methods for detecting and describing gestalt clusters," IEEE Trans. Computers, Vol. C-20, pp. 68-86, January 1971.

3. Shepard, R.N. and Carroll, J.D., "Parametric representation of non-linear data structures," in Multivariate Analysis, P. Krishnaiah, Ed., New York: Academic Press, 1966.

4. Sammon, Jr., J.W., "A nonlinear mapping for data structure analysis", IEEE Trans. Computers, Vol. C-18, pp. 401-409, May 1969.

5. Kruskal, J.B., "Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new 'index of condensation'," in Statistical Computation, R.C. Milton and J.A. Nelder, Ed., New York, Academic Press, 1969.

6. Kruskal, J.B., "Linear transformation of multivariate data to reveal clustering," in Multidimensional Scaling: Theory and Application in the Behavorial Sciences, Vol. 1, Theory, New York and London, Seminar Press, 1972.

7. Rosenbrock, H.H., "An automatic method for finding the greatest or least value of a function," Comp. J., Vol. 3, pp. 175-184, 1960.

8. Powell, M.J.D., "An efficient method for finding the minimum of a function of several variables without calculating derivates," Comp. J., Vol. 7, pp. 155-162, 1964.

9. Maltson, R.L. and Dammann, J.E., "A technique for determining and coding subclasses in pattern recognition problems," IBM Journal, Vol. 9, pp. 294-302, July 1965.

10. Film: "PRIM-9", produced by Stanford Linear Accelerator Center, Stanford, California, Bin 88 Productions, April 1973.

11. Friedman, J.H. and Steppel, S., "Non-linear constraint elimination in high dimensionality through reversible transformations," Stanford Linear Accelerator Center, Stanford, California, Report, SLAC-PUB-1292, August 1973.

12. Derenzo, S., "MINF-68 - A general minimizing routine," Lawrence Berkeley Laboratory, Berkeley, California, Group A Technical Report P-190, 1969.

13. Brent, R.P., "Algorithms for minimization without derivatives," Englewood Cliffs, N.H.,: Prentice-Hall, 1973.

14. Golub, G.H., "Numerical methods for solving linear least squares problems," Numer. Math., Vol. 7, pp. 206-216, 1965.

15. Fisher, R.A., "Multiple measurements in taxonomic problems," Contributions to Mathematical Statistics, New York: Wiley.

16. Sammon, Jr., J.W., "An optimal discriminant plane," IEEE Trans. Computers, Vol. C-19, pp. 826-829, September 1970.

17. Ballam, J., Chadwick, G.B., Guiragossian, Z.C.G., Johnson, W.B., Leith, D.W.G.S. and Moriyasu, K., "Van Hove analysis of the reactions $\pi^- p \rightarrow \pi^- \pi^- \pi^+ p$ and $\pi^+ p \rightarrow \pi^+ \pi^+ \pi^- p$ at 16 GeV/C, Phys. Rev., D Vol, 4, pp. 1946-1947, Oct. 1971.

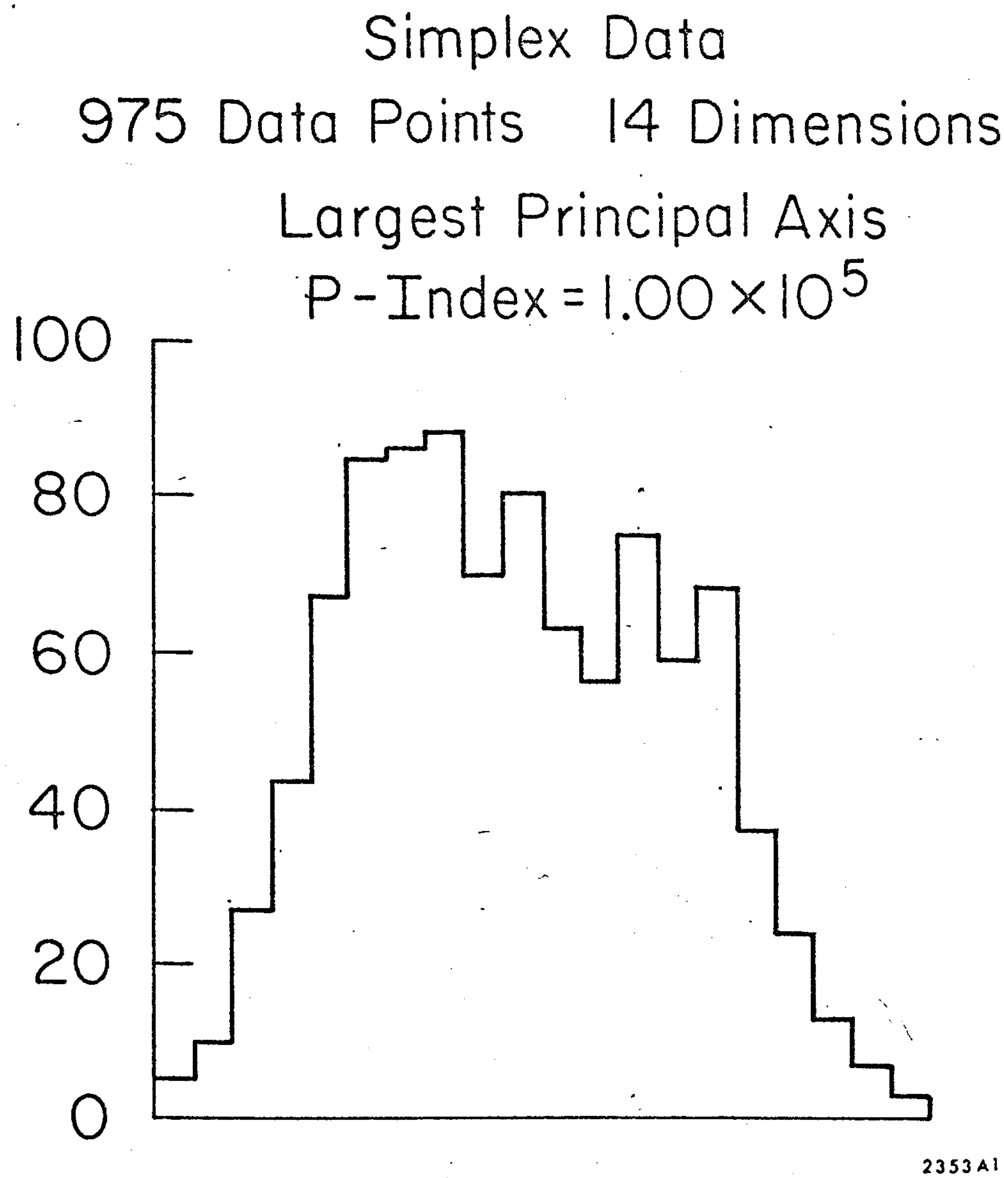
18. Box, M.J., Computer Journal 9, p. 67-77, (1966).

SPHERICALLY RANDOM DATA

14 Dimensions 975 Data Points

Search No.	P-Index Starting	P-Index Solution	Search No.	P-Index Starting	P-Index Solution
1	151.8	156.3	15	150.6	155.8
2	150.7	156.7	16	150.6	156.9
3	149.4	156.0	17	151.0	157.0
4	152.4	159.7	18	152.2	155.6
5	152.3	159.2	19	152.1	160.2
6	151.0	154.4	20	152.9	159.0
7	152.6	155.6	21	150.0	160.8
8	149.6	155.4	22	150.0	157.9
9	150.0	156.1	23	150.7	155.4
10	151.8	153.0	24	151.5	158.4
11	154.0	154.6	25	151.0	155.3
12	151.4	156.7	26	152.8	159.6
13	149.7	154.1	27	151.8	157.0
14	152.1	154.2	28	152.1	157.9

TABLE 1



Simplex Data
975 Data Points 14 Dimensions
Largest Principal Axis
P - Index = 1.00 × 10^5
100
80
60
40
20
0
2353A1

FIGURE 1a

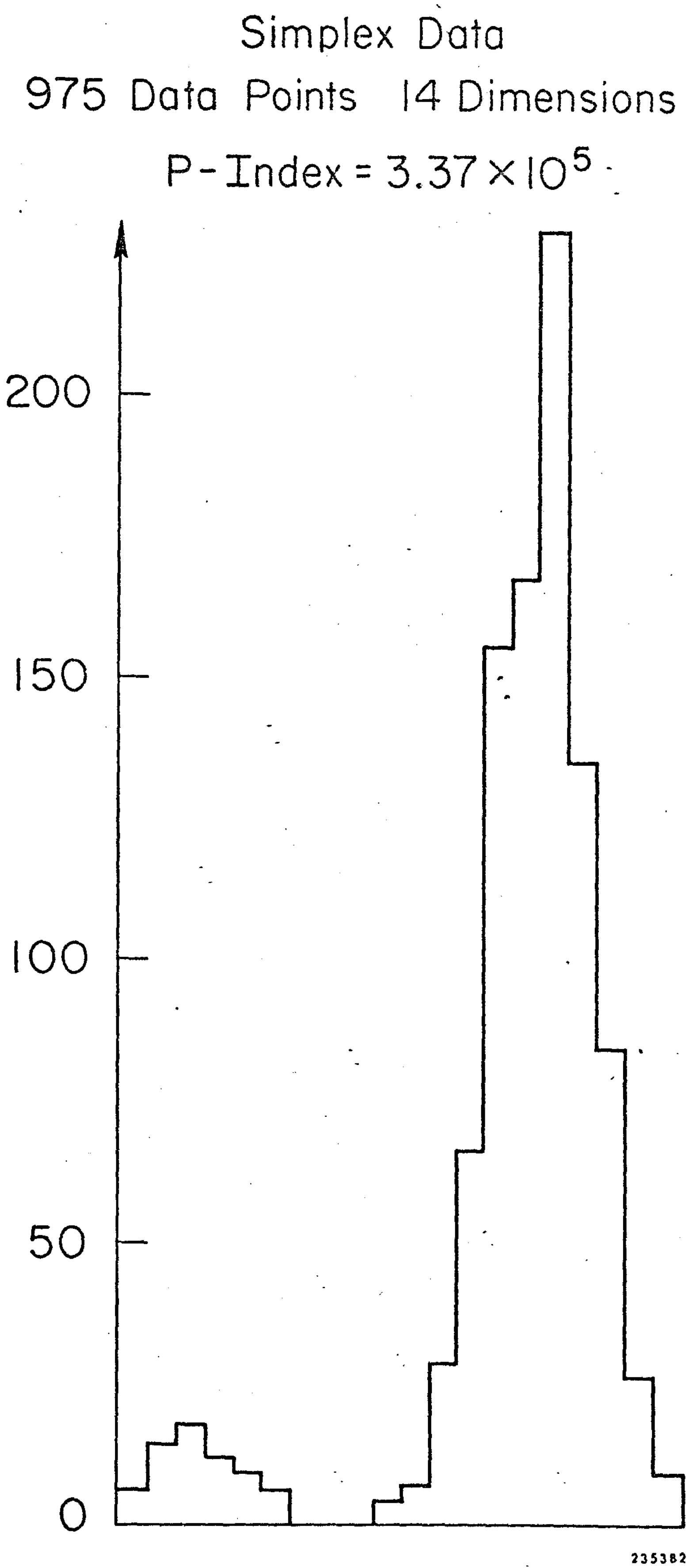

Simplex Data
975 Data Points 14 Dimensions
P-Index = 3.37 × 10^5
200
150
100
50
0
235382

FIGURE 1b

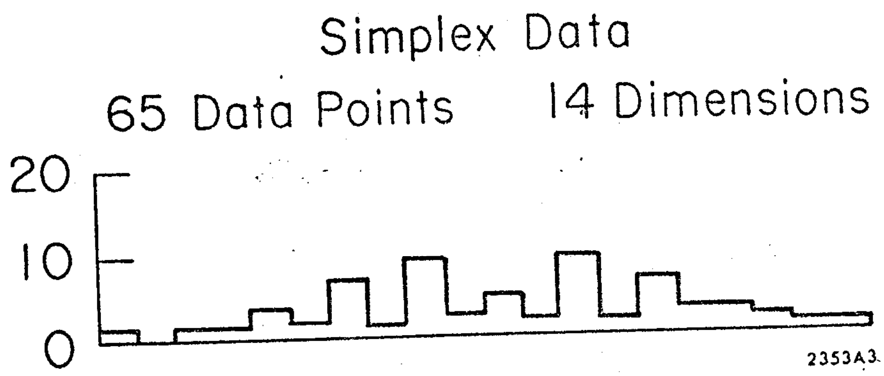


FIGURE 1c

Simplex Data

910 Data Points 14 Dimensions

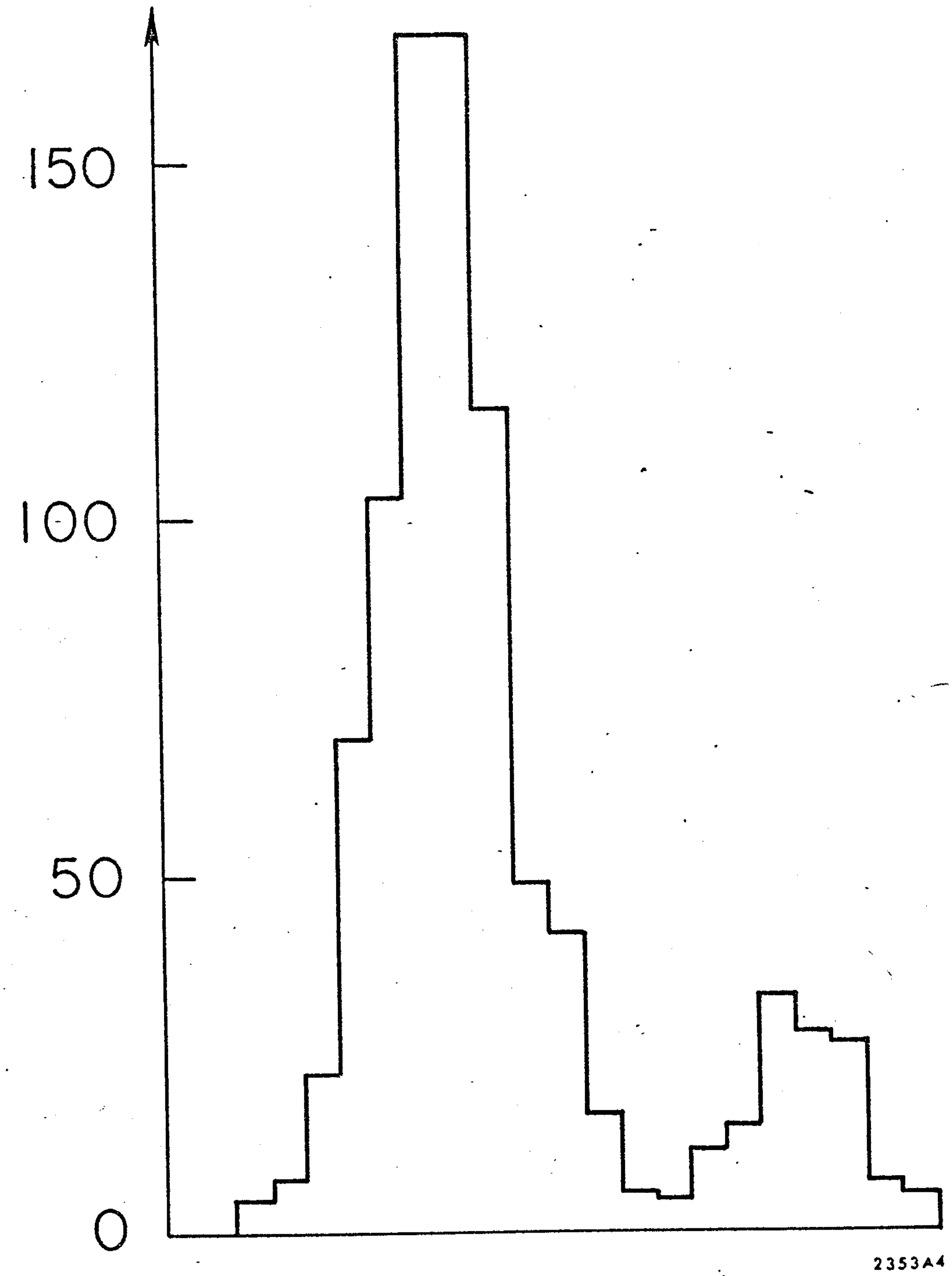


FIGURE 1d

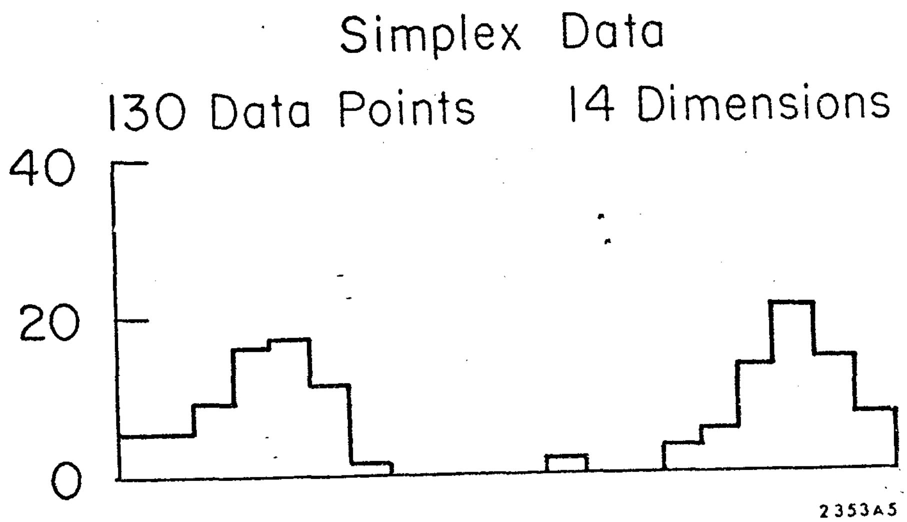


FIGURE 1e

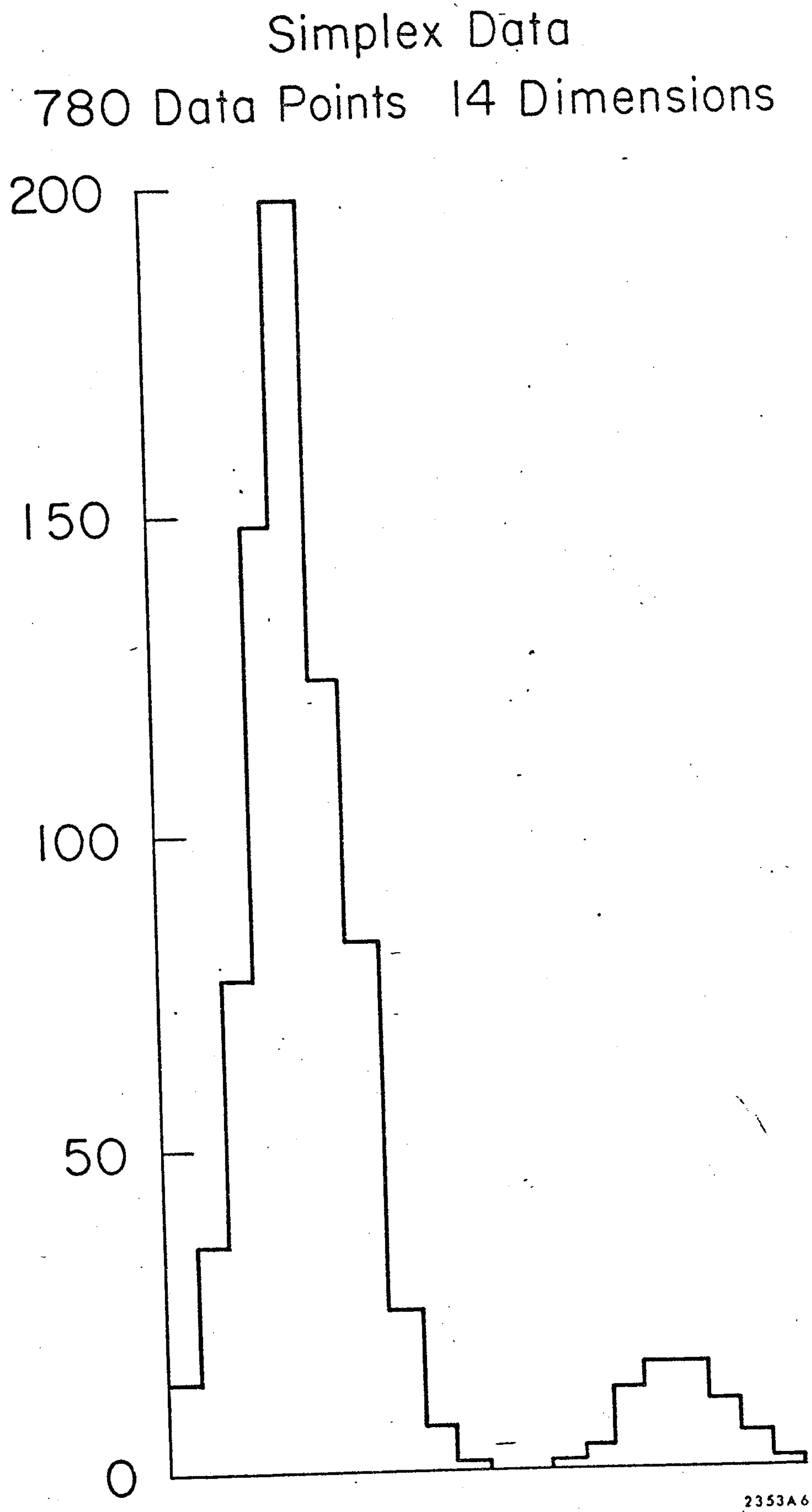

Simplex Data
780 Data Points 14 Dimensions
200
150
100
50
0
2353A6

FIGURE 1f

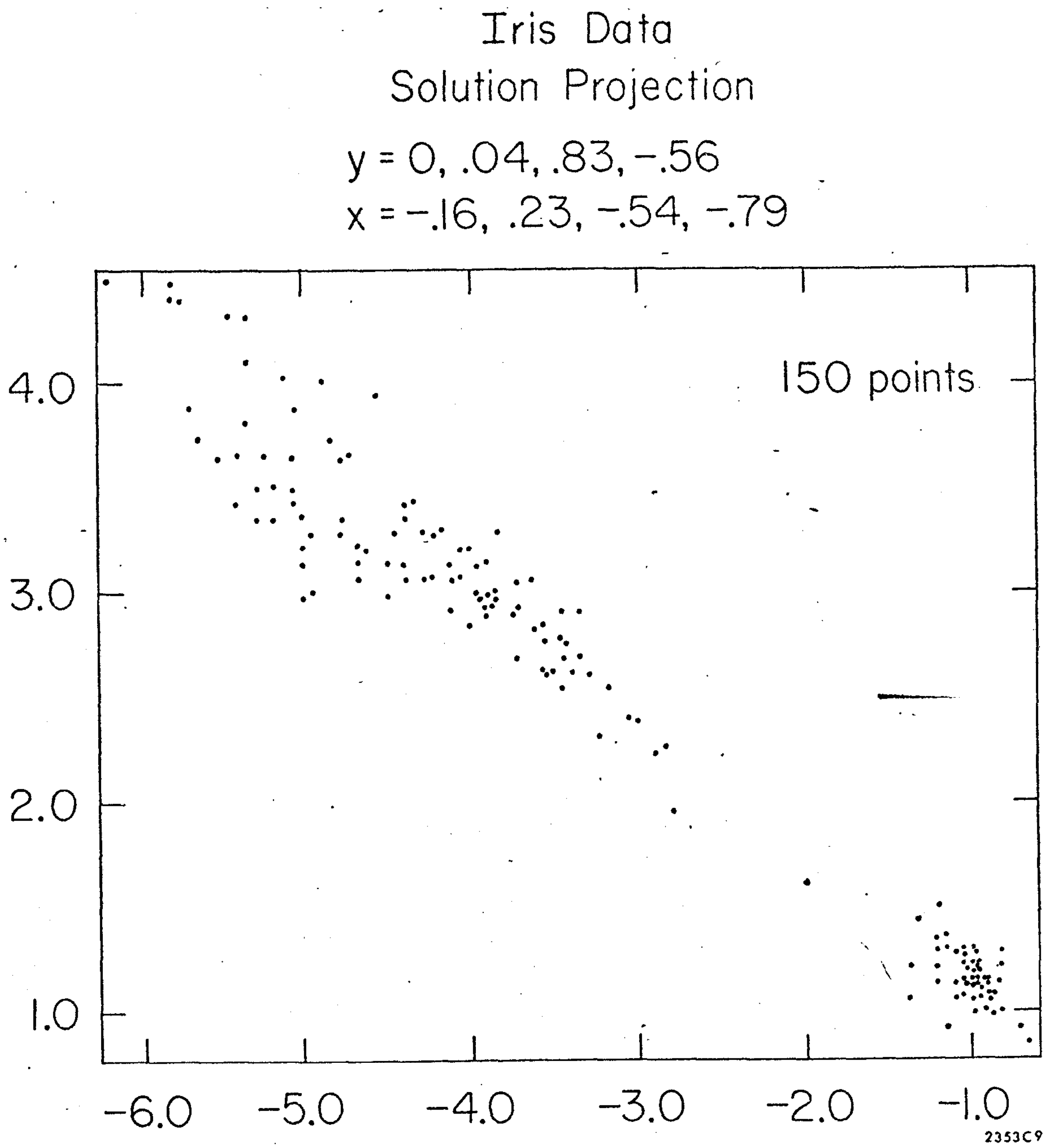

Iris Data
Solution Projection
y = 0, .04, .83, −.56
x = −.16, .23, −.54, −.79
150 points
4.0
3.0
2.0
1.0
−6.0
−5.0
−4.0
−3.0
−2.0
−1.0
2353C9

FIGURE 2a

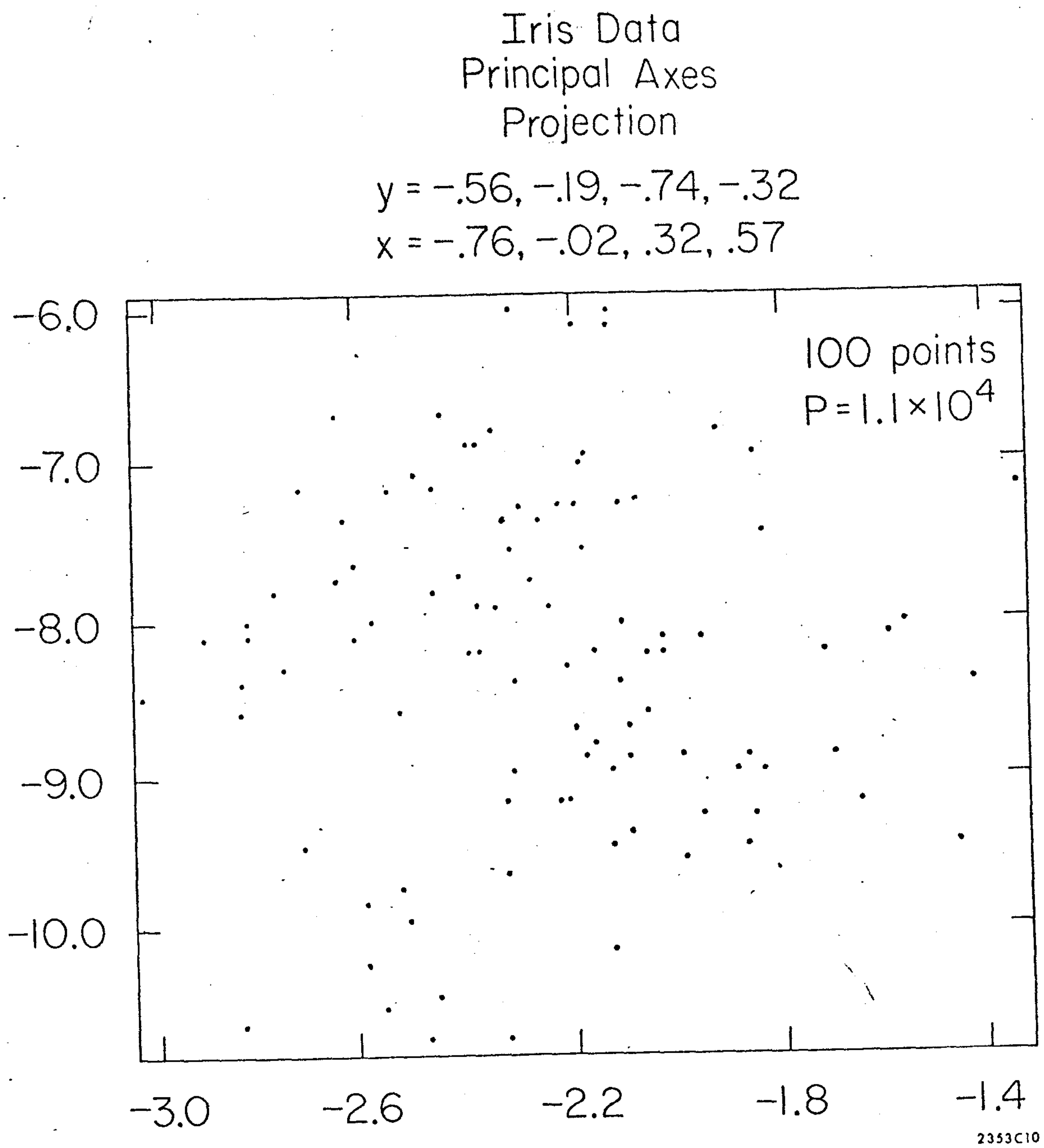
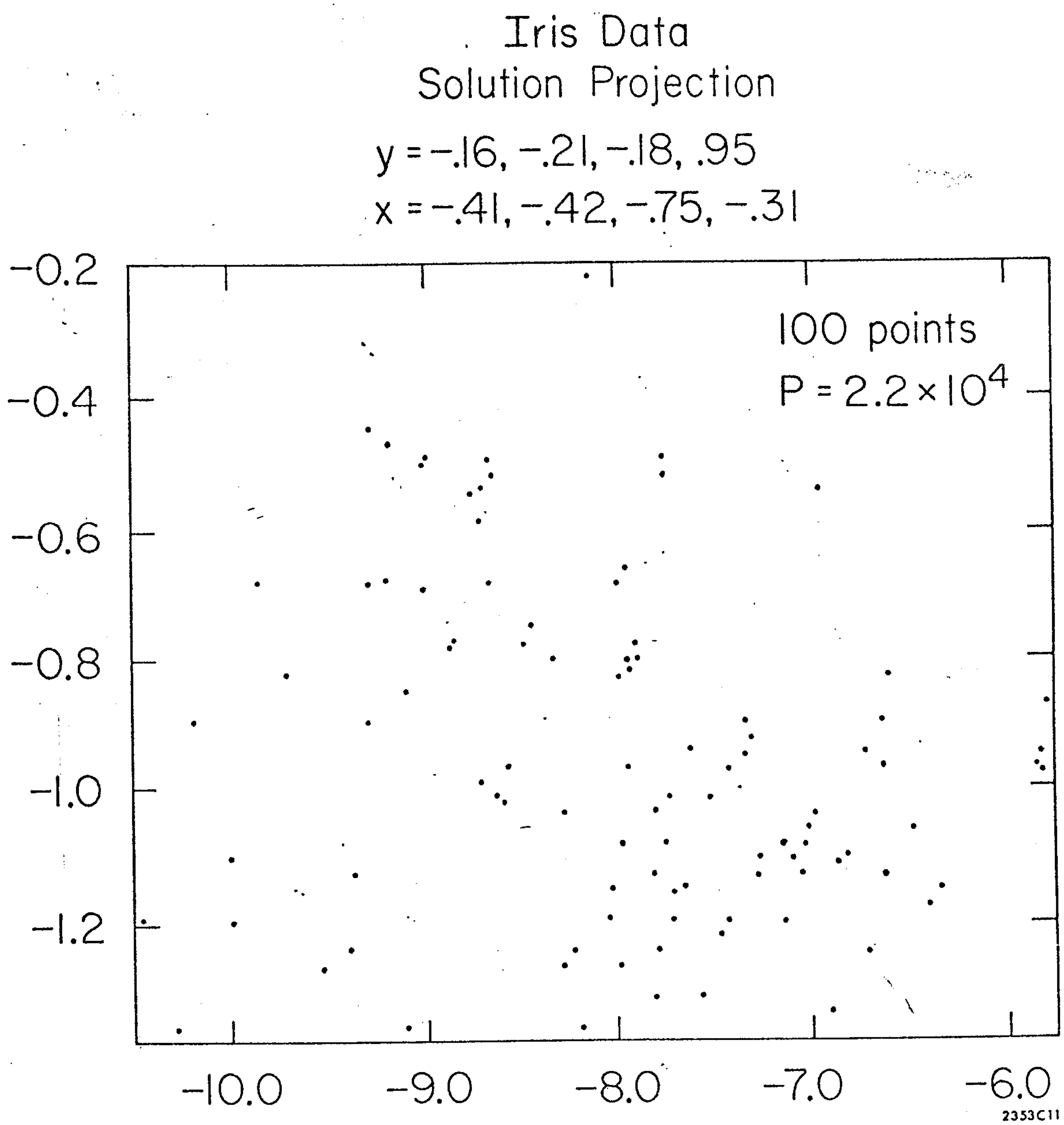
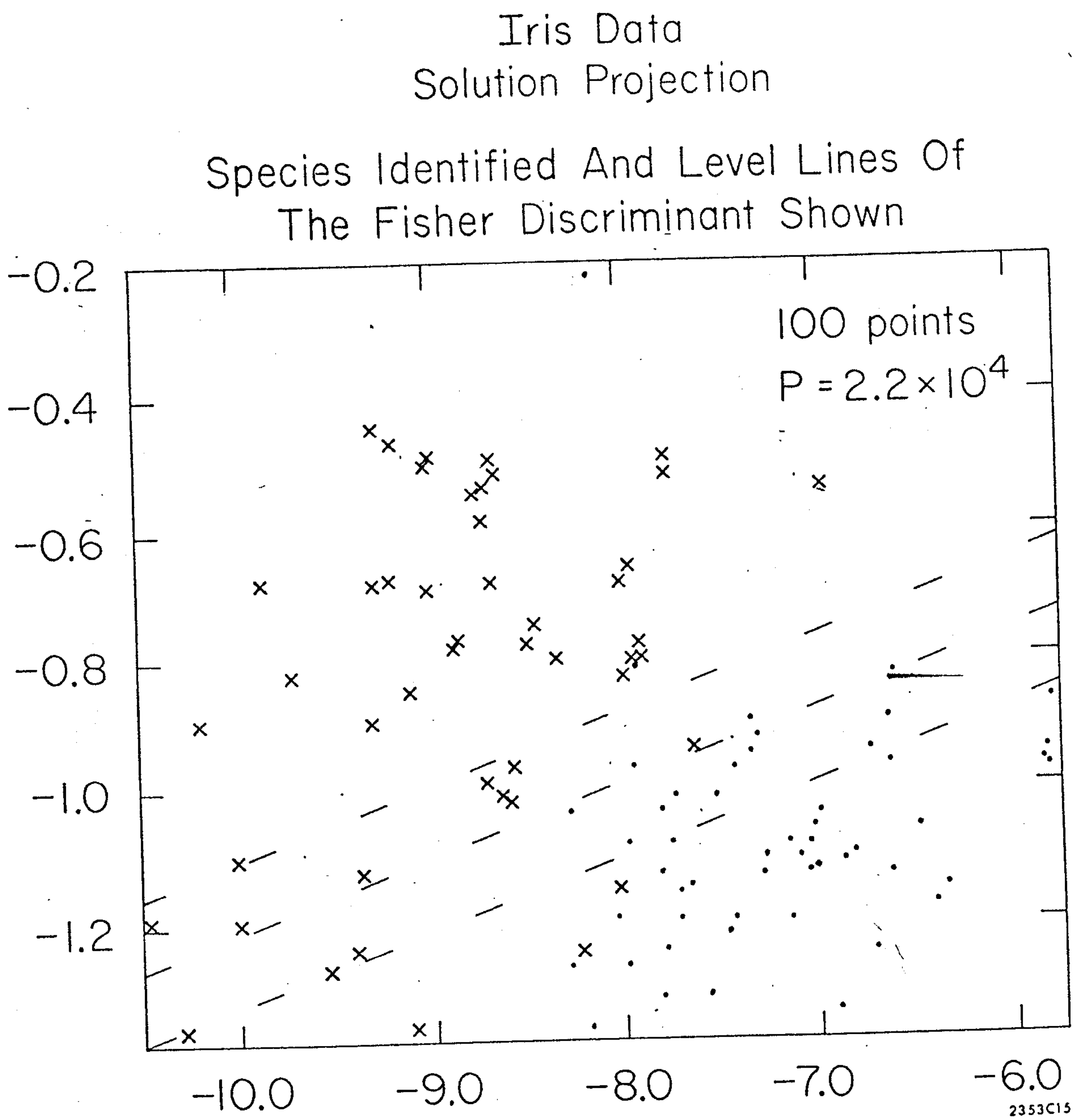

Iris Data
Principal Axes
Projection
y = −.56, −.19, −.74, −.32
x = −.76, −.02, .32, .57
100 points
P = 1.1×10^4
−6.0
−7.0
−8.0
−9.0
−10.0
−3.0
−2.6
−2.2
−1.8
−1.4
2353C10

FIGURE 2b



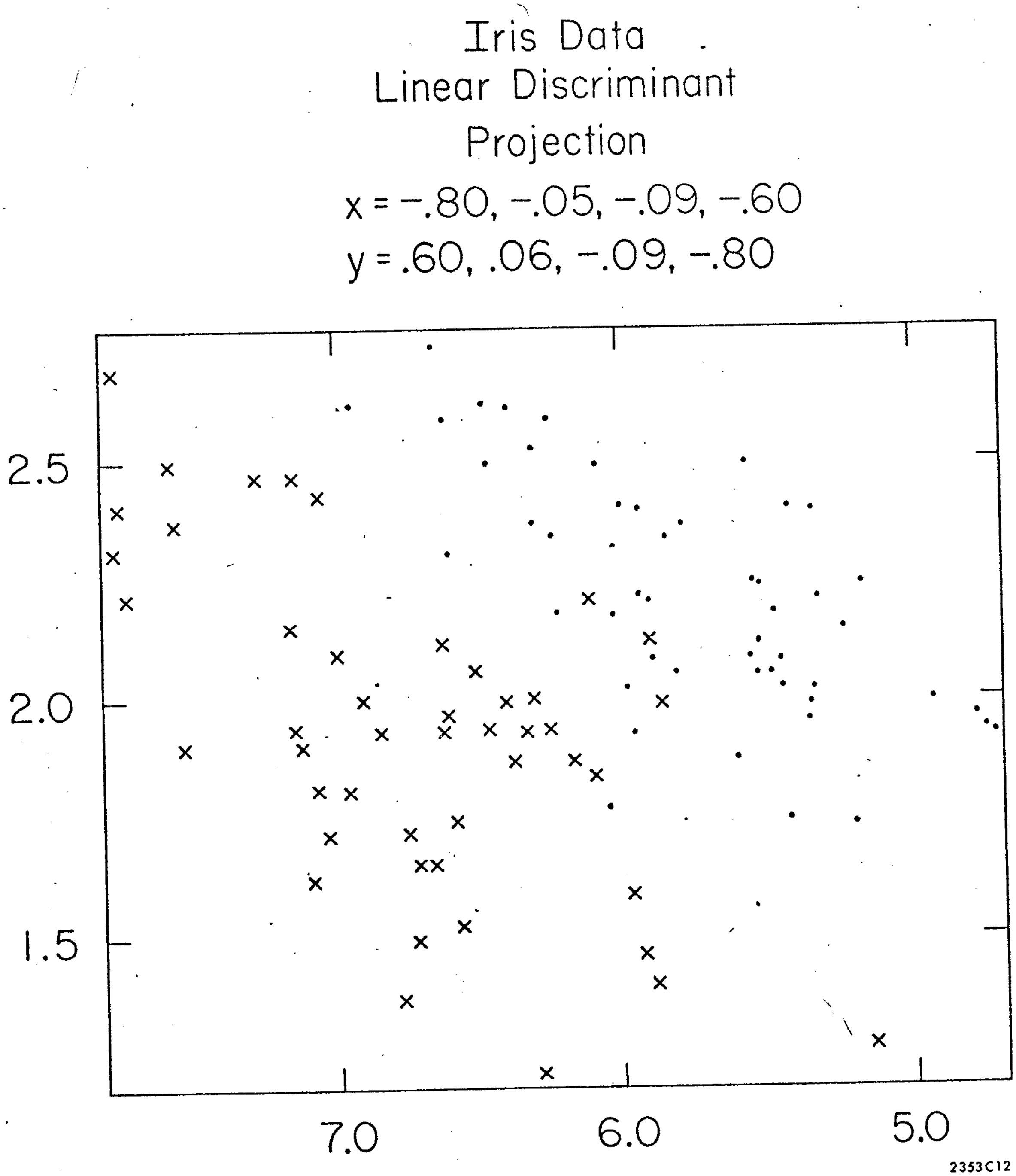
Iris Data
Solution Projection
y = -.16, -.21, -.18, .95
x = -.41, -.42, -.75, -.31
100 points
P = 2.2×10^4
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
-10.0
-9.0
-8.0
-7.0
-6.0
2353C11

FIGURE 2c



Iris Data
Solution Projection
Species Identified And Level Lines Of
The Fisher Discriminant Shown
100 points
P = 2.2×10^4
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
-10.0
-9.0
-8.0
-7.0
-6.0
2353C15

FIGURE 2d



Iris Data
Linear Discriminant
Projection
x = −.80, −.05, −.09, −.60
y = .60, .06, −.09, −.80
2.5
2.0
1.5
7.0
6.0
5.0
2353C12

FIGURE 2e

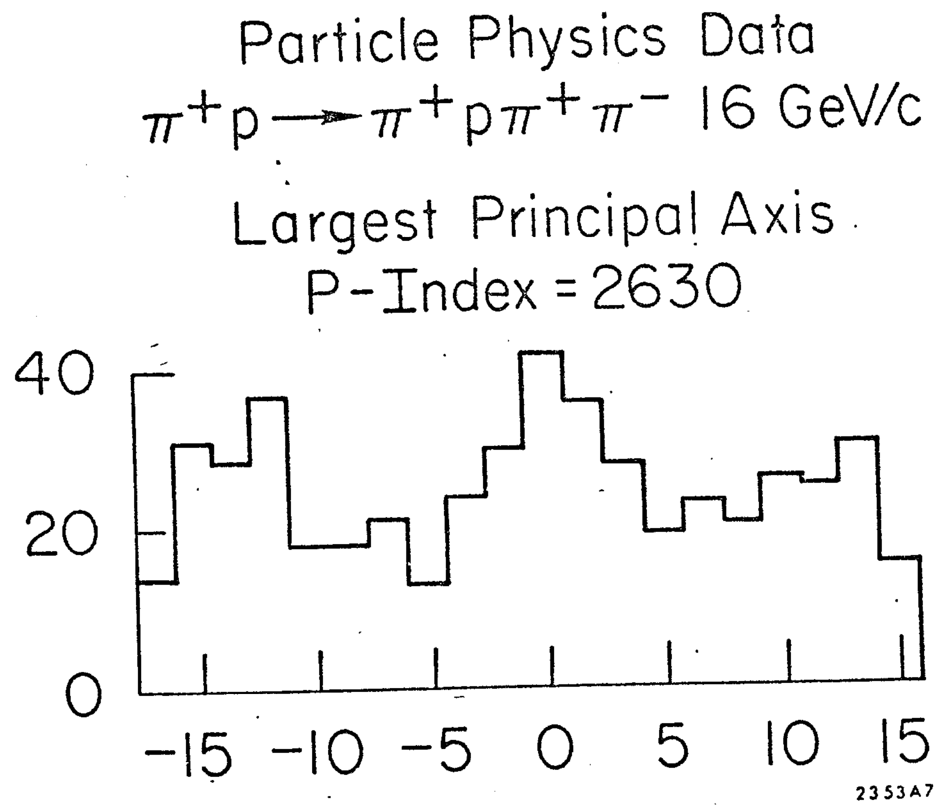

Particle Physics Data
$\pi^+ p \longrightarrow \pi^+ p \pi^+ \pi^-$ 16 GeV/c
Largest Principal Axis
P-Index = 2630
40
20
0
−15 −10 −5 0 5 10 15
2353A7

FIGURE 3a

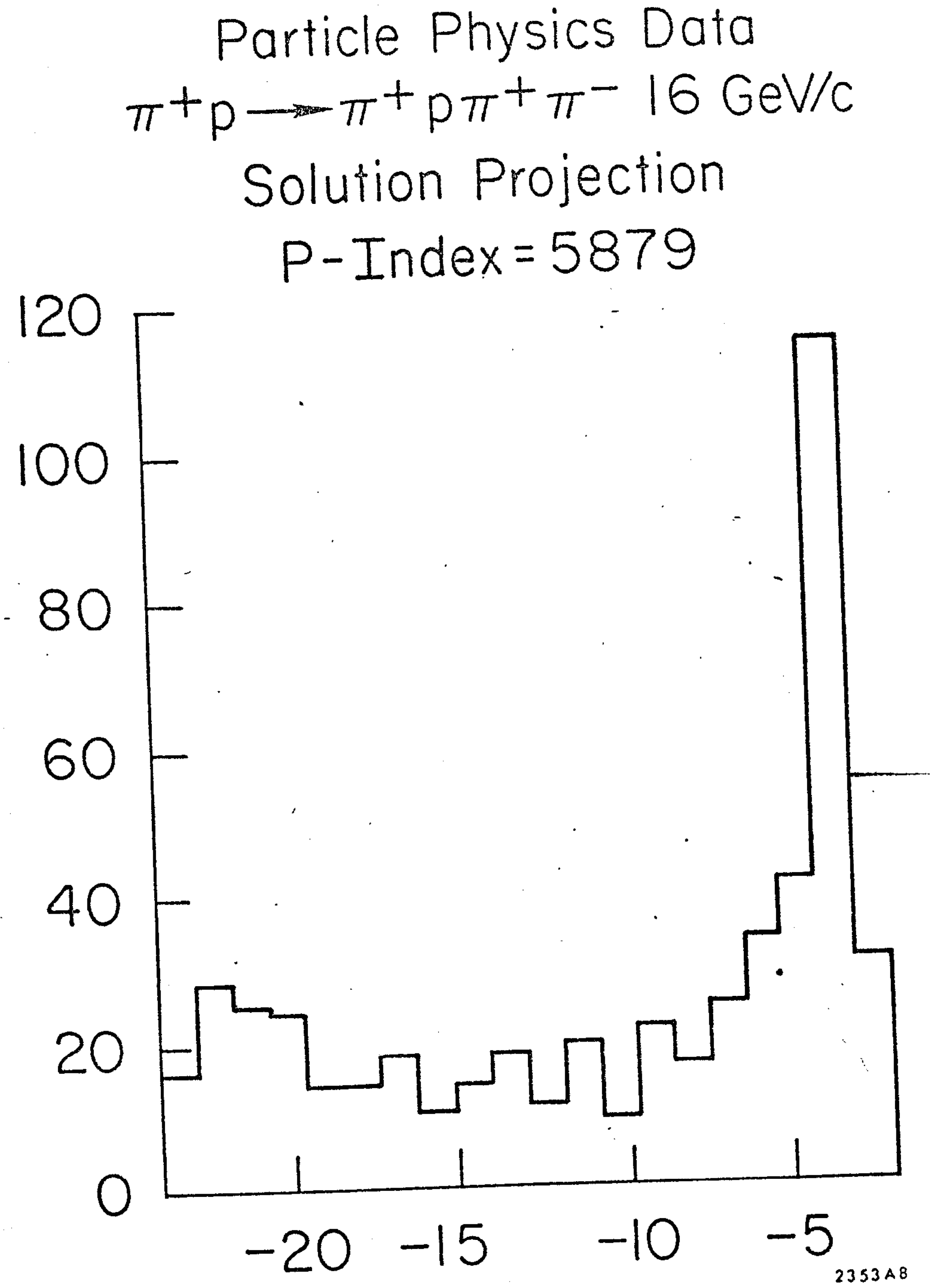

Particle Physics Data
$\pi^+ p \longrightarrow \pi^+ p \pi^+ \pi^-$ 16 GeV/c
Solution Projection
P-Index = 5879
120
100
80
60
40
20
0
−20
−15
−10
−5
2353A8

FIGURE 3b

Particle Physics Data

$\pi^+ p \longrightarrow \pi^+ p \pi^+ \pi^-$ 16 GeV/c

Largest Principal Axes

Projection

P-Index = 45.3

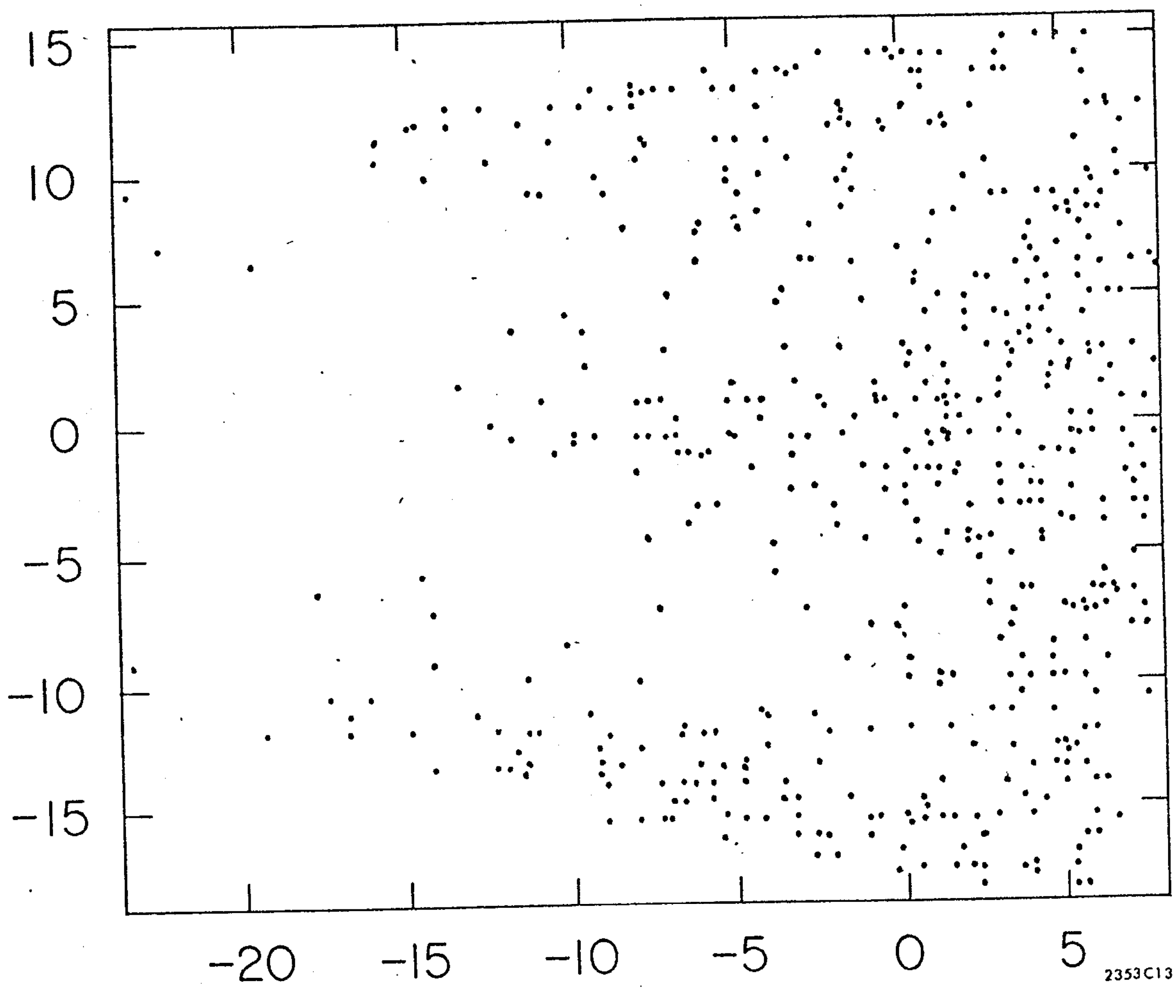


FIGURE 3c

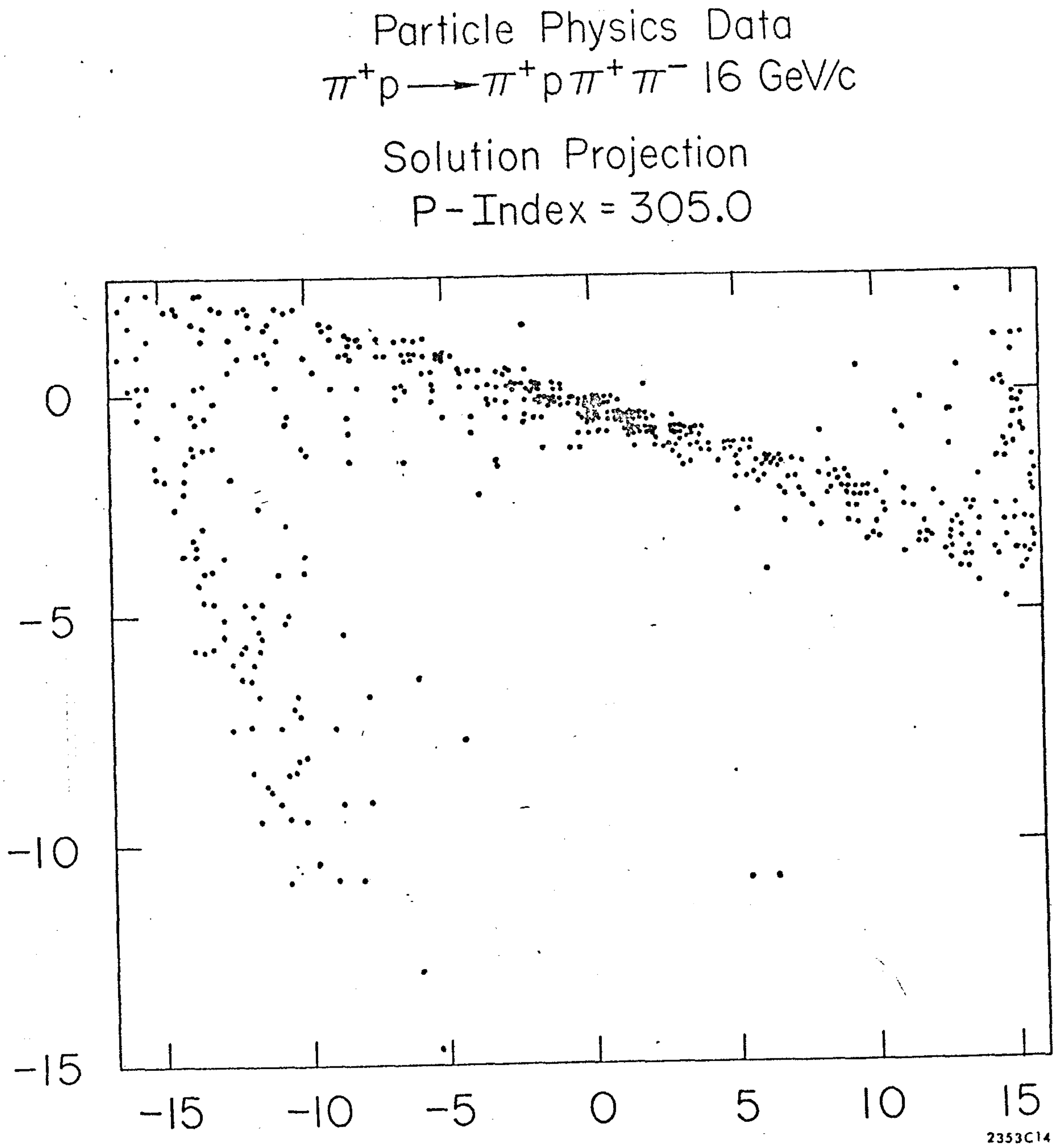

Particle Physics Data
$\pi^+ p \longrightarrow \pi^+ p \pi^+ \pi^-$ 16 GeV/c
Solution Projection
P-Index = 305.0
0
-5
-10
-15
-15
-10
-5
0
5
10
15
2353C14

FIGURE 3d