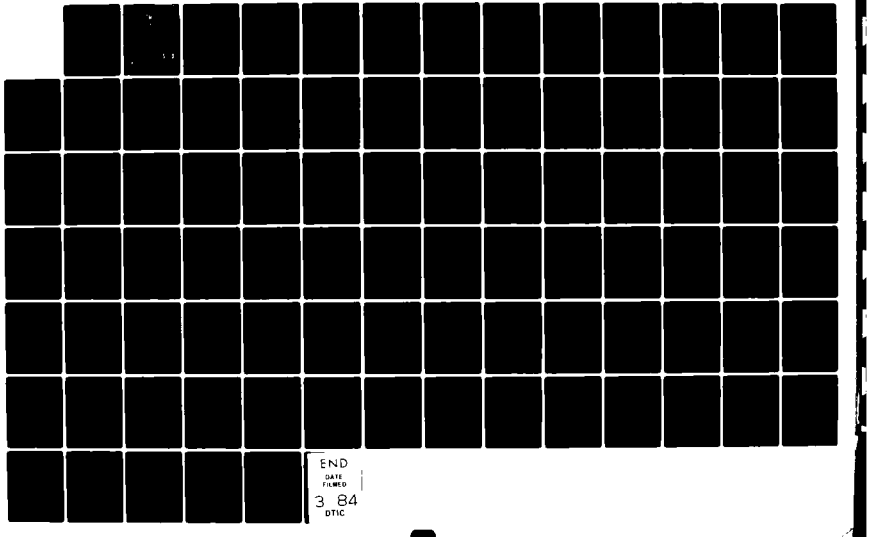
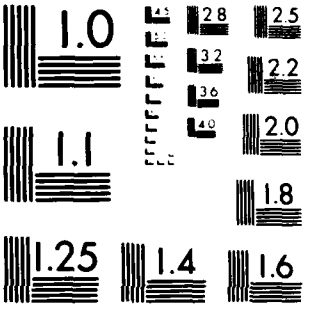
AD-A138 194 GRID-BASED LINE DRAWING QUANTIZATION(U) AIR FORCE INST 1/1
OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
E A THOMPSON DEC 83 AFIT/GCS/EE/83D-20

UNCLASSIFIED F/G 9/2 NL





DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/EE/83D-20

GRID BASED LINE DRAWING QUANTIZATION

THESIS

AFIT/GCS/EE/83D-20 Edward A. Thompson
Capt. USAF

Approved for public release; distribution unlimited.

AFIT/GCS/EE/83D-20

GRID BASED LINE DRAWING QUANTIZATION

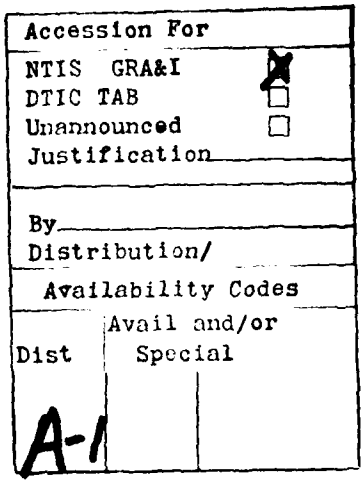
THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Edward A. Thompson, B.S.
Capt USAF
Graduate Computer Science
December 1983

Acknowledgments

I would like to thank my advisor, Major Ken Castor of the Air Force Institute of Technology, for proposing this thesis topic. I am grateful for his consistent support throughout the project. I would also like to thank Dr David Barr for his help with the math, and my wife and friends for their constant encouragement.



Contents

List of Figures

List of Tables

Abstract

This paper documents a quantitative analysis of the performance of the generalized chain codes when used to quantize waveforms of specific degrees of curvature. The analysis was performed using a software simulation of the chain coding system on periodic circular and sinusoid waves. The performance of the codes was measured in terms of the number of encoded bits per length of the function and the area error per length.

The analytic measure of the curvature of the circular and sine wave was developed so that the performance of the codes could be evaluated as a function of the curvature of the waveform being quantized. Comparisons on the circular wave provided a baseline of a constant curvature function. The performance of the coding system was then compared to the results when sine functions of similar curvature were encoded to determine if curvature is a valid predictor of the performance of the chain code system.

I. Introduction

The recent trend of modeling images for computer processing or storage has been toward discrete quantization. The classical two dimensional sampling, however, is inefficient for virtually all types of images due to the inherent duplication of patterns in the image field. Several methods have been examined in attempts to utilize the basic properties of images as a guide to the development of an efficient data representation model.

Saghri and Freeman [3,13] have reported on a family of encoding schemes for use on a specific type of image known as a line drawing. A line drawing is an image consisting of lines of a single intensity on a contrasting background. (Contour maps, graphs, and even printed text are examples of line drawings.) These coding schemes are collectively referred to as Grid Based Line Drawing Quantization Methods, and take advantage of line segments in the image. The performance of these codes is apparently affected by the shape or content of the image being quantized.

There are several measures which have been used to characterize the performance of these codes. In a 1982 AFIT thesis, Lt Keith R. Jones developed a computer program based

on the grid based line drawing quantization method to empirically examine the performance of the chain codes when used to quantize periodic circular and sinusoidal wave functions [8].

Lt Jones measured the area error per unit length of the function to quantify the accuracy and precision of the code. He used the average number of quantized bits per unit length to measure the efficiency of the codes. This thesis will continue the examination of the grid based line drawing quantization method. My objective is to investigate the degree of correlation between the curvature of a line drawing and the performance of the quantization, particularly in relation to the grid size used.

This investigation will be primarily concerned with the results obtained by quantizing circular and sine waves. These two families of functions were chosen because the circular wave has a constant curvature and the sine wave has a continuously variable curvature. Also, these functions can be simulated in a straightforward manner by the computer.

Although the analytical development of this thesis includes a phase term, only sine and circular waves of zero initial phase will be examined. Jones has shown that different initial phases affect the quantization system very little, at least for periodic sinusoidal and circular waves.

In chapter two of this thesis, I will introduce the algorithms used for quantization. (For a more complete discussion of the family of grid based quantization schemes,

the reader may wish to study reference 10.) Chapter three discusses the mathematic aspects of curvature and how they relate to the particular functions examined in this thesis.

Chapter four discusses the implementation details of Lt Jones' FORTRAN program on the AFIT VAX 11/780. The particular changes involved to convert the routines originally produced on the CDC Cyber and their effects are reviewed.

Chapter five is the analysis of the coding scheme when used on functions of constant curvature (i.e. periodic circular waves). It will introduce new statistics such as the ratio of grid size to curvature and how these can be used to measure performance. Specific conclusions with respect to circular functions will be presented.

Chapter six analyzes the coding scheme's performance when used to encode sinusoidal waves. The specific observations of these results will be reported. Chapter seven will report the general conclusions of this thesis. Based on the experience with functions of constant and varying curvature, the results will be generalized to show how well the curvature of a function can be used to predict the performance of the quantization system on that function. Recommendations for further study will be made based on these conclusions.

II. Background of the Chain Codes

This chapter will introduce the basic concepts of the chain codes which are examined in this thesis. Certain results and recommendations of a previous AFIT thesis by Lt Keith R. Jones will be reviewed [8]. A number of different coding algorithms will be described and the performance criteria used to evaluate these codes will be reviewed.

The Generalized Chain Codes

The chain codes are used to quantize a function or line drawing by superimposing a rectangular cartesian grid on the function and identifying node points "near" the intersection of the function and the grid lines. The sequence of these node points is used as a model of the original analog function, image or line drawing. There are many possible ways of encoding these sequences of intersection points. The simplest of these will be discussed, followed by a generalization to the more complex coding schemes.

Grid Intersections

The points at which the horizontal and vertical lines of the grid intersect are called nodes. It is these nodes

which are assigned the discrete values used for quantization. The simplest quantization scheme, called the grid intersect code, merely specifies the nodes closest (in terms of physical distance; the midpoint can be arbitrarily assigned to either side) to each intersection of the drawing and the grid lines. The node points are specified sequentially as the drawing or segments of the drawing are traversed from beginning to end. Since each node is adjacent to eight other nodes (the use of other than these eight nodes has been researched by Rosen [12]), three bits are used to identify each node. (See Figure II-1.)

A function or line drawing is quantized by starting at one end of the line and tracing the line through the grid. The values of successive nodes are recorded as a sequential chain since each value represents the position of a point in relation to the previous point. Thus for a given point, the next point must be one of the eight immediately adjacent nodes.

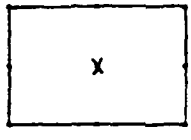


Figure II-1. Possible next nodes

Figure II-2 shows a straightforward encoding scheme for these nodes, as introduced by Freeman [3]. The specific

ordering of these nodes is arbitrary.

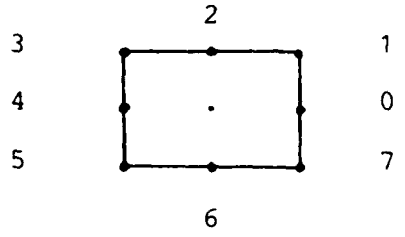


Figure II-2. Encoded Values of Grid Intersect Code

Using this encoding scheme, the encoded representation of the line drawing in Figure II-3 would be:

start - 001 000 000 111 111 001 001

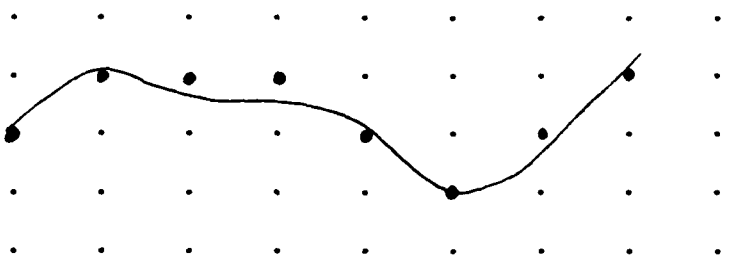


Figure II-3. An Encoded Example

A variation of the grid intersect code was introduced by Saghri [13] and referred to as the (1) code. The (1) code selects for quantization only the node closest to the intersection of the line drawing and the ring boundary (the grid intersect code selects all the grid intersect points). Thus for certain line drawing segments such as in Figure II-

4, the (1) code would not include certain nodes which are in the grid intersect quantization.

According to Freeman [5], since the set of nodes selected using the (1) code is a subset of the set selected by the generalized grid intersect code, the (1) code will perform at least as well (generally better) as the grid intersect code in terms of the total number of bits used for quantization. The fewer nodes identified in the quantization may however mean that the area error of the quantization will be greater for the (1) code.

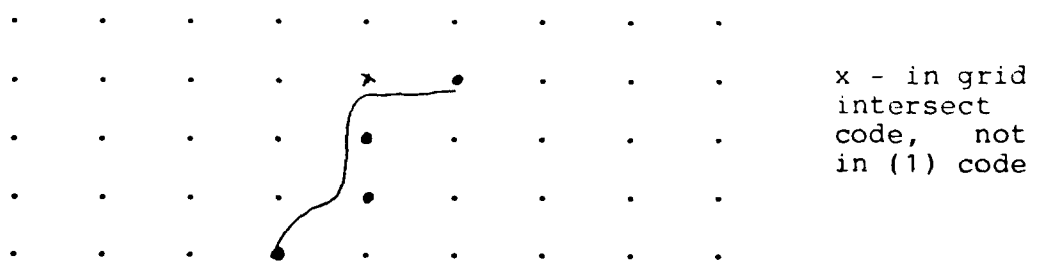


Figure II-4. Example Encoding Using Grid Intersects

The precision and accuracy of both codes are highly dependent on the grid size and the characteristics of the particular function being quantized. If a function rapidly changes direction within a grid square, this change may not be encoded and may be lost. Curvature is one measure of how rapidly a function changes direction which can be readily calculated for circular and sinusoidal waves. By calculating curvature, we can determine whether a function

can arc within a given grid size. As shown in Figure II-5, drawings with large curvature can be lost if the grid size is too small. Because of this, only grid sizes of at least twice the minimum curvature will be used in the quantization experiments. The primary emphasis of this thesis will be to examine this relationship between grid size versus the curvature of a function and how well the encoding system quantizes functions with various degrees of curvature. (Chapter three will discuss curvature in more detail.)

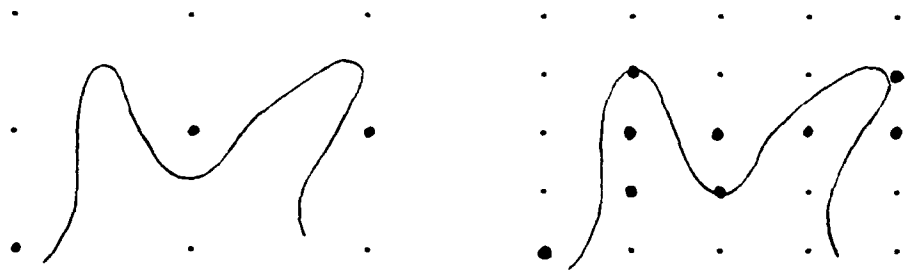


Figure II-5. Combined Effects of Grid Size and Curvature.

Higher Order Codes

The chain coding scheme can be generalized to include the set of nodes other than those immediately adjacent to the present node. Each set of boundary nodes at equal grid distance from the current node is called a ring. Ring 1 includes the nodes immediately adjacent to the current point, ring 2 is the next outer ring and so on as shown in Figure II-6. For example, a (1,2,3) code uses rings 1, 2, and 3 for encoding. A particular coding system may use any number of these rings, but it must be consistent for proper

encoding and decoding.

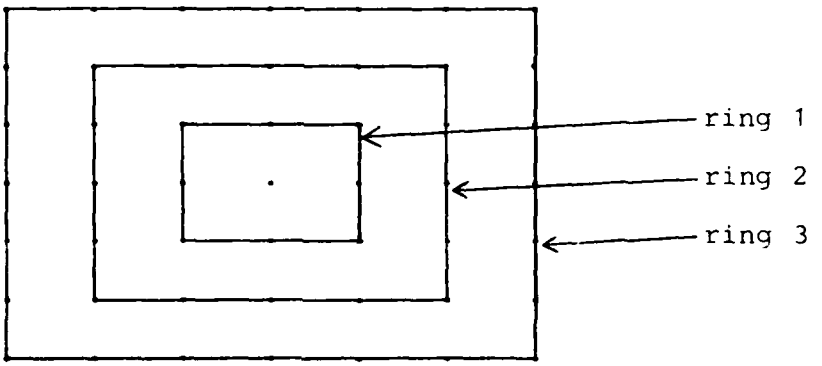


Figure II-6. Encoding Rings

The higher order codes take advantage of trends in lines by utilizing the largest ring possible to quantize the next node while maintaining a given error performance. The choice of which ring to use also involves examining each ring and utilizing the outermost ring which meets the minimum performance requirement. If the outermost ring fails the error test, the next inner ring is checked and so on until the innermost ring is used. The higher order codes are loosely adaptive therefore, in that they utilize an optimum coding sequence subject to the area of precision constraint.

There are several tests which can be used to determine the optimum ring level, such as maximum deviation, or total area error. Lt Jones used an area of precision rule. As shown in Figure II-7, the area of precision is the

enclosed region bounded by the outer ring line segment within plus and minus one-half grid length on either side of the proposed node and the line segments connecting these points to the current node.

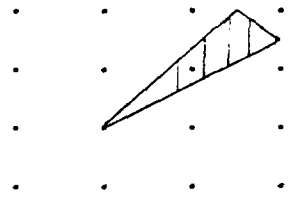
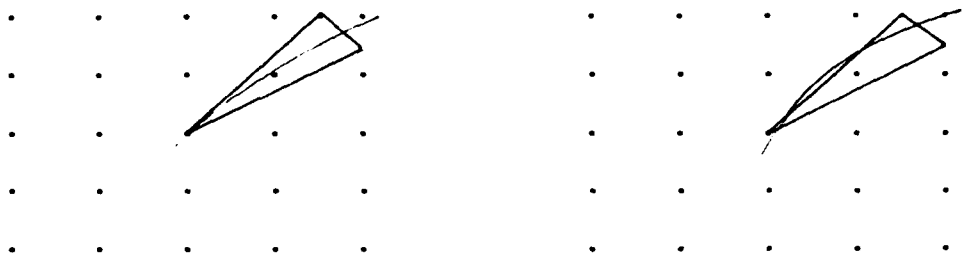


Figure II-7. Area of Precision

The use of this particular area of precision rule to determine the proper ring level has also been referred to as triangular quantization in reports by other researchers [13]. Figure II-8 shows how the area of precision applies to two line drawings. Since the drawing on the left passes outside the specified triangle, the outside ring fails the test and will not be used. The entire drawing segment under consideration on the right lies within the triangle, and the third ring will be used.



passes fails

Figure II-8. Examples of Area of Precision Rule

If all the line drawing-grid intersections are within the area of precision, the proposed ring level and node are used for the next encoding point. If not, the next inner ring is evaluated with a similar area of precision; the process continues until a node passes the test or the innermost ring is reached. The maximum performance in terms of minimum number of bits required to encode the line is obtained by choosing the outermost ring which passes the area of precision rule.

Performance Standards

Other research on chain codes has produced families of general chain codes with good and poor performance [2,3,5,12]. At this time, no adequate mathematical models have been reported which can quantitatively predict the performance of the chain codes. The lack of such a model has required that the performance of these codes be established by computer simulations.

Evaluating the General Chain Codes

Five criteria for comparison of general chain codes have been suggested [3]: (1) compactness, (2) precision, (3) smoothness, (4) ease of encoding and decoding, and (5) facility for processing. The relative importance of these criteria is somewhat subjective and is highly dependent on the particular application. An increase in performance of one criteria is usually obtained at the expense of another.

The compactness of a code refers to how many bits are required to store or transmit a given picture. A higher order code may result in fewer bits for one picture if it includes relatively straight lines, or the computation overhead may become excessive if too many rings are evaluated. This evaluation also requires some additional amount of memory for node comparisons. A lower order code will require the quantization of at least as many nodes as a higher order code.

Precision is a measure of how well the coding scheme correctly quantizes small changes of a function. In applications where data are displayed, the smoothness of the quantization may be an important factor. Display smoothness often affects recognition of certain patterns in visible data [7,8].

The major tradeoffs of these criteria are compactness versus precision and speed versus ease of encoding and decoding. The latter might influence hardware/software design decisions of an entire system.

Performance Criteria Used in this Thesis

Lt Jones used two metrics to quantify the performance of the codes. These same metrics will be used to compare the performance of the codes produced in this thesis. The efficiency or compactness of the code is measured as the number of bits per unit length. The precision of the code is computed as the area error per unit length.

The compactness or bits per unit length is computed by dividing the number of bits required to encode the line up to a given point by the current length of the code at a given point. Similarly, the precision is computed as the accumulated area error of the quantization divided by the current length. The quantization software prints out the value of each of these statistics at intervals equal to 10 grid lengths.

The accuracy of the code measured how well the code represented the original line drawing. The quantization error was obtained by integrating the area between the original line and the quantized line. Lt Jones' results indicate that increased performance in terms of area error requires more bits for encoding (where differences were significant).

This thesis investigates the nature of the relationship between the curvature of a line drawing, the grid size, and the area error of quantization. The primary area of concern is the evaluation of the performance of the chain code with

respect to the compactness and precision of the codes.

Summary

This chapter has reviewed the basic concepts of the chain code system, the results of previous research in this area, and the evaluation criteria generally used to measure the performance of the codes. In the next chapter, the curvature of a function will be introduced as an additional parameter to be used to quantitatively measure the performance of the codes.

III. Curvature

This chapter will introduce the basic mathematic concepts of the curvature of a function. It was stated in chapter two that functions which rapidly change direction within a quantization grid square may not be quantized well by the grid based line drawing quantization system. This chapter will introduce curvature as a measure of how quickly such a function or drawing changes direction. Curvature will provide an objective, tractable parameter against which the chain codes will be evaluated. After reviewing the general background and definitions of curvature, the basic concepts will be extended to show how the curvature of periodic circular and sinusoidal waves are evaluated in this thesis.

Basic Curvature

Curvature is a measure of how rapidly a function changes direction, or how sharply it turns. The analytic definition is the rate of change of the tangent vector of the function [14:585]. Since the length of the tangent vector is always one, curvature measures the change of

direction of the tangent vector. The physical interpretation of curvature agrees closely with this analytic definition. The tangent vector in Figure III-1 changes rapidly around point A, but changes slowly between points B and C. Note that the derivations in this chapter will only deal with two dimensional drawings and functions. The same concepts can be generalized to higher dimensions with similar results.

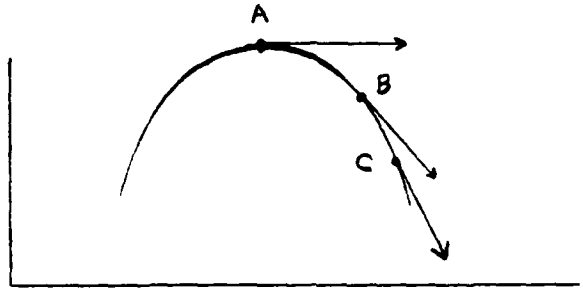


Figure III-1. Tangent Vector

The mathematical definition of curvature is:

$$k = [\, d\theta / ds \,]$$

where theta is the angle to the tangent vector and s is the arc length (See Figure III-2). (The braces [] are used to indicate the absolute value.) Theta changes from point to point along the line drawing; its sign is taken to be the same as the sign of ds/dx. Thus k is positive if the concave side of the curve is on the left and negative if the curve is concave to the right (left and right are from the perspective of tracing the line from beginning to end). For a straight line, the angle theta is constant and thus

$d\theta/ds$ is equal to zero.

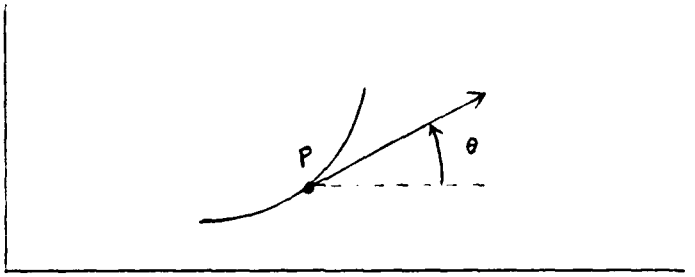


Figure III-2. Tangent vector at point P

A circle of curvature (see Figure III-3) is the unique circle whose center is on the concave side of a function and is tangent to the function at a particular point P. The radius of this circle can also be used as a measure of the curvature of a function and is equal to 1/[k]. (This result will be derived in the next section.)

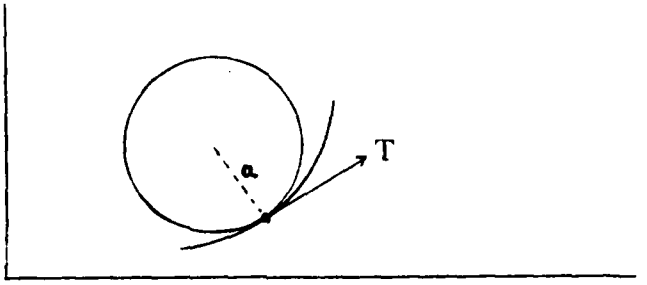
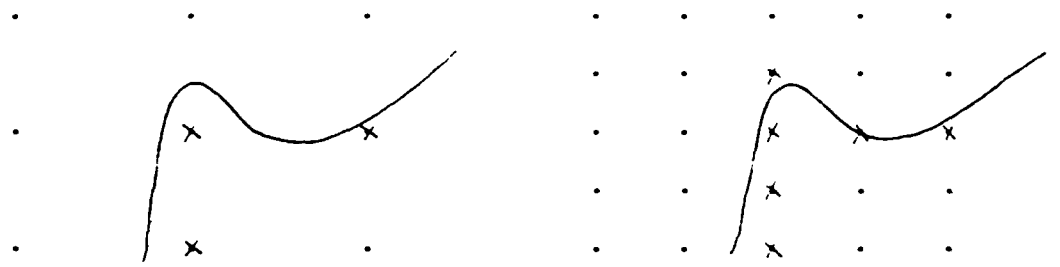


Figure III-3. Circle of Curvature

In this study, curvature will be examined to determine how well the chain codes work with functions of different degrees of curvature. If the grid size is much smaller than the smallest radius of curvature of a line drawing then any inflection of the curve will cross a grid line and will

be encoded. If the radius of curvature is much smaller than the grid size, rapid changes in the function may not be recognized by the encoding process, as in Figure III-4.



x - quantized nodes

Figure III-4. Effect of Grid Size on Quantization Nodes

Derivation of Curvature

Thomas [15:546] develops curvature by noting (refer back to Figure III-3):

$$\tan \theta = dy/dx \tag{1}$$

$$\theta = \arctan(dy/dx) \tag{2}$$

$$d\theta/dx = \frac{(d^2y/dx^2)}{(1+(dy/dx)^2)} \tag{3}$$

Now s is the arc length at point P, so that:

$$ds = (dx^2 + dy^2)^{\frac{1}{2}} \tag{4}$$

This leads to:

$$ds/dx = (1 + (dy/dx)^2)^{\frac{1}{2}} \qquad (5)$$

By combining (3) and (5),

$$k = d\theta/ds = (d\theta/dx) / (ds/dx)$$

$$= \frac{[d^2y/dx^2]}{(1 + (dy/dx)^2)^{\frac{1}{2}}} \qquad (6)$$

In order to more easily extend the calculations to the circular waves, the curvature of the parametric form of the two dimensional function is developed.

$$x = f(t)$$

$$y = g(t)$$

$$= \arctan((dy/dt) / (dx/dt))$$

$$d\theta/dt = \frac{(dx/dt)(d^2y/dt^2) - (dy/dt)(dx^2/dt^2)}{(1 + ((dy/dt) / (dx/dt))^2)}$$

$$= \frac{(dx/dt)(d^2y/dt^2) - (dy/dt)(dx^2/dt^2)}{(dx/dt)^2 + (dy/dt)^2} \qquad (7)$$

The derivative of the arc length s is:

$$d\theta/dt = \left((dx/dt)^2 + (dy/dt)^2 \right)^{\frac{1}{2}} \qquad (8)$$

Combining (7) and (8) yields:

$$k = (d\theta/dt) \,/\, (ds/dt)$$

$$= \frac{(dx/dt)(d^2y/dt^2) - (dy/dt)(dx^2/dt^2)}{\left((dx/dt)^2 + (dy/dt)^2 \right)^{1.5}} \qquad (9)$$

Generalized Curvature of a Circle

This formula can now be used to express the curvature of a circle, and of the generalized circular function. First the circle is parameterized by:

$$x = a\cos(t)$$

$$y = a\sin(t)$$

The derivatives are:

$$dx/dt = -a\sin(t)$$

$$dy/dt = a\cos(t)$$

$$d^2x/dt^2 = -a\cos(t)$$

$$d^2y/dt^2 = -a \sin(t)$$

Substituting into the parametric curvature formula (9),

$$k = \frac{(-a \sin(t))(-a \sin(t)) - (a \cos(t))(-a \cos(t))}{(a^2 \sin^2(t) + a^2 \cos^2(t))} \quad (10)$$

$$= \frac{a^2}{a^3}$$

$$k = \frac{1}{a} \quad (11)$$

Therefore, the curvature of a circle whose radius is a is 1/a. The smaller the radius of a circle, the greater is the curvature. Also, the curvature of the periodic circular function is constant throughout its domain.

Curvature of the Sinusoidal Function

The curvature of the sinusoid function can be evaluated in the same manner by using the parametric equations:

$$x = f(t) = t$$

$$y = g(t) = a \sin(wt + p)$$

Now the derivatives are:

$$dx/dt = 1$$

$$dy/dt = aw \cos(wt + p)$$

$$d^2x/dt^2 = 0$$

$$d^2y/dt^2 = -aw^2 \sin(wt + p)$$

Substituting into equation (9),

$$k = \frac{[\ aw^2 \sin(wt + p)\ ]}{(\ 1 + (aw)^2 \cos^2(wt + p)\ )^{1.5}} \qquad (12)$$

This equation can be used to evaluate the curvature of a given sinusoid function at any point, with any given amplitude, period and phase.

Maximum Curvature of the Sinusoidal Function

A closer examination of equation (12) reveals that the maximum curvature for a sinusoid function occurs at the extreme points of the sine function. At $n \cdot \pi/2$ (n = 1,3,5 ...), the numerator equals aw^2 as the sine goes to

one. At the same time, the cosine value in the denominator goes to zero, with the denominator itself going to one. Therefore the maximum curvature of the sinusoid function is aw^2 .

Average Curvature of the Sinusoidal Function

Now that the instantaneous curvature of the sine function has been evaluated, it would be desirable to be able to find the average curvature of a given sine wave. This will be done by integrating the instantaneous curvature over a full period, and dividing by the length of the period.

$$C = \int_0^{2\pi/w} \frac{[aw^2 \sin(wt + p)]}{(1 + a^2 w^2 \cos^2 (wt + p))^{1.5}} dt \quad (13)$$

The instantaneous curvature of the sine wave is symmetrical about $\pi/2$ and $\pi/4$, as is shown in Figure III-5.

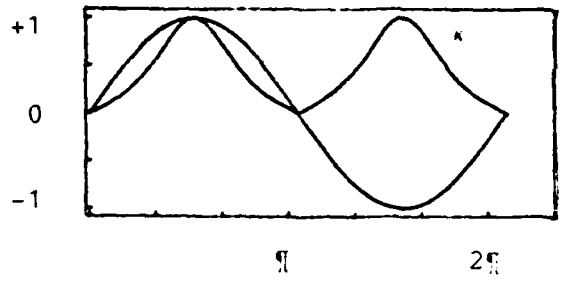


Figure III-5. Symmetry of the Curvature of the Sine Wave

This symmetry is shown analytically by noting that:

$$\sin(wt + p + \pi/2) = \sin(wt + p - \pi/2)$$

and

$$\cos^2(wt + p + \pi/2) = \cos^2(wt + p - \pi/2)$$

for all values of t and p.

Symmetry allows the average curvature integral to be evaluated by:

$$C = 4 \int_0^{\pi/2w} \frac{aw^2 \sin(wt + p)}{(1 + a^2 w^2 \cos^2(wt + p))^{1.5}} dt \quad (14)$$

By changing variables with:

$$A = wt + p$$

$$dA = wdt$$

$$C = 4 \int_{p}^{\pi/2 + p} \frac{aw \sin(A)}{(1 + a^2 w^2 \cos^2(A))^{1.5}} dA \quad (15)$$

Due to symmetry, the integral over a complete period is the same regardless of the starting point (or phase) so that:

$$C = 4 \int_{0}^{\pi/2} \frac{aw \sin(A)}{(1 + a^2 w^2 \cos^2(A))^{1.5}} dA \quad (16)$$

Substituting the variables:

$$u = aw \cos(A)$$

$$du = -aw \sin(A)\, dA$$

$$C = 4 \int_{aw}^{0} \frac{-1}{(1 + u^2)^{1.5}}\, du \qquad (17)$$

$$= 4 \int_{0}^{aw} \frac{1}{(1 + u^2)^{1.5}}\, du \qquad (18)$$

The final variable substitution of:

$$u = \tan(B)$$

$$du = \sec^2(B)$$

yields the integral:

$$C = 4 \int_{0}^{\arctan(aw)} \frac{\sec^2(B)}{(\sec^2(B))^{1.5}} \, dB \qquad (19)$$

$$C = 4 \int_{0}^{\arctan(aw)} \cos(B) \, dB \qquad (20)$$

$$= 4 \sin(B) \Big|_{0}^{\arctan(aw)} \qquad (21)$$

$$= \frac{4aw}{(\P) \; (1 + a^2 w^2)^{\frac{1}{2}}} \qquad (22)$$

The average curvature is this area divided by the width of the period, or

$$\text{avg curv} = \frac{2\,a\,w^2}{\pi\,(1 + a^2\,w^2)^{\frac{1}{2}}} \qquad (22)$$

Now that an expression for the average value of the curvature of a sine wave has been obtained, I will examine how these functions of curvature affect the performance of the quantization system. It is also interesting to note that the expression of the average curvature supports the assumption that the curvature of a sine function is not dependent on phase.

Summary

This chapter has reviewed the general concepts of the mathematic quantity of curvature of a function. The curvature of a function has been introduced as a parameter to be used to quantitatively evaluate the performance of the chain codes. The formulas used to evaluate the curvature of continuous circular and sinusoidal waveforms were reviewed. Finally, equations for determining the instantaneous and average curvature of an arbitrary sinusoidal function were developed.

IV. Implementation

This chapter discusses the implementation details of Lt Jones' FORTRAN program and the changes required to make it run on the AFIT SSC VAX 11/780. The changes are of two basic types: 1) the obvious compiler specific changes and 2) changes required to increase the numerical precision of the standard VAX arithmetic routines. The specific changes are discussed with the associated problem areas and symptoms, followed by a description of the test runs used to verify the correctness of the resulting program. Recommendations are made concerning possible further analysis and modifications to the program.

Background

The quantization program was written for the FORTRAN 5 compiler on the CDC Cyber computer. This compiler is an extension of the 1966 FORTRAN standard and is documented as a superset of the FORTRAN 77 standard. The VAX compiler is the VAX UNIX f77 version of FORTRAN 77.

The two machines have a significantly different word size; consequently, the accuracy of the single precision arithmetic operations is different. The basic word of the

CDC is 64 bits, providing a floating point range of approximately plus or minus 10 exp 308 with 15 significant decimal digits. The VAX single precision d-format range is plus or minus 10 exp 38 with 7 significant digits [9,11].

Errors and Software Changes

This basic difference led to the only compiler specific change in the program. The ZERO function uses an error tolerance test of 10 exp -100 when searching for the zero point of a function. The exponent was identified as a floating point exception by the f77 compiler. Analysis of the routine indicated that only a "small tolerance" was required. Since this tolerance is also used as a divisor, 10 exp -25 was chosen to replace the 10 exp -100 to provide some leeway.

The resulting program ran, but the results produced two error symptoms: 1) a run time error "NTOL EXCEEDED" and 2) all the BPL (BPL is the bits per unit length metric as discussed in chapter two) results were the same.

The NTOL variable is used to place a limit on the number of iterations used in the search for the function zero point. Since this loop was being repeated too many times, I concluded that the decreased precision of the VAX was accumulating errors significant enough to invalidate the search loop. The obvious fix was to use double precision for all the REAL variables. (The f77 compiler uses the VAX type g floating point format for double precision operations.)

To aid in debugging, I included forced data typing statements for all variables. This led to the identification of another syntax problem: the generic (or untyped) name for the sine function on the VAX was not the same as that used on the CYBER. Changing the spelling corrected this problem.

The results of the double precision runs were again that the BPL output showed no change. (The X variable did agree to all eight printed digits.) The use of the symbolic debugger sdb produced two results: 1) the identification of multiple variables in the same COMMON BLOCKS having the same address, and 2) the conclusion that the data input file was not being read correctly. The real problem here was that a variable was being input as zero, causing all the results to be the same.

I corrected the data input file and the program ran properly (I reported the COMMON BLOCK problem to the system administrator as a possible sdb or f77 error.) The output from this run agreed identically with the corresponding run produced on the CYBER. The results from the next run identified a classic numerical problem in that the quotient LOG(8)/LOG(2) on the CYBER was 3.000000 but on the VAX the same operation produced a result of 2.999999. I corrected this by adding 10 exp -20 to the result and rounding.

With these corrections, the program was tested on both sine and circular waves using the (1), (2), (1,2), (1,3) and (1,2,3) codes with amplitudes of 0, 10 and 20 and periods of

0 and 0.2. All the output from these runs agreed exactly with the results from the CYBER.

Summary and Recommendation

This chapter has reviewed the software changes I used to make Lt Jones' quantization analysis program work on the VAX. By utilizing double precision for all real variables, the calculations agree exactly with the Cyber results.

Use of the double precision variables more than doubles the calculation time for these variables. Although not prohibitive, run times of eight to twelve CPU minutes were common. Double precision accuracy might not be required in further studies of this kind unless specific parameters are being evaluated. The trends of the runs are usually apparent in the first two or three significant digits, and single precision may provide the necessary resolution.

Two areas should be analyzed to evaluate whether single precision is sufficient for the program: 1) effect of accumulated errors in loops and 2) effect of precision on the determination of the zero points. As discussed previously in this chapter, the accumulated errors may cause the zero function not to find the correct zero point.

A possible solution to both of these problem areas is to use double precision only in the repeated major loops or in the zero function. Even with double precision in these two areas, a significant decrease in CPU run time would be expected.

V. Results For Circular Waves

In this chapter I will present the results observed when quantizing circular waves. First I will describe what the quantizing program's input parameters are and how they define the function being quantized. Next I will present plots of the performance of these functions and relate the performance to the curvature of the circular waves. Finally, I will summarize the results for the quantization system when used to encode circular waves.

Input Parameters

Lt Jones' quantization program was written to be general enough to quantize sine and circular functions of arbitrary amplitude, period, and phase. He also made provision for allowing different grid sizes, the number of periods to run in the experiment and multiple ring levels for the various types of codes.

In order to be able to easily compare all my results for both the sine and circular functions, I decided to use a uniform grid size of one and run each experiment for one thousand periods. As I have previously explained in chapter III, I only examined sinusoid and circular

functions with zero initial phase. The only remaining input parameters for circular waves are the type of code to use for quantization, and the period since the amplitude of a circular wave is one fourth the period. Therefore to test a circular wave, one need only specify the period and the type of code to be used.

With these inputs, the program produces a listing of the independent variable, the total area error, the area error per unit length of the function and the number of quantized bits per unit length. Using the "S" data manipulation package on the VAX, I produced the plots of the quantization results included in Appendix A.

Discussion of Plots

For each of the plots, the upper graph is the area error per unit length (or aepl) as a function of the independent variable x (which was used to generate the circular function). The lower plot is the number of encoded bits per unit length used to encode the function (or bpl) as a function of the independent variable x. These are the accumulated statistics so that the aepl is the total error area up to each point divided by the total length of the function up to that point.

The objective of these experiments is to determine the performance characteristics of the quantization schemes when used to quantize circular waves. I generated computer runs for each of the (1), (2), (3), (1,2), (1,3) and (1,2,3) codes with periods of 5, 10, 15, 20, 30, 40, 50,

and 100 units. To correlate the results I graphed the performance of each of the codes against the same amplitude (Figures A3 through A10).

These graphs are very similar in shape. Most of the curves (both aepl and bpl) are flat, which means that the performance settles down to a relatively constant level. Another interesting observation is that the ordering of the performance of the codes is the same across all the different amplitudes. Also, the expected classical tradeoff is apparent: increasing the accuracy (or decreasing the aepl) causes an increase in the number of bits required to encode the function.

Generally, these charts show that the (1,3) code has a very low aepl and it has an average bpl. This holds across all the different curvatures which were examined. The other general observations are that the (3) code always had the lowest bpl and the highest aepl, followed by the (2) code and the (1) code.

Curvature

These results tend to confirm the prediction of performance based on the analysis of the curvature of the circular function. Since (for the circular wave, as was discussed in chapter III) the curvature of a circle is equal to 1/a (where a is the radius of the circle) and constant, one would expect a fairly constant performance plot. For the circular waves just discussed, the curvature data are presented in Table V-1.

Per	5	10	20	30	40	50	100
Amp	1.25	2.5	5.0	7.5	10	12.5	25
Curv	.8	.4	.2	.13	.10	.08	.04

Table V-1. Radius of Curvature of Circular Waves

A large curvature value indicates a relatively sharp turn of the curve; a smaller value indicates a more straight function. Since the instantaneous curvature is constant, so also is the average value. Therefore, based on the knowledge that the grid based line drawing system models functions as sequences of line segments, one would expect that the aepl of a circular wave of low curvature would be lower than the aepl of a wave with a large curvature (based on either instantaneous, maximum or average curvature). This trend is seen in the plots by noting that the aepl generally increases as the curvature increases (although this trend is not monotonic). This trend was gradual from periods of 100 down to 20 with a marked increase in the neighborhood of 15. At about 5, the error doubled that at 15; note that this corresponds to the curvature approaching the grid size in the quantization. The charts on pages A-11 and A-12 show how the asymptotic values of the aepl change as related to the curvature of the function, with the single ring codes presented on A-11 and the multiple ring codes on A-12.

Note that the behavior of the single ring codes and the multiple ring codes are different at the lower end of the graphs. The charts on A-13 and A-14 show the corresponding bpl plots for the same curvature points. The bpl plots are more constant than the bpl, but do change behavior somewhere in the neighborhood of radius 15.

In order for the bpl to be low, the quantization must be able to utilize the outer rings for portions of the curve, while also being able to utilize the inner rings where necessary to reduce the error. This can be seen in the graphs by observing that the (1,3) code is relatively low in both aepl and bpl. My interpretation of this is that when the section of the functions where the tangent is parallel to either of the grid axes, the third ring is used, while the first ring is used when the tangent is at a larger angle to the axes, such as in Figure V-1. Apparently there are not enough of the intermediate points to make the second ring necessary since the (1,2,3) code performance is not that much better than the (1,3) code. Another notable result is that the single ring codes ((1), (2), and (3) codes) have a very nearly constant bpl regardless of the curvature. This means that the code is using a straight line approximation parallel to the grid axes for the majority of the points. (Proffitt concludes that this trend will continue and that only a four way coding system is necessary for any straight line [12:79].)

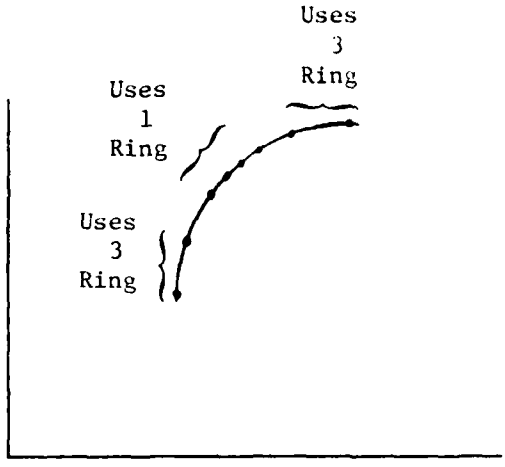


Figure V-1. Ring Utilization of the (1,3) Code

Summary

In this chapter I have reviewed what the quantization program's input parameters are and how these parameters affect the curvature of the resultant circular function. I have reviewed the performance of the grid based line drawing quantization system when used to encode circular waves. These results indicate that analysis of curvature provides some general indications of performance such as code choice and grid size, but it does not strictly predict the performance of the quantization system, at least when used on circular waves.

VI. Results for Sine Waves

This chapter reviews the results obtained when sine functions of various curvatures were encoded with the line drawing quantization system. Observations are made comparing several measures of curvature and how these measures relate to the performance of the coding system. The performance of the codes against circular waves and sine waves are compared and generalizations concerning these results are made.

The experiments which produced these results were begun with the same motivations as the circular waves. Based on the results from Chapter III, the curvature of sine waves can easily be evaluated. Using the formulas for maximum curvature of the sine wave as a guide, several computer runs were made and the resulting curvatures are listed in Table VI-1.

These sine waves are generated with inputs similar to those discussed in Chapter V for the circular waves. Specifically, the inputs are amplitude, period, phase, grid size, number of periods to run, and the type of code to use for quantization. As with the circular wave, the phase was set to zero, a grid size of one and a length of one

thousands periods was used in all the runs. The same codes ((1), (2), (3), (1,2), (1,3) and (1,2,3)) were run with periods of 5, 10, 20, and 50, each with amplitudes ranging from 5 to 100. The graphs of these results are Figures A-15 through A-24.

The results of the sine runs were quite similar to those of the circular runs although not as regular. The rankings of the bpl metric was as expected, with the (1) code consistently least efficient and the (3) code most efficient. The aepl for these runs was more noisy, but had an observable trend in the ranking of the codes.

Based on these similarities, the analysis was limited to the study of the (1,3) code. This code was in the middle of the grouping of both aepl and bpl. Also, the ability of this code to use the third ring to take advantage of long straight sections and also use the first ring to react to areas of more curvature is theoretically appealing. As was noted in chapter V, the single ring codes produced a very nearly constant value of bpl when used on circular waves as well.

After these preliminary runs produced no surprising results, the results were evaluated based on a comparison of the minimum radius of curvature (or maximum curvature) function. As this description implies, the function returns the value of the radius of curvature of the sine wave at its peak. Table VI-1 indicates that several of the runs have exactly the same maximum curvature and that there is a wide range of values in the table.

AMP	PER5	PER10	PER20	PER50
.5	.7896	.1974	.0493	.0079
1.0	1.5791	.3948	.0987	.0158
5.0	7.8957	1.9739	.4935	.0790
10.0	15.7914	3.9478	.9870	.1579
20.0	31.5827	7.8957	1.9739	.3158
30.0	47.3741	11.8435	2.9609	.4737
40.0	63.1655	15.7914	3.9478	.6317
50.0	78.9568	19.7392	4.9348	.7896
100.0	157.9137	39.4784	9.8696	1.5791

Table VI-1. Maximum Curvature of Sine Wave

The graphs on pages A-24 and A-25 are plots of these runs of similar minimum radius of curvature for the (1,3) codes. In each of these plots, the aepl and bpl are quite close, even though the characteristics of the sine curves are quite different in terms of scale. (Of course, the shapes are similar as both are sinusoids.)

This clustering of performance is the most consistent result in this thesis. Note that by evaluating the absolute minimium radius of curvature of a function, one can guarantee that any inflection of the curve will not be removed by the coding system as long as the grid size is chosen to be at least as large as the minimum curvature.

I proceeded to investigate whether some other functions of curvature would lead to a more definite or tighter clustering of results. The average curvature of the sine waves was evaluated and is presented in Table

VI-2. This average is not a strong function of the amplitude, and therefore seems to be an undesirable statistic.

AMP	PER5	PER10	PER20	PER50
.5	.4256	.1199	.0310	.0050
1.0	.6260	.2128	.0599	.0100
5.0	.7901	.3812	.1687	.0426
10.0	.7975	.3950	.1906	.0626
20.0	.7994	.3987	.1975	.0743
30.0	.7997	.3994	.1989	.0773
40.0	.7998	.3997	.1994	.0785
50.0	.7999	.3998	.1996	.0790
100.0	.8000	.3999	.1999	.0797

Table VI-2. Average Curvature of Sine

Although at first it seemed advantageous from an engineering point of view to see if the average curvature might be a guide to estimated performance, Table VI-2 highlighted the fact that the averaging of the curvature would mask out many of the inflections of a curve. Also, for an arbitrary function with varying curvature, one would not expect an area of many small inflections to be encoded with performance similar to a fairly straight function with an equal average value of curvature.

One would, however, expect that sine waves of the same amplitude/period ratio to result in similar performance since they have a similar shape. This statistic however, produces a similar ordering as the average curvature, and the results showed no specific patterns. I also evaluated

the ratio of maximum curvature to the square root of the sum of amplitude square and period square, and the ratio of maximum curvature to the square root of amplitude period product, with the same result of no apparent pattern.

Finally, I compared the quantization results of circular waves with the results of sine waves with a maximum curvature equal to the curvature of the circular wave. One would expect that the sine wave would have a similar or better aepl because every part of the sine function is straighter except for the portion of the curve at the extreme points.

The graphs on pages A-26 and A-27 show that the results for sine and circular waves are quite similar, but I was not able to run enough experiments to completely investigate this comparison. Because the quantization program does not allow complete independence of the DEL, AMP and PER parameters for the grid size, amplitude and period, I was not able to generate circular waves of curvatures greater than 0.8 (see Table V-1). (There was some discussion about choosing a minimum DEL and adjusting the AMP and PER parameters by a specified scale factor to achieve the same results, but I preferred to explicitly state the desired inputs.)

Summary

This chapter has reviewed the results of evaluating the performance of the chain codes when used to evaluate sine waves based on the criteria of curvature. The results indicate that the minimum curvature can be used to specify the grid size and that this choice limits the error of the quantization. Based on this study, the (1,3) code provides a good balance between bit rate and error performance.

VII. Conclusions and Recommendations

In this chapter, the general results of the study are reviewed and discussed. Recommendations for further study are made.

The overall performance of the quantization system was very consistent in this study. Over the various amounts of curvature which were examined, both the average error rate and the bit rate converged. In nearly all cases, the bpl ranking of the codes agreed to the expected order. Likewise, the ranking of the codes in order of aepl was consistent, although not as consistent as was the bpl. The bit rate of the single ring codes was found to be nearly constant for both sine and circular waves, regardless of the curvature of the function being quantized.

The objective of this thesis was to determine whether the curvature of a function could be used to predict the performance of the grid based quantization system. The results indicated that the minimum curvature of a function can be used as a guide to the grid size for the quantization system and that the (1,3) code seems to provide a good combination of low error and bit rate. (At least where the minimum radius of curvature is greater than

the grid size and less than one hundred times the grid size). The study also indicates that average curvature is not a very good predictor of performance.

There are some areas which still need to be further examined however. Because the relative importance of these results may be application dependent, other aspects of curvature may be more important to characterize a signal than the minimum value. Also, the computer program used to evaluate the codes could be modified to produce more useful results. My specific recommendations are to:

1) Evaluate the (1,4) and higher order curves to determine if the trends continue to increase performance with only marginally more coding bits.

2) Modify the program to produce statistics showing the relative usage of the different ring levels. This would help to explain the results from recommendation 1.

3) Evaluate the codes using non-integral values of the grid size for the amplitude and period. (This is related to recommendation 5 below.)

4) Improve the efficiency of the program by limiting the use of Double Precision. This should drastically reduce run times, and several runs exceeded thirty minutes of CPU time.

5) Change the quantization algorithm to follow along the curve for finding grid intersections. This should also

decrease run time because the search for the grid intersections using the ZERO function does not appear to be very efficient.

6) Change the program so that the grid size, amplitude, and period are totally independent. As discussed in chapter VI, this would allow the evaluation of circular waves with curvature greater than one. When this is done, the comparison of circular waves and sine waves with the same curvature should be continued.

BIBLIOGRAPHY

1. Castor, K. G. and Neuhoff, D. L. "A Rate and Distortion Analysis for Grid Intersect Encoding of Line Drawings", IEEE 1981 Pattern Recognition & Image Processing Proceedings, IEEE Computer Society Conference on Pattern Recognition and Image Processing. Dallas, Texas. August 1981.

2. Cederberg, R. L. T. "Chain Link Coding and Segmentation for Raster Scan Devices." Computer Graphics and Image Processing. Volume 10, number 3. July 1979.

3. Freeman, Herbert. The Generalized Chain Code for Map Data Encoding and Processing. Technical Report CRL-59. Air Force Office of Scientific Research. June 1978.

4. Freeman, Herbert. "Computer Processing of Line Drawing Images", Computing Surveys. Volume 6, number 1, pp 57-59. March 1974.

5. Freeman, H. and Frauenknecht, P. J. "An Efficient Computer Representation Scheme for Linear Map Data." IEEE 1982 Pattern Recognition and Image Processing Proceedings.

6. Freeman, H. "A Review of the Relevant Problems in the Processing of Line Drawing Data. Automatic Interpretation and Classification of Images." Academic Press Inc. New York, NY. 1979.

7. Hildreth, E. C. "The Detection of Intensity Changes by Computer and Biological Vision Systems." Computer Vision, Graphics and Image Processing. Volume 22, number 1. April 1983.

8. Jones, Keith R. "Grid-Based Line Drawing Quantization." Master's Thesis AFIT/GE/EE/82D-41. Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio. Dec 1982.

9. Lemone, K. A. and Kaliska, M. E. Assembly Language Programming for the VAX-11. Little, Brown computer systems series. pp 147-150. Boston, Mass. 1983.

10. Maxwell, P. C. "The Perception and Description of Line Drawings by Computer", Computer Graphics and Image Processing, volume 1, pp 31-46. April 1972.

11. PDP-11 Processor Handbook. Digital Equipment Corporation, Maynard Mass. 1978.

12. Proffitt, D. and Rosen, D. "Metrication Errors and Coding Efficiency of Chain Encoding Schemes for Representation of Lines and Edges." Computer Graphics and Image Processing, Volume 10, number 4. August 1979.

13. Saghri, John A. "Efficient Encoding of Line Drwaing Data with Generalized Chain Codes." Rensselaer Polytechnic Institute, Troy, New York. August 1979.

14. Thomas, George B. Jr. Calculus and Analytic Geometry, 3rd edition. Addison-Wesley, Reading Mass, June 1962. pp 585-591.

15. Thomas, G. B. Jr., and Finney, R. L. Calculus and Analytic Geometry, 5th ed. Addison-Wesley, Reading Mass, May 1979. pp 546-556.

Appendix A

This appendix contains the plots of the computer runs referred to in this thesis. The plots are in four major categories. Figures A-3 through A-10 show the relationship between curvature and the performance of the quantization system when used on circular waves at a given curvature, using the six codes examined in this thesis. Figures A-15 through A-23 are similar plots with sinusoidal waves. Figures A-11 thru A-14 show the performance of each of the codes with circular waves of different curvatures. Figures A-24 and A-25 show the performance of the (1,3) coding system with sine waves of the same curvature. Figures A-26 and A-27 compare the performance of circular waves with sine waves of the same minimum curvature, using only the (1,3) code.

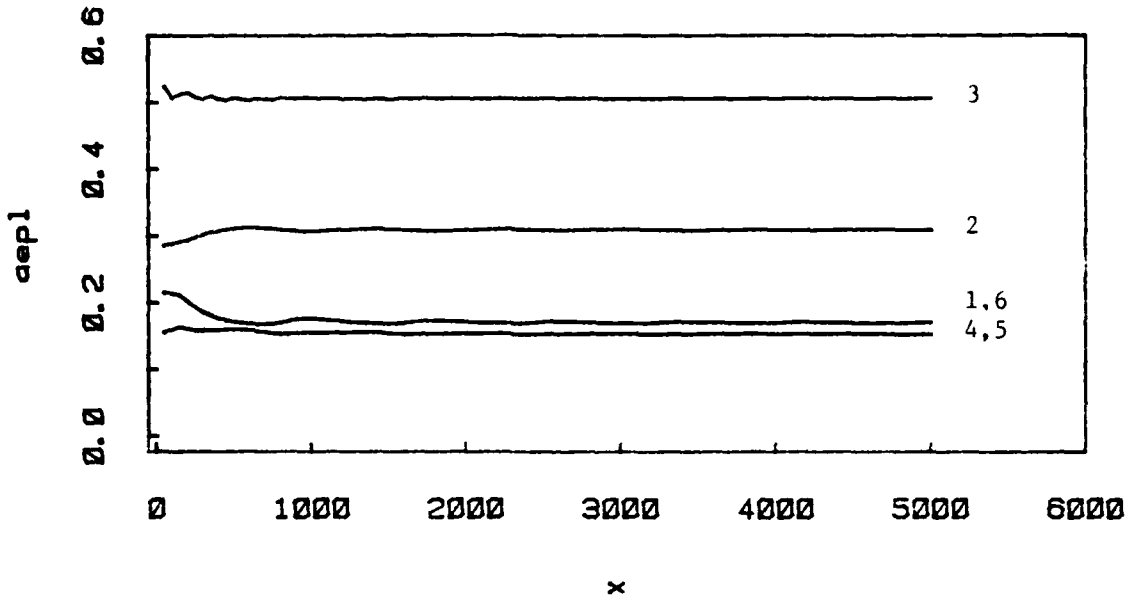
On these plots the AEPL is the area error per length of the function and the BPL is the number of encoded bits per unit length of the function. In both cases the independent variable is the length of the function, and the statistics are the accumulated error and bit rate up to that point. This use of normalized mecrics makes it easier to compare the codes when used on curves of different lengths.

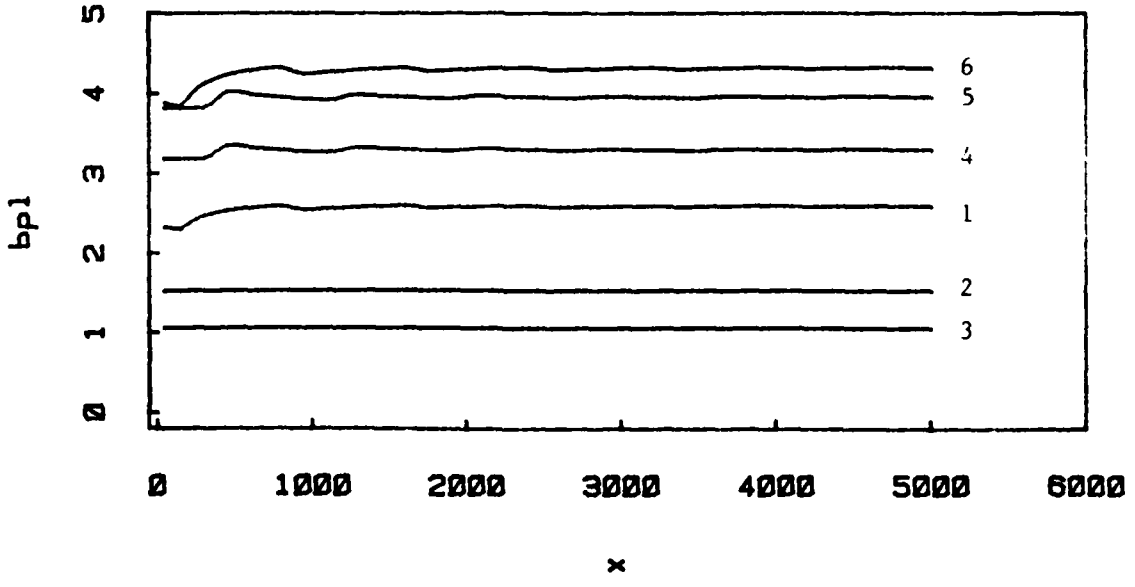
In order to easily identify the codes on the plots, I have marked them with the following numbering system:

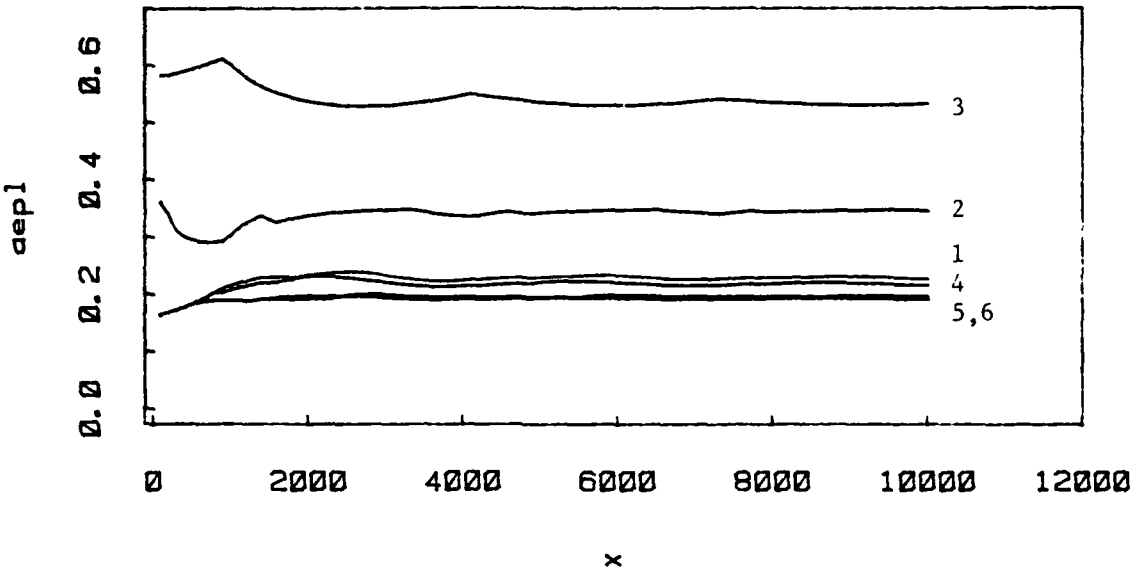
1	(1) code
2	(2) code
3	(3) code
4	(1,2) code
5	(1,2,3) code
6	(1,3) code

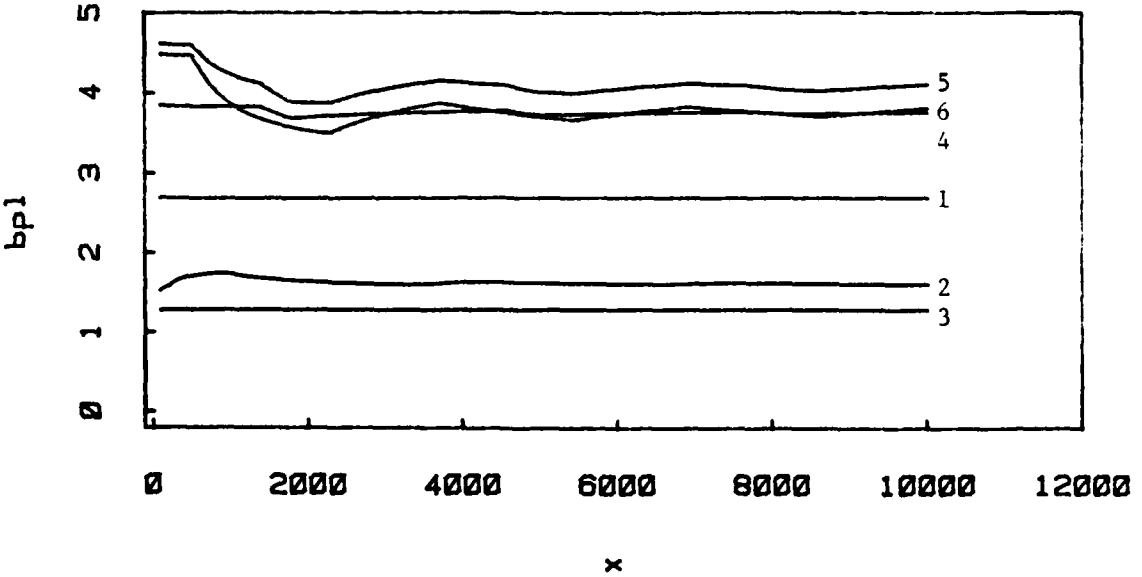
Table of Performance Plots

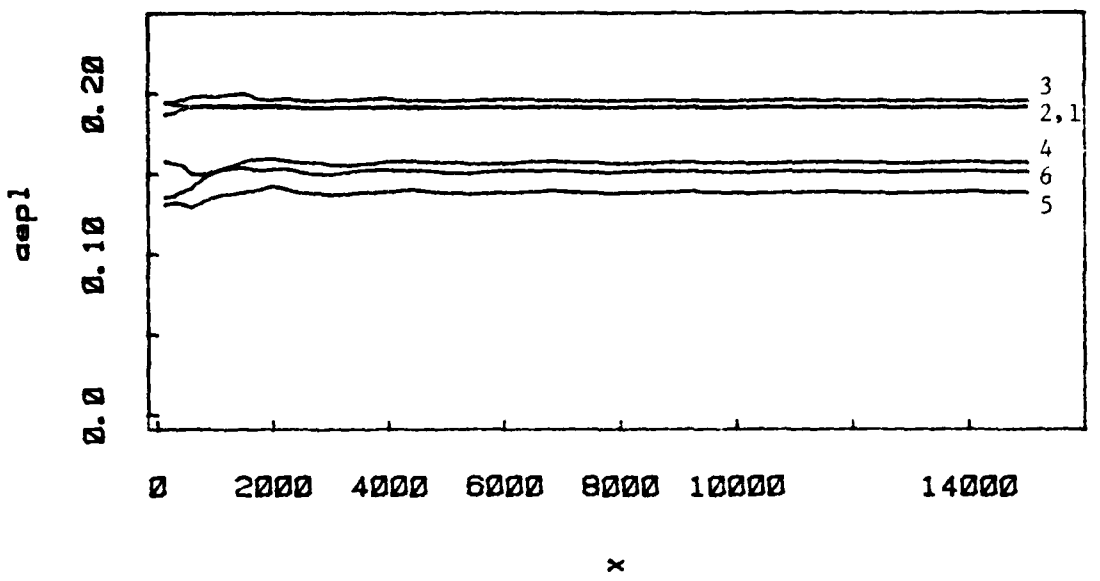
Page	Function	Curvature	Description
A-3	Circular	1.25	All codes Vs Curvature
A-4	Circular	2.5	All codes Vs Curvature
A-5	Circular	3.75	All codes Vs Curvature
A-6	Circular	5.0	All codes Vs Curvature
A-7	Circular	7.5	All codes Vs Curvature
A-8	Circular	10.0	All codes Vs Curvature
A-9	Circular	12.5	All codes Vs Curvature
A-10	Circular	25.0	All codes Vs Curvature
A-11	Circular		Aepl Vs Curvature
A-12	Circular		Aepl Vs Curvature
A-13	Circular		Bpl Vs Curvature
A-14	Circular		Bpl Vs Curvature
A-15	Sine	15.79	All codes Vs Curvature
A-16	Sine	1.97	All codes Vs Curvature
A-17	Sine	3.95	All codes Vs Curvature
A-18	Sine	15.79	All codes Vs Curvature
A-19	Sine	0.49	All codes Vs Curvature
A-20	Sine	1.97	All codes Vs Curvature
A-21	Sine	3.95	All codes Vs Curvature
A-22	Sine	0.47	All codes Vs Curvature
A-23	Sine	0.79	All codes Vs Curvature
A-24	Sine	2.0	(1,3) code Vs Curvature
A-25	Sine	15.79	(1,3) code Vs Curvature
A-26	Both	5.0	(1,3) code Vs Curvature
A-27	Both	12.0	(1,3) code Vs Curvature

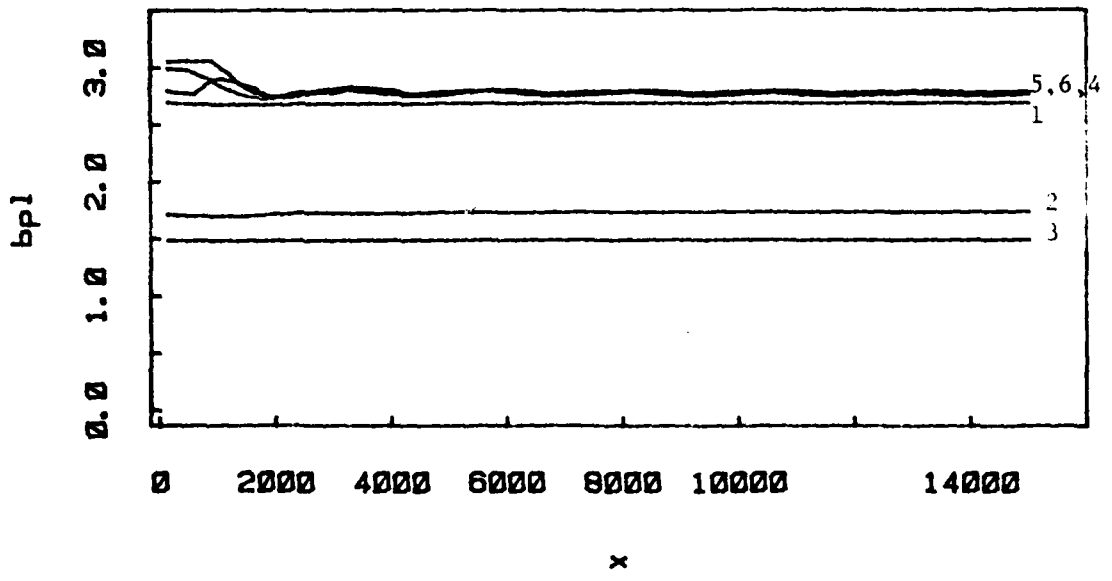

0.6
0.4
0.2
0.0
aepl
3
2
1,6
4,5
0
1000
2000
3000
4000
5000
6000
x

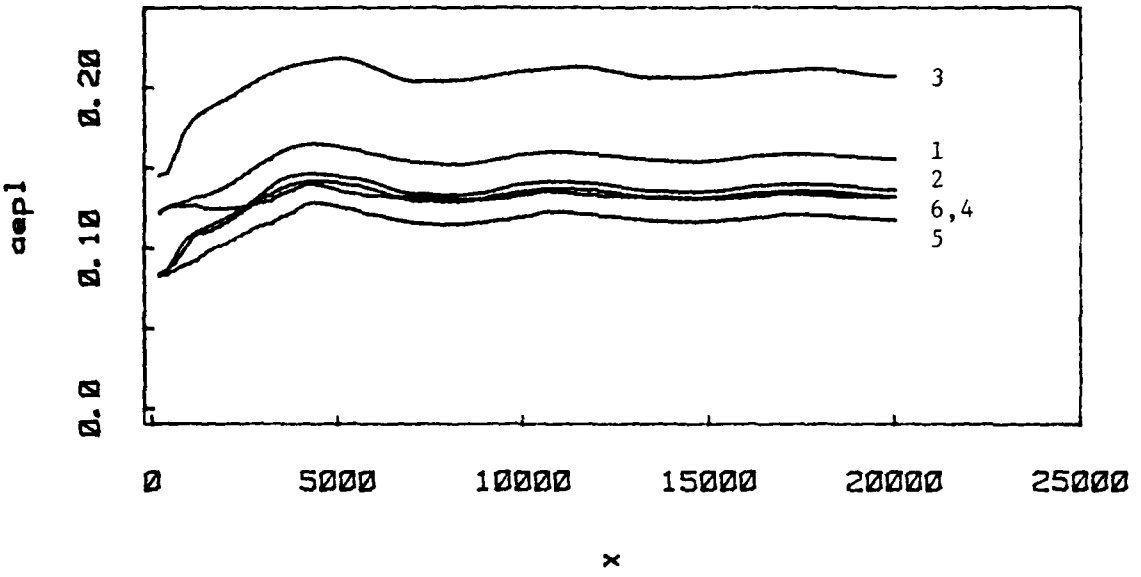

5
4
3
2
1
0
bpl
6
5
4
1
2
3
0
1000
2000
3000
4000
5000
6000
x

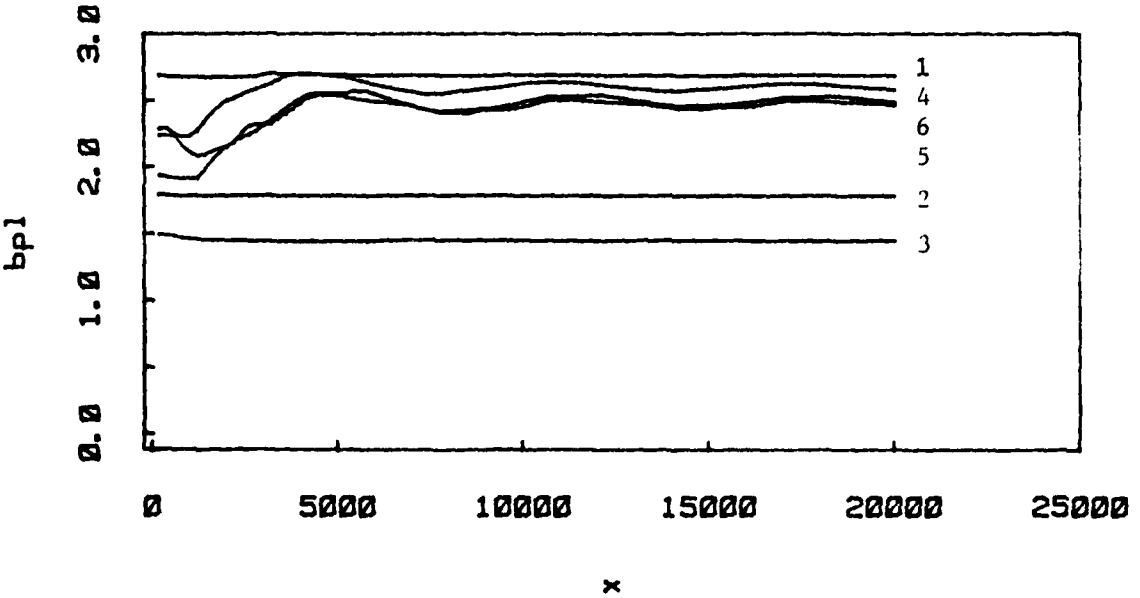

aepl
0.0
0.2
0.4
0.6
0
2000
4000
6000
8000
10000
12000
x
3
2
1
4
5,6


bpl
0
1
2
3
4
5
0
2000
4000
6000
8000
10000
12000
x
5
6
4
1
2
3

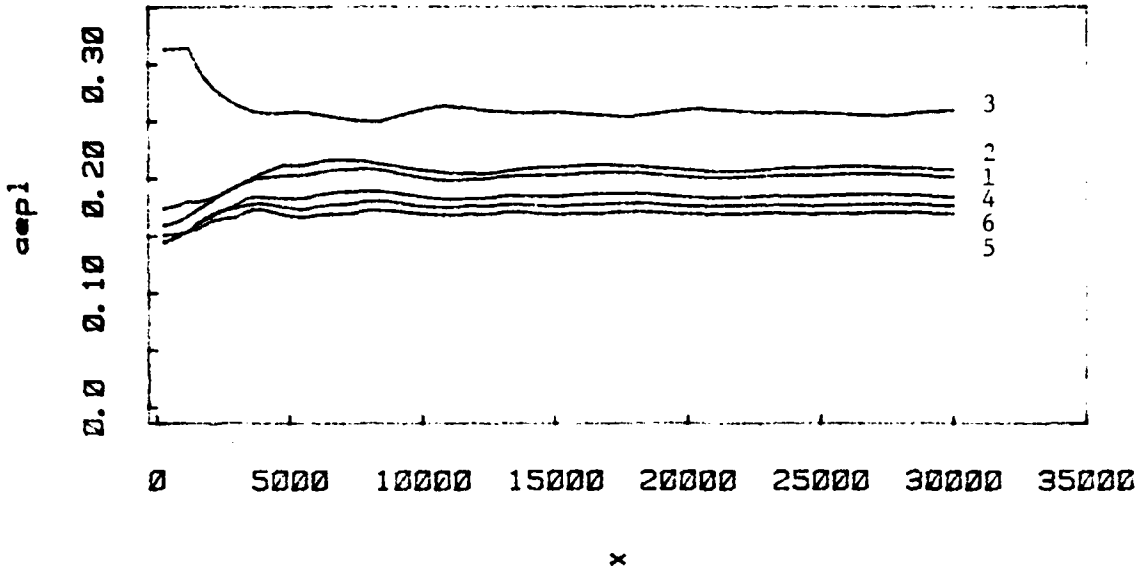

capl
0.0
0.10
0.20
3
2,1
4
6
5
0
2000
4000
6000
8000
10000
14000
x

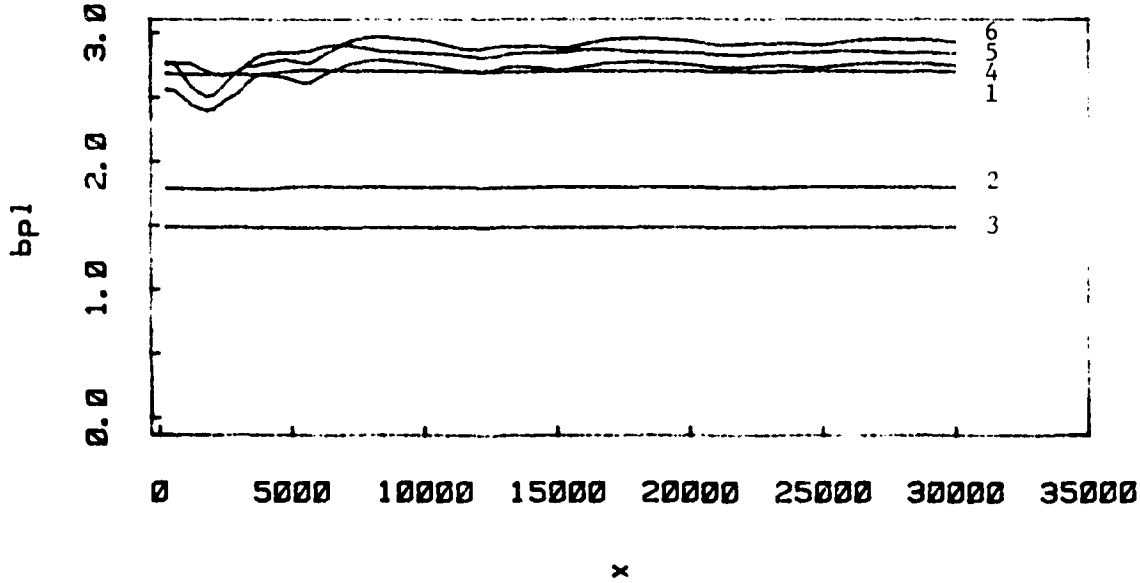

bpl
0.0
1.0
2.0
3.0
5,6,4
1
2
3
0
2000
4000
6000
8000
10000
14000
x

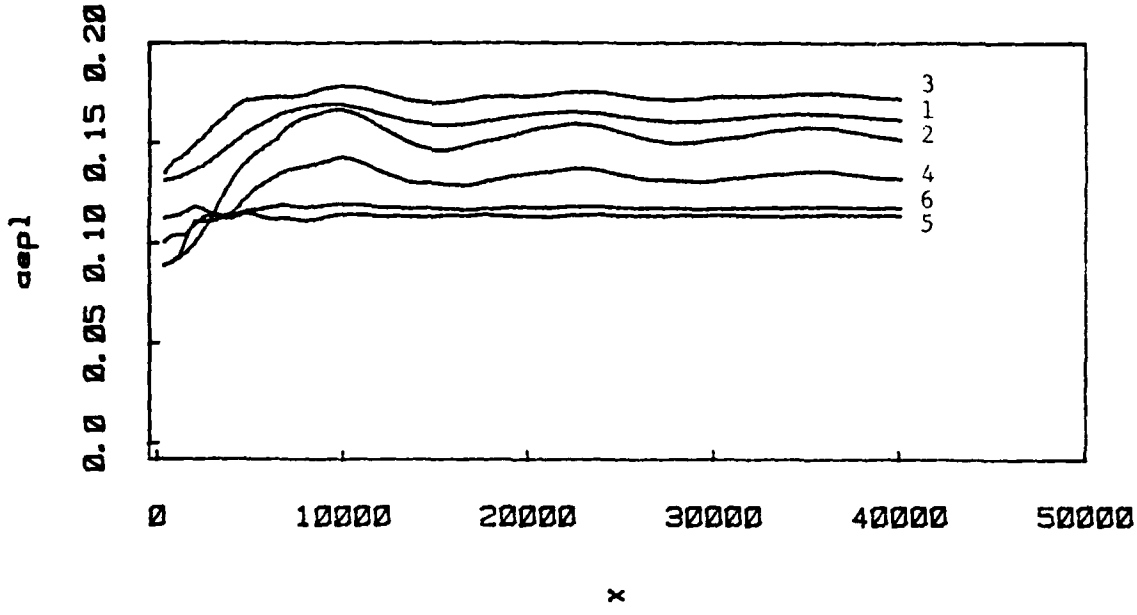

aepl
0.0
0.10
0.20
0
5000
10000
15000
20000
25000
x
3
1
2
6,4
5

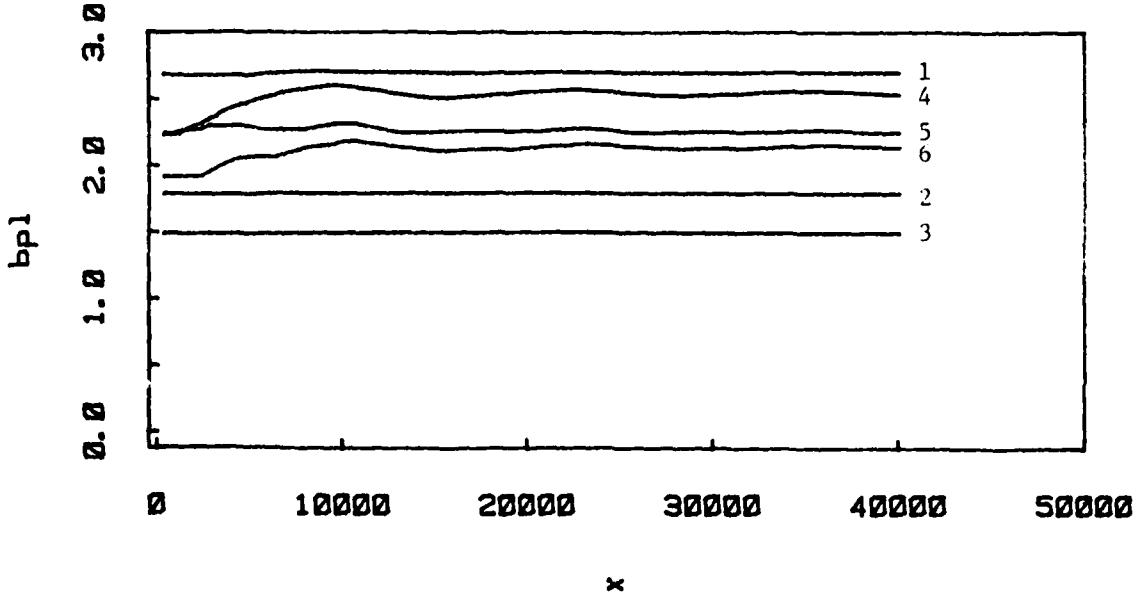

bpl
0.0
1.0
2.0
3.0
0
5000
10000
15000
20000
25000
x
1
4
6
5
2
3

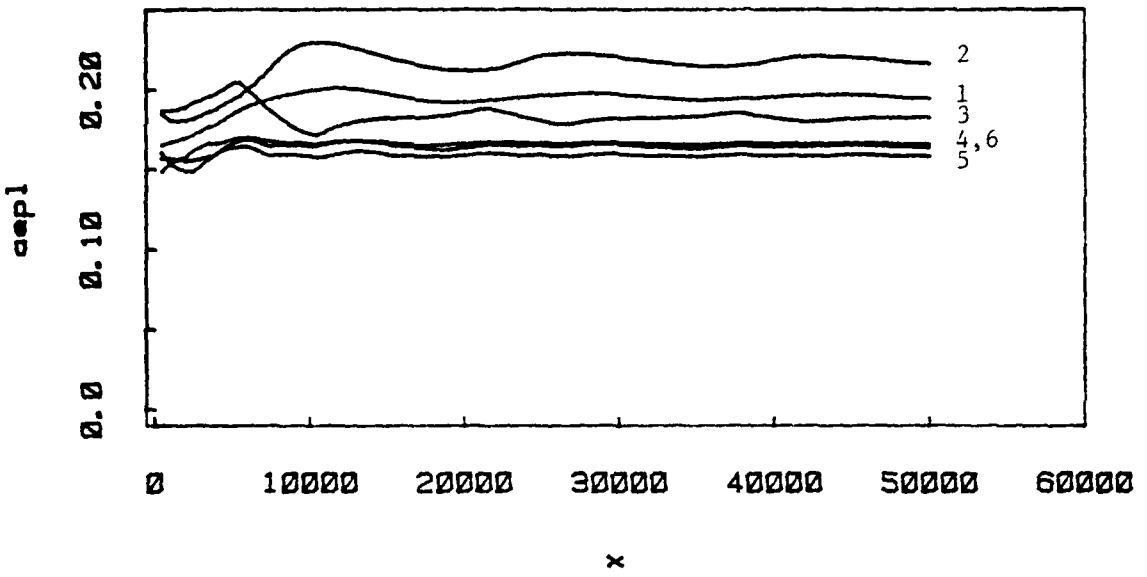
12

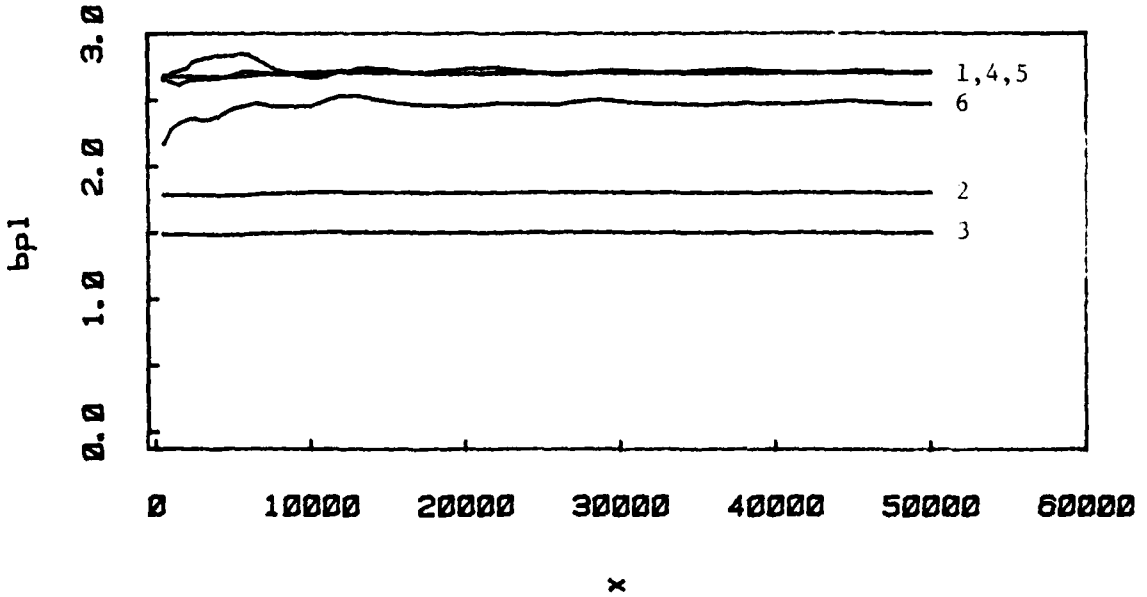

aapl
0.0
0.10
0.20
0.30
3
2
1
4
6
5
0
5000
10000
15000
20000
25000
30000
35000
x

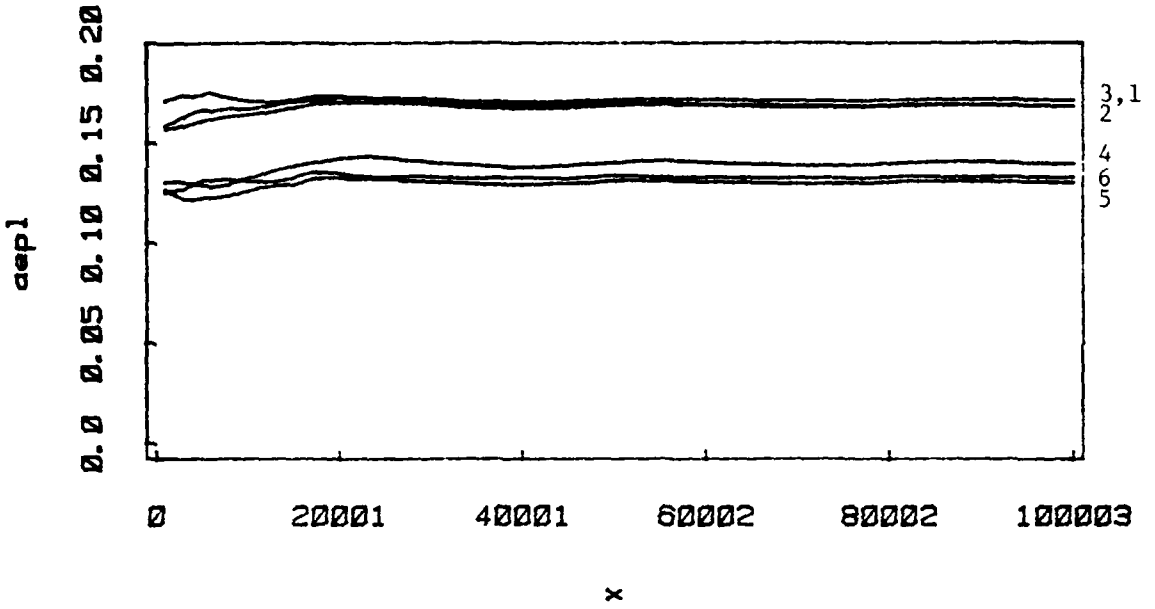

bpl
0.0
1.0
2.0
3.0
6
5
4
1
2
3
0
5000
10000
15000
20000
25000
30000
35000
x

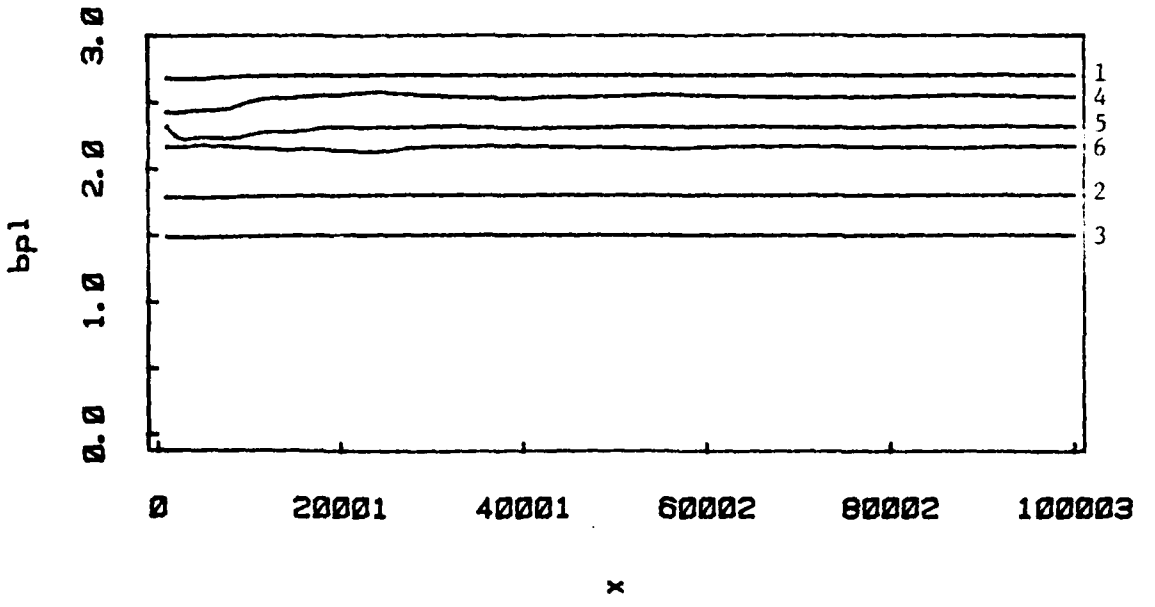

aepl
0.0 0.05 0.10 0.15 0.20
3
1
2
4
6
5
0 10000 20000 30000 40000 50000
x

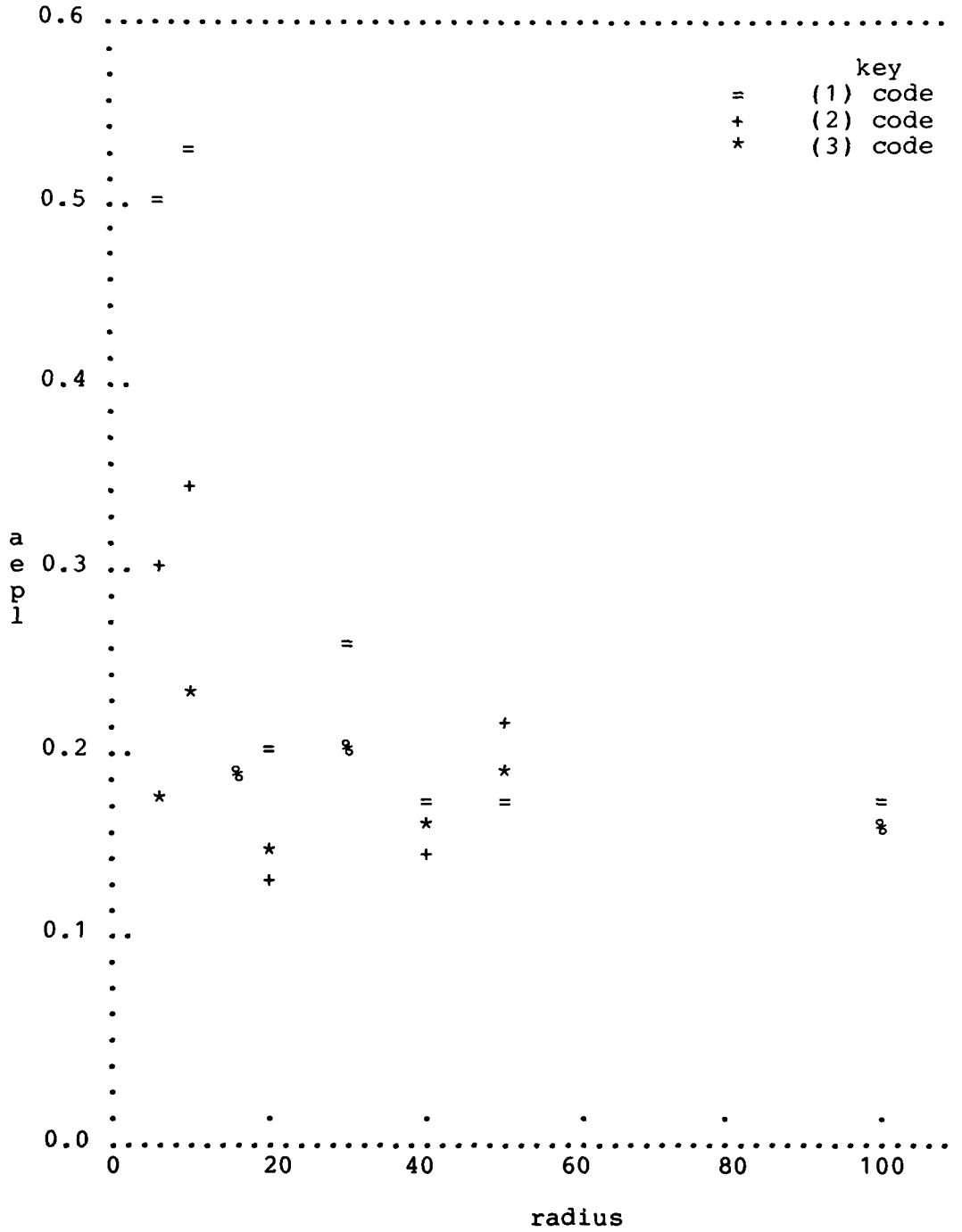

bpl
0.0 1.0 2.0 3.0
1
4
5
6
2
3
0 10000 20000 30000 40000 50000
x

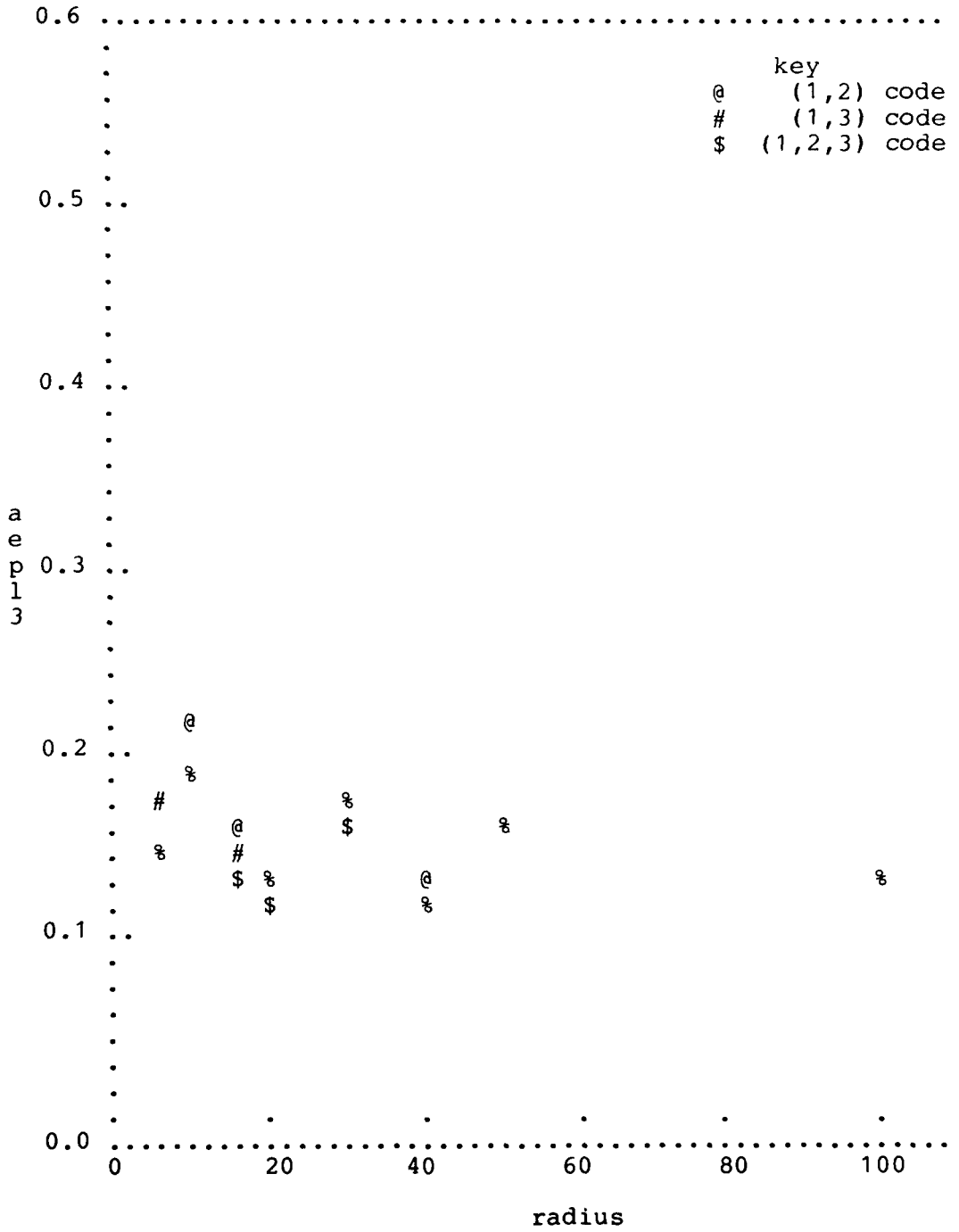

0.20
0.10
0.0
cepl
2
1
3
4,6
5
0
10000
20000
30000
40000
50000
60000
x

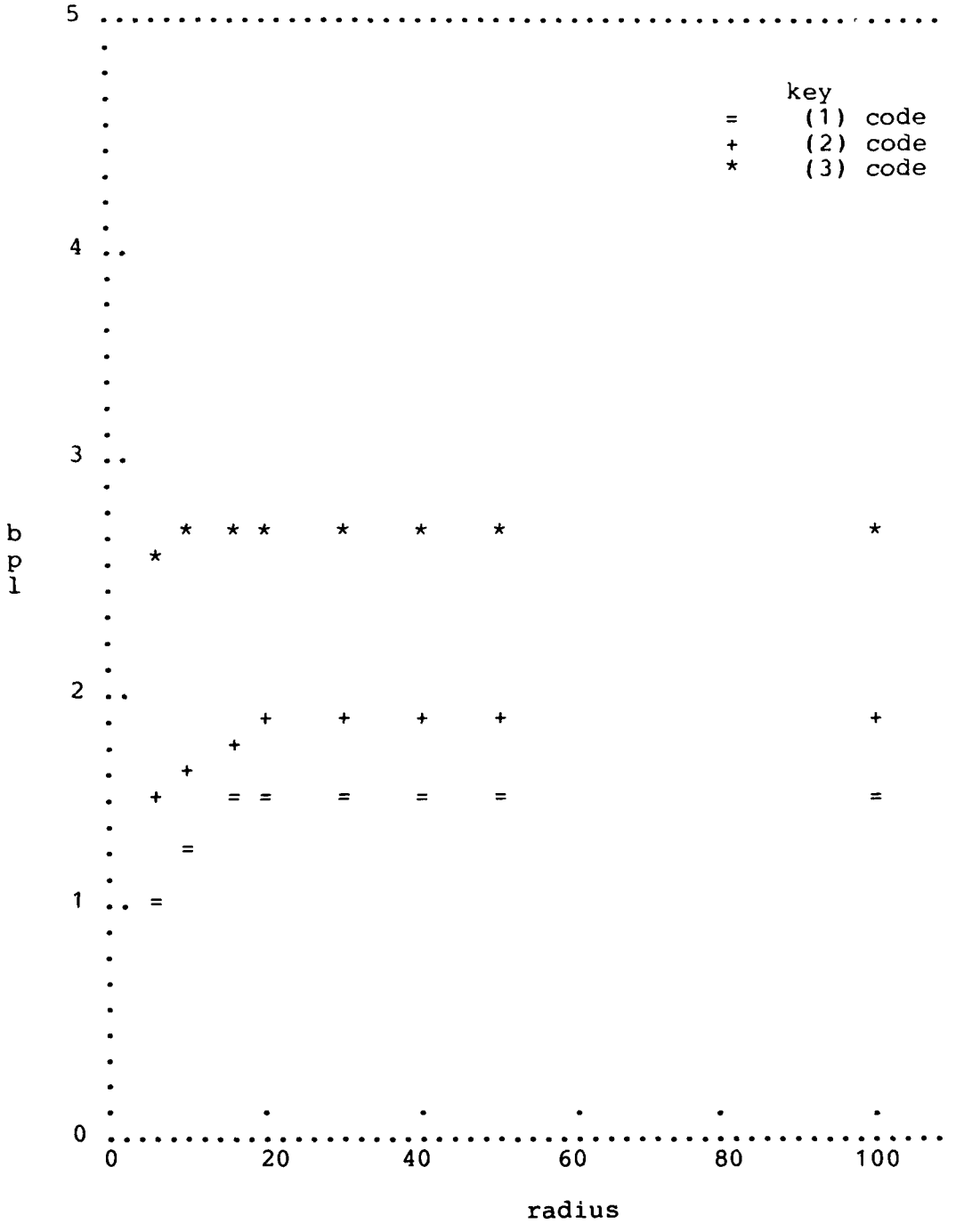

3.0
2.0
1.0
0.0
bpl
1,4,5
6
2
3
0
10000
20000
30000
40000
50000
60000
x

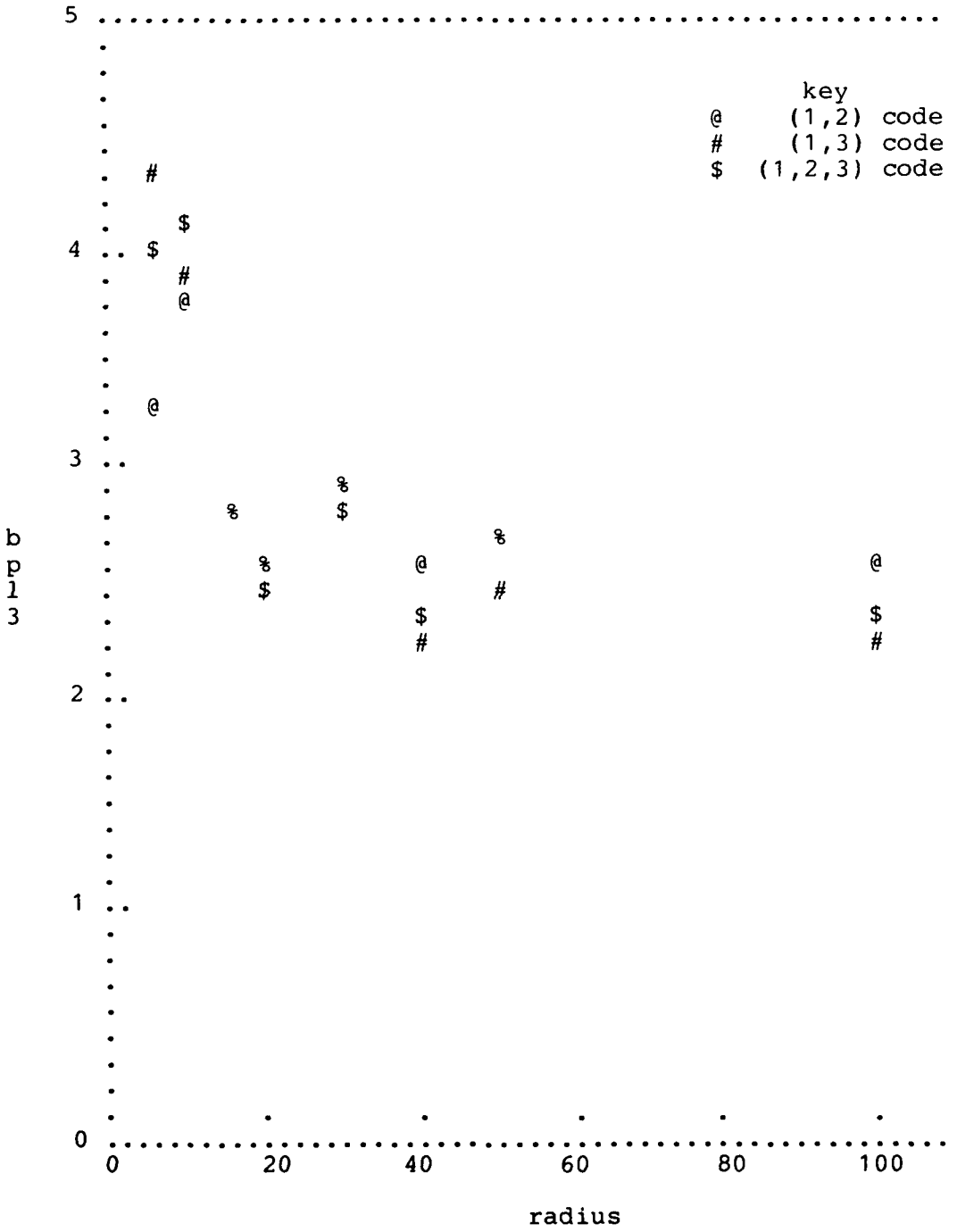

aepl
0.0 0.05 0.10 0.15 0.20
3,1
2
4
6
5
0 20001 40001 60002 80002 100003
x

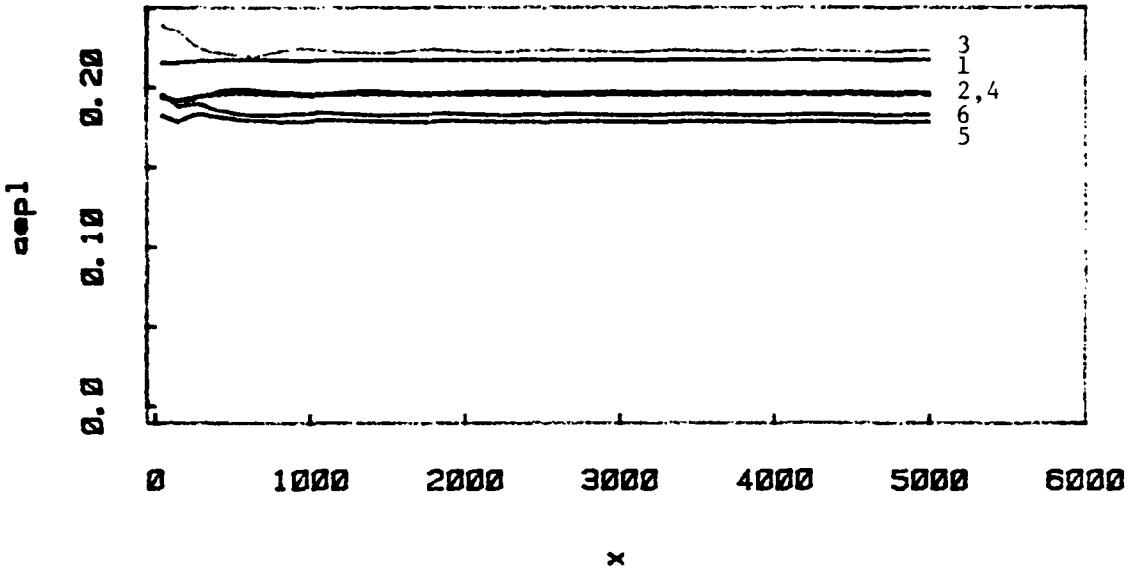

bpl
0.0 1.0 2.0 3.0
1
4
5
6
2
3
0 20001 40001 60002 80002 100003
x

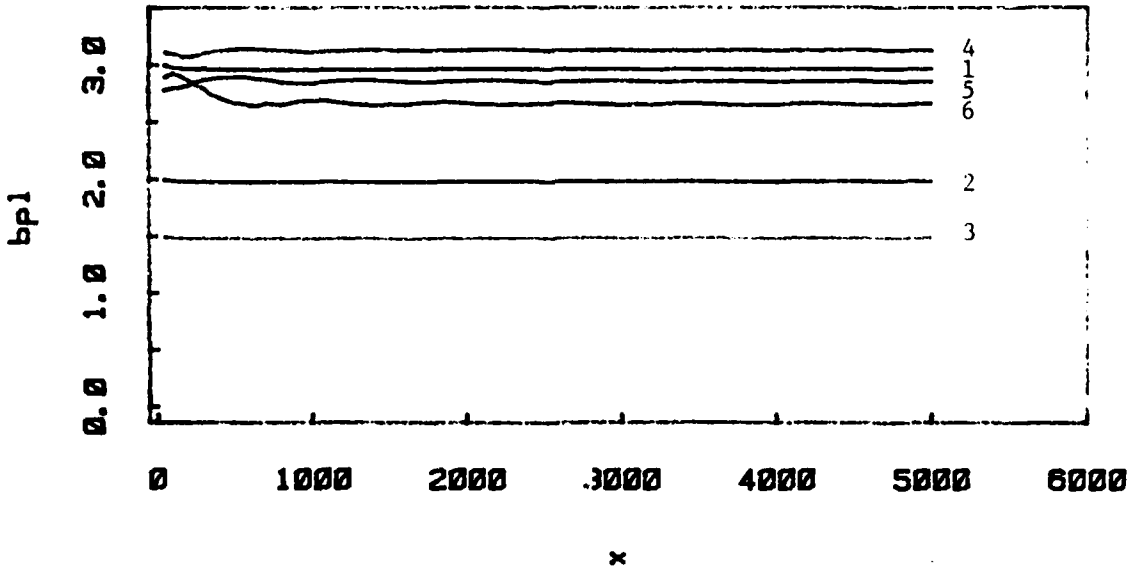

key
= (1) code
+ (2) code
* (3) code
0.6
0.5
0.4
0.3
0.2
0.1
0.0
a
e
p
l
0
20
40
60
80
100
radius

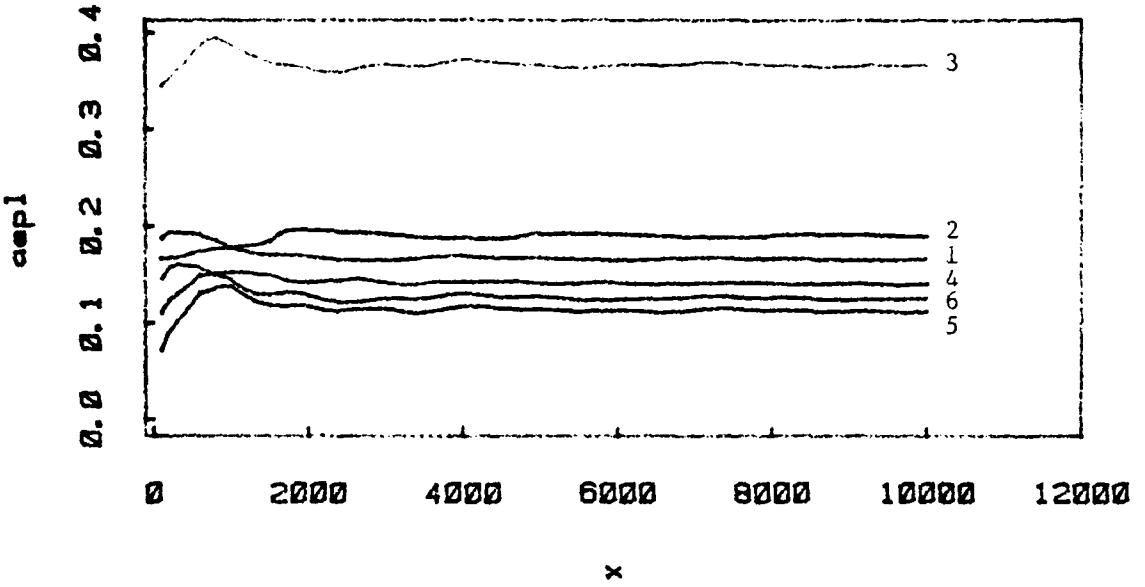

key
@ (1,2) code
# (1,3) code
$ (1,2,3) code
0.6
0.5
0.4
0.3
0.2
0.1
0.0
aepl3
0
20
40
60
80
100
radius

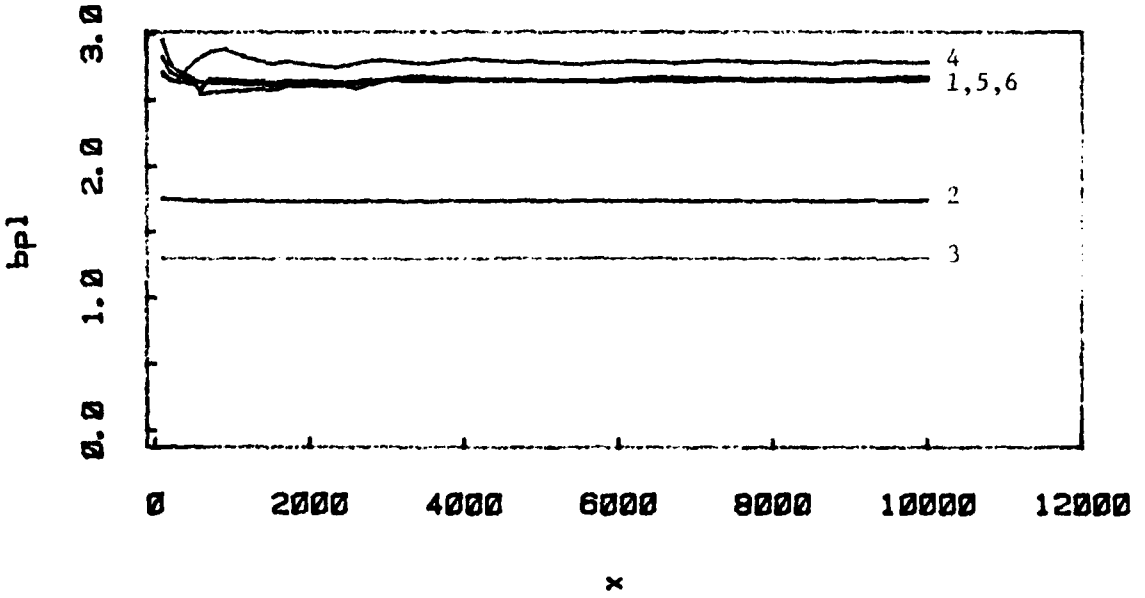

key
= (1) code
+ (2) code
* (3) code
5
4
3
2
1
0
b
p
l
0
20
40
60
80
100
radius

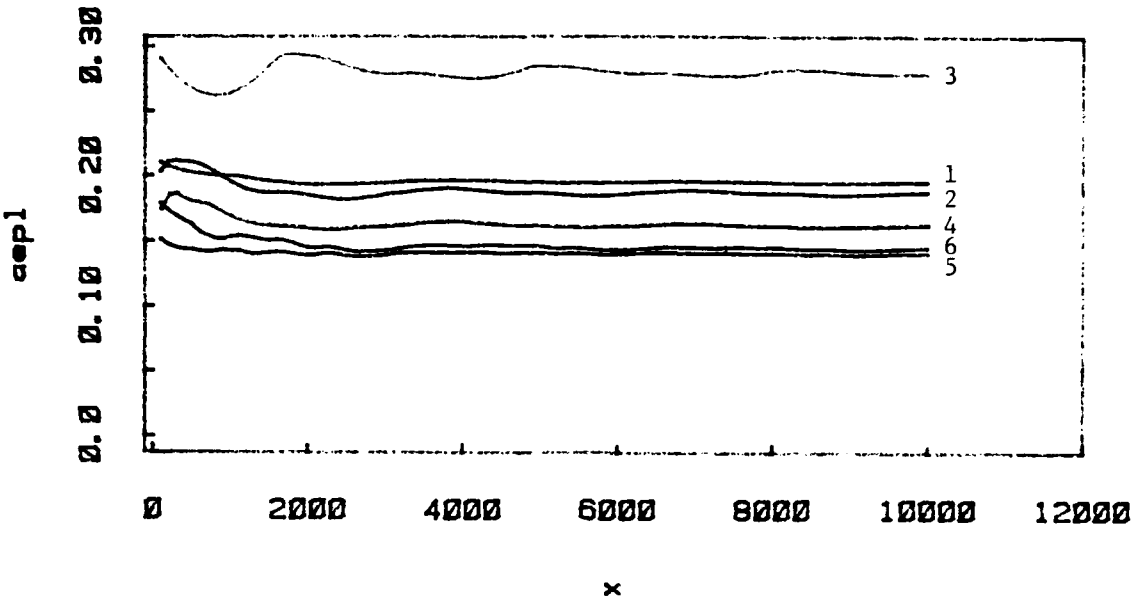

key
@ (1,2) code
# (1,3) code
$ (1,2,3) code
b
p
l
3
5
4
3
2
1
0
0
20
40
60
80
100
radius

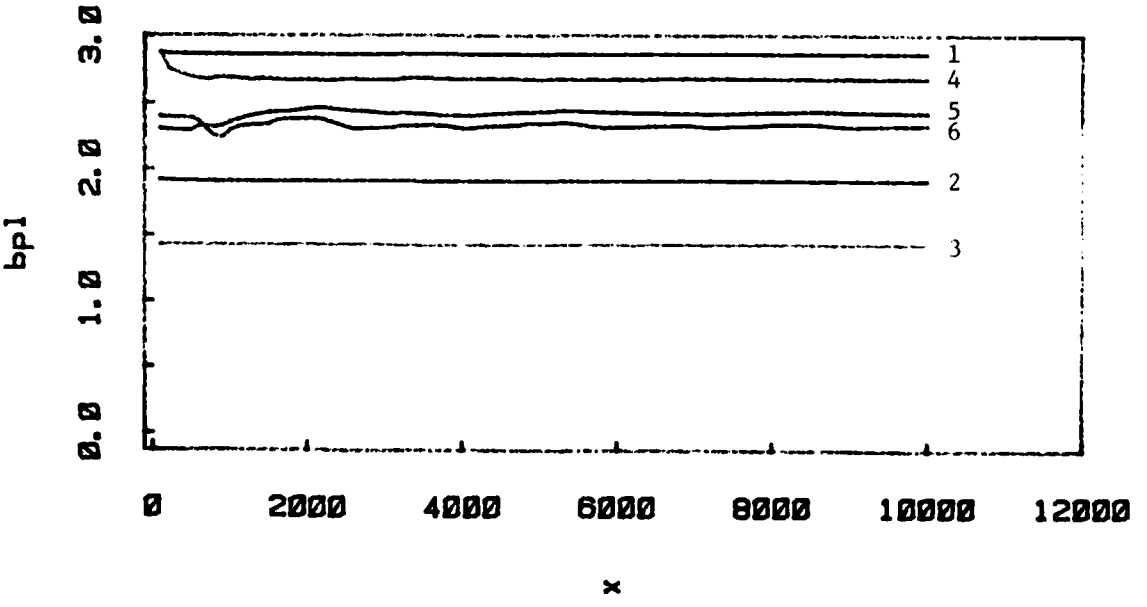

3
1
2,4
6
5
0.0
0.10
0.20
0
1000
2000
3000
4000
5000
6000
x

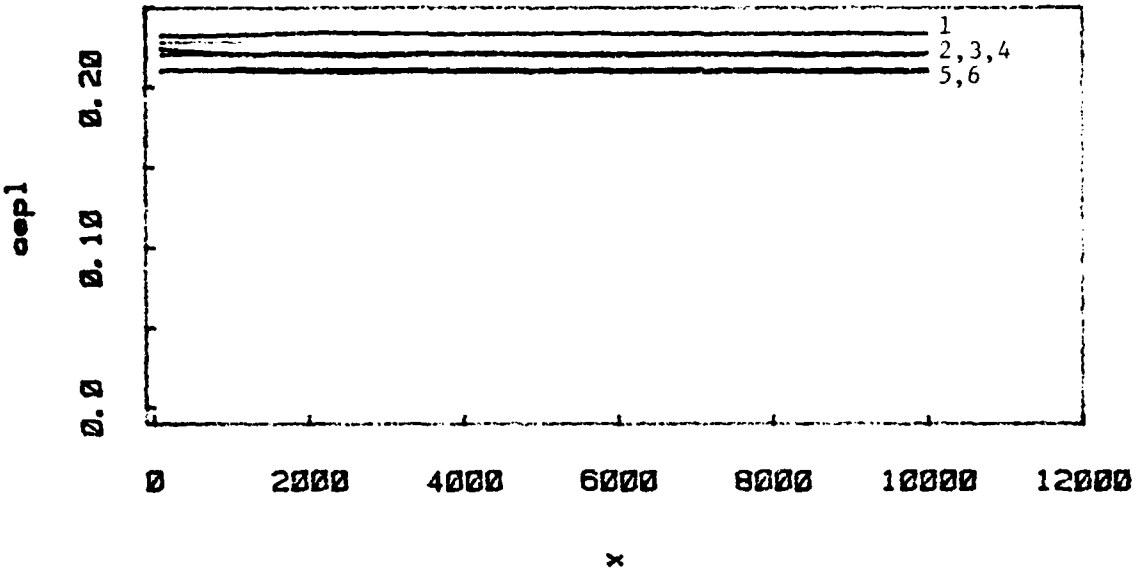

4
1
5
6
2
3
0.0
1.0
2.0
3.0
0
1000
2000
3000
4000
5000
6000
x

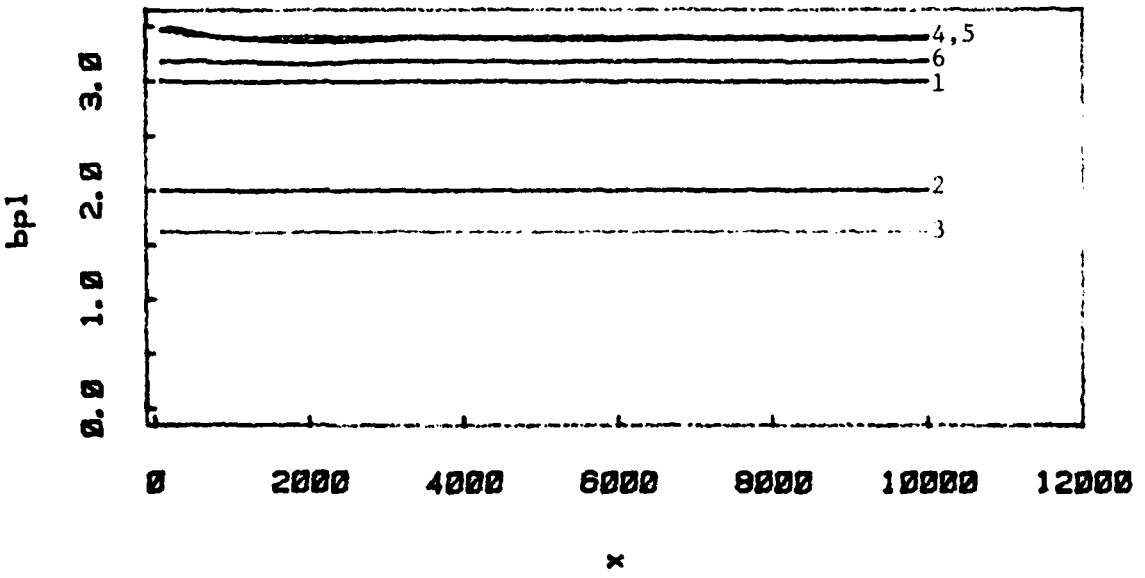

aepl
0.0
0.1
0.2
0.3
0.4
3
2
1
4
6
5
0
2000
4000
6000
8000
10000
12000
x

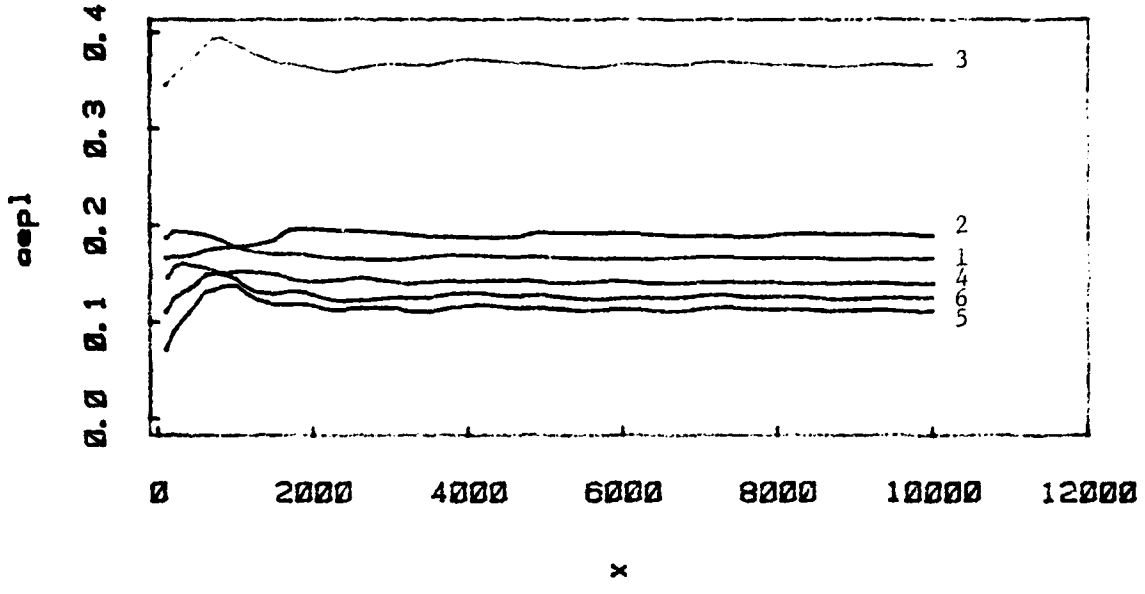

bpl
0.0
1.0
2.0
3.0
4
1,5,6
2
3
0
2000
4000
6000
8000
10000
12000
x

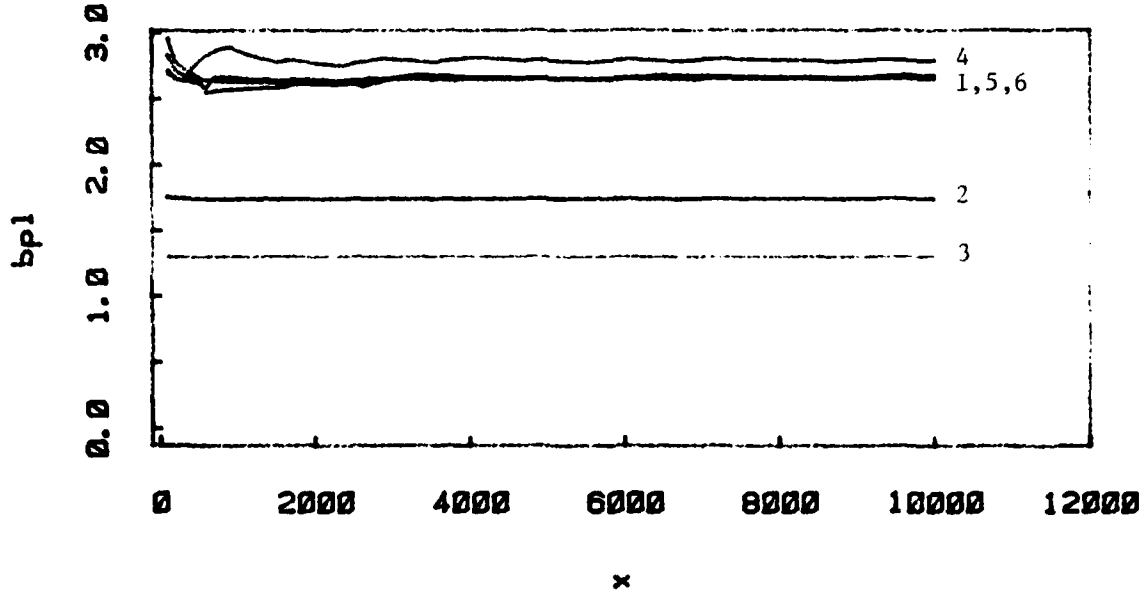

aapl
0.0
0.10
0.20
0.30
0
2000
4000
6000
8000
10000
12000
x
3
1
2
4
6
5

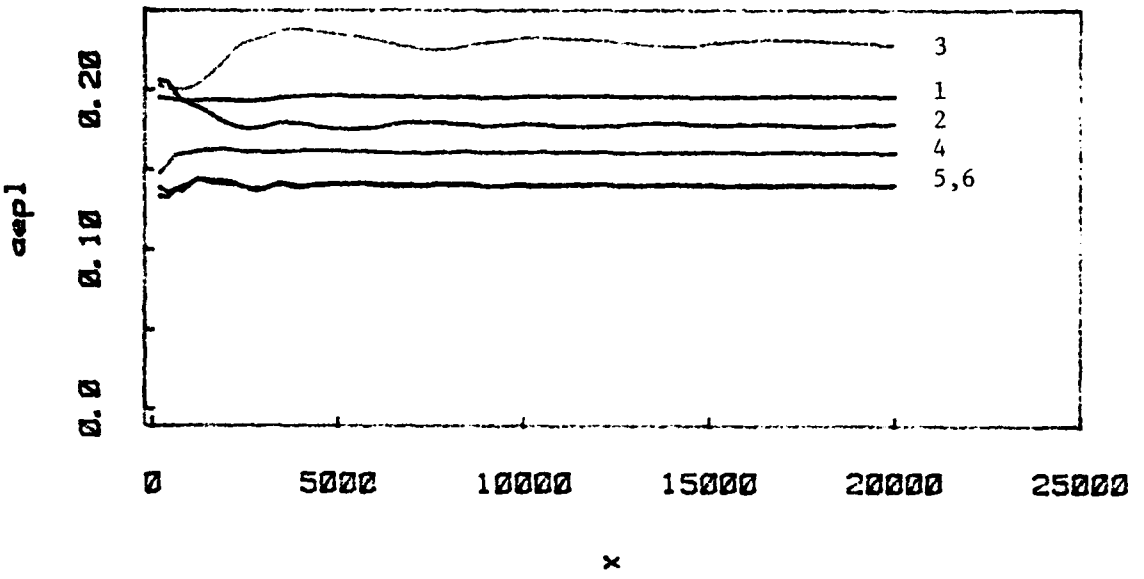

bpl
0.0
1.0
2.0
3.0
0
2000
4000
6000
8000
10000
12000
x
1
4
5
6
2
3

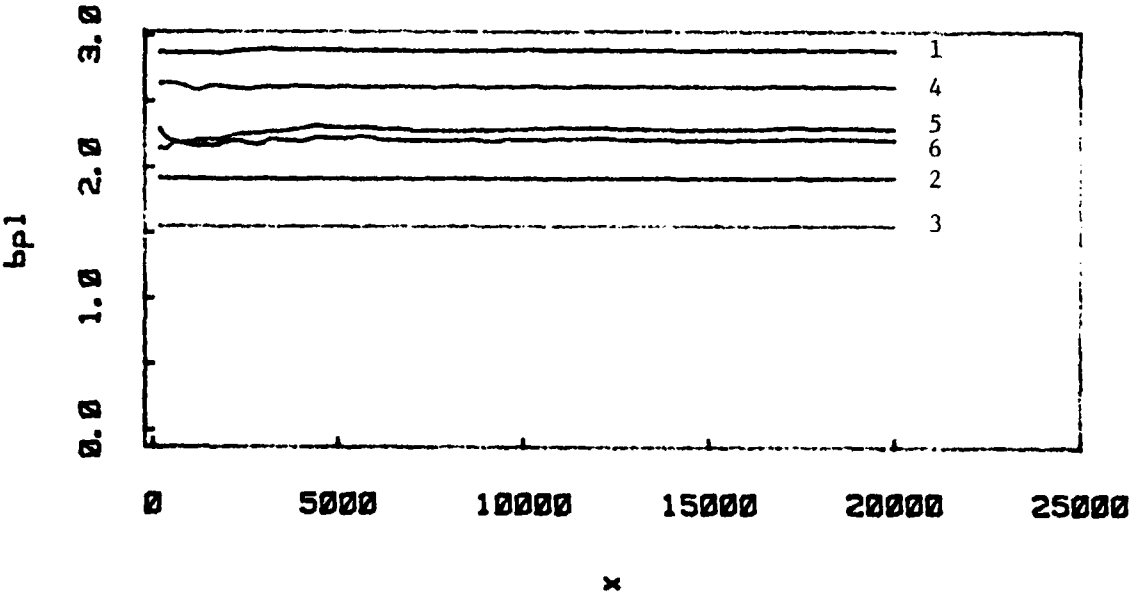

epl
0.0
0.10
0.20
1
2,3,4
5,6
0
2000
4000
6000
8000
10000
12000
x

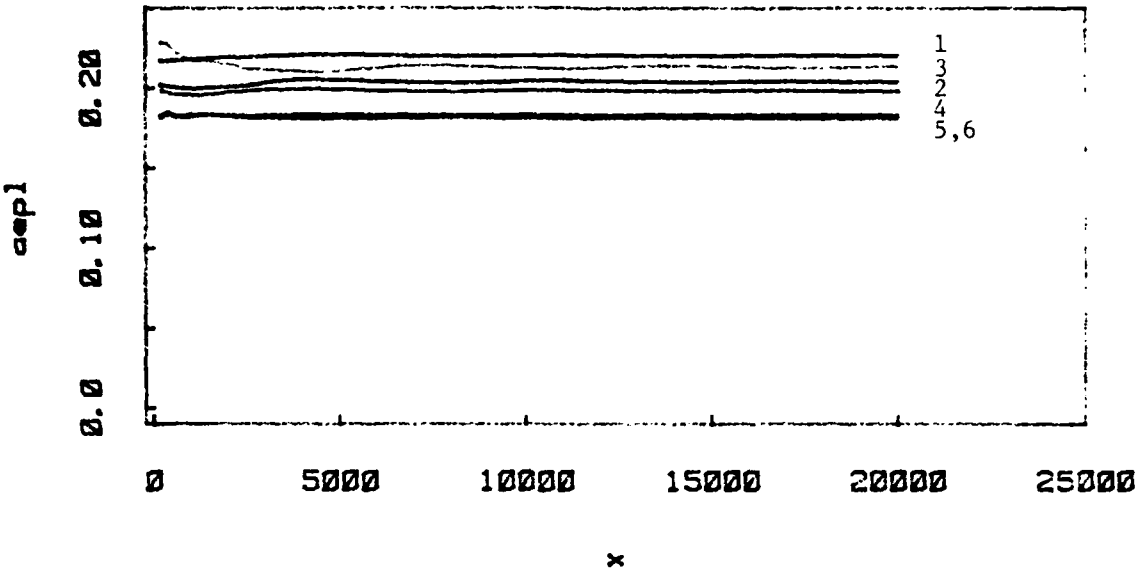

bpl
0.0
1.0
2.0
3.0
4,5
6
1
2
3
0
2000
4000
6000
8000
10000
12000
x

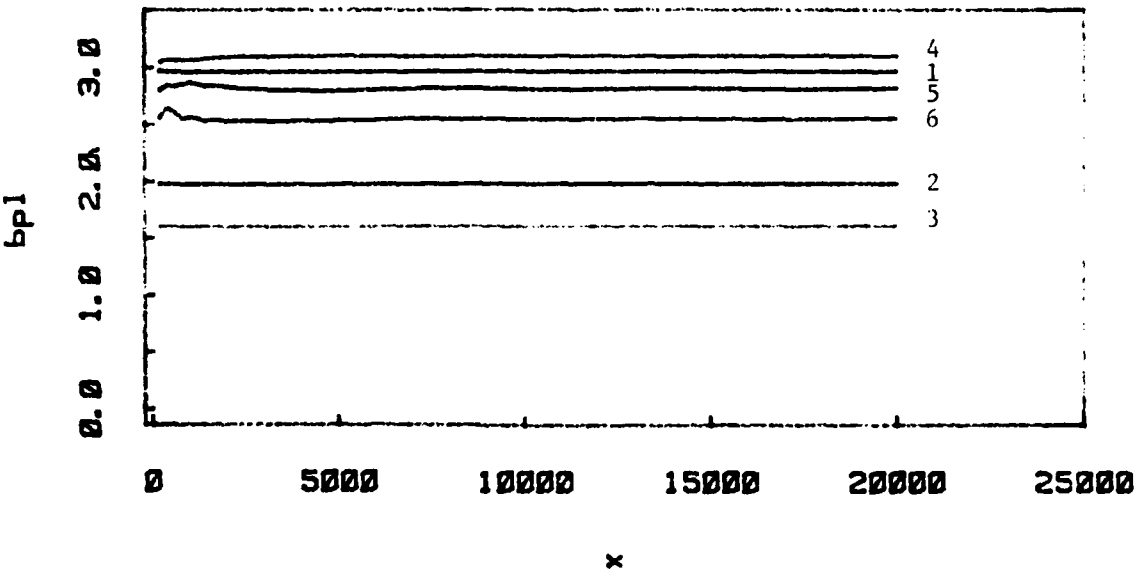

aepl
0.0
0.1
0.2
0.3
0.4
3
2
1
4
6
5
0
2000
4000
6000
8000
10000
12000
x

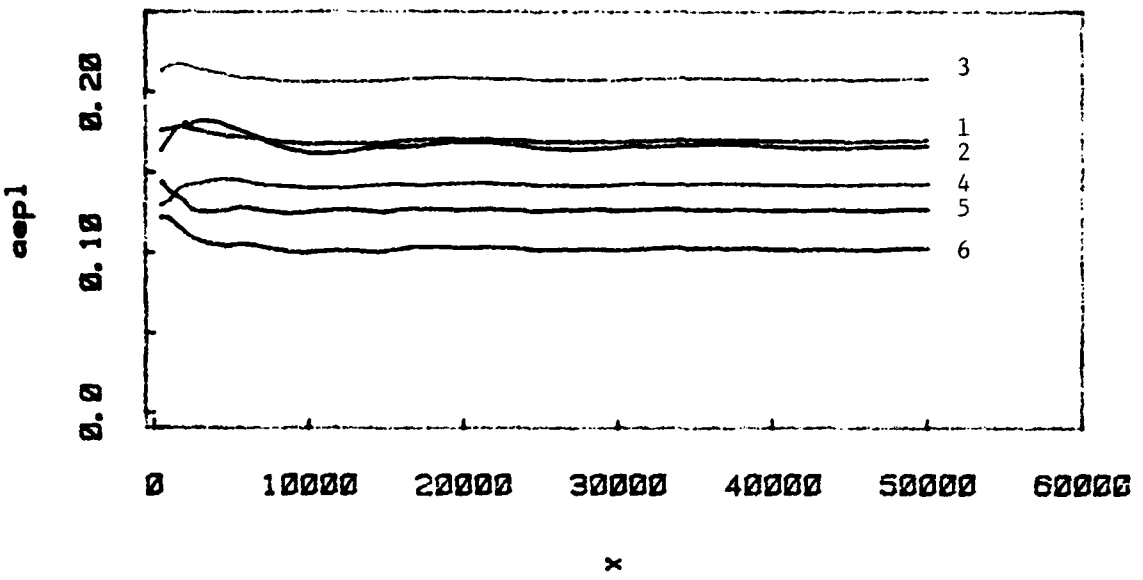

bpl
0.0
1.0
2.0
3.0
4
1,5,6
2
3
0
2000
4000
6000
8000
10000
12000
x

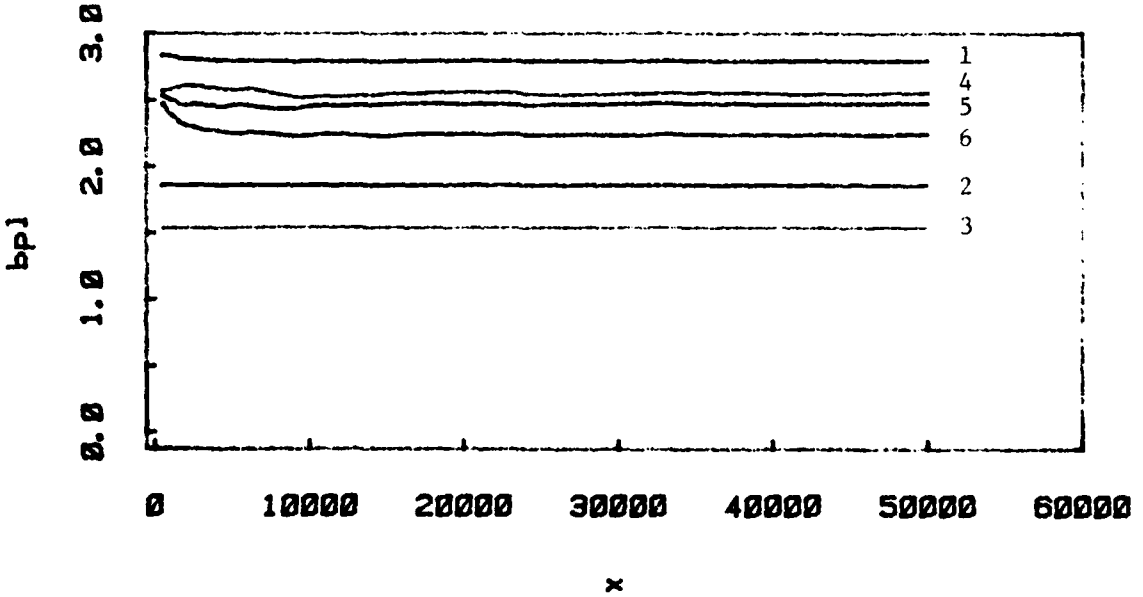

aepl
0.0
0.10
0.20
0
5000
10000
15000
20000
25000
x
3
1
2
4
5,6

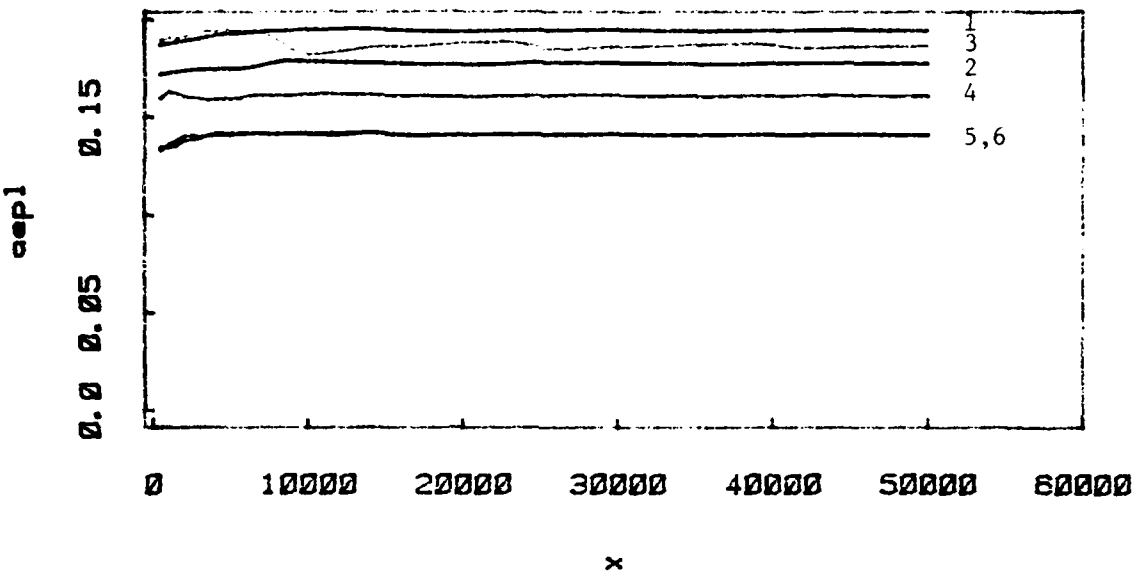

bpl
0.0
1.0
2.0
3.0
0
5000
10000
15000
20000
25000
x
1
4
5
6
2
3

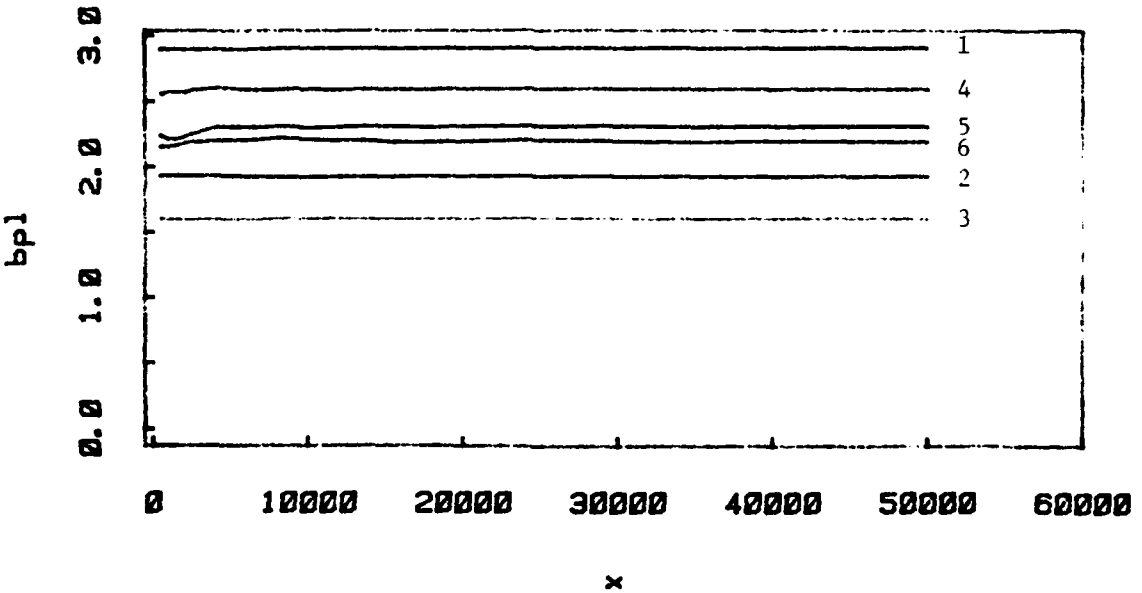

capl
0.0
0.10
0.20
0
5000
10000
15000
20000
25000
x
1
3
2
4
5,6

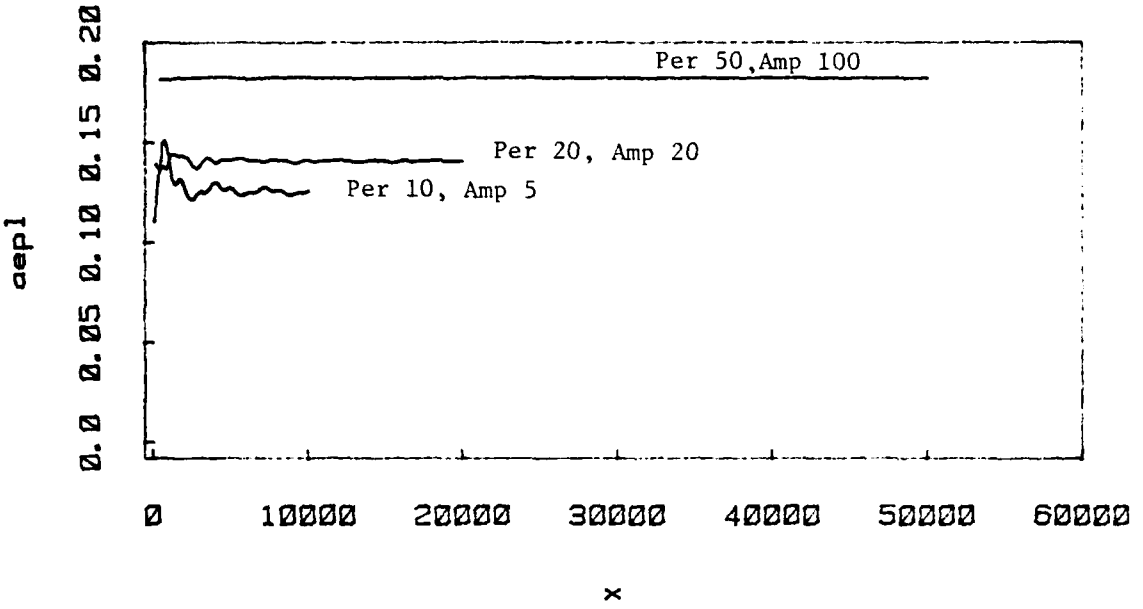

bpl
0.0
1.0
2.0
3.0
0
5000
10000
15000
20000
25000
x
4
1
5
6
2
3

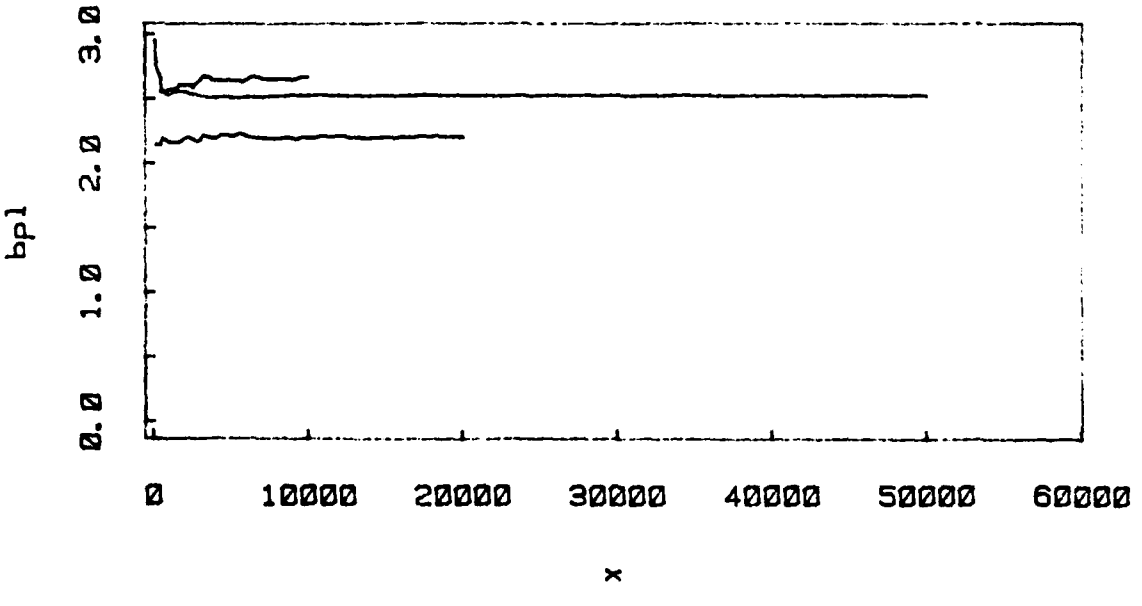

aepl
0.0
0.10
0.20
0
10000
20000
30000
40000
50000
60000
x
1
2
3
4
5
6

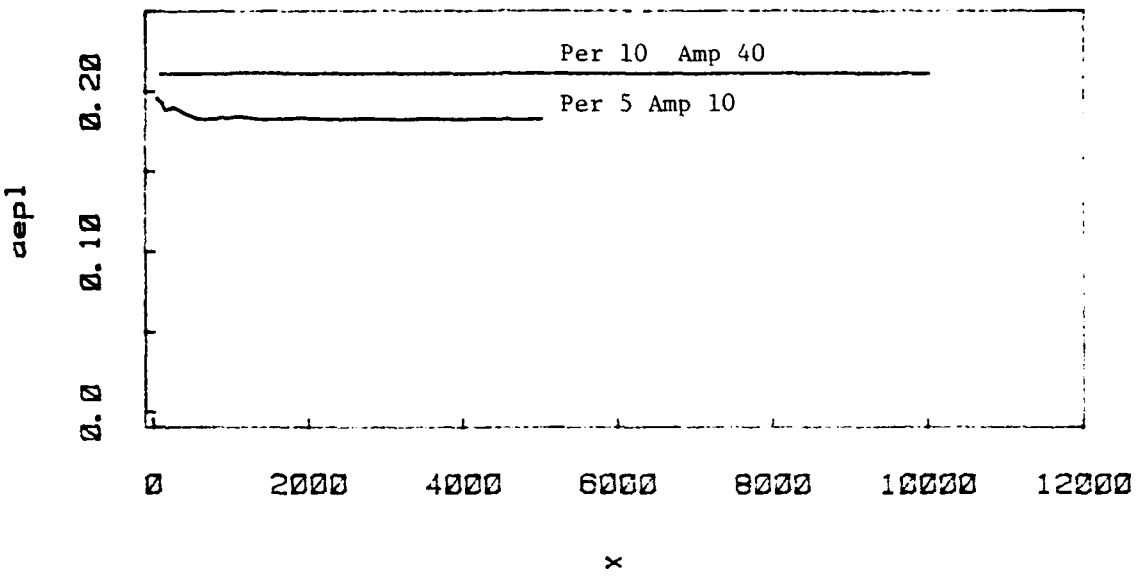

bpl
0.0
1.0
2.0
3.0
0
10000
20000
30000
40000
50000
60000
x
1
4
5
6
2
3

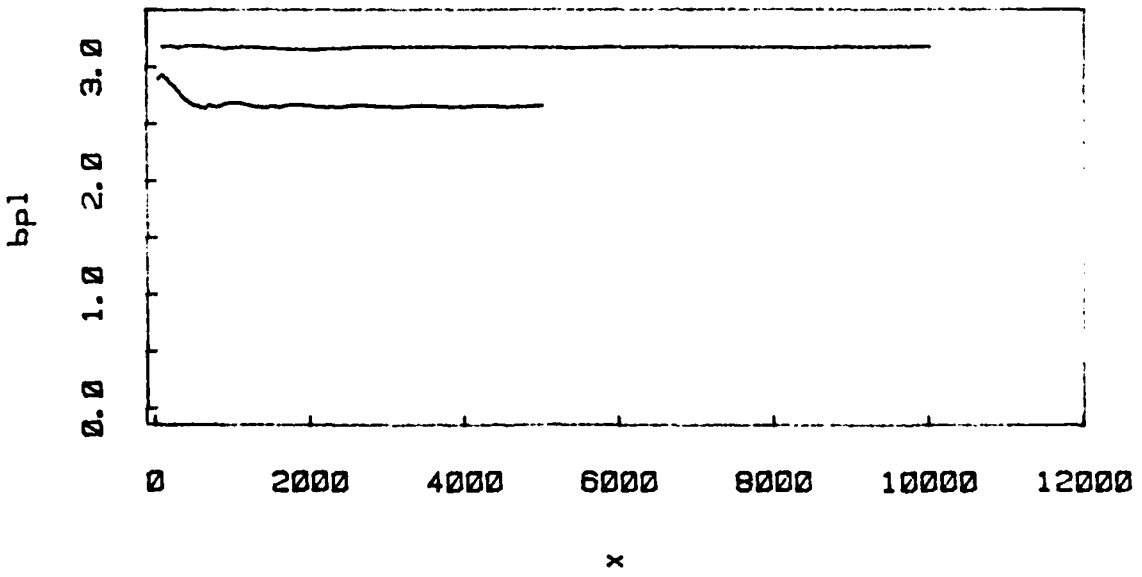

aepl
0.0
0.05
0.15
0
10000
20000
30000
40000
50000
60000
x
1
3
2
4
5,6

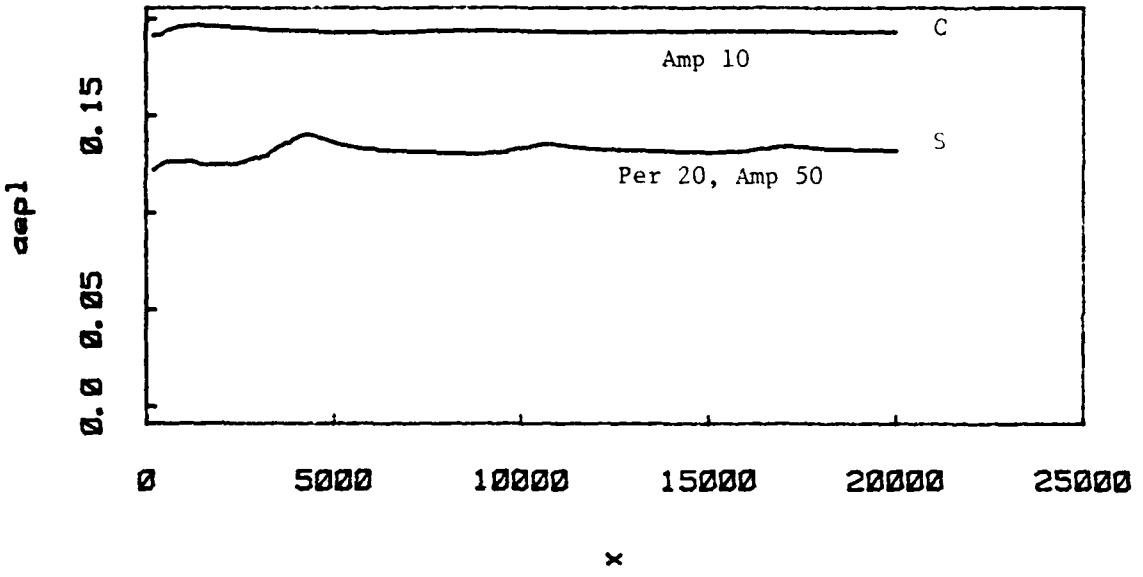

bpl
0.0
1.0
2.0
3.0
0
10000
20000
30000
40000
50000
60000
x
1
4
5
6
2
3

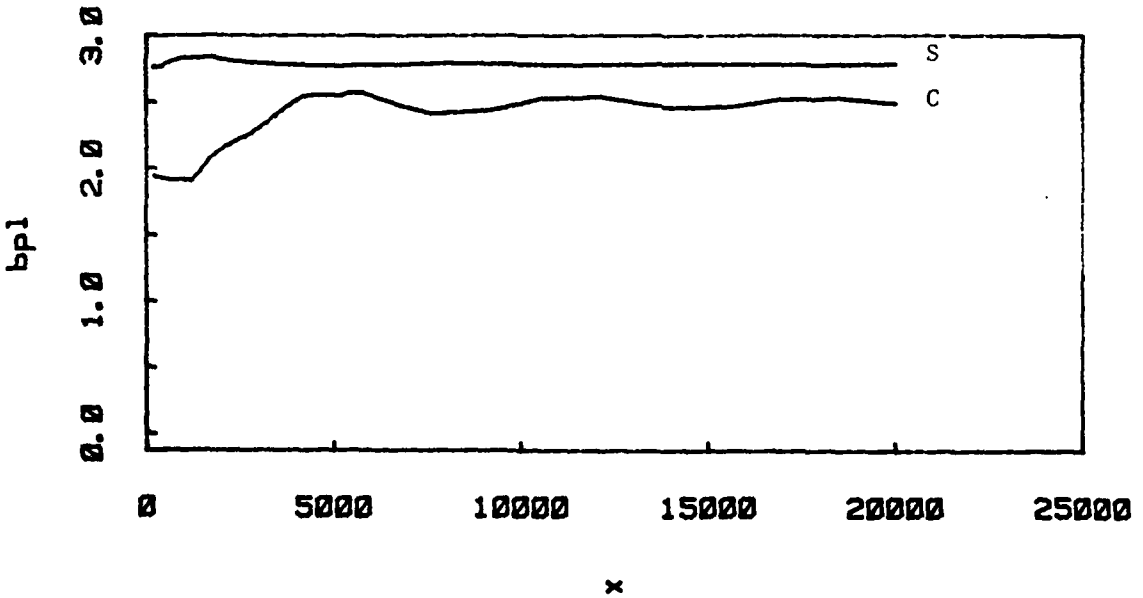

aepl
0.0 0.05 0.10 0.15 0.20
Per 50,Amp 100
Per 20, Amp 20
Per 10, Amp 5
0 10000 20000 30000 40000 50000 60000
x

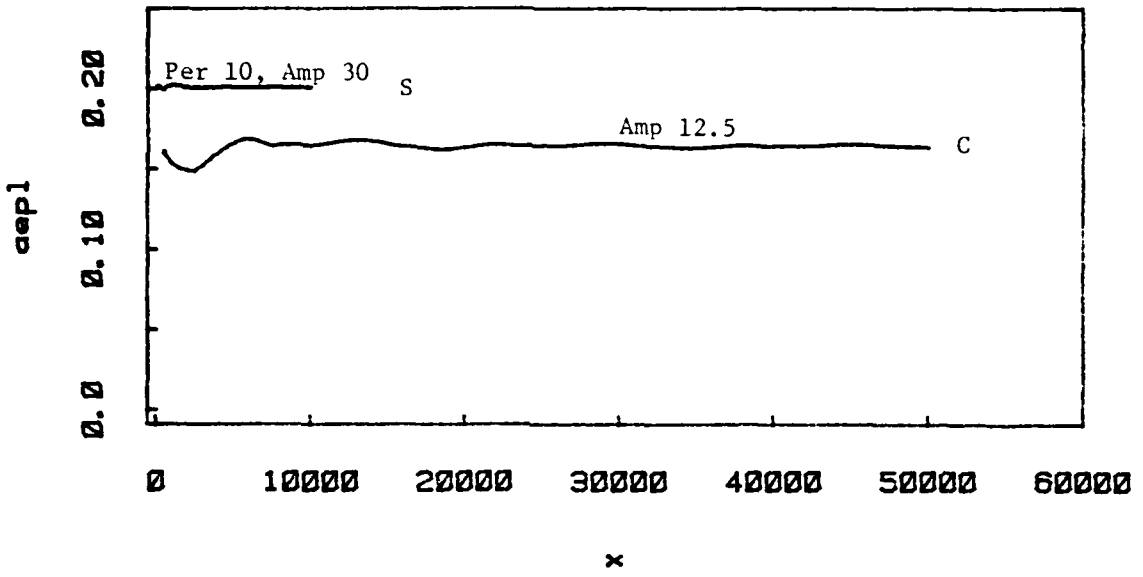

bpl
0.0 1.0 2.0 3.0
0 10000 20000 30000 40000 50000 60000
x

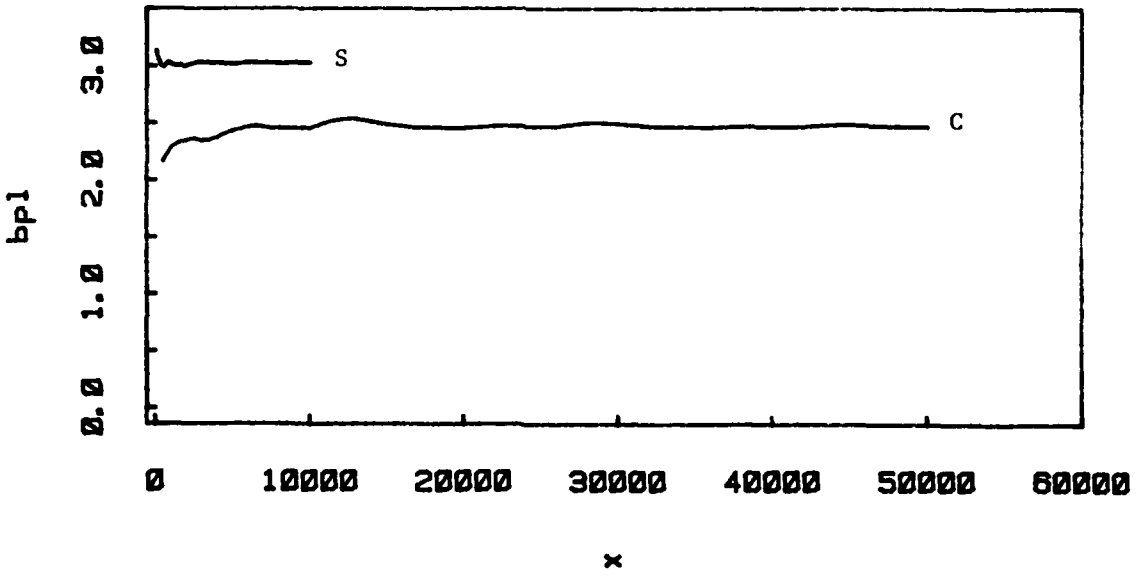

Per 10 Amp 40
Per 5 Amp 10
aepl
0.0
0.10
0.20
0
2000
4000
6000
8000
10000
12000
x

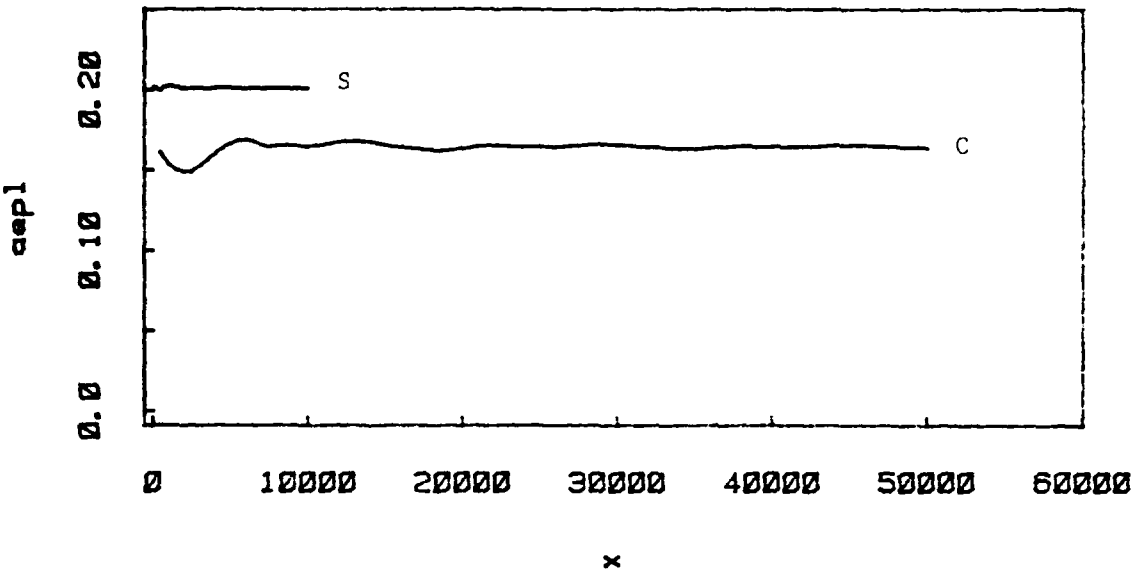

bpl
0.0
1.0
2.0
3.0
0
2000
4000
6000
8000
10000
12000
x

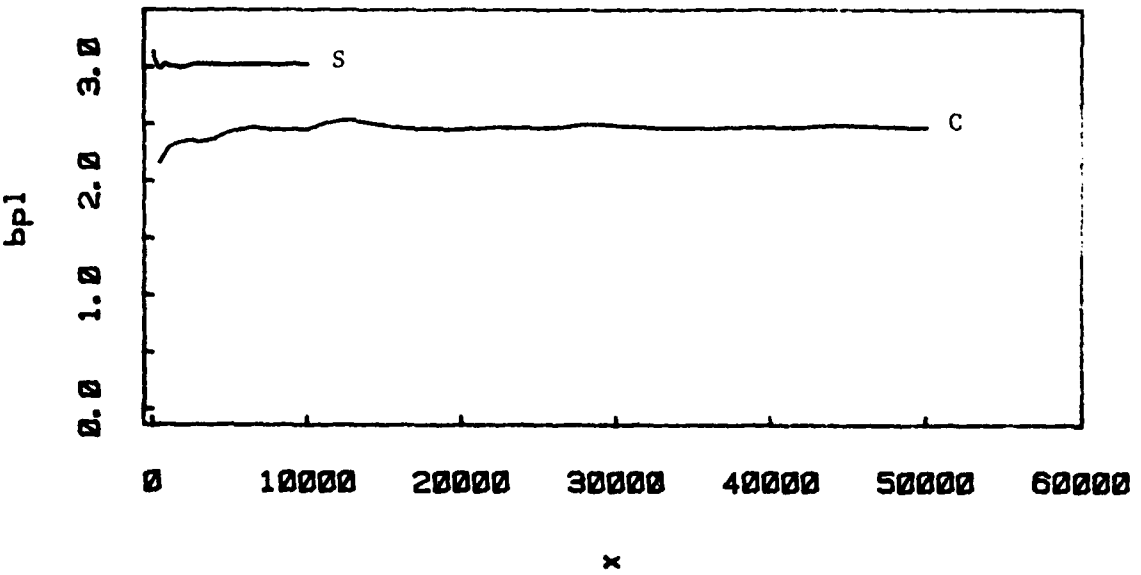

C
Amp 10
S
Per 20, Amp 50
aapl
0.0
0.05
0.15
0
5000
10000
15000
20000
25000
x


S
C
bpl
0.0
1.0
2.0
3.0
0
5000
10000
15000
20000
25000
x


Per 10, Amp 30
S
Amp 12.5
C
apl
0.0
0.10
0.20
0
10000
20000
30000
40000
50000
60000
x


S
C
bpl
0.0
1.0
2.0
3.0
0
10000
20000
30000
40000
50000
60000
x


aepl
0.0
0.10
0.20
S
C
0
10000
20000
30000
40000
50000
60000
x


bpl
0.0
1.0
2.0
3.0
S
C
0
10000
20000
30000
40000
50000
60000
x

Vita

Edward Allen Thompson was born on 23 February 1956 in Paragould, Arkansas, where he graduated from high school in 1974. He graduated from the United States Air Force Academy in 1978 with a Bachelor of Science. His first assignment was to Air Force Systems Command Space Division at Los Angeles AFS, where he was a software engineer for the Defense Dissemination System Program Office until his assignment to the School of Engineering, Air Force Institute of Technology in June 1982.

Permanent Address: Route 5, Box 275
Paragould, AR 72450

unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	DISTRIBUTION UNLIMITED
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GCS/EE/83D-20	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Air Force Institute of Technology 6b. OFFICE SYMBOL (If applicable) AFIT	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio, 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) GRID BASED LINE DRAWING QUANTIZATION (U)	
12. PERSONAL AUTHOR(S) Edward A. Thompson, Capt, USAF	
13a. TYPE OF REPORT MS Thesis 13b. TIME COVERED FROM Jun 82 TO Dec 83 14. DATE OF REPORT (Yr., Mo., Day) 1983 Dec 5 15. PAGE COUNT 82	
16. SUPPLEMENTARY NOTATION Approved for public release: IAW AFR 190-17. LYNN E. WOLAVER 7 Feb 84 Dean for Research and Professional Development Air Force Institute of Technology (AU) Wright-Patterson AFB OH 45433	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Line drawing Chain codes Computer Graphics

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper documents a quantitative analysis of the performance of the generalized chain codes when used to quantize waveforms of specific degrees of curvature. The analysis was performed using a software simulation of the chain coding system on periodic circular and sinusoid waves. The performance of the codes was measured in terms of the number of encoded bits per length of the function and the area error per length.

The analytic measure of the curvature of the circular and sine wave was developed so that the performance of the codes could be evaluated as a function of the curvature of the waveform being quantized. Comparisons on the circular wave provided a baseline of a constant curvature function. The performance of the coding system was then compared to the results when sine functions of similar curvature were encoded to determine if curvature is a valid predictor of the performance of the chain code system.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Kenneth G. Castor, Maj, USAF	22b. TELEPHONE NUMBER (Include Area Code) 513-255-5533 22c. OFFICE SYMBOL ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE