

ADAPTIVE ALGORITHMIC RESPONSE SURFACE ESTIMATION

Dr. Thomas W. Sager
The University of Texas at Austin

and

Dr. M.W. Hemphill
Texas Air Control Board
Austin, Texas

ABSTRACT

By *response surface estimation* is commonly meant the fitting of a quadratic function of two variables to a third and the graphical display of the fit by a perspective plot. PROC RSREG of SAS implements the calculation of such a fit, and PROC G3D can display the results. This paper considers a more general and more flexible procedure. Fitting quadratic response surfaces can distort the data, suppressing important real structures and creating non-existent ones, because of the limited types of surfaces that the quadratic form permits. In this paper we introduce a nonparametric response surface method that adapts to the data, rather than forcing the data to adapt to the limited quadratic model. The algorithm is implemented in SAS code that we have written. There are four major steps in the algorithm: (1) principal components transformation of the data, (2) density estimation of the principal components, (3) estimation of the response on a grid, (4) plotting the response surface with PROC G3D. The principal components transformation sidesteps the "curse of dimensionality" and thereby makes the technique computationally feasible on a routine basis. Each of the steps in the algorithm is discussed, together with some of the basic mathematical ideas underlying the technique. The technique is applied to estimating ozone as a response to two related air pollutants that are feedstocks to the process of ozone formation. The insights that can be obtained from interpretation of plots of the response surface are illustrated.

Response surface estimation aims to predict the value of one variable Y as a function of other variables $X_1, X_2, \dots, X_k$. Often, $k = 2$ and the predicted values of Y are graphed as a surface over the plane of values of X_1 and X_2 . When displayed on paper or on the monitor of a computer, visual inspection of the graph can reveal interesting features about the relationship between Y and its predictors X_1 and X_2 (cf. the Figures and discussion in the text to follow). This is particularly true if the perspective of the display can be rotated through different angles. Many commercial statistics/graphics packages now offer the capability of animating such rotations (e.g., MacSPIN and JMP for the Macintosh). Mainframe and PC-SAS offer the ability to produce a series of rotations of three-dimensional views through PROC G3D in the SAS/GRAPH module. We shall utilize PROC G3D for display of the images we produce, although our primary interest in this paper is not on the display of the images, *per se*. Our main interest here is on the calculation of the values that the image represents. SAS offers abundant capabilities to meet our programming needs.

Usually the value of Y that is predicted and graphed is an average for the population of Y 's expected to occur at the X_1 and X_2 associated with that population. Thus, response surface estimation can be viewed as a special case of the general regression model, $Y = f(X_1, X_2, \dots, X_k) + \epsilon$, where k is usually 2 and the error term ϵ is assumed to have mean zero, and, possibly, constant variance and a normal distribution.

The relationship f between Y and the X 's need not be linear, although most applications of the regression model either assume f to be linear, or take it to be linear after a transformation, or approximate it by a locally linear transformation (e.g., the routine in PROC NLIN). In SAS's implementation of response surface methodology (PROC RSREG), the relationship f is taken to be quadratic in the two predictors, X_1 and X_2 , including the cross-product term X_1X_2 between the predictors. This is the usual case in response surface methodology, and may almost be taken as a definition of the method. PROC RSREG offers a contour plot as an option, and the output of RSREG may be input into PROC G3D for a three-dimensional perspective.

Although the quadratic response models of usual response surface estimation may be adequate for some problems, the assumption of a quadratic relationship between response and predictors is often rather limiting. Unless the data actually fit one of the few forms permitted by the quadratic class, important features in the data may be crushed out of existence or artifacts be induced in the chopping and squeezing and padding required to fit the data to the Procrustean bed of a quadratic model. A more flexible class of possible relationships is re-

quired, one that will adapt to the data, rather than requiring the data to adapt so strongly to the model.

We propose a nonparametric approach to this problem. This is not a novel idea. The statistical literature on nonparametric regression is enormous. Almost every issue of a statistics journal will have at least one article related to it. What we offer here is two-fold: (1) A limited focus on two predictor variables; (2) A practical implementation that exploits some ideas for reducing computing time without gutting the spirit of the nonparametrical approach.

The practical application that motivated this work originated in contract work between the Texas Air Control Board and the Center for Statistical Sciences of the University of Texas at Austin. In Houston, monitored ambient levels of ozone regularly exceed the National Ambient Air Quality Standard. The Clean Air Act, as amended, requires States with regions in violation of the ozone standard to develop plans, called SIPs, to bring their regions into compliance. It is an objective of control strategy to develop means for bringing ozone levels into compliance. Heretofore, most control strategy, including that sanctioned by the Environmental Protection Agency, aimed to reduce ozone levels by reducing related pollutants (so-called "precursor" pollutants) that feed the ozone formation process. These strategies focussed on reducing volatile organic compounds (VOC's), represented in our work by non-methane organic compounds (NMOC), and largely ignored nitrogen oxides (NOX), another implicated set of precursors. The cost of implementing these strategies has been estimated to exceed \$100,000,000 annually in Houston. Recently, work by Sager et al. (1988, 1991) suggests that NOX may be more important than NMOC for control strategy. There are important policy implications for such a finding, for NMOC and NOX originate from different sources. Insights can be gained by displaying the relationship between ozone on the one hand and NMOC and NOX on the other as a response surface. The relationship is complex and not suitable for the limited set of models available in quadratic response surface estimation. Our adaptive response surface algorithm was motivated by the need to understand the ozone-NMOC-NOX relationship. The version discussed in this paper represents a different and improved approach over Sager et al. (1988, 1991).

In order to understand the algorithm, it is desirable to understand some of the mathematical ideas underlying it. We shall denote the response variable by Z and the two predictors by X and Y . For specificity, the reader may think of Z as ozone, X as NMOC, and Y as NOX, although the discussion applies generally. The data for the problem consist of 365 triples (X, Y, Z) measured in Houston during the four months of high ozone (June through September)

for five-year period 1985-1989. We want to estimate the average (mean) value of Z that corresponds to every possible combination of X and Y , including combinations (X,Y) that were never measured in the data. In practice, it suffices for graphical displays to limit the calculation to a reasonable number of combinations that encompass the range of all the combinations that were actually measured. We have found that for a wire-frame display, such as is available in PROC G3D, a grid formed from about 60 evenly spaced cells for X and the same for Y provides adequate detail without becoming too busy. This means 61 tick-marks per dimension, so $61 \times 61 = 3,721$ (X,Y) combinations, for each of which Z must be estimated.

To say that we want to estimate the mean of Z at each such (X,Y) pair means that we want to estimate the conditional mean of Z given X and Y , i.e., $E(Z|X, Y)$. Mathematically, this is $\int z f(z|x, y) dz$, where $f(z|x, y)$ is the probability density function of the conditional distribution of Z given X and Y . But $f(z|x, y) = g(x,y,z)/h(x,y)$, where $g(x,y,z)$ is the joint probability density function of (X,Y,Z) and $h(x,y)$ is the joint probability density function of (X,Y) alone (without Z). Thus, if we can estimate these two joint density functions, we can calculate $E(Z|X,Y)$ by numerical integration as $\sum_i g(x,y,z_i)/h(x,y) \Delta z$, where the summation can be thought of as proceeding up a column of small cubes stacked on top of the given (X,Y) cell in the plane and Δz is the length of each cell in the Z dimension.

By this device, the nonparametric regression problem becomes a problem in nonparametric density estimation. Again, this is not an original idea with us, and the statistical literature on density estimation is quite large, although not as large as that on nonparametric regression (cf. Silverman, 1986). Out of the available density estimation methods, we selected the kernel method, which is the most commonly used and the most investigated. The kernel method estimates the density at any point as a weighted average of the distance from the point to the data, with the weights decreasing as the distance from the data increases. But the kernel and all other density estimation methods suffer from the "curse of dimensionality", which means that the straightforward and unmodified application of the estimator requires computational resources that increase exponentially with the dimension of the density. Estimation of a one-dimensional density is relatively quick. Estimation of the density of a pair (such as (X,Y)) is not too bad. But higher-dimensional densities are very taxing. For example, to compute the density of (X,Y,Z) in a straightforward manner with the kernel method on a $61 \times 61 \times 61$ grid would require 226,981 computations, each being a sum of distances to n data ($n = 365$ in the ozone example). (Cf. Scott, 1992, for difficulties of multidimensional density estimation.)

We chose to simplify this process by looking for interesting ways to decompose the density. In particular, it is known that if X , Y , and Z are independent, then their joint density $g(x,y,z)$ factors into the product of the densities of the three individual variables. Independence would afford an enormous simplification, for the computation of three univariate densities is much faster than the computation of one three-variable density (e.g, $61 + 61 + 61 = 183$ computations rather than $61 \times 61 \times 61 = 226,981$). Unfortunately, ozone, NMOC, and NOX are not independent. However, any set of variables can be linearly transformed into a set of new, uncorrelated variables. Although uncorrelated variables are not necessarily independent, they are often approximately independent. (In fact, if they are normally distributed, then uncorrelated variables *are* independent.) We therefore chose to replace our original set of (X,Y,Z) with a new set of three variables (U,V,W) , each of which is a linear combination of (X,Y,Z) and each of which is uncorrelated with the other two. We then treat (U,V,W) as though they are independent, compute the density of each separately, and multiply the three separate densities to obtain the joint density of (U,V,W) . The result of this is not, however, the joint density of (X,Y,Z) , because in the transformation, the space of (X,Y,Z) is stretched into the space of (U,V,W) , and this stretches the density as well. But the density of (X,Y,Z) can be obtained by multiplying the density of (U,V,W) by an adjustment factor, called the Jacobian of the transformation. Since the transformation is linear, the Jacobian is a constant and is easy to compute. We also separately transformed (X,Y) , without Z , in a similar manner to obtain two additional uncorrelated variables that we treated as independent in order to get the density $h(x,y)$ of (X,Y) .

This means that our response surface estimation algorithm requires the computation of five one-dimensional densities, rather than one three-dimensional and one two-dimensional density. But there are many (in fact, infinitely many) ways to transform (X,Y,Z) linearly into uncorrelated variables. Out of the available candidates, we chose to use the principal components provided by PROC PRINCOMP. Principal components are convenient because they are already provided in SAS, and they are commonly used for getting uncorrelated predictors in ordinary least squares regression. PROC PRINCOMP provides as output not only a dataset containing the values of the transformed data, but also a dataset containing the scoring coefficients so that other data sets can be transformed. It should be noted that PROC PRINCOMP by default standardizes the data to zero mean and unit variance before producing the principal components. So the original (X,Y,Z) data first undergo a linear transformation in the form of standardization, then the results undergo a second linear transformation that yields the principal components. But a linear transformation of a linear transformation results in a linear trans-

formation of the original data. So the density of (X,Y,Z) can still be obtained from that of (U,V,W) by the transformation of variable technique with appropriate Jacobian.

Other ideas may be used to find interesting transformations of the data. One that is undergoing a lot of investigation currently in the statistical literature is projection pursuit. This is a computer-intensive technique that attempts to find structure in multivariate data by looking for (pursuing) linear combinations (projections) of the data on which interesting structures are revealed. This technique has proven successful in identifying multiple modes in data.

After obtaining the principal components, our algorithm next runs the univariate kernel density estimator on each of the five transformed variables. The kernel density estimate at any given point u can be calculated as $\hat{f}(u) = (nh)^{-1} \sum_{i=1}^n K((u_i - u)/h)$. In this expression, n is the sample size ($n = 365$ in our ozone example), the u_i 's ($u_1, \dots, u_n$) are the transformed data for one of the five transformed variables, K is a kernel function, and h is a "tuning" constant.

The kernel function provides weights for the data, depending on the distance from u to the data. K is a nonnegative function which integrates to 1 and is usually symmetric. Thus, any probability density function could be a kernel. A popular kernel is the standard normal probability density function. But we chose to use the quartic, or biweight, kernel, which has the advantages of computational simplicity, because of its finite support and simpler functional form (i.e., no exponentials), and properties similar to the normal kernel. The formula for this kernel is $K(x) = 15/16(1 - x^2)^2_+$, where the "+" denotes the positive part, which assigns $K(x) = 0$ if $|x| > 1$. That is, only the data within h distance of u contribute anything to the value of $\hat{f}$, as opposed to the normal kernel, for which every datum contributes and must be calculated.

The tuning constant (or "window width" or bandwidth) h determines how rough or smooth the density estimate will be. One must be careful not to choose h too small, in which case random clumping in the data will be emphasized and the density will be too rough -- or too large, in which case real features in the data will be smashed down and the density will be too smooth. Considerable literature exists which purports to provide mathematical rules for choosing h , but none is yet as good as the eye of an experienced data analyst. Theoretically, the value of h that optimizes one measure of fit called mean integrated squared error is $(n/35)^{-1/5} [\int |f''(x)|^2 dx]^{-1/5}$ for the quartic kernel. But this expression depends on the unknown density f which is being estimated.

The SAS code for the kernel density estimator is shown below:

```

%MACRO DENSITY;
%DO VAR = 1 %TO 5;
DATA DS&VAR(KEEP=PDF&VAR);
  ARRAY F{0:&BINS} F1-F&BINS1;
  DO I=0 TO &BINS;
    F{I} = 0;
  END;
  H = &&H&VAR;
  HINVSQ= (1/H)*(1/H);
  DO UNTIL (ENDDATA);
    SET TRANDATA END=ENDDATA;
    X = BINVAR&VAR;
    DO I=0 TO &BINS;
      Z= &&BEG&VAR + I * &&DEL&VAR;
      IF ABS(Z - X) LT H THEN DO;
        F{I} = F{I} +
          (1 - (Z - X)*(Z - X)*HINVSQ) *
          (1 - (Z - X)*(Z - X)*HINVSQ)*15/16;
      END;
    END;
  END;
  DO I=0 TO &BINS;
    PDF&VAR = F{I}/&N/H;
  OUTPUT;
  END;
%END;
%MEND;

```

TRANDATA is a SAS dataset containing five columns of values. These columns are the separate transformations of (X,Y,Z,) and of (X,Y). &H1 - &H5 are the tuning constants for the five densities. &BINS is the number of regularly spaced cells, or bins, into which the macro divides the range of the five transformed variables. The macro calculates the &BINS1 (= &BINS + 1) boundary points of these cells for each of the densities from user-specified begin-

ning values (&BEG1 - &BEG5) and cell lengths (&DEL1 - &DEL5). DS1 - DS5 are five SAS datasets containing the output density estimates in the variables PDF1 - PDF5, respectively. The algorithm takes each data point (outer DO UNTIL (ENDDATA)) and finds those grid points (inner DO loop) that the datum is close to (within distance h), and adds the contribution of the datum to the running totals ($F\{I\}$) for those grid points. The algorithm is relatively efficient. Although the comparison $ABS(Z - X) \leq h$ is performed $\&N * \&BINS$ times ($= 365 * 61 = 22,265$ times in the ozone example) for each density, the DO computation that follows is performed for only a fraction of the bins --the number of DO computations depending on the selected value of h .

The third major step in the adaptive response surface algorithm is calculation of the estimated response at each (X,Y) grid point. The two Jacobians are calculated with the DETerminant function in PROC IML. The values of important constants like the Jacobians that are produced in one section of the code and used in others are placed in macro variables for global accessibility by means of the CALL SYMPUT function in a DATA step. For example, here is how the principal component coefficients for (X,Y,Z), produced by PROC PRINCOMP and stored in output SAS dataset WEIGHT3, are transferred to macro variables for use in Step 3 of the algorithm:

```
DATA _NULL_;          /* (X,Y,Z) are the original variables */
  SET WEIGHT3;         /* scored by PRINCOMP. WEIGHT3 has */
  IF _TYPE_ = 'SCORE'; /* their coefficients stored under */
  INPUT A $ B $ C $;   /* their names. The coefficient */
  CALL SYMPUT(A,X);    /* values are transferred to the macro */
  CALL SYMPUT(B,Y);    /* variables whose names are read into */
  CALL SYMPUT(C,Z);    /* the data step variables A,B,C. */
CARDS;
A31X B31Y C31Z
A32X B32Y C32Z          /* The names of SAS macro variables */
A33X B33Y C33Z          /* for the PRINCOMP coefficients. */
RUN;
```

The grid of $61 \times 61 = 3,721$ (X,Y) pairs is created and stored in a SAS dataset by the usual double nested DO loop used to create grids for PROC G3D. The five univariate densities

of the transformed variables are read into five ARRAYS. Then each of the 3,721 pairs of grid coordinates is processed: For a given grid point (x_i, y_j) , numerical integration of $E(Z|x_i, y_j)$ is performed by taking a grid of Z values, $z_1, z_2, \dots, z_m$, calculating the principal component transformation (u_i, v_j, w_k) of the standardized (x_i, y_j, z_k) , and "looking up" u_i in the ARRAY for the univariate density of U, "looking up" v_j in the ARRAY for V, "looking up" w_k in the ARRAY for W. The value of the density for the U closest to u_i is multiplied by the value of the density for the V closest to v_j , which is multiplied by the value of the density for the W closest to w_k . This lookup is direct and does not involve multiple comparisons: the value of u_i is directly converted into the index of the closest U, and this index references the correct element in the U-ARRAY. The product of these three terms, when multiplied by the appropriate Jacobian, yields the density $g(x_i, y_j, z_k)$. The same process is performed with the remaining two of the five density arrays to yield $h(x_i, y_j)$. The ratio $g(x_i, y_j, z_k)/h(x_i, y_j) = f(z_k|x_i, y_j)$. This process is repeated for each of $z_1, \dots, z_m$. Then $E(Z|x_i, y_j)$ is numerically approximated by $\sum_{k=1}^m z_k f(z_k|x_i, y_j) \Delta z$. Some obvious computational improvements are possible in this exposition of the method, for example, locating in outer loops certain operations uninvolved in the inner loop on Z .

The fourth and final step in the algorithm is to run the 3,721 (X,Y,Z) triples just produced through PROC G3D. This step is straightforward and standard. The user may wish to try additional runs, "tweaking" some of the parameters, particularly with different values for the tuning constant h and the number of bins for the density estimate. The objective should be a graph that looks neither too smooth nor too rough. We experienced additional difficulty with estimates of Z that were unbelievably high in the tails of the (X,Y) distribution, where the data are sparse and the density of (X,Y) is low. Estimated values of ozone were obtained that were several times higher than any that had been measured in the data. Investigation showed that these values were occurring where the density $h(x,y)$ was estimated to be very small. Since $h(x,y)$ enters the estimate of Z through the denominator, the result was a ballooning of the ratio. We adopted an *ad hoc* solution for this problem by censoring extremely small values of $h(x,y)$ and truncating the Z -axis of the PROC G3D display where ozone = 0.36 parts per million. (The highest measured value of ozone with a complete set of all three pollutants was .25.) A further refinement of the algorithm would address this problem in a more consistent manner. We believe that the tuning parameter h needs to be larger in the tails of the (X,Y) distribution, so that more smoothing can occur there. Such *variable kernel* methods continue to receive attention in the literature and have been advocated, for example, by Silverman (1986).

Examining the graphical output is instructive. We illustrate the interpretation here to show how useful it can be. Figure 1 shows that the highest ozone levels occur along a ridge where NOX levels are high and that these high ozone levels are present for most NMOC values. Thus the highest levels of ozone appear to depend less strongly upon NMOC than upon NOX. A secondary and much lower peak occurs where both NMOC and NOX are relatively high, around $\text{NMOC} = 2.81$ and $\text{NOX} = 11.89$. Curiously, another secondary peak occurs close to the origin, where both pollutants are absent! Chemically, higher levels of NMOC are supposed to scavenge NOX, thus depriving the ozone formation mechanism of a critical feedstock. This phenomenon is visible in Figure 2, which shows that as NMOC increases from right to left along a slice of the response surface fixed at a given value of NOX, the ozone level tends to diminish. This is manifested for slices at low levels of NOX and at high levels of NOX, but not at middle levels, where ozone first rises and then falls with increasing NMOC. Figure 3 shows that ozone ultimately rises dramatically as NOX increases at any level of NMOC. At slices for low NMOC, ozone first drops before rising; but for higher levels of NMOC, ozone rises, dips slightly, then rises sharply. Figures 2 and 3 were produced with the XYTYPE option in PROC G3D. Finally, Figure 4 (note rotation of the X and Y axes) shows that most of the data are found in the region bounded by the origin and the tick-marks at $\text{NMOC} = 1.41$ and $\text{NOX} = 11.89$.

REFERENCES

1. Sager, T. W. and Hemphill, M.W. (1988). "A surface response analysis of ozone and its precursors", paper 88-44.1. *Proceedings of annual meeting of Air Pollution Control Association*.
2. Sager, T.W., McAllister, R., Gise, J.P., and Hemphill, M.W. (1991). "Importance of NOx control suggested by NMOC-NOx-Ozone data analysis", paper 91-75.1. *Proceedings of annual meeting of Air and Waste Management Association*.
3. SAS/STAT User's Guide, Version 6 Edition. Cary, NC: SAS Institute, Inc., 1990.
4. SAS/GRAPH User's Guide, Version 6 Edition. Cary, NC: SAS Institute, Inc., 1990.

5. Scott, D. W. Multivariate Density Estimation. New York: Wiley, 1992.
6. Silverman, B. W. Density Estimation for Statistics and Data Analysis. London: Chapman and Hall, 1986.

Ozone as Function of NMOC and NOX

Houston, 1985-1989

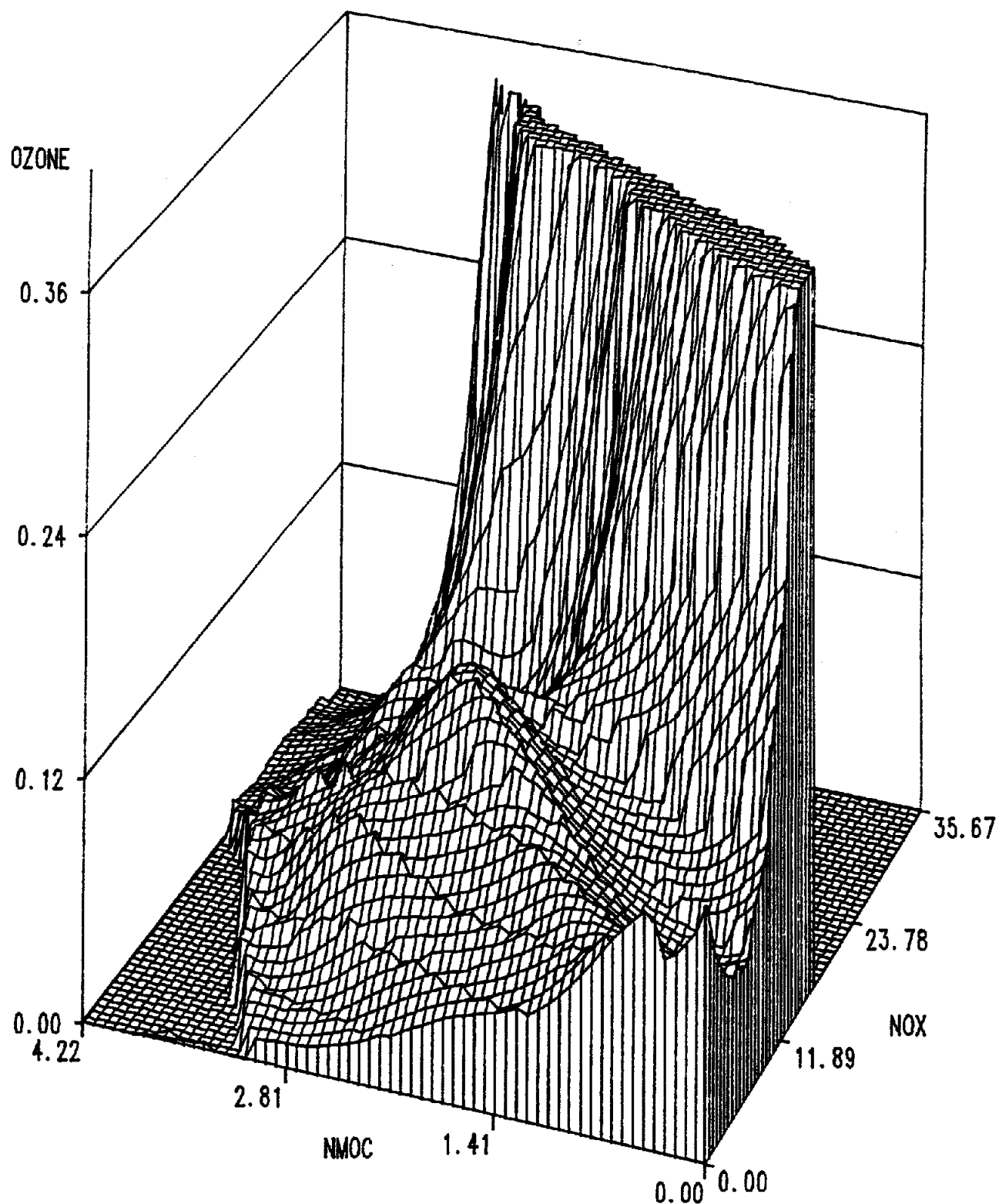


FIGURE 1

Ozone as Function of NMOC and NOX

Houston, 1985-1989

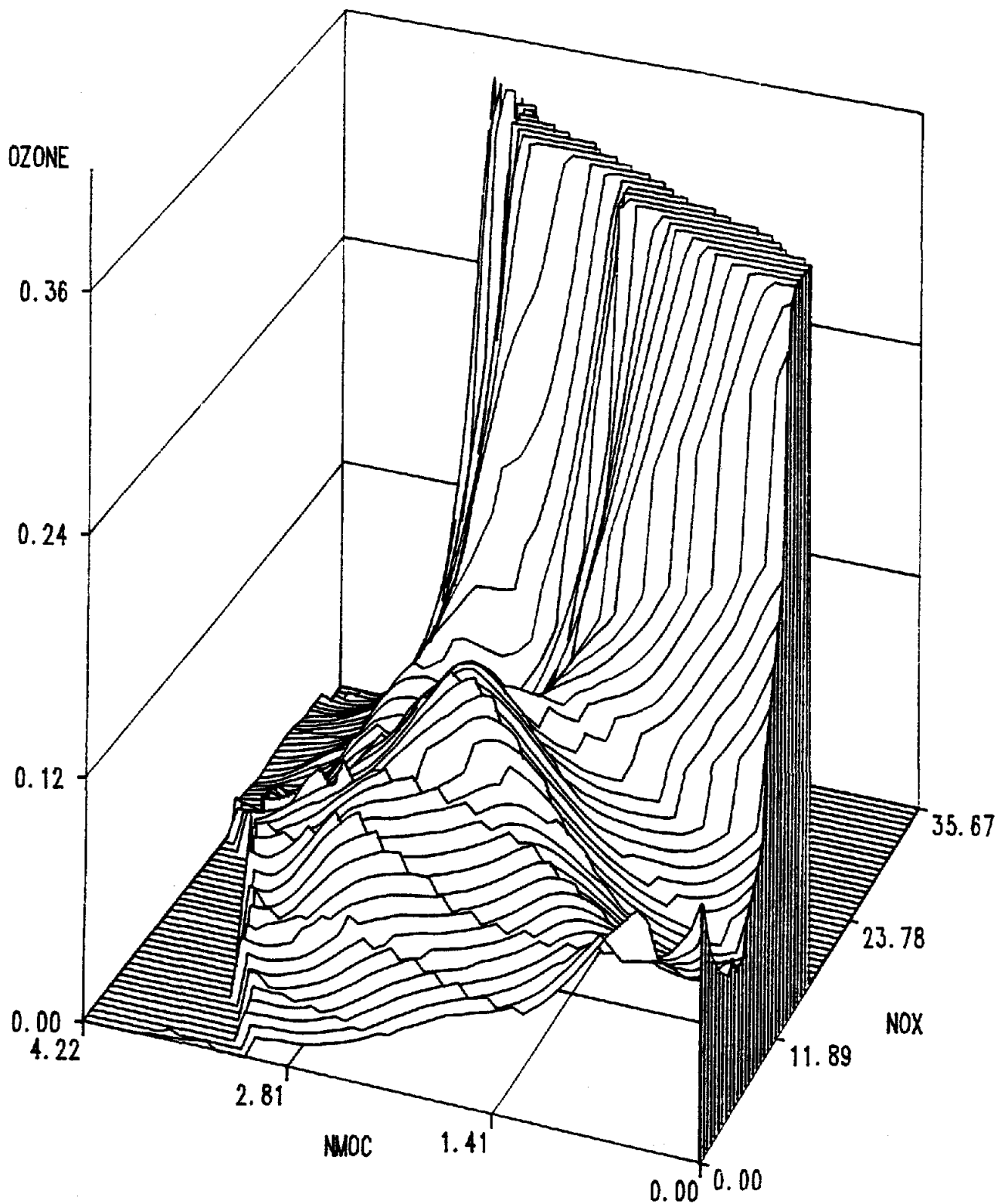


FIGURE 2
264

Ozone as Function of NMOC and NOX

Houston, 1985-1989

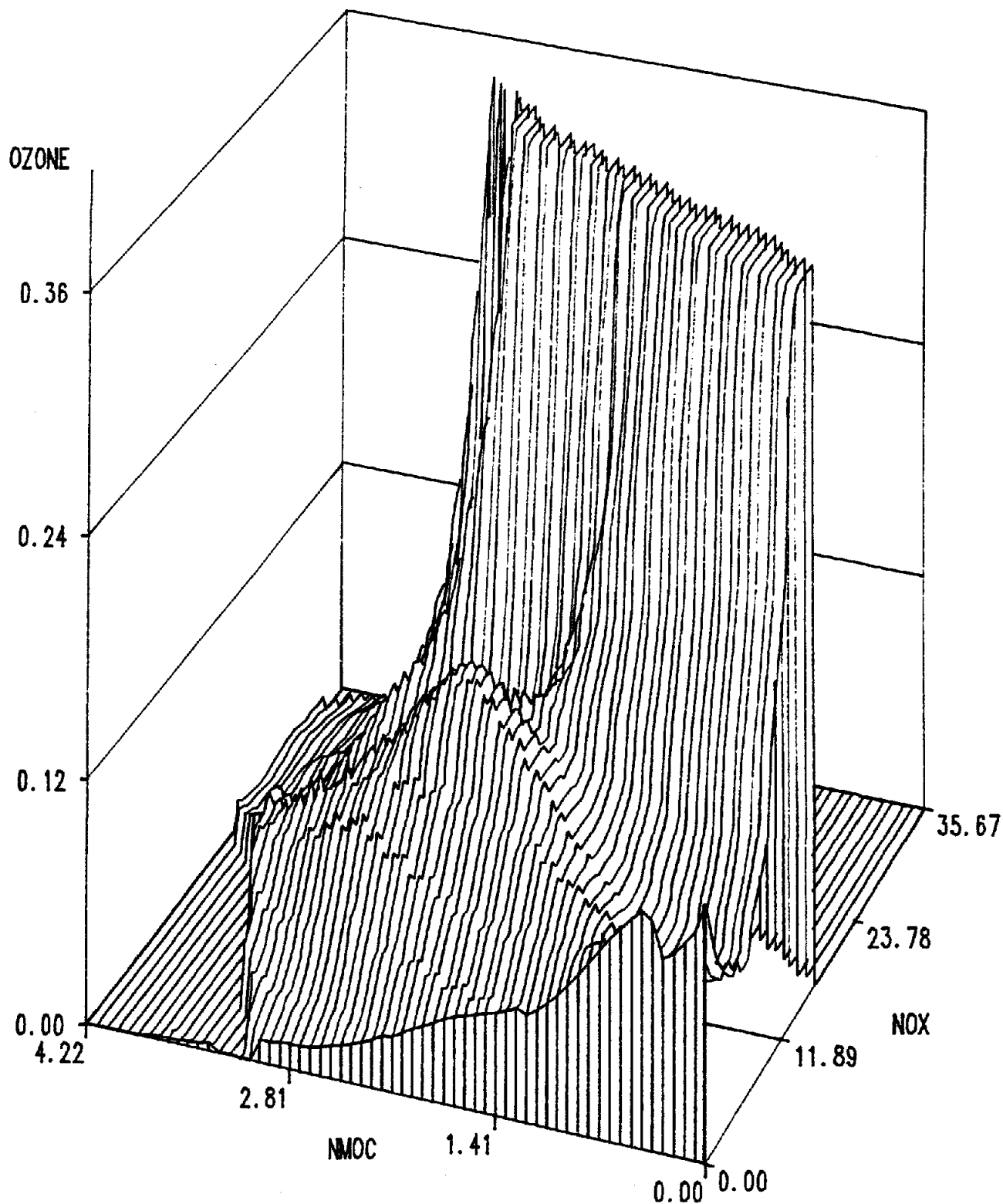


FIGURE 3

Ozone as Function of NMOC and NOX

Houston, 1985-1989

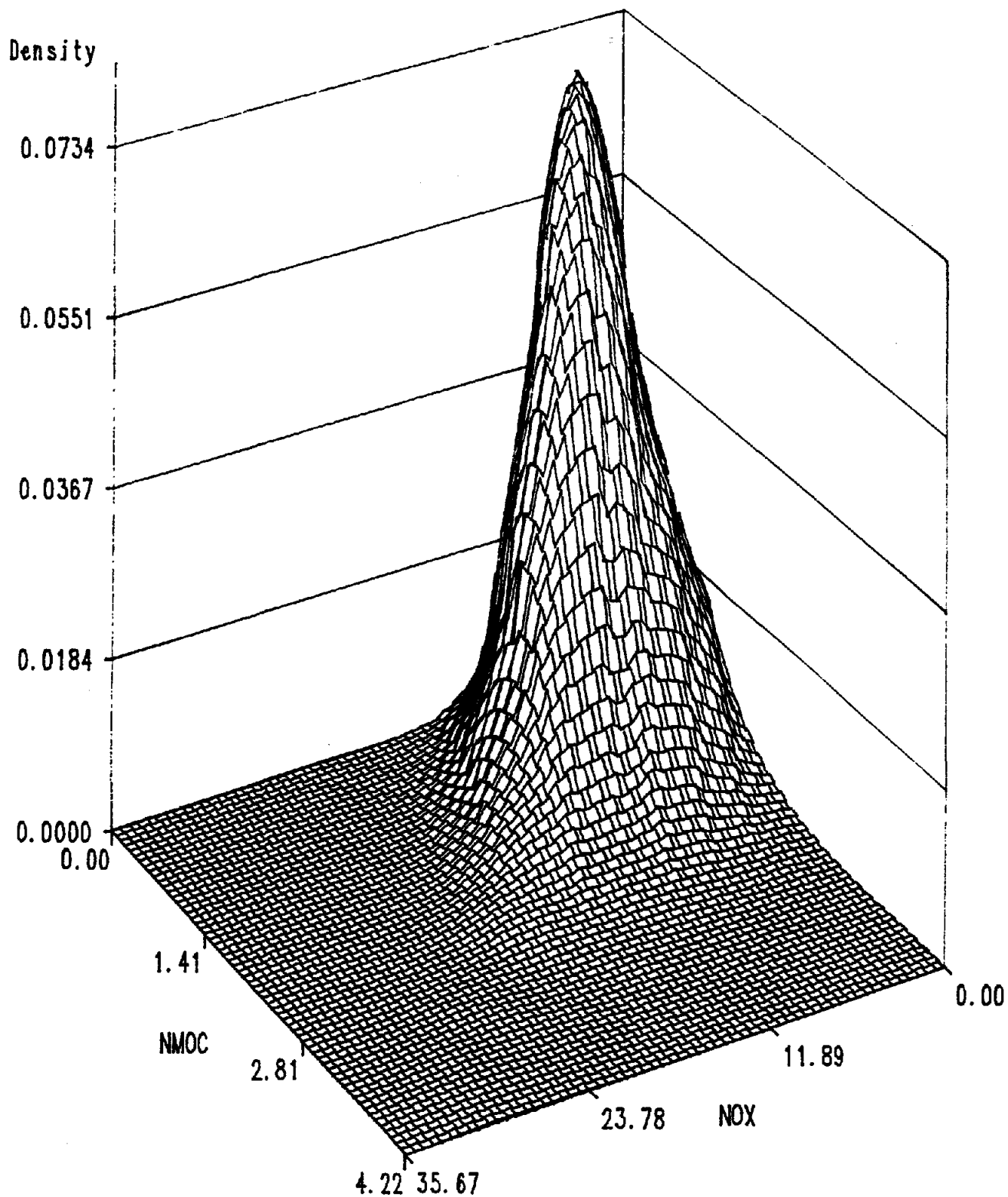


FIGURE 4