POLYNOMIAL APPROXIMATION OF CIRCUIT RESPONSES FOR OPTIMAL DESIGN OF VLSI CIRCUITS

A Thesis

by

MUSTAFA HANI KHAMMASH

Department of Electrical Engineering

Submitted in Partial Fulfillment of the Requirements of the
University Undergraduate Fellows Program

1985-86

Approved by:

Dr. M. A. Styblinski
(Advisor)

Dr. R. Biernacki
(Advisor)

May 1986

ABSTRACT

Polynomial Approximation of Circuit Responses for

Optimal Design of VLSI Circuits. (May 1986)

Mustafa Hani Khammash

An approximation scheme that uses 2^{nd} order polynomials is presented. The approximation is used to obtain circuit responses without having to continuously refer to circuit simulation programs which are very time consuming. In building up the coefficients of the approximating polynomial, a number of points in the circuit parameter space are sampled and the circuit responses are evaluated using a circuit analysis program. Once the polynomial is determined substution of circuit parameters in the polynomial gives the approximated value for the circuit response to which the polynomial corresponds. The approximation scheme allows the approximation to be updated and makes use of all available sampled and analyzed points without being restrained by having exactly the right number of points necessary for unique interpolation, either linear or quadratic. The method is based on an interpolation where the unique polynomial is obtained by adding the constraint that the function should be maximally flat while at the same time satisfying all the available sampled and analyzed points. The approximation presented behaves much better than the linear and even the full quadratic interpolation when only a few additional points more than the number needed for the rather inaccurate linear interpolation are added. The savings in the computer time needed for VLSI circuit optimization gained when using this method are invaluable. Software implementation and a practical example demonstrating the approximation scheme are presented in this thesis.

To my Mother and Father

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

CHAPTER I

INTRODUCTION

A. Background

Modern world has witnessed great advancements made in the field of Electrical Engineering. A most drastic example is seen in the miniaturization of large electronic circuits into literally microscopic sizes. Recent achievements in the field of solid state electronics have made possible the integration of millions of circuit elements into a single functional silicon chip which occupies an area of about only one square inch. This integration is termed VLSI (Very Large Scale Integration) and is itself an entire area of study in Electrical Engineering.

The extremely large size of VLSI circuits makes the statistical approach of VLSI circuit design and optimization very convenient. Optimization involves choosing the nominal values of circuit elements for a particular circuit design in such a way to maximize the number of manufactured functional circuits that meet their pre-assigned specifications. The problem of producing circuits that do not meet their design specifications arises from the fact that elements tolerences cannot be avoided in the production stage. The final design for every circuit that will be produced assigns nominal values to all the elements in that circuit. The set of all such nominal values is called the *nominal point.* The existence of element tolerences arising in the production stage will cause the produced circuits to deviate slightly from their nominal point. It is this deviation that will cause the final performance of some of the circuits produced to be unacceptable. With this problem in mind, we are lead to the definition of *yield* which is the ratio of all the functional produced circuits

---

Journal model is *IEEE Transactions on Automatic Control.*

with an acceptable performance to the total number of produced circuits. From its definition, it is seen that yield is an important economical factor in VLSI circuit production, and its maximization is a crucial step towards cost reduction.

The circuit element tolerences encountered in circuit production are not totally arbitrary. In fact, element values can be considered as random variables and assigned probability density functions. By moving the nominal point locally and evaluating the Yield at each point an optimal nominal point which gives maximum value for the Yield can be located. This is called *design centering.*

B. Problem Definition

A common method used to find the optimal nominal point is the *Monte Carlo* method [1-5]. This method requires that for different nominal points the circuit responses be evaluated. This is done using circuit analysis programs that simulate the circuit performance, and it is repeated for many nominal points until the optimal one is found as mentioned earlier. In spite of the fact that these simulation programs are very accurate and give almost exact results, the large size of VLSI circuits makes them very time consuming to run especially since this analysis procedure has to be repeated many times. If this computer analysis were to be repeated hundreds of times for every circuit, valuable computer time is consumed, and efficiency which is the ultimate goal for optimization is not maintained. Therefore, an efficient and fairly accurate method for evaluating the circuit responses is needed in order to cut down the large but necessary computer CPU time needed for the optimization purposes.

C. Objective

The objective of this work is to develop a computer program that will implement an algorithm to approximate circuit behavior in a manner that will reduce the continuous dependency on simulation programs without giving up a lot of the accuracy associated with the full use of these programs. In addition, application to a practical example is necessary to demonstrate the properties and effectiveness of this approximation.

CHAPTER II

LITERATURE REVIEW

Polynomial approximations have been used in the literature because of their desirable properties and well known bahavior. The efforts have been directed to constructing a polynomial function that will provide a fairly accurate approximation of circuit responses as a function of the circuit parameters. However, in order to find the polynomial in any of the methods, it is necessary to completely determine the coefficients of that polynomial. The discussion of some of the methods in use utilizing polynomial approximation has to be preceeded by some knowledge of coefficient determination.

A. Coefficient Determination

The method used for coefficent determinationn in all polynomial approximations involves sampling the parameter space and evaluating the response at the sampled points. Given a circuit configuration with n parameters (resistors, capacitors, inductors, ...etc) let $\mathbf{X}^{nom} = (x_1^{nom}, x_2^{nom}, \ldots, x_n^{nom})$ denote the nominal point for that particular circuit, where x_i^{nom} in this set is the nominal value for the ith circuit parameter. Also let $\mathbf{Y}^{nom} = (y^{nom})$ denote the value of the response (gain, voltage, ...etc) for the above mentioned nominal point. To sample the parameter space a random number generator is used to generate points around the nominal point. So let $\mathbf{X}^i = (x_1^i, x_2^i, \ldots, x_n^i)$ denote the ith generated point. Now the response of each of the n generated points, denoted by $\mathbf{Y}^i = (y^i)$ for an arbitrary point, can be evaluated using circuit analysis programs. The points and their corresponding responses form a set of linear simultaneous equations. If the system of equations is linearly independent, it can be solved for the coefficients and the polynomial is determined uniquely. Since these points are used to construct the polynomial coefficients, they are called *base points*. Finally, once the polynomial has been determined, any point may be substituted in the polynomial, and the approximated response is found.

B. Previous Polynomial Interpolation Schemes

Linear Interpolation

In this method of approximation, the approximating polynomial takes the form

$$P(x_1, x_2, \ldots, x_n) = a_o + \sum_{i=1}^{n} a_i x_i.$$

Since there are $m = n + 1$ coefficients, exactly m number of sampled points is needed to completely determine the polynomial. Although this number of samples and their responses is relatively small, the accuracy of this method is hardly sufficient.

Full Quadratic Interpolation

The full quadratic approximation uses an approximating polynomial that has the form:

$$P(x_1, x_2, \ldots, x_n) = a_o + \sum_{i=1}^{n} a_i x_i + \sum_{\substack{i=1 \\ j \geq 1}}^{n} a_{ij} x_i x_j$$

The number of coefficients in this method is equal to $m = (n + 1)(n + 2)/2$. This method, although more accurate than the linear approximation, requires a substantially larger number of sampled and analyzed points, m, for a complete determination of its coefficients. For example, a 20 variable problem requires 231 sampled and analyzed points for the full quadratic case as compared to only 21 in the linear one. This is the major drawback of this method.

It may be seen from the above description of the two methods that there is a tradeoff between the number of analyzed points used and accuracy. Large number of sampled and analyzed points gives better results in terms of accuracy but is very costly when computer time is a factor. The smaller number of analyzed points associated with linear approximation, on the other hand, lacks sufficient accuracy.

CHAPTER III

PROPOSED APPROXIMATION METHOD

A. Philosophy

In order to overcome the drawbacks of the linear and full quadratic approximations, Dr. R. Biernacki [6],[7] showed that it is possible to make use of all available sampled and analyzed points even if their number is between that needed for linear and full quadratic approximations. In similar methods used previously the available sampled and analyzed points above those needed for the linear approximation were discarded if their number was not enough to allow a full quadratic solution (i.e.$n + 1 < m < (n + 1)(n + 2)/2$). Since the additional sampled and analyzed points carry extra information about the approximated response function, a polynomial that makes use of them is expected to have an improved accuracy over that associated with the linear approximation.

The actual procedure used in the construction of the polynomial may be divided into two steps. In the first step, $n + 1$ sampled and analyzed points are used to determine the linear coefficients by solving the corresponding system of linear equations. The second step involves the actual build up of the 2nd order terms of the quadratic polynomial. The sampled and analyzed point following the first $n + 1$ points is used to create the higher order coefficients and to update the linear ones. As more points are obtained, they are used to update the linear as well as the higher order coefficients.

The underlying principle of the suggested procedure relies on the fact that the total number of available sampled and analyzed points does not uniquely determine the coefficients due to the existence of an infinite number of solutions. In other words, the available system of equations is underdetermined, and more unknowns are available than equations. Therefore in order to select one solution, a *least prejudiced* approach is taken. Since the nonlinearity of the actual response is not known, it is reasonable to require that the coefficients of the higher order terms be minimized in the least square sense. It is this constraint that will allow the inclusion of an extra $m - (n+1)$ points in the process of coefficient determination resulting in the overall improvement in accuracy.

B. Theoretical Development [6],[7]

Let $\mathbf{X}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})$ be a base point in the parameter space. Also let $\mathbf{Y}^{(i)} = (y^{(i)})$ be the exact response corresponding to that base point. Now, if $n+1$ base points are available along with their responses, the linear polynomial has $n+1$ coefficients as mentioned earlier. Accordingly, this polynomial has the form:

$$P(x_1, x_2, \ldots, x_n) = a_o + \sum_{i=1}^{n} a_i x_i \tag{1}$$

where the a's constitute the coefficients. The a coefficients can be determined by setting up the following system of equations:

$$\begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & & \vdots & \\ 1 & x_1^{(n+1)} & x_2^{(n+1)} & \cdots & x_n^{(n+1)} \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(n+1)} \end{pmatrix} \tag{2}$$

or

$$\mathbf{XA} = \mathbf{B} \tag{3}$$

where $\mathbf{X}$ is the base points matrix, and $\mathbf{B}$ is the responses matrix. Solving the above system of equations leads to the determination of the coefficient matrix. The solution is:

$$\mathbf{A} = \mathbf{X}^{-1}\mathbf{B} \tag{4}$$

The polynomial to be used here takes the form:

$$h(x_1, \ldots, x_n) = a_o + \sum_{i=1}^{n} a_i x_i + \sum_{\substack{i=1 \\ j \geq i}}^{n} a_{ij} x_i x_j \tag{5}$$

The addition of the higher order terms increases the number of unknown coefficients and will require more than $n+1$ points. Let the number of available base points be m such that

$$n + 1 < m < (n+1)(n+2)/2$$

For each of the base points the polynomial (5) is used to relate each of the base points to its response. As an example, for the ith base point

$$b^{(i)} = a_o + a_1 x_1 + \cdots + a_n x_n^{(i)} + a_{(11)}(x_1^{(i)})^2 + a_{(12)} x_1^{(i)} x_2^{(i)} + \cdots + a_n (x_n^{(i)})^2 \tag{6}$$

Repeating this process for all the base points, the set of all m such equations can be represented by the following matrix relation:

$$\mathbf{QC} = \mathbf{B} \tag{7a}$$

or

$$\begin{pmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix} \tag{7b}$$

where

$$\mathbf{Q}_{11} = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n+1)} & x_2^{(n+1)} & \cdots & x_n^{(n+1)} \end{pmatrix}$$

$$\mathbf{Q}_{12} = \begin{pmatrix} x_1^{(1)}x_1^{(1)} & x_1^{(1)}x_2^{(1)} & \cdots & x_n^{(1)}x_n^{(1)} \\ x_1^{(2)}x_1^{(2)} & x_1^{(2)}x_2^{(2)} & \cdots & x_n^{(2)}x_n^{(2)} \\ \vdots & \vdots & & \vdots \\ x_1^{(n+1)}x_1^{(n+1)} & x_1^{(n+1)}x_2^{(n+1)} & \cdots & x_n^{(n+1)}x_n^{(n+1)} \end{pmatrix}$$

$$\mathbf{Q}_{21} = \begin{pmatrix} 1 & x_1^{(n+2)} & x_2^{(n+2)} & \cdots & x_n^{(n+2)} \\ 1 & x_1^{(n+3)} & x_2^{(n+3)} & \cdots & x_n^{(n+3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$$

$$\mathbf{Q}_{22} = \begin{pmatrix} x_1^{(n+2)}x_1^{(n+2)} & x_1^{(n+2)}x_2^{(n+2)} & \cdots & x_n^{(n+2)}x_n^{(n+2)} \\ x_1^{(n+3)}x_1^{(n+3)} & x_1^{(n+3)}x_2^{(n+3)} & \cdots & x_n^{(n+3)}x_n^{(n+3)} \\ \vdots & \vdots & & \vdots \\ x_1^{(m)}x_1^{(m} & x_1^{(m)}x_2^{(m)} & \cdots & x_n^{(m)}x_n^{(m)} \end{pmatrix}$$

$$\mathbf{C}_1 = \begin{pmatrix} a_o \\ a_1 \\ \vdots \\ a_n \end{pmatrix}$$

$$\mathbf{C}_2 = \begin{pmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{nn} \end{pmatrix}$$

$$\mathbf{B}_1 = \begin{pmatrix} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(n+1)} \end{pmatrix}$$

$$\mathbf{B}_2 = \begin{pmatrix} b^{(n+2)} \\ b^{(n+3)} \\ \vdots \\ b^{(m)} \end{pmatrix}$$

If the system of equations (7) is underdetermined, which is always the case when the number of points available is not sufficient for full quadratic interpolation, the following constraint is added to obtain a unique solution:

$$\underset{\mathbf{C}_2}{minimize} \; \|\mathbf{C}_2\|^2$$

$$\textit{subject to } \mathbf{QC} = \mathbf{B} \tag{8}$$

The solution of (8) will minimize the coefficients of the quadratic terms and simultaneously satisfy the set of equations. Rewriting (7a) we have

$$\mathbf{Q}_{11}\mathbf{C}_1 + \mathbf{Q}_{12}\mathbf{C}_2 = \mathbf{B}_1 \tag{9}$$

$$\mathbf{Q}_{21}\mathbf{C}_1 + \mathbf{Q}_{22}\mathbf{C}_2 = \mathbf{B}_2 \tag{10}$$

Solving for $\mathbf{C}_1$ in the first equation we have

$$\mathbf{C}_1 = \mathbf{Q}_{11}^{-1}(\mathbf{B}_1 - \mathbf{Q}_{12}\mathbf{C}_2) \tag{11}$$

Substituting for $\mathbf{C}_1$ in the first equation we have

$$\mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{B}_1 - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}\mathbf{C}_2 + \mathbf{Q}_{22}\mathbf{C}_2 = \mathbf{B}_2 \tag{12}$$

or

$$(\mathbf{Q}_{22} - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})\mathbf{C}_2 = \mathbf{B}_2 - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{B}_1 \tag{13}$$

Now let

$$\mathbf{A} = \mathbf{Q}_{22} - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12} \tag{14}$$

and

$$\mathbf{D} = \mathbf{B}_2 - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{B}_1 \tag{15}$$

Equation (13) can therefore be written as follows

$$\mathbf{AC}_2 = \mathbf{D} \tag{16}$$

Since $\mathbf{C}_1$ and $\mathbf{C}_2$ are related by (11), once $\mathbf{C}_2$ is solved for, $\mathbf{C}_1$ can be determined.

The problem of minimization reduces to the following:

$$\underset{\mathbf{C}_2}{minimize} \ \|\mathbf{C}_2\|^2$$

$$subject\ to\ \mathbf{A}\mathbf{C}_2 = \mathbf{D} \tag{17}$$

The solution to this problem is well know and takes the form

$$\mathbf{C}_2 = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{D} \tag{18}$$

Whenever a new base point and its response become available the coefficient matrix $\mathbf{C}$ has to be updated to account for the new information gained by these extra points. While using equation (18) to solve for $\mathbf{C}_2$ is mathematically correct, for every new base point the $\mathbf{A}$ and $\mathbf{D}$ matrices have to be recalculated, and therefore the problem has to be solved from the beginning. This may be very time consuming since it will have to be done every time a new base point becomes available. To avoid unnecessary calculations, a new efficient algorithm is used to update the coefficients for every additional base point. The solution will have to be identical to that obtained using equation (18).

Let us assume that we have already found the coefficients for $(m-1)$ base points $(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m-1)})$ and their corresponding responses $(\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m-1)})$. Now when the mth base point and its response are available the matrices $\mathbf{Q}_{21}$ and $\mathbf{Q}_{22}$ will have an additional row added to them. Because $\mathbf{Q}_{11}$ and $\mathbf{Q}_{12}$ are unaffected by this addition, $\mathbf{A}$ will have an additional row since $\mathbf{A} = \mathbf{Q}_{22} - \mathbf{Q}_2\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}$. The new row added to $\mathbf{A}$, denoted by $\mathbf{a}^{m^T}$, will have the form

$$\mathbf{a}^{(m)^T} = \mathbf{q}_{22}^{(m)^T} - \mathbf{q}_{21}^{(m)^T}\mathbf{E} \tag{19}$$

where $\mathbf{q}_{21}^{(m)^T}$ is the new row in matrix $\mathbf{Q}_{21}$, $\mathbf{q}_{22}^{(m)^T}$ is the new row in matrix $\mathbf{Q}_{22}$, and $\mathbf{E} = \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}$. In a similar manner a new row will be added to the matrix $\mathbf{D}$, and according to equation (15) it will have the form

$$\mathbf{d}^{(m)} = \mathbf{b}^{(m)} - \mathbf{q}_{21}^{(m)}\mathbf{Q}_{11}^{-1}\mathbf{B}_1 \tag{20}$$

where $\mathbf{b}^{(m)}$ is the response at the new base point and forms the new row of $\mathbf{B}_2$.

The constraints given by (16) can be expressed as

$$\mathbf{A}^{(m-1)}\mathbf{C}_2 = \mathbf{D}^{(m-1)} \tag{21}$$

and

$$\mathbf{A}^{(m)^T}\mathbf{C}_2 = d^{(m)} \tag{22}$$

The solution $\mathbf{C}_2^{(m-1)}$ of (21) for the $(m-1)$ step is orthogonal to the hyperplane $\mathbf{A}^{(m-1)}\mathbf{C}_2 = \mathbf{D}^{(m-1)}$. Therefore, any point satisfying (21) and (22) can be expressed as

$$\mathbf{C}_2 = \mathbf{C}_2^{(m-1)} + \Delta\mathbf{C}_2 \tag{23}$$

such that $\Delta\mathbf{C}_2$ is orthogonal to $\mathbf{C}_2^{(m-1)}$. Accordingly it can be said that

$$min\|\mathbf{C}_2^m\|^2 = \|\mathbf{C}_2^{(m-1)}\|^2 + \underset{\Delta\mathbf{C}_2}{min}\|\Delta\mathbf{C}_2\|^2 \tag{24}$$

but since $\|\mathbf{C}_2^{(m-1)}\|^2$ is already determined, to find $\|\mathbf{C}_2^{(m)}\|^2$ we have the following problem:

$$\underset{\Delta\mathbf{C}_2}{minimize}\ \|\Delta\mathbf{C}_2\|^2 \tag{25a}$$

$$subject\ to\ \mathbf{A}^{(m)}C_2 = \mathbf{D}^{(m)} \tag{25b}$$

where $\mathbf{C}_2 = \mathbf{C}_2^{(m-1)} + \Delta\mathbf{C}_2$.

Since $\mathbf{A}^{(m-1)}\mathbf{C}_2 = \mathbf{D}^{(m-1)}$ we have

$$\mathbf{A}^{(m)}\Delta\mathbf{C}_2 = \mathbf{e}^m \tag{26}$$

where $\mathbf{e}^{(m)^T} = \left[0 \;\; 0 \;\; 0 \;\; \ldots \;\; (d^{(m)} - \mathbf{a}^{(m)^T}\mathbf{C}_2^{(m-1)})\right]$.

The solution of (25a) subject to the constraint (25b) is

$$\Delta\mathbf{C}_2^{(m)} = \mathbf{A}^{(m)^T}(\mathbf{A}^{(m)}\mathbf{A}^{(m)^T})^{-1}\mathbf{e}^{(m)} \tag{27}$$

Calculating (27) can be done using LU decomposition methods. $\mathbf{G}^{(m)}$ can be formed from $\mathbf{G}^{(m-1)}$ by adding a new row and a new column. Since $\mathbf{G}^{(m-1)} = \mathbf{A}^{(m-1)}\mathbf{A}^{(m-1)^T}$ we have

$$\begin{aligned} \mathbf{G}^{(m)} = \mathbf{A}^{(m)}\mathbf{A}^{(m)^T} &= \begin{pmatrix} \mathbf{A}^{(m-1)} \\ \mathbf{a}^{(m)^T} \end{pmatrix} \begin{pmatrix} \mathbf{A}^{(m-1)^T} & \mathbf{a}^{(m)} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{A}^{(m-1)}\mathbf{A}^{(m-1)^T} & \mathbf{A}^{(m-1)}\mathbf{a}^{(m)} \\ \mathbf{a}^{(m)^T}\mathbf{A}^{(m-1)^T} & \|\mathbf{a}^{(m)}\|^2 \end{pmatrix} \end{aligned} \tag{28}$$

Let $\mathbf{A}^{(m-1)}\mathbf{a}^{(m)}$ be denoted by $\mathbf{g}^{(m)}$ and $\|\mathbf{a}^{(m)}\|^2$ by $s^{(m)}$. $(\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{G}^{-1}$ can be decomposed into its UL decomposition, and therefore

$$\mathbf{G}^{-1} = \mathbf{U}_G^{-1}\mathbf{L}_G^{-1} = \mathbf{U}\mathbf{L} \tag{29}$$

where $\mathbf{U}_G$ is the upper triangular matrix in the decomposition of $\mathbf{G}$, and $\mathbf{L}_G$ is the lower triangular matrix in the decomposition of $\mathbf{G}$ with 1's across its diagonal. The updating of $\mathbf{L}^{(m-1)}$ and $\mathbf{U}^{(m-1)}$ is in effect the same as forming $\mathbf{L}^{(m)}$ and $\mathbf{U}^{(m)}$. The latter matrices can be formed as follows:

$$\mathbf{U}^{(m)} = \begin{pmatrix} \mathbf{U}^{(m-1)} & \mathbf{z}^{(m)}\mathbf{l}^{(m)} \\ \mathbf{0} & \mathbf{z}^{(m)} \end{pmatrix} \tag{30}$$

and

$$\mathbf{L}^{(m)} = \begin{pmatrix} \mathbf{L}^{(m-1)} & \mathbf{0} \\ \mathbf{l}^{(m)^T} & 1 \end{pmatrix} \tag{31}$$

where $\mathbf{l}^{(m)} = -\mathbf{U}^{(m-1)}\mathbf{L}^{(m-1)}\mathbf{g}^{(m)}$ and $\mathbf{z}^{(m)} = \frac{1}{\mathbf{s}^{(m)} + \mathbf{g}^{(m)^T}\mathbf{l}^{(m)}}$ $\Delta\mathbf{C}_2^{(m)}$ can therefore be calculated according to (27) to give

$$\Delta\mathbf{C}_2^{(m)} = \mathbf{A}^{(m)^T}\begin{pmatrix} \mathbf{r}^{(m)}\mathbf{l}^{(m)} \\ \mathbf{r}^{(m)} \end{pmatrix} \tag{32}$$

where $\mathbf{r}^{(m)} = \mathbf{z}^{(m)}(\mathbf{d}^{(m)} - \mathbf{a}^{(m)^T}\mathbf{C}_2^{(m-1)})$. Once $\Delta\mathbf{C}_2$ is calculated, $\mathbf{C}_2^{(m)}$ can be formed by

$$\mathbf{C}_2^{(m)} = \mathbf{C}_2^{(m-1)} + \Delta\mathbf{C}_2^{(m)} \tag{33}$$

and finally

$$\mathbf{C}_1^{(m)} = \mathbf{C}_1^{(m-1)} - \mathbf{Q}^{-1}\mathbf{Q}_{12}\Delta\mathbf{C}_2^{(m)} \tag{34}$$

Therefore, the coefficient matrix $\mathbf{C}$ composed of $\mathbf{C}_1$ and $\mathbf{C}_2$ has been completely determined.

CHAPTER IV

SOFTWARE IMPLEMENTATION

The actual application of the theoretical results discussed earlier in this thesis takes the form of a computer program written in *FORTRAN 77* language. To conveniently discuss the main structure of the computer implementation, the following description takes two forms. First, the approximation subroutine that directly applies the theoretical ideas is discussed in a rather detailed fashion since it contains the heart of this program. This discription follows more or less the actual steps taken when actual execution takes place. The other part of developed software to be discussed is the program and supporting subroutines used to call the approximation routine. This part of software makes the connection between the interactive user, the particular function or circuit example to be approximated, and the approximation routine. Since this *calling* program and its supporting routines are only *one* way to apply the more universal approximation routine, detailed discussion is not crucial, but rather a functional discription of the important structures is more appropriate.

A. Approximation routine

Input/output parameter discription

The approximation routine receives as its input a set of m sampled points stored in the double precision one dimensional array DXSAM. Each of these points are composed of n elements, where n is the number of variables of the problem. Furthermore, corresponding to each of the points mentioned above is m exact circuit response values which are passed in another double precision, one dimensional array, DXSAM_VALU, in which the elements are arranged in the same order as that of their corresponding sampled points. DXSAM, DXSAM_VALU, the number of variables n, and the number of previous updatings KL form the input of this routine. The output, on the other hand, consists of the array of polynomial coefficients COEF that have been calculated along with their number NCOEF.

Internal Storage

All matrices and work space used in this routine are stored in *SPACE*, a single one dimensional array, using double precision. Two dimensional matrices are stored by columns. To allow ease of access to any of the matrices stored, integer variables are assigned values equal to the location in SPACE in which the first element of that matrix is stored. By passing these *pointers*, along with the exact size of the desired matrix to the subroutines, only the referenced matrix is accessed and operated on. These pointers are described in detail in the documentation part of the routine in the appendix.

Execution Steps

Almost all subroutines used in the approximation routine are especially created for that routine, and are appended to its main body as shown in the routine listing in the appendix. The only two exceptions are AXB and INVERT which are library subroutines.

The execution steps followed by the approximation subroutine may be outlined as follows:

Step 1 : The updating number KL is checked to find out if this is a new approximation problem or if this is merely an updating procedure. If this is a new problem, execution continues at the next step; otherwise it is transfered to **Step 7**. In either case the value of KL is updated to account for the current call of the routine.

Step 2 : All pointers associated with the matrices to be used in the routine including working space matrices for temporary usage are defined.

Step 3 : By calling subroutine LOAD, matrices $\mathbf{Q}_{11}$, $\mathbf{Q}_{12}$, and $\mathbf{B}_1$ are created using data recieved from DXSAM and DXSAM_VALU. Then they are stored in their appropriate location.

Step 4 : $\mathbf{C}_1^{(n+1)} = \mathbf{Q}_{11}^{-1}\mathbf{B}_1$ is calculated by calling subroutine AXB and INVERT, then using the same two subroutines $\mathbf{E} = \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}$ is calculated and both matrices are stored in proper locations for later use.

Step 5 : If linear approximation is all that is required then COEF matrix is loaded with the contents of $\mathbf{C}_1$ and set the number of coefficients NCOEF is set to the appropriate value after which control is returned to the main program. If updating is also required, then execution continues at the next step.

Step 6 : $\mathbf{C}_2$, $\mathbf{U}$, $\mathbf{L}$, and $\mathbf{A}$ matrices are initialized to zero by calling subroutine ZERO. Also an updating counter IUP is initialized to 1.

Step 7 : Calling subroutine GENMAT, matrices $\mathbf{q}_{21}$ and $\mathbf{q}_{22}$ are computed and stored in proper location. Then $\mathbf{a}^{(m)}$ and $\mathbf{d}^{(m)}$ are calculated directly by calling subroutines MATRAN, AXB, and MATSUB. Results are stored.

Step 8 : Matrix $\mathbf{A}$ is created only if this is the first time this step is executed for a particular problem; otherwise it is updated. The subroutine that performs these operations is UPDTA.

Step 9 : $||\mathbf{a}^{(m)}||$ is calculated by calling subroutine NORM. The resulting value is stored in the location assigned for $\mathbf{z}^{(m)}$.

Step 10 : If this is the first updating, $\mathbf{L}$ and $\mathbf{U}$ matrices are calculated and then **Step 13** is executed next; otherwise execution is proceeded at the next step.

Step 11 : Matrices $\mathbf{l}^{(m)}$, $\mathbf{g}^{(m)}$, and $\mathbf{z}^{(m)}$ are calculated by calling subroutines SMULT, MATADD, AXB. The three matrices are stored.

Step 12 : Matrices **L** and **U** are updated by adding a new row to **L** and a new column to **U**. The subroutines which perform these functions are subroutines UPDTL and UPDTU. The updated matrices **L** and **U** are then stored for additional future updating.

Step 13 : $\Delta\mathbf{C}_2$ is calculated and stored by calling subroutines MATRAN, AXB, and SMULT.

Step 14 : The matrix $\mathbf{C}_2$ is created or updated if it has already been created. This matrix contains the quadratic coefficients of the polynomial.

Step 15 : The coefficient matrix of the linear terms $\mathbf{C}_1$ is updated by using subroutines AXB and MATSUB. Results are stored.

Step 16 : The coefficient array COEF is loaded with the linear and higher order coefficients, and the number of coefficients NCOEF is assigned its proper value.

Step 17 : The number of updating IUP is incremented. If the number of desired updating has been reached, control is returned the main program; otherwise execution proceeds at **Step 7**.

B. Interfacing Main and Supporting Routines

This part of software consists of a main program that performs the subroutine calls and all other supporting subroutines that are necessary to interface the approximation program with what other routines are needed for interactive use. All subroutines used were especially developed for the purpose of use in this approximation program with two exceptions. These exceptions are ISCAS and SAMPLE routines which have been provided by Dr. M. A. Styblinski and have been developed by Mr. L. J. Opalski.

A brief description of all the subroutines used is presented here with the exception of the approximation subroutine which has been discussed in detail earlier.

1. ISCAS subroutine: This subroutine calls any of 14 other subroutines (ISCAS1-ISCAS14) which provide a number of different pre-specified circuit responses corresponding to a set of circuit parameters operating at different frequencies. The fourteen (ISCAS1-14) routines correspond to an equal number of different circuit examples any of which can be accessed by specifying the desired circuit example to ISCAS which in turn calls the correct ISCAS1-14 routine.

2. SAMPLE subroutine: This is a random number generator that generates a sample point of NX elements around a given nominal value. Normal or Gaussian distributions are allowed. The desired nominal point is passed to the subroutine along with the tolerances for each element of the nominal point. The subroutine generates one point for each call and has to be reset before the first call.

3. INDAT subroutine: This subroutine is responsible for displaying a message on the screen prompting the user to choose between the 14 examples available. Once the choice has been made, the subroutine reads the appropriate input data file for that example. The data files should be named EX1-14 with each file corresponding to the circuit example with the same number. The subroutine reads the nominal point, the type of distribution, the tolerance for either distribution, and the frequencies at which the responses are evaluated. This information is passed back to the main program.

4. TRAFIC subroutine: This subroutine prompts the user to enter the number of base points he/she wants included in the approximation after displaying the possibilities. According to the choice of the user the subroutine decides how many

times should the updating loop in the main program be repeated. The subroutine also decides if this is the first time the approximation is taking place for that problem or if this is just an updating of the present problem. If this is the first time, it assigns a positive value to NREP whose absolute value carries the number of base points that need to be sampled, analyzed, and passed to the approximation subroutine; otherwise it assigns a negative value to NREP whose absolute value carries the same meaning as explained before.

5. DOUBLE subroutine: This subroutine converts the responses for various outputs and frequencies obtained from ISCAS routines from single to double precision and stores the result for all the sampled points in the 3 dimensional array C. The order in the 3 dimensional array in which the responses were stored is: output number, frequency number, and sample number. So to access the response for the 2nd output, 3^{rd} frequency, and 10^{th} sample, one can address C(2,3,10).

6. AVERAGE subroutine: AVERAGE subroutine calculates the average of all the sampled points used for a single run of the program.

7. UPDTAVG subroutine: This subroutine updates the average calculated by AVERAGE subroutine if the program generates additional sample points for the purpose of updating the approximation. This subroutine simply takes into account the new generated points and the old calculated average and calculates the average for all the sample points used.

8. DEV subroutine: This subroutine calculates the deviations of all the sample points generated from the average. In addition, it calculates the relative deviation from the average. The deviation values are stored in the two dimensional array EPSILON, while the relative deviation values are stored in RELEPS array.

9. NORMAL subroutine: NORMAL subroutine normalizes all of the generated

sample points by dividing each by the nominal point. Moreover, the subroutine normalizes all the circuit responses by dividing them by the corresponding nominal point responses. The normalization procedure allows the used sample points and their responses to be of reasonable distance from each other which is not a characteristic of the unnormalized points since they may vary greatly in magnitude.

10. MPRINT subroutine: This subroutine prints to the output file OUTPUT.DAT the nominal point, the used sample points before normalization, the sample points average, deviation from average, and the relative deviation from average.

11. PNTRESP subroutine: PNTRESP subroutine prints to the output file OUTPUT.DAT the values predicted by the approximation for the base points used to build up the coefficients of the polynomial as well as for randomly generated points.

12. PNTCOEF subroutine: This subroutine prints the generated coefficients of the approximating polynomial for the number of sampled points used. The order in which the coefficients are printed is important. First, the coefficients of the linear terms are printed in the order $a_o, a_1, a_2, \ldots, a_n$. Following the linear terms immediately are the square terms coefficients arranged in the order $a_{11}, a_{22}, \ldots, a_{nn}$. Finally, the mixed terms coefficients are printed in the order $a_{12}, a_{13}, \ldots, a_{1n}, a_{23}, a_{24}, \ldots, a_{n-1\ n}$.

13. SUBS subroutine: This subroutine, as its name might suggest, simply substitutes any number of points into a quadratic polynomial given its coefficients are passed to the subroutine. For every point specified, there is a response obtained by the polynomial substitution which is in effect the response for that point predicted by that polynomial. D option may be used in the beginning of that subroutine to provide a driver program for testing purposes or for running the subroutine separately if so desired.

CHAPTER V

RESULTS AND DISCUSSION

Before the approximation program can be applied to a practical example, it is desired to know just how well it performs its intended task by applying it to a known example. This test ought to indicate any errors in the implementation and if no errors are present to uncover the limitations of the algorithm. Once sufficient knowledge about the algorithm and its implementaion has been acquired, one can proceed to the practical implementation of a real problem with confidence.

A. Application to a quadratic

If a full quadratic function is used to obtain the response for the sampled points, the implemented algorithm should theoretically yield the coefficients of a polynomial that eventually converges to the same quadratic function used. This provides a good test to the algorithm. For that purpose, the computer program APPTEST has been developed. The program uses an actual quadratic function to evaluate the responses of the sampled points. The quadratic function used has the following form:

$$P(x_1, \ldots, x_n) = 1 + \sum_{i=1}^{n} x_i + x_i^2$$

It is noted that all the coefficients of the linear and square coefficients of that polynomial are 1, while the coefficients of the mixed terms are zero. The program runs using samples around the nomimal point $\mathbf{X}_{nom} = (1, 1, \ldots, 1)$ generated by the random number generator. The resulting coefficients are compared with those of the full quadratic. This procedure is repeated for a different number of variables to show that the algorithm is still applicable as the dimensionality of the problem is changed.

To provide a second basis for comparison, the coefficients of the full quadratic were solved for by solving the resulting system of equations (3) by directly evaluating the inverse matrix as in (4). The only error in this method is that created when the inverse is evaluated due to rounding in the computer. Otherwise, the results should be identical.

Table I shows results obtained using the two different methods discussed above. The program was run for 3 variables and a tolerance value of 0.05. This tolerance is the sigma value fed to the random number generator, since a normal distribution was used as a basis for sampling the points around the nominal point. It is observed from this table that the approximation method does in fact converge to the actual values used for the quadratic. Despite the fact this convergence is not exact, it is very encouraging.

It can also be observed that results obtained with the approximation algorithm were very close to those obtained by solving the system of equations (4). The small difference can be attributed to round off errors in both methods. On the other hand, both methods displayed a relatively much larger error when compared to the actual value of the coefficients. Since the direct solution of the problem by (4) and the approximation method agree closely but differ from the actual values, it can be concluded that this error is not due to the approximation method itself.

TOLERANCE = 0.001		
Actual value	Approximation method	Direct solution
1.0	0.927460	0.927397
1.0	1.048116	1.048154
1.0	1.067927	1.067999
1.0	1.029160	1.029178
1.0	0.991967	0.991962
1.0	0.990856	0.990841
1.0	0.991042	0.991040
0.0	-0.03521	-0.03524
0.0	0.003095	0.003096
0.0	-0.01441	0.014427

Table I

To explain what is happening, it is important to realize that since a relatively small tolerance was used for sampling, the sampled points are expected to be close to one another. This closeness can be observed when looking at the sampled points for this value of tolerance in the appendix. Even though these points are in fact linearly independent, the problem might be *ill-conditioned* due to the small distance between the points. So in order to check if this indeed is the case, results for larger values of tolerance were obtained. Tables II and III are obtained for tolerances of 0.05 and 0.1 respectively.

TOLERANCE = 0.05		
Actual value	Approximation method	Direct solution
1.0	1.009675	1.009649
1.0	0.997591	0.997607
1.0	0.996041	0.996070
1.0	0.986992	0.987000
1.0	1.000629	1.000626
1.0	0.998776	0.998768
1.0	1.002211	1.002210
0.0	-0.00054	-0.00055
0.0	0.001678	0.001678
0.0	0.006946	0.006944

Table II

TOLERANCE = 0.1		
Actual value	Approximation method	Direct solution
1.0	1.000741	1.000701
1.0	0.999746	0.999767
1.0	0.999165	0.999212
1.0	0.999555	0.999570
1.0	1.000037	1.000033
1.0	1.000066	1.000058
1.0	0.999958	0.999956
0.0	0.000171	0.000155
0.0	0.000020	0.000021
0.0	0.000540	0.000525

Table III

Tables II and III show that as the tolerance was increased, great improvement in the results were obtained. The results were accurate to the fourth or in some cases to the fifth decimal point. As the tolerance increases, the problem becomes less ill conditioned and better results are obtained. Another reason of equal importance for this improvement in accuracy is that while the responses of the points with smaller tolerances were evaluated using single precision and thus leading to the loss of some significant digits, the larger tolerances allowed the inclusion of more significant digits in the response evaluation which led to an improvement in accuracy. Finally, it is worthwhile to remember that a 3 variable example was shown here because of its suitability for tabulation, the same conclusions can be drawn for dimensions higher than 3. As a matter of fact the behavior described above is more obvious for higher dimensions, and more drastic sensitivity to tolerance change can be observed.

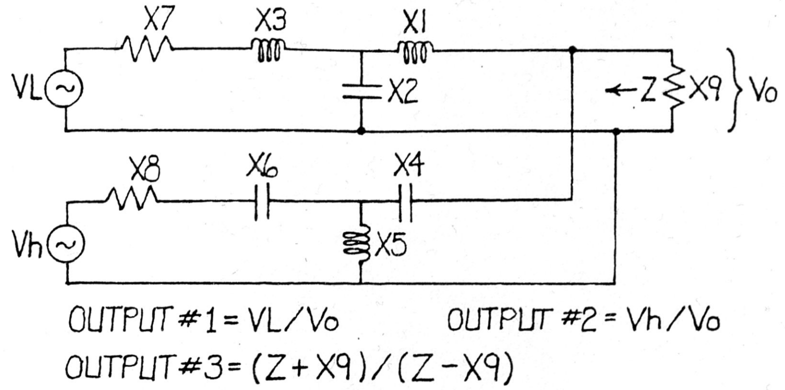
B. Application to a practical example

To demonstrate the properties of the approximation method, it was applied to the circuit example shown in Figure I. The circuit in the figure is based on the standard Test Example # 4 of Wehrhahn and Spence [5]. This example is one of various examples used by Wehrhahn and Spence to compare the performance of different yield optimization algorithms. The nominal value for each circuit element and the tolerance for each value are given and shown in the figure along with the output constraints and the cicuit diagram. Constraints for five operating frequencies and three outputs are given, but it turns out that only frequencies # 3 and 5 of the 3^rd output are really critical for the production yield at that particular nominal point. Output # 3 is the return loss coefficient, given by the formula

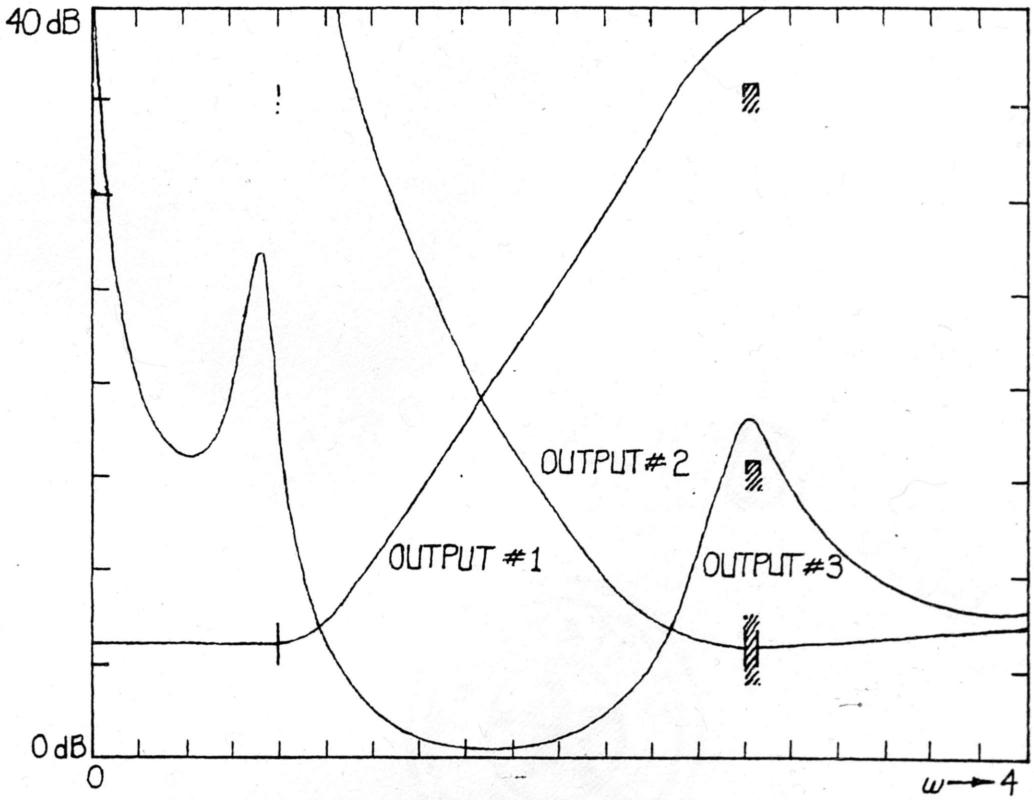
$$\rho(jw) = \frac{Z(jw) + x_9}{Z(jw) - x_9}.$$

It can be seen that $Z(jw)$ is a function of all the circuit paramters except for x_9 but that $\rho(jw)$ depends on x_9 and $Z(jw)$. The return loss coefficient is therefore a function of all the circuit parameters in that circuit and is a good test for the approximation algorithm.

Since this problem has 9 variables the number of sampled and analyzed points needed for linear approximation is 10. So the approximation starts with 10 samples



#	Nominal values	Sigma
1	19.03	.006
2	.003	.03
3	13.11	.006
4	.00056	.03
5	4.57	.006
6	.00056	.03
7	75.	.006
8	75.	.006
9	75.	.006



#	Ang. freq.	Return loss in dB	
		Upper const.	Lower const.
1	4.9951	none	16.
2	5.026	none	16.
3	5.059	none	16.
4	17.5929	none	16.
5	18.0327	none	16.

Figure I. (a) circuit diagram, (b) nominal values, and (c) output response constraints for Example 4. (see Reference [5]) obtained by calling the random number generator SAMPLE. A normal distribution is assumed for the sampling probability density function and zero correlation between the elements. Then the return loss coefficient is evaluated by calling subroutine ISCAS4 which gives the exact value of the return loss coefficient for the first 10 sampled points. The sampled points and their responses are then used to calculate the coefficients of the linear polynomial. After the polynomial has been determined more points are sampled by the random number generator and they are substituted in the polynomial to find out what are the values predicted for those points by the linear polynomial as compared to the actual values. Of course, the points that have been *used* to build up the coefficients should give exact results. After the newly sampled points have been substituted in the polynomial and their predicted value evaluated, they are used along with their exact responses obtained from ISCAS4 to update the polynomial which should now have second order coefficients. It is not necessary to use the same sampled points and their responses for testing the linear polynomial and other sampled and analyzed points could be used instead. The same points are used here because of their availability, a fact that slightly simplifies programming.

Once the new polynomial is obtained, still another set of points is found through sampling and the function response for the new points is evaluated by using the approximation program. Again the results are compared to the actual responses given by ISCAS4.

Tables IV, V, VI, VII, VIII show a comparison between the actual and approximated value for 10, 11, 15, 20, and 35 samples respectively.

#OF SAMPLES USED=10	
Actual value	Approximated value
14.65965	16.87664
16.64453	17.13360
18.41751	22.81373
17.15001	17.98259
17.15209	15.66887

Table IV

#OF SAMPLES USED=11	
Actual value	Approximated value
16.64453	16.60087
18.41751	18.01246
17.15001	17.48724
17.15209	17.27151
16.13204	16.09066

Table V

#OF SAMPLES USED=15	
Actual value	Approximated value
16.13204	16.16760
15.66544	15.61548
13.83851	14.33121
17.29080	17.44162
15.86370	15.72512

Table VI

#OF SAMPLES USED=20	
Actual value	Approximated value
17.33571	17.22661
20.20018	19.68150
16.26776	16.09655
19.64002	19.99169
16.74227	16.72752

Table VII

#OF SAMPLES USED=35	
Actual value	Approximated value
17.09623	16.53985
16.52122	15.59449
15.77145	15.94141
17.08289	17.03423
17.63427	16.96043

Table VIII

From the tables, it is obvious that the approximation for 10 samples is the worst out of all samples. Since 10 is the number corresponding to the linear approximation, it is clear that when one additional base point was added creating all the 2^{nd} order terms a considerable improvement in the approximation was noticed. Actually, this sudden improvement is rather shocking at the first glance, since only one base point was added. But this clearly shows that the additon of the 2^{nd} order terms coefficients in the approximating polynomial allows the polynomial to account much more accurately for the nonlinearities of the approximated function at the very small cost of only *one* extra base point. The linear approximation alone could not provide this flexibility since it is limited to a planar representation of the approximated function. As more points are added, the approximation fluctuates but is still significantly better than the linear approximation. Despite the fact that the addition of extra points provides additional information about the approximated function which should in effect improve the approximation, it is noted that the approximation tends to become slightly worse as the unique full quadratic solution is approached. This may be attributed to the fact that less degrees of freedom are available and the ill-conditioning of the problem noted earlier is starting to take effect. Nothing can be done about this since the tolerances are prespecified and cannot be changed to better condition the problem. This will always be the problem of the full quadratic approximation, which still remains more accurate than the linear approximation alone. The approximation algorithm used is therefore a great improvement over *both* the linear and full quadratic approximations obtained at a very small cost.

CHAPTER VI

CONCLUSION

It has been demonstrated that the presented polynomial approximation gives a much better function representation than that provided by the linear approximation at the small cost of a few additional base points. The power of this approximation lies in its ability to contain additional information about the approximated function and use it in a manner to update the approximating polynomial. It is this desirable feature that eliminates the necessity of being restricted by only a particular number of base points, an obvious disadvantage of the unique linear and full quadratic approximations. So while the linear approximation discards all base points with a number higher than $n + 1$, the presented approximation scheme makes use of them to update the approximation and, as it turns out, to significantly improve the accuracy. Full quadratic approximation, although still better than the linear approximation, suffers from the unavoidable problem of ill-conditioning especially if the approximation is local. Therefore, even with fewer base points used, the ability of the presented approximation to accomodate any number of points less than that required by the unique full quadratic approximation enables it to avoid the ill-conditioning problem and as a result to provide an even better approximation.

As far as computer time is concerned, once the polynomial is obtained, the time needed for substitution in the polynomial to obtain an approximation for a circuit response is negligible by any computer standard compared to the time needed for circuit analysis. In addition, the updating scheme presented reduces the time required to update the approximating polynomial therefore avoiding the repetition of calculations when more base points become available. It can generally be concluded that the accuracy of any approximating polynomial is indeed its greatest limitation. The approximation scheme presented was demonstrated to be more accurate than the previously used polynomial approximation schemes. This accuracy was obtained with a very small additional number of base points if not fewer. This provides a powerful tool for the optimal design of VLSI circuits and is sure to cut down dramatically the computer time needed for this purpose.

REFERENCES

[1] J.W. Bandler, H.L. Abdel-Malek, "Optimal Centering, Tolerancing and Yield Determination Via Updated Approximations and Cuts" IEEE Tran. Circ. and Syst.,CAS-25, 1978, pp.853-871.

[2] D.E. Hocevar, P.F. Cox and P.Yang, "Parametric Yield Optimization for VLSI", Digest IEEE Int. Conf. on Comp.-Aided Design (ICCAD-85), Nov. 18-21, 1985, Santa Clara, Ca., pp.312-314.

[3] M.A. Styblinski and A. Ruszczynski, "Stochastic approximation approach to production yield optimization," in *25-th Midwest Symposium on Circuits and Systems*, Houghton, Mich., Aug. 30-31, 1982.

[4] M.A. Styblinski and A. Ruszczynski, "Stochastic approximation approach to statistical circuit design," *Electr. Letters*, vol. 19, No. 8, pp 300–302, 1983.

[5] E. Wehrhahn, R. Spence, "The performance of some design centering methods", *Proc. 1984 IEEE Int'l Symp. Circ. and Syst.*, Montreal, Canada, May 1984 pp.1424-1438.

[6] R. Biernacki, "Approximation of Response Functions by Updated Interpolation Based on Available Information," author's manuscript, March 26, 1985.

[7] R.M. Biernacki and M.A. Styblinski,"Statistical Circuit Design With a Dynamic Constraint Approximation Scheme,"*IEEE Int. Symp. on Circuits and System*,San Jose, Ca.,1986.

APPENDIX A

PROGRAM AND SUBROUTINES LISTING

```
      PARAMETER(NWMAX=20,NOUTMAX=3)
      DIMENSION  W(NWMAX),XSAM(50),OUT(NWMAX,NOUTMAX),XNOM(43),
     *NXRO(43),TOLE(43),IDIS(43),XNRESP(NWMAX,NOUTMAX)
      REAL*8  DXSAM(2700),C(140,NWMAX,NOUTMAX),COEF(140),
     *ANSWER(140),EPSILON(2700),RELEPS(2700),AVG(20)
C
C
C
      CALL  INDAT(IPRO,ICON,NX,XNOM,NW,W,NOUT,NXRO,TOLE,IDIS,
     *KOMEGA,KOUT)
C
C        find  the  responce  of  the  nomimal  point  and  store  in  XNRESP
C
      CALL  ISCAS(XNOM(1),NX,W,NW,XNRESP(1,1),NWMAX,NOUTMAX,IRUB,
     *IPRO,IFLAG)
      PRINT*,'FLAG',IFLAG
      IFLAG=0
      ISTAT=0
      CALL  TRAFIC(NX,ISTAT,NREP,INTER)
   10 DO  K=1,IABS(NREP)
              CALL  GDXVEC(ISTAT,ICON,NX,XNOM,TOLE,NXRO,IDIS,XSAM)
              CALL  ISCAS(XSAM(1),NX,W,NW,OUT(1,1),NWMAX,NOUTMAX,IRUB,
     *        IPRO,IFLAG)
              CALL  DOUBLE(NX,ISTAT,NW,NOUT,OUT,NWMAX,C,140,DXSAM,
     *        XSAM)
              ISTAT=ISTAT+1
      END  DO
      IOSTAT=ISTAT-IABS(NREP)
C
C
C
      IF(IOSTAT.EQ.0)THEN
         CALL  AVERAGE(NX,NREP,DXSAM,AVG)
      ENDIF
      CALL  DEV(DXSAM(1),NX,ISTAT,AVG(1),EPSILON(1),RELEPS(1))
      CALL  MPRINT(NX,ISTAT,IOSTAT,XNOM,AVG,DXSAM,EPSILON,RELEPS)
      IF(IOSTAT.NE.0)THEN
         CALL  UPDTAVG(AVG,IOSTAT,DXSAM(IOSTAT*NX+1),NX,IABS(NREP))
      ENDIF
      CALL  NORMAL(NX,NW,NOUT,ISTAT,IOSTAT,DXSAM(1),C,140,NWMAX,
     *XNRESP,XNOM(1))
      CALL  SUBS(NX,COEF,DXSAM(1),ISTAT,ANSWER)
      IF(IOSTAT.NE.0)THEN
          CALL  PNTRESP(DXSAM,NX,ISTAT,IOSTAT,ANSWER,
     *    XNRESP(KOMEGA,KOUT),C(1,KOMEGA,KOUT))
      ENDIF
C
C
C
      DO  I=1,ISTAT
```

```
      PRINT*,ANSWER(I)*XNRESP(KOMEGA,KOUT),C(I,KOMEGA,KOUT)*
     *XNRESP(KOMEGA,KOUT)
      END DO
      PRINT*,'========================================'
99    CALL APP(NX,NREP,DXSAM(1),C(1,KOMEGA,KOUT),COEF(1),NCOEF)
      CALL PNTCOEF(COEF,NCOEF,W,I,J,ISTAT,INTER)
      CALL SUBS(NX,COEF,DXSAM(1),ISTAT,ANSWER)
      DO I=1,ISTAT
         PRINT*,ANSWER(I)*XNRESP(KOMEGA,KOUT),C(I,KOMEGA,KOUT)*
     *XNRESP(KOMEGA,KOUT)
      END DO
      CALL TRAFIC(NX,ISTAT,NREP,INTER)
      IF(NREP.NE.0)GO TO 10
      END
```

```
C*********************************************************************
C                              SUBROUTINE PNTRESP
C*********************************************************************
      SUBROUTINE PNTRESP(DXSAM,NX,ISTAT,IOSTAT,ANSWER,XNRESP,C)
      REAL*8 DXSAM(NX,1),ANSWER(1),C(1)
      WRITE(11,50)
   50 FORMAT(' ','==============================================')
      WRITE(11,60)
   60 FORMAT('0',5X,'BASE POINTS',/,' ',5X,'-----------')
      WRITE(11,100)
  100 FORMAT('0',5X,'PREDICTED RESPONSE',10X,'ACTUAL RESPONSE',
     */,' ',5X,'==================',10X,'===============')
      DO I=1,IOSTAT
         WRITE(11,200)ANSWER(I)*DBLE(XNRESP),C(I)*DBLE(XNRESP)
  200    FORMAT(' ',4X,E14.7,15X,E14.7)
      END DO
      WRITE(11,50)
      WRITE(11,70)
   70 FORMAT('0',5X,'RANDOM POINTS',/,' ',5X,'-------------')
      WRITE(11,100)
      DO I=IOSTAT+1,ISTAT
      WRITE(11,200)ANSWER(I)*DBLE(XNRESP),C(I)*DBLE(XNRESP)
      END DO
      RETURN
      END
C*********************************************************************
C                              SUBROUTINE NORMAL
C*********************************************************************
      SUBROUTINE NORMAL(NX,NW,NOUT,ISTAT,IOSTAT,DXSAM,
     *C,NR,NWMAX,XNRESP,XNOM)
      DIMENSION XNOM(1),XNRESP(NWMAX,1)
      REAL*8 C(NR,NWMAX,1),DXSAM(NX,1)
      DO I=IOSTAT+1,ISTAT
         DO J=1,NX
            DXSAM(J,I)=DXSAM(J,I)/DBLE(XNOM(J))
         END DO
         DO J=1,NW
            DO K=1,NOUT
              C(I,J,K)=C(I,J,K)/DBLE(XNRESP(J,K))
            END DO
         END DO
      END DO
      RETURN
      END
C*********************************************************************
C.                             SUBROUTINE DOUBLE
C*********************************************************************
      SUBROUTINE DOUBLE(NX,ISTAT,NW,NOUT,OUT,NWMAX,C,L,DXSAM,
     *XSAM)
      DIMENSION XSAM(1),OUT(NWMAX,1)
```

```
      REAL*8 DXSAM(1),C(L,NWMAX,1)
      DO I=1,NW
        DO J=1,NOUT
           C(ISTAT+1,I,J)=DBLE(OUT(I,J))
        END DO
      END DO
      DO I=1,NX
        DXSAM(ISTAT*NX+I)=DBLE(XSAM(I))
      END DO
      RETURN
      END
C*********************************************************************
C                      SUBROUTINE MPRINT
C*********************************************************************
      SUBROUTINE MPRINT(NX,ISTAT,IOSTAT,XNOM,AVG,DXSAM,EPSILON,
     *RELEPS)
      DIMENSION XNOM(1)
      REAL*8 AVG(1),DXSAM(NX,1),EPSILON(NX,1),RELEPS(NX,1)
      OPEN(UNIT=11,FILE='OUTPUT',STATUS='NEW')
      WRITE(11,100)ISTAT,IOSTAT
  100 FORMAT(' ',50X,'PRESENT NUMBER OF SAMPLES USED = ',I2,/,' ',
     *50X,'PREVIOUS NUMBER OF SAMPLES USES = ',I2)
      WRITE(11,200)
  200 FORMAT('0',5X,'NOMINAL POINT',10X,'AVERAGE POINT',/,' ',
     *5X,'=============',10X,'=============')
      DO I=1,NX
      WRITE(11,300)XNOM(I),AVG(I)
  300 FORMAT(' ',4X,E14.7,9X,E14.7)
      END DO
      WRITE(11,400)
  400 FORMAT('0',5X,'BASE POINTS',14X,'DEVIATION FROM AVG',
     *10X,'RELATIVE DEVIATION',/,' ',5X,'=============',12X,
     *'=================',10X,'=================')
      DO I=IOSTAT+1,ISTAT
        DO J=1,NX
           WRITE(11,500)DXSAM(J,I),EPSILON(J,I),
     *     RELEPS(J,I)
  500      FORMAT(' ',4X,E14.7,13X,E14.7,13X,E14.7)
        END DO
        WRITE(11,600)
  600   FORMAT(' ',4X,'===============',11X,'==============',
     *  13X,'=============')
      END DO
      RETURN
      END
C*********************************************************************
C                      SUBROUTINE UPDTAVG
C*********************************************************************
C
C INPUT:
```

```
C          NX=NUMBER OF VARIALBES
C          AVG=ARRAY OF NX ELEMENTS
C          NAVG=NUMBER OF VECTORS USED TO CALCULATE AVG ARRAY
C          DXSAM=ARRAY CONTAING NEW VECTORS THAT WILL UPDATE AVG
C          NUMBER=NUMBER OF VECTORS TO BE INCLUDED IN UPDATING AVG
C
C OUTPUT:
C          AVG=UPDATED AVG ARRAY
C
C
      SUBROUTINE UPDTAVG(AVG,NAVG,DXSAM,NX,NUMBER)
      REAL*8 DXSAM(NX,1),AVG(1)
      DO I=1,NX
        TEMP=0.0
          DO J=1,NUMBER
            TEMP=TEMP+DXSAM(I,J)
          END DO
        AVG(I)=(AVG(I)*NAVG+TEMP)/(NAVG+NUMBER)
      END DO
      RETURN
      END
C**********************************************************************
C                          SUBROUTINE DEV
C**********************************************************************
      SUBROUTINE DEV(DXSAM,NX,NUMBER,AVG,EPSILON,RELEPS)
      REAL*8 DXSAM(NX,1),EPSILON(NX,1),RELEPS(NX,1),AVG(1)
      DO I=1,NX
         DO J=1,NUMBER
            EPSILON(I,J)=DXSAM(I,J)-AVG(I)
            RELEPS(I,J)=EPSILON(I,J)/AVG(I)
         END DO
      END DO
      RETURN
      END
C**********************************************************************
C                          SUBROUTINE AVERAGE
C**********************************************************************
      SUBROUTINE AVERAGE(NX,NREP,DXSAM,AVG)
      REAL*8 DXSAM(NX,1),AVG(1)
      DO I=1,NX
       AVG(I)=0
      END DO
      DO I=1,NX
        DO J=1,NREP
          AVG(I)=AVG(I)+DXSAM(I,J)
        END DO
        AVG(I)=AVG(I)/NREP
      END DO
      RETURN
      END
```

```
C*********************************************************************
C                              SUBROUTINE GDXVEC
C*********************************************************************
      SUBROUTINE GDXVEC(ISTAT,ICON,NTOT,XNOM,TOLE,NXRO,IDIS,XSAM)
      DIMENSION XNOM(1),NXRO(1),TOLE(1),IDIS(1),XSAM(1)
      IF(ISTAT.EQ.0)THEN
            TEMP=ICON
            ICON=0
      ENDIF
   10 CALL SAMPLE(XSAM,PROB,XNOM,TOLE,NXRO,IDIS,NTOT,ICON)
      IF (ICON.EQ.0)THEN
         ICON=TEMP
         GO TO 10
      ENDIF
      RETURN
      END
C*********************************************************************
C                              SUBROUTINE INDAT
C*********************************************************************
      SUBROUTINE INDAT(IPRO,ICON,NX,XNOM,NW,W,NOUT,NXRO,
     *TOLE,IDIS,KOMEGA,KOUT)
      DIMENSION XNOM(1),W(1),NXRO(1),TOLE(1),IDIS(1)
      CHARACTER*4 FILENAME
   20 PRINT*,'WHICH ISCAS# DO YOU WISH TO USE THE APPROXIMATION ON ?'
      PRINT*,' '
      PRINT*,'ENTER THE NUMBER (1 - 14)'
      READ*,IPRO
      PRINT*,'ENTER KOMEGA'
      READ*,KOMEGA
      PRINT*,'ENTER KOUT'
      READ*,KOUT
      IF(IPRO.LT.1.OR.IPRO.GT.14)GO TO 20
      IF(IPRO.EQ.1)THEN
         NOUT=1
         NX=2
         FILENAME='EX1'
      ELSEIF(IPRO.EQ.2)THEN
         NOUT=1
         NX=2
         FILENAME='EX2'
      ELSEIF(IPRO.EQ.3)THEN
         NOUT=1
         NX=10
         FILENAME='EX3'
      ELSEIF(IPRO.EQ.4)THEN
         NOUT=3
         NX=9
         FILENAME='EX4'
      ELSEIF(IPRO.EQ.5)THEN
         NOUT=1
```

```
   NX=28
   FILENAME='EX5'
ELSEIF(IPRO.EQ.6)THEN
   NOUT=1
   NX=40
   FILENAME='EX6'
ELSEIF(IPRO.EQ.7)THEN
   NOUT=1
   NX=11
   FILENAME='EX7'
ELSEIF(IPRO.EQ.8)THEN
   NOUT=1
   NX=13
   FILENAME='EX8'
ELSEIF(IPRO.EQ.9)THEN
   NOUT=1
   NX=43
   FILENAME='EX9'
ELSEIF(IPRO.EQ.10)THEN
   NOUT=1
   NX=2
   FILENAME='EX10'
ELSEIF(IPRO.EQ.11)THEN
   NOUT=1
   NX=6
   FILENAME='EX11'
 ELSEIF(IPRO.EQ.12)THEN
   NOUT=1
   NX=2
   FILENAME='EX12'
ELSEIF(IPRO.EQ.13)THEN
   NOUT=1
   NX=2
   FILENAME='EX13'
ELSEIF(IPRO.EQ.14)THEN
   NOUT=2
   NX=3
   FILENAME='EX14'
ENDIF
OPEN(UNIT=9,FILE=FILENAME,STATUS='OLD')
READ(9,*)ICON
READ(9,*)NX
DO I=1,NX
     READ(9,*)XNOM(I),NXRO(I),TOLE(I),IDIS(I)
END DO
READ(9,*)NW
DO I=1,NW
     READ(9,*)W(I)
END DO
CLOSE(UNIT=9)
```

```
      RETURN
      END
C******************************************************************
C                        SUBROUTINE TRAFIC
C******************************************************************
      SUBROUTINE TRAFIC(NX,ISTAT,NREP,INTER,IPRINT)
      CHARACTER*1 RESP
      IFQ=NX+1+(NX*(NX+1))/2
      IF (ISTAT.EQ.0)THEN
  600    PRINT*,'INPUT THE NUMBER OF VECTORS YOU WANT INCLUDED '
         PRINT*,'IN THE APROXIMATION'
         PRINT*,'THE NUMBER ENTERED SHOULD BE IN THE FOLLOWING RANGE'
         PRINT*,' '
         PRINT*,NX+1,'       -',IFQ
         PRINT*,' '
         PRINT*,NX+1,'      FOR LINEAR APPROXIMATION'
         PRINT*,IFQ,'      FOR FULL QUADRATIC APPROXIMATION'
         READ*,K
         IF(K.LT.NX+1.OR.K.GT.IFQ)GO TO 600
         NREP=K
         CALL OUTFORM(INTER,IPRINT)
         RETURN
      ENDIF
C       IF(ISTAT.EQ.IFQ)STOP'FULL QUADRATIC APPROXIMATION IS REACHED'
      PRINT*,'DO YOU WISH TO UPDATE THE APPROXIMATION ?(Y/N)'
      READ(5,100)RESP
  100 FORMAT(A)
      IF(RESP.EQ.'Y'.OR.RESP.EQ.'y')THEN
  200    PRINT*,' '
         PRINT*,'THE NUMBER OF VECTORS NEEDED FOR LINEAR APPROXIMATION'
         PRINT*,'IS :',NX+1
         PRINT*,'THE NUMBER OF VECTORS NEEDED FOR FULL QUADRATIC '
         PRINT*,'APPROXIMATION IS :',IFQ
         PRINT*,' '
         PRINT*,'THE NUMBER OF VECTORS ALREADY USED IS :',ISTAT
         PRINT*,'ENTER THE NUMBER OF VECTORS YOU WANT INCLUDED '
         PRINT*,'THE NUMBER SHOULD BE IN THE FOLLOWING RANGE :'
         PRINT*,ISTAT+1,'       -',IFQ
         READ*,K
         IF(K.LT.ISTAT+1.OR.K.GT.IFQ)GO TO 200
         NREP=ISTAT-K
         CALL OUTFORM(INTER,IPRINT)
         RETURN
      ENDIF
      NREP=0
      RETURN
      END
C******************************************************************
C                        SUBROUTINE PNTCOEF
C******************************************************************
```

```
      SUBROUTINE PNTCOEF(A,N,W,NOMEGA,NUMOUT,ISTAT,INTER)
      DIMENSION W(1)
      REAL*8 A(1)
      IF(INTER.EQ.0.OR.INTER.EQ.2)THEN
          OPEN(UNIT=8,FILE='POLOUT',STATUS='NEW')
          WRITE(8,100)W(NOMEGA)
100       FORMAT('1',/,'0',/,'0',50X,'FREQUENCY = ',F12.5,'RADIANS')
          WRITE(8,200)NUMOUT
200       FORMAT(' ',50X,'OUTPUT NUMBER = ',I1)
          WRITE(8,250)ISTAT
250       FORMAT(' ',50X,'NUMBER OF SAMPLES USED = ',I2,'SAMPLES')
          WRITE(8,300)
300       FORMAT('0',/,'0','THE POLYNOMIAL COEFFICIENTS ARE:')
          DO I=1,N
              WRITE(8,400)A(I)
400           FORMAT(' ',E23.16)
          END DO
      ENDIF
      IF(INTER.EQ.1.OR.INTER.EQ.2)THEN
          PRINT*,'W=',W(NOMEGA)
          PRINT*,'OUTPUT=',NUMOUT
          PRINT*,'SAMPLES USED=',ISTAT
          PRINT*,'THE POLYNOMIAL COEFFICIENTS ARE:'
          DO I=1,N
              PRINT*,A(I)
          END DO
      ENDIF
      RETURN
      END
C*********************************************************************
C                            SUBROUTINE OUTFORM
C*********************************************************************
      SUBROUTINE OUTFORM(INTER,IPRINT)
   99 PRINT*,' '
      PRINT*,'DO YOU WISH THE COEFFICIENTS DISPLAYED ON THE SCREEN '
      PRINT*,'OR DO YOU WANT THEM STORED IN AN OUTPUT FILE ?'
      PRINT*,' '
      PRINT*,'ENTER 0 FOR OUTPUT FILE OPTION'
      PRINT*,'ENTER 1 FOR SCREEN DISPLAY OPTION'
      PRINT*,'ENTER 2 FOR BOTH OPTIONS'
      READ*,INTER
      IF(INTER.NE.0.AND.INTER.NE.1.AND.INTER.NE.2)GO TO 99
      PRINT*,' '
      PRINT*,'ENTER 1 IF YOU WANT THE SAMPLE POINTS AND RELATED'
      PRINT*,'DATA PRINTED TO A FILE'
      PRINT*,'ENTER ANY OTHER NUMBER IF OTHERWISE'
      READ*,IPRINT
      RETURN
      END
C*********************************************************************
```

```
C                   SUBROUTINE SUBS
C*************************************************************
C
C         THE STATEMENTS STARTING WITH A D PERFORM THE FUNCTION
C         OF A SUBROUTINE DIRVER PROGRAM FOR SUBROUTINE TESTING
C         PURPOSES.
C
D       REAL*8 COEF(10),VECTORS(30),ANSWER(10)
D       PRINT*,'INPUT NX'
D       READ*,NX
D       PRINT*,'INPUT COEFFICIENTS ARRAY '
D       DO I=1,NX+1+(NX*(NX+1))/2
D        READ*,COEF(I)
D       END DO
D       PRINT*,'INPUT NVEC'
D       READ*,NVEC
D       PRINT*,'INPUT THE VECTORS ARRAY IN ORDER'
D       DO I=1,NVEC
D          DO J=1,NX
D           READ*,VECTORS((I-1)*NX+J)
D          END DO
D       END DO
D       CALL SUBS(NX,COEF,VECTORS,NVEC,ANSWER)
D       DO I=1,10
D           PRINT*,ANSWER(I)
D       END DO
D       END
C
C           THIS SUBROUTINE CALCULATES THE VALUE OF A
C         POLYNOMIAL BY SUBSTITUING IN THE VECTOR AND PUTS
C       THE ANSWER IN THE ARRAY ANSWER.
C
C       ON INPUT:
C       NX=NUMBER OF VARIABLES (X1,X2,...,XNX)
C       COEF=ARRAY OF POLYNOMIAL COEFFICIENTS
C       NCOEF=NUMBER OF ELEMENTS IN COEF
C       VECTORS=ARRAY OF VECTORS TO BE SUBSTITUTED IN THE POLYNOMIAL
C             [(X1,X2,...XNX),.........,(X1,X2,...,XNX)]
C       NVEC=THE NUMBER OF VECTORS IN THE ARRAY VECTORS
C
C       ON OUTPUT:
C       ANSWER=ARRAY OF NVEC ELEMENTS
C
C
        SUBROUTINE SUBS(NX,COEF,VECTORS,NVEC,ANSWER)
        REAL*8 COEF(1),VECTORS(1),ANSWER(1)
        DO K=1,NVEC
        TEMP=COEF(1)
        DO I=1,NX
          TEMP=TEMP+COEF(I+1)*VECTORS(I+(K-1)*NX)
```

```
    TEMP=TEMP+COEF(NX+1+I)*VECTORS(I+(K-1)*NX)
**VECTORS(I+(K-1)*NX)
  END DO
  ICOUNT=1
  DO I=1,NX-1
     DO J=I+1,NX
         TEMP=TEMP+VECTORS(I+(K-1)*NX)*VECTORS
*(J+(K-1)*NX)*COEF(2*NX+1+ICOUNT)
         ICOUNT=ICOUNT+1
     END DO
  END DO
      ANSWER(K)=TEMP
  END DO
  RETURN
  END
```

```
C*****************************************************************************
C
C       NAME:          MUSTAFA HANI KHAMMASH
C
C       ADVISORS:      Dr. R. BIERNACKI
C                      Dr. M. STYBLINSKI
C
C       DEPARTMENT: ELECTRICAL ENGINEERING
C
C       PROGRAM:       POLYNOMIAL INTERPOLATION OF REAL FUNCTIONS WITH n VARIABLES
C
C*****************************************************************************
C
C           This program determines the (a) coefficients of a polynomial that
C       interpolates real functions with n variables. An interpolated function
C       which has the form y=f(X1,X2,...,Xn) maps vectors in n dimensional
C       space (Rn) into the set of real numbers R. The program uses as its
C       input different vectors in the function's domain (one such vector has
C       the form (X1,X2,...,Xn)) as well as each of these vectors' corresponding
C       value (y) under the mapping of the function. Naturally,the accuracy
C       of the interpolating polynomial increases as more vectors and their
C       values are available.
C           The interpolating polynomial used has the form:
C
C                 Pn(X1,...,Xn) =   a0 + (a1)X1 + (a2)X2 + ... + (an)Xn   +
C       (a11)X1**2 + (a22)X2**2 + ... + (ann)Xn**2 + (a12)X1*X2 + (a13)X1*X3
C       + ... + (a1n)X1*Xn + (a23)X2*X3 + (a24)X2*X4 + ... + (a2n)X2*Xn +
C       ........ + (a(n-1)n)Xn-1*Xn
C
C            The desired values are the (a) coefficients and they constitute
C       the output of the program. The program is divided into two main parts.
C       The first part determines the n+1 coefficients of the linear portion
C       of the polynomial (a1,a2,...,an). In order to determine n+1 unique
C       coefficients, n+1 different vectors must be provided along with each
C       vector's corresponding (y) value. The 2nd part of this pogram makes
C       use of the extra information about the function gained by the addi-
C       tional vectors to build up the quadratic terms and continuously update
C       all the coefficients of the polynomial. The vector immediately following
C       the first n+1 vectors is used along with its function value to build
C       up the quadratic part of the interpolating polynomial and to update
C       the linear coefficients. Once the quadratic coefficients are formed,
C       additional vectors and their function values are used to update all
C       the (a) coefficients and thus continuously improving the accuracy.
C
C             The input vectors are arranged in the one dimensional array
C       DXSAM starting with the 2nd element in the array, with the first
C       n+1 vectors designated for finding the linear coefficients. The first
C       element in DXSAM contains the value of n, or the dimension of the vector
C       space. The function values of each of the vectors stored in DXSAM is
C       stored in DXSAM_VALU in the same order as that of their corresponding
```

```
C     vectors.
C
C           Aside from DXSAM and DXSAM_VALU which constitute the input arrays
C     to this program, all other matrices used are stored in a single one
C     dimensional array called SPACE. To fascilitate ease of access, each of
C     the matrices stored in SPACE has a pointer pointing to the first location
C     in the space assigned for that matrix. Each of these pointers is an
C     integer constant with its name begining with the letter I, followed by
C     the name of the matrix it is associated with. The size and order of
C     arrangement of the matrices in SPACE is illustrated below.
C
C
C
C
C                  ILLUSTRATION OF THE CONTENTS AND ORGANIZATION
C                               OF THE ARRAY SPACE
C
C
C
C        SPACE(IQ11)     =======>[ ]
C                                 :     N1XN1          N1 = n+1
C        SPACE(IQ12)     =======>[ ]
C                                 :     L1XN1          L1 = ((n+2)*(n+1)/2)-(n+1)
C        SPACE(IB1)     ========>[ ]
C                                 :     N1X1
C        SPACE(IQINV)     ======>[ ]
C                                 :     N1XN1
C        SPACE(IALFN1)     =====>[ ]
C                                 :     N1X1
C        SPACE(ID)     =========>[ ]
C                                 :     L1XN1
C        SPACE(ILQ21)     ======>[ ]
C                                 :     N1X1
C        SPACE(IALPHM)     =====>[ ]
C                                 :     N1X1
C        SPACE(ILBM)     =======>[ ]
C                                 :     1X1
C        SPACE(ILDM)     =======>[ ]
C                                 :     1X1
C        SPACE(IZM)     ========>[ ]
C                                 :     1X1
C        SPACE(IXM)     ========>[ ]
C                                 :     L1X1
C        SPACE(IDELXM)     =====>[ ]
C                                 :     L1X1
C        SPACE(IA)     =========>[ ]
C                                 :     L1XL1
C        SPACE(IU)     =========>[ ]
C                                 :     L1XL1
C        SPACE(IL)     =========>[ ]
C                                 :     L1XL1
```

```
C          SPACE(ILQ22)  ======>[ ]
C                                 :    L1X1
C          SPACE(ILAM)   =======>[ ]
C                                 :    L1X1
C          SPACE(ILGM)   =======>[ ]
C                                 :    L1X1
C          SPACE(ILLM)   =======>[ ]
C                                 :    L1X1
C          SPACE(IWORK1)   =====>[ ]
C                                 :    L1XL1
C          SPACE(IWORK2)   =====>[ ]
C                                 :
C                                 :
C                                 :
C
C
C
C
C
C
      SUBROUTINE  APP(N,KL,DXSAM,DXSAM_VALU,COEF,NCOEF)
      PARAMETER(LENGTH=100000)
      REAL*8  SPACE(LENGTH),COEF(1),DXSAM(1),DXSAM_VALU(1)
      IF(KL.GT.0)THEN
        KLM=KL-(N+1)
      ELSEIF(KL.LT.0)THEN
        KLM=IUP-KL
        IUP=IUP+1
        IF  (IUP.EQ.1)GO  TO  88
        GO  TO  99
      ENDIF
C
C              INPUT  FIRST  n+1  VECTORS  AND  THEIR  CORRESPONDING  FUNCTION
C              VALUES,  AND  LOAD  THEM  INTO  DXSAM  AND  DXSAM_VALU  ARRAYS
C              RESPECTIVELY.
C
C              DELETED
C
C              DEFINE  N1  AND  L1
C
      N1=N+1
      L1=((N+2)*(N+1)/2)-(N+1)
C
C              DEFINE  POINTERS
C
      IQ11=4
      IQ12=IQ11+N1*N1
      IB1=IQ12+L1*N1
      IQINV=IB1+N1
      IALFN1=IQINV+N1*N1
```

```
      ID=IALFN1+N1
      ILQ21=ID+L1*N1
      IALPHM=ILQ21+N1
      ILBM=IALPHM+N1
      ILDM=ILBM+1
      IZM=ILDM+1
      IXM=IZM+1
      IDELXM=IXM+L1
      IA=IDELXM+L1
      IU=IA+L1*L1
      IL=IU+L1*L1
      ILQ22=IL+L1*L1
      ILAM=ILQ22+L1
      ILGM=ILAM+L1
      ILLM=ILGM+L1
      IWORK1=ILLM+L1
      IWORK2=IWORK1+L1*L1
C
C             LOAD MATRICES {Q11}, {Q12}, AND {B1} USING DATA AVAILABLE
C             IN DXSAM AND DXSAM_VALU ARRAYS.
C
      CALL LOAD(N1,L1,SPACE(IQ11),SPACE(IQ12),SPACE(IB1),DXSAM(1),
     *DXSAM_VALU(1))
D     PRINT*,'************************************* Q11 ******************
D    ************************'
D     CALL PNTMAT(N1,N1,SPACE(IQ11))
D     PRINT*,'************************************* Q12 ******************
D    ************************'
D     CALL PNTMAT(N1,L1,SPACE(IQ12))
D     PRINT*,'************************************* B1 *******************
D    ************************'
D     CALL PNTMAT(N1,1,SPACE(IB1))
C
C             CALCULATE {ALFN1} MATRIX AND STORE IT IN PROPER LOCATION
C             IN SPACE ARRAY.
C
      CALL INVERT(N1,N1,N1,SPACE(IQ11),SPACE(IQINV),SPACE(IWORK1),
     *SPACE(IWORK2))
D     PRINT*,'************************************* Q11 INV ***************
D    ************************'
D     CALL PNTMAT(N1,N1,SPACE(IQINV))
      CALL AXB(N1,N1,N1,N1,N1,1,SPACE(IQINV),SPACE(IB1),SPACE(IALFN1))
D     PRINT*,'********************************** ALPHA n+1 **************
D    ************************'
D     CALL PNTMAT(N1,1,SPACE(IALFN1))
C
C             CALCULATE {D} MATRIX AND STORE IT IN ASSIGNED LOCATION IN
C             SPACE ARRAY.
C
      CALL AXB(N1,N1,N1,N1,N1,L1,SPACE(IQINV),SPACE(IQ12),SPACE(ID))
```

```
D       PRINT*,'*************************************** D ******************
D      ************************'
C
C       FIND COEFFICIENTS FOR LINEAR INTERPOLATION AND RETURN IF REQUESTED
C
        IF(KLM.EQ.0)THEN
           IUP=0
           DO I=1,N1
              COEF(I)=SPACE(IALFN1+I-1)
           END DO
           NCOEF=N1
           RETURN
        ENDIF
C
C       BEGIN UPDATING PROCESS IF NEEDED
C
C
C            INITIALIZE {X}, {U}, {L}, AND {A} MATRICES TO ZERO
C
   88 CALL ZERO(L1,1,SPACE(IXM))
      CALL ZERO(L1,L1,SPACE(IA))
      CALL ZERO(L1,L1,SPACE(IU))
      CALL ZERO(L1,L1,SPACE(IL))
C
C            INITIALIZE UPDATING NUMBER (IUP) TO 1
C
      IUP=1
C
C            IF THE INCLUSION OF ANOTHER VECTOR AND ITS (y) VALUE IN THE
C            CREATION OR UPDATING(IF ALREADY CREATED) OF THE QUADRATIC
C            COEFFICIENTS IS DESIRED, THEN ENTER THAT VECTOR AND ITS (y)
C            VALUE; OTHERWISE STOP.
C
   99 CALL GENMAT(IUP,N1,SPACE(ILQ21),SPACE(ILQ22),SPACE(ILBM),DXSAM(1),
     *DXSAM_VALU(1))
D      PRINT*,'****** q21 ******'
D      CALL PNTMAT(N1,1,SPACE(ILQ21))
D      PRINT*,'****** q22 ******'
D      CALL PNTMAT(L1,1,SPACE(ILQ22))
C
C            CALCULATE {LAM} MATRIX AND STORE IT IN PROPER LACATION IN SPACE
C            ARRAY.
C
      CALL MATRAN(N1,L1,SPACE(ID),SPACE(IWORK1))
      CALL AXB(L1,N1,L1,L1,N1,1,SPACE(IWORK1),SPACE(ILQ21),
     *SPACE(IWORK2))
      CALL MATSUB(L1,1,SPACE(ILQ22),SPACE(IWORK2),SPACE(ILAM))
D      PRINT*,'******* am *******'
D      CALL PNTMAT(L1,1,SPACE(ILAM))
C
```

```
C           CALCULATE {LDM} MATRIX AND STORE IT IN ASSIGNED LOCATION IN
C           SPACE ARRAY.
C
      CALL AXB(1,N1,1,1,N1,1,SPACE(ILQ21),SPACE(IALFN1),SPACE(IWORK1))
      CALL MATSUB(1,1,SPACE(ILBM),SPACE(IWORK1),SPACE(IWORK2))
      CALL AXB(1,L1,1,1,L1,1,SPACE(ILAM),SPACE(IXM),SPACE(IWORK1))
      CALL MATSUB(1,1,SPACE(IWORK2),SPACE(IWORK1),SPACE(ILDM))
D     PRINT*,'******* dm *******'
D     CALL PNTMAT(1,1,SPACE(ILDM))
C
C           UPDATE {A}
C
D     PRINT*,'************************************** Am-1 ************
D    ***************************'
D     CALL PNTMAT(L1,L1,SPACE(IA))
      CALL UPDTA(L1,IUP,SPACE(IA),SPACE(ILAM))
D     PRINT*,'*************************************** Am *************
D    ****************************'
D     CALL PNTMAT(L1,L1,SPACE(IA))
C
C           CALCULATE NORM OF {LAM} AND STORE IT IN THE LOCATION ASSIGNED
C           ASSIGNED FOR {ZM}.
C
      CALL NORM(L1,SPACE(ILAM),SPACE(IZM))
C
C           IF UPDATING NUMBER IS 1, CREATE {L} AND {U} MATRICES AND STORE
C           THEM IN THEIR ASSIGNED LOCATION IN SPACE ARRAY.
C
      IF(IUP.EQ.1)THEN
      SPACE(IL)=1.0
      SPACE(IU)=1.0/SPACE(IZM)
      CALL MATRAN(L1,L1,SPACE(IA),SPACE(IWORK2))
      CALL SMULT(L1,L1,L1,1,SPACE(ILDM)/SPACE(IZM),SPACE(IWORK2),
     *SPACE(IDELXM))
D     PRINT*,'******* Lm *******'
D     CALL PNTMAT(1,1,SPACE(IL))
D     PRINT*,'******* Um *******'
D     CALL PNTMAT(1,1,SPACE(IU))
C
C           IF UPDATING NUMBER IS DIFFERENT FROM ZERO, CALCULATE THE MATRIX
C           {LGM} FOR THE NEW VECTOR AND STORE IT IN ASSIGNED LOCATION IN
C           SPACE ARRAY.
C
      ELSE
      CALL AXB(L1,L1,L1,IUP-1,L1,1,SPACE(IA),SPACE(ILAM),SPACE(ILGM))
D     PRINT*,'************************************* gm ******************
D    *************************'
D     CALL PNTMAT(L1,1,SPACE(ILGM))
C
C           CALCULATE THE MATRIX {LM} FOR THE NEW VECTOR AND STORE IT IN
```

```
C            ASSIGNED LOCATION IN SPACE ARRAY.
C
       CALL AXB(L1,L1,IUP-1,IUP-1,IUP-1,IUP-1,SPACE(IU),SPACE(IL),
     *SPACE(IWORK1))
       CALL AXB(IUP-1,L1,L1,IUP-1,IUP-1,1,SPACE(IWORK1),SPACE(ILGM),
     *SPACE(ILLM))
       CALL SMULT(L1,L1,IUP-1,1,-1.0,SPACE(ILLM),SPACE(ILLM))
D      PRINT*,'************************************ lm ****************
D    **************************'
D      CALL PNTMAT(L1,1,SPACE(ILLM))
C
C            CALCULATE THE MATRIX {ZM} FOR THE NEW VECTOR AND STORE IT
C            IN PROPER LOCATION IN SPACE ARRAY.
C
       CALL AXB(1,L1,1,1,IUP-1,1,SPACE(ILGM),SPACE(ILLM),
     *SPACE(IWORK1))
       CALL MATADD(1,1,1,1,1,SPACE(IZM),SPACE(IWORK1),SPACE(IZM))
D      PRINT*,'************************************ Zm ****************
D    **************************'
D      CALL PNTMAT(1,1,SPACE(IZM))
C
C            UPDATE {L}, AND {U} MARICES.
C
       CALL SMULT(L1,L1,L1,1,1.0/SPACE(IZM),SPACE(ILLM),
     *SPACE(IWORK1))
       SPACE(IWORK1+IUP-1)=1.0/SPACE(IZM)
       CALL UPDTL(IUP,L1,SPACE(ILLM),SPACE(IL))
       CALL UPDTU(IUP,L1,SPACE(IWORK1),SPACE(IU))
D      PRINT*,'************************************ L ****************
D    **************************'
D      CALL PNTMAT(L1,L1,SPACE(IL))
D      PRINT*,'************************************ U ****************
D    **************************'
D      CALL PNTMAT(L1,L1,SPACE(IU))
C
C            CALCULATE {DELXM} MATRIX AND STORE IT IN ASSIGNED LOCATION
C            IN SPACE ARRAY.
C
       CALL MATRAN(L1,L1,SPACE(IA),SPACE(IWORK2))
       CALL AXB(L1,L1,L1,L1,IUP,1,SPACE(IWORK2),SPACE(IWORK1),
     *SPACE(IDELXM))
       CALL SMULT(L1,L1,L1,1,SPACE(ILDM),SPACE(IDELXM),
     *SPACE(IDELXM))
       ENDIF
D      PRINT*,'********************************* DELTA Xm **********
D    ******************************'
D      CALL PNTMAT(L1,1,SPACE(IDELXM))
C
C            CREATE/UPDATE(IF ALREADY CREATED) THE MATRIX {XM}. ELEMENTS
C            IN THIS MATRIX ARE THE COEFFICIENTS OF THE QUADRATIC TERMS
```

```
C             OF THE POLYNOMIAL. (PROGRAM OUTPUT)
      CALL MATADD(L1,L1,L1,L1,L1,1,SPACE(IDELXM),SPACE(IXM),SPACE(IXM))
D      PRINT*,'************************************ Xm **************
D     ****************************'
D      CALL PNTMAT(L1,1,SPACE(IXM))
C
C
C        FIND COEFFICENTS AND STORE THEM IN COEF
      CALL AXB(N1,L1,N1,N1,L1,1,SPACE(ID),SPACE(IXM),SPACE(IWORK1))
      CALL MATSUB(N1,1,SPACE(IALFN1),SPACE(IWORK1),SPACE(IALPHM))
      DO I=1,N1
         COEF(I)=SPACE(IALPHM+(I-1))
      END DO
      DO I=1,L1
         COEF(I+N1)=SPACE(IXM+I-1)
      END DO
      NCOEF=N1+L1
C
C
C              IF FULL QUADRATIC APPROXIMATION HAS BEEN REACHED THEN STOP;
C              OTHERWISE REPEAT PROCEDURE STARTING WITH FORTRAM STATEMENT
C              NUMBER 10.
C
      IF(KLM.EQ.IUP)RETURN
C
C              INCREMENT UPDATING NUMBER (IUP) BY 1.
C
      IUP=IUP+1
      GO TO 99
      END
C*****************************************************************************
C              END OF MAIN PROGRAM.
C*****************************************************************************
      SUBROUTINE DATAIN(A,B,N)
      REAL*8 A(10),B(10)
      PRINT*,'ENTER THE NUMBER OF VARIABLES N >'
      READ(5,*)N
      PRINT*,'ENTER THE COEFFICIENTS MATRIX A (SIZE=N+1 X N)'
      DO I=1,N+1
           DO J=1,N
                WRITE(6,200)I,J
 200            FORMAT(1X,'A(',I2,',',I2,')=')
                L=(I-1)*N+J
                READ(5,*)A(L)
           END DO
      END DO
      PRINT*,'ENTER THE ELEMENTS OF MATRIX B (SIZE=N+1 X 1)'
      DO I=1,N+1
           WRITE(6,300)I
 300       FORMAT(1X,'B(',I2,')=')
```

```
        READ(5,*)B(I)
    END DO
    RETURN
    END
    SUBROUTINE MSG1
    PRINT*,'THE COEFFICIENTS OF THE LINEAR TERMS ARE ALPHA n+1; SECOND
   *ORDER COEFFICIENTS ARE ALL ZERO.'
    RETURN
    END
    SUBROUTINE ZERO(MS,NS,A)
    REAL*8 A(MS,1)
         DO I=1,MS
              DO J=1,NS
                   A(I,J)=0.0
              END DO
         END DO
    RETURN
    END
    SUBROUTINE PNTMAT(MS,NS,A)
    REAL*8 A(MS,1)
    OPEN(UNIT=9,FILE='PIOUT',STATUS='NEW')
    DO I=1,MS
    WRITE(6,10)(A(I,J),J=1,NS)
10  FORMAT(1X,10(E10.3,1X))
    END DO
    RETURN
    END
    SUBROUTINE LOAD(MS,NS,A,B,C,D,E)
    REAL*8 A(MS,1),B(MS,1),C(MS,1),D(1),E(MS,1)
    DO I=1,MS
    DO J=1,MS-1
    A(I,J+1)=D((I-1)*(MS-1)+J)
    B(I,J)=A(I,J+1)*A(I,J+1)
    END DO
    A(I,1)=1.0
    END DO
    DO I=1,MS
    ICOUNT=1
         DO J=2,MS-1
              DO K=J+1,MS
                   B(I,(MS-1)+ICOUNT)=A(I,J)*A(I,K)
                   ICOUNT=ICOUNT+1
              END DO
         END DO
    END DO
    DO I=1,MS
         C(I,1)=E(I,1)
    END DO
    RETURN
    END
```

```
SUBROUTINE GENMAT(MS,NS,A,B,C,D,E)
REAL*8 A(1),B(1),D(1),E(1)
DO I=2,NS
    A(I)=D((NS)*(NS-1)+(NS-1)*(MS-1)+I-1)
    B(I-1)=A(I)*A(I)
END DO
A(1)=1.0
ICOUNT=1
DO I=2,NS-1
     DO J=I+1,NS
          B((NS-1)+ICOUNT)=A(I)*A(J)
          ICOUNT=ICOUNT+1
     END DO
END DO
C=E(NS+MS)
RETURN
END
SUBROUTINE UPDTL(MS,NS,A,B)
REAL*8 A(1),B(NS,1)
      DO I=1,MS-1
           B(MS,I)=A(I)
           B(I,MS)=0.0
      END DO
B(MS,MS)=1.0
RETURN
END
SUBROUTINE UPDTU(MS,NS,A,B)
REAL*8 A(1),B(NS,1)
     DO I=1,MS
          B(MS,I)=0.0
          B(I,MS)=A(I)
     END DO
RETURN
END
SUBROUTINE MATRAN(MS,NS,A,B)
REAL*8 A(MS,1),B(NS,1)
     DO I=1,MS
          DO J=1,NS
               B(J,I)=A(I,J)
          END DO
     END DO
RETURN
END
SUBROUTINE SMULT(LA,LB,MS,NS,SCALAR,A,B)
REAL*8 A(LA,1),B(LB,1)
     DO I=1,MS
          DO J=1,NS
               B(I,J)=A(I,J)*SCALAR
          END DO
     END DO
```

```
RETURN
END
SUBROUTINE MATMUL(LS,MS,NS,A,B,C)
REAL*8 A(LS,1),B(MS,1),C(LS,1)
TEM=0.0
DO I=1,NS
     DO J=1,LS
          DO K=1,MS
               TEM=A(J,K)*B(K,I)
               C(J,I)=C(J,I)+TEM
          END DO
     END DO
END DO
RETURN
END
SUBROUTINE MATADD(LA,LB,LC,MS,NS,A,B,C)
REAL*8 A(LA,1),B(LB,1),C(LC,1)
     DO I=1,MS
          DO J=1,NS
               C(I,J)=A(I,J)+B(I,J)
          END DO
     END DO
RETURN
END
SUBROUTINE MATSUB(MS,NS,A,B,C)
REAL*8 A(MS,1),B(MS,1),C(MS,1)
DO I=1,MS
     DO J=1,NS
          C(I,J)=A(I,J)-B(I,J)
     END DO
END DO
RETURN
END
SUBROUTINE UPDTA(MS,NS,A,B)
REAL*8 A(MS,1),B(1)
DO I=1,MS
     A(NS,I)=B(I)
END DO
RETURN
END
SUBROUTINE NORM(MS,A,B)
REAL*8 A(1)
B=0.0
DO I=1,MS
     TEMP=A(I)*A(I)
     B=B+TEMP
END DO
RETURN
END
```

APPENDIX B

BASE POINTS AND PREDICTED VS. ACTUAL RESPONSES FOR VARIOUS NUMBER OF SAMPLES

ISCAS4 EXAMPLE

FREQUENCY = 18.0237 RADIANS
OUTPUT NUMBER = 3
NUMBER OF SAMPLES USED = 10SAMPLES

THE POLYNOMIAL COEFFICIENTS ARE:

-0.1145999729966857E+02
0.4643872926568869E+01
0.8725256656477007E+00
-0.3020990221212755E-01
-0.1579132149006098E+01
-0.6057252493583944E+01
0.1422832983243309E+01
0.4619595238991765E+01
0.5893422959728621E+01
0.2745146168154251E+01

ISCAS4 OUTPUT#3 FREQUENCY#5

PRESENT NUMBER OF SAMPLES USED = 10
PREVIOUS NUMBER OF SAMPLES USES = 0

NOMINAL POINT	AVERAGE POINT
0.1903000E+02	0.1884715E+02
0.3000000E-02	0.2965740E-02
0.1311000E+02	0.1314617E+02
0.5600000E-03	0.5656865E-03
0.4570000E+01	0.4565217E+01
0.5600000E-03	0.5514964E-03
0.7500000E+02	0.7499015E+02
0.7500000E+02	0.7470435E+02
0.7500000E+02	0.7515597E+02

BASE POINTS	DEVIATION FROM AVG	RELATIVE DEVIATION
0.1863364E+02	-0.2135098E+00	-0.1132849E-01
0.2986408E-02	0.2066826E-04	0.6969006E-02
0.1314856E+02	0.2386761E-02	0.1815555E-03
0.5613285E-03	-0.4358019E-05	-0.7703948E-02
0.4578350E+01	0.1313248E-01	0.2876638E-02
0.5801924E-03	0.2869595E-04	0.5203288E-01
0.7487767E+02	-0.1124779E+00	-0.1499902E-02
0.7545962E+02	0.7552658E+00	0.1011006E-01
0.7493042E+02	-0.2255547E+00	-0.3001154E-02
0.1876527E+02	-0.8187981E-01	-0.4344412E-02
0.3045191E-02	0.7945125E-04	0.2678969E-01
0.1335950E+02	0.2133271E+00	0.1622732E-01
0.5422755E-03	-0.2341096E-04	-0.4138504E-01
0.4614855E+01	0.4963818E-01	0.1087312E-01
0.5708377E-03	0.1934128E-04	0.3507054E-01
0.7475592E+02	-0.2342278E+00	-0.3123447E-02
0.7476848E+02	0.6412659E-01	0.8584050E-03
0.7605817E+02	0.9021996E+00	0.1200436E-01
0.1898621E+02	0.1390541E+00	0.7377990E-02
0.2965448E-02	-0.2920860E-06	-0.9848673E-04
0.1320794E+02	0.6176634E-01	0.4698427E-02
0.6073826E-03	0.4169611E-04	0.7370887E-01
0.4604646E+01	0.3942862E-01	0.8636745E-02
0.5170516E-03	-0.3444487E-04	-0.6245711E-01
0.7479639E+02	-0.1937538E+00	-0.2583724E-02
0.7561869E+02	0.9143387E+00	0.1223943E-01
0.7445711E+02	-0.6988670E+00	-0.9298889E-02
0.1897599E+02	0.1288345E+00	0.6835755E-02
0.2789617E-02	-0.1761232E-03	-0.5938593E-01
0.1340513E+02	0.2589528E+00	0.1969796E-01
0.5367783E-03	-0.2890821E-04	-0.5110287E-01
0.4549951E+01	-0.1526651E-01	-0.3344094E-02
0.5311570E-03	-0.2033947E-04	-0.3688051E-01
0.7613561E+02	0.1145464E+01	0.1527486E-01
0.7448629E+02	-0.2180618E+00	-0.2918998E-02
0.7386514E+02	-0.1290839E+01	-0.1717547E-01

0.1899977E+02	0.1526134E+00	0.8097427E-02
0.2795901E-02	-0.1698394E-03	-0.5726711E-01
0.1313946E+02	-0.6715107E-02	-0.5108031E-03
0.5790695E-03	0.1338298E-04	0.2365794E-01
0.4533521E+01	-0.3169641E-01	-0.6943025E-02
0.5603894E-03	0.8893001E-05	0.1612522E-01
0.7412605E+02	-0.8641029E+00	-0.1152289E-01
0.7431868E+02	-0.3856720E+00	-0.5162644E-02
0.7685626E+02	0.1700288E+01	0.2262345E-01
═══	═══	═══
0.1845476E+02	-0.3923923E+00	-0.2081971E-01
0.2803926E-02	-0.1618142E-03	-0.5456114E-01
0.1313054E+02	-0.1563864E-01	-0.1189596E-02
0.5609239E-03	-0.4762621E-05	-0.8419187E-02
0.4475001E+01	-0.9021578E-01	-0.1976155E-01
0.5596126E-03	0.8116220E-05	0.1471672E-01
0.7562456E+02	0.6344093E+00	0.8459902E-02
0.7467970E+02	-0.2465668E-01	-0.3300568E-03
0.7571716E+02	0.5611809E+00	0.7466883E-02
═══	═══	═══
0.1895757E+02	0.1104172E+00	0.5858560E-02
0.2950601E-02	-0.1513900E-04	-0.5104627E-02
0.1290614E+02	-0.2400344E+00	-0.1825888E-01
0.5032141E-03	-0.6247237E-04	-0.1104364E+00
0.4580147E+01	0.1492968E-01	0.3270310E-02
0.5481148E-03	-0.3381656E-05	-0.6131781E-02
0.7544382E+02	0.4536766E+00	0.6049816E-02
0.7464386E+02	-0.6049194E-01	-0.8097513E-03
0.7480563E+02	-0.3503487E+00	-0.4661621E-02
═══	═══	═══
0.1907677E+02	0.2296207E+00	0.1218331E-01
0.3135680E-02	0.1699395E-03	0.5730087E-01
0.1311959E+02	-0.2658777E-01	-0.2022472E-02
0.5856979E-03	0.2001143E-04	0.3537548E-01
0.4524822E+01	-0.4039536E-01	-0.8848507E-02
0.5658669E-03	0.1437052E-04	0.2605732E-01
0.7372791E+02	-0.1262243E+01	-0.1683212E-01
0.7350036E+02	-0.1203993E+01	-0.1611677E-01
0.7546471E+02	0.3087395E+00	0.4107983E-02
═══	═══	═══
0.1904937E+02	0.2022121E+00	0.1072905E-01
0.3017769E-02	0.5202868E-04	0.1754324E-01
0.1308291E+02	-0.6326704E-01	-0.4812582E-02
0.5659783E-03	0.2918416E-06	0.5159069E-03
0.4599957E+01	0.3473940E-01	0.7609583E-02
0.5367087E-03	-0.1478768E-04	-0.2681374E-01
0.7503989E+02	0.4973831E-01	0.6632646E-03
0.7461190E+02	-0.9245148E-01	-0.1237565E-02
0.7413230E+02	-0.1023673E+01	-0.1362065E-01
═══	═══	═══
0.1857218E+02	-0.2749702E+00	-0.1458948E-01
0.3166860E-02	0.2011202E-03	0.6781449E-01
0.1296198E+02	-0.1841901E+00	-0.1401093E-01
0.6142163E-03	0.4852981E-04	0.8578923E-01
0.4590923E+01	0.2570572E-01	0.5630777E-02

0.5450331E-03	-0.6463286E-05	-0.1171954E-01
0.7537366E+02	0.3835167E+00	0.5114228E-02
0.7495595E+02	0.2515961E+00	0.3367890E-02
0.7527285E+02	0.1168755E+00	0.1555105E-02

BASE POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1682850E+02	0.1682849E+02
0.1698779E+02	0.1698780E+02
0.1266971E+02	0.1266971E+02
0.1633964E+02	0.1633964E+02
0.1564811E+02	0.1564810E+02
0.1687192E+02	0.1687192E+02
0.1870902E+02	0.1870903E+02
0.1539803E+02	0.1539803E+02
0.1475305E+02	0.1475304E+02
0.1334546E+02	0.1334545E+02

RANDOM POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1687664E+02	0.1465965E+02
0.1713360E+02	0.1664453E+02
0.2281373E+02	0.1841751E+02
0.1798259E+02	0.1715001E+02
0.1566887E+02	0.1715209E+02

FREQUENCY = 18.0327 RADIANS
OUTPUT NUMBER = 3
NUMBER OF SAMPLES USED = 11SAMPLES

THE POLYNOMIAL COEFFICIENTS ARE:
0.5854458302911873E+01
0.6287490111971690E+00
-0.1784497732139059E+01
0.1029560764466579E+00
-0.6238896070340901E+01
-0.2518386130053006E+01
0.1834916658632215E+00
-0.2105503495409401E+00
0.1604803591829005E+01
0.1236512631785992E+00
-0.3672938664325465E-01
0.5378375306815863E+00
0.5940805972857609E-01
0.1838176137535177E+01
-0.5079431301810026E-01
0.2749274998199349E+00
0.3919868867922632E-01
0.2440954524650237E-01
-0.6621251184875558E-02
-0.2694762875041406E+00
0.3449473736369830E-01
-0.1775853190617884E+00
-0.7472384388290466E-01
0.5408556177580093E-01
0.1006096973215514E-01
-0.2356877192775863E-01
0.4745927606118920E-01
-0.2142936939869026E+00
0.8693829779858075E+00
-0.1302751988188022E+00
0.3889618652332641E+00
-0.5474832715608846E-01
-0.1925840848236109E-02
0.1788716266251757E+00
-0.2162254223425418E-01
-0.1916954060806779E-01
-0.1347633060076393E+00
0.2526467172503168E-01
-0.8734838924116187E-02
-0.6119596797953795E-01
0.4718411914242567E-01
0.1204058765945847E+00
-0.1530258236886534E+00
0.2682250895892767E-01
0.1870588772061682E+00
0.3610363211991482E-01
0.2639889714128803E-01
0.1860258160189752E-01

```
 0.2830702818968158E-01
-0.9614230201942328E-01
-0.1898573152856356E-01
-0.2532537381338813E-01
 0.8148946794687435E-02
-0.4119026340263776E-01
-0.9733070478931648E-02
```

ISCAS4 OUTPUT#3 FREQUENCY= 18.0327 RAD PRESENT NUMBER OF SAMPLES USED = 11
PREVIOUS NUMBER OF SAMPLES USES = 0

NOMINAL POINT	AVERAGE POINT
0.1903000E+02	0.1886809E+02
0.3000000E-02	0.2969745E-02
0.1311000E+02	0.1315171E+02
0.5600000E-03	0.5682607E-03
0.4570000E+01	0.4564525E+01
0.5600000E-03	0.5514224E-03
0.7500000E+02	0.7502286E+02
0.7500000E+02	0.7477738E+02
0.7500000E+02	0.7512242E+02

BASE POINTS	DEVIATION FROM AVG	RELATIVE DEVIATION
0.1863364E+02	-0.2344449E+00	-0.1242547E-01
0.2986408E-02	0.1666318E-04	0.5610980E-02
0.1314856E+02	-0.3143657E-02	-0.2390304E-03
0.5613285E-03	-0.6932188E-05	-0.1219896E-01
0.4578350E+01	0.1382412E-01	0.3028599E-02
0.5801924E-03	0.2876997E-04	0.5217411E-01
0.7487767E+02	-0.1451874E+00	-0.1935242E-02
0.7545962E+02	0.6822350E+00	0.9123548E-02
0.7493042E+02	-0.1919979E+00	-0.2555800E-02
0.1876527E+02	-0.1028149E+00	-0.5449144E-02
0.3045191E-02	0.7544617E-04	0.2540493E-01
0.1335950E+02	0.2077967E+00	0.1579998E-01
0.5422755E-03	-0.2598513E-04	-0.4572748E-01
0.4614855E+01	0.5032982E-01	0.1102630E-01
0.5708377E-03	0.1941530E-04	0.3520949E-01
0.7475592E+02	-0.2669373E+00	-0.3558079E-02
0.7476848E+02	-0.8904197E-02	-0.1190761E-03
0.7605817E+02	0.9357563E+00	0.1245642E-01
0.1898621E+02	0.1181190E+00	0.6260251E-02
0.2965448E-02	-0.4297165E-05	-0.1446981E-02
0.1320794E+02	0.5623592E-01	0.4275941E-02
0.6073826E-03	0.3912195E-04	0.6884507E-01
0.4604646E+01	0.4012025E-01	0.8789578E-02
0.5170516E-03	-0.3437085E-04	-0.6233125E-01
0.7479639E+02	-0.2264633E+00	-0.3018591E-02
0.7561869E+02	0.8413079E+00	0.1125083E-01
0.7445711E+02	-0.6653103E+00	-0.8856348E-02
0.1897599E+02	0.1078994E+00	0.5718619E-02
0.2789617E-02	-0.1801283E-03	-0.6065447E-01
0.1340513E+02	0.2534224E+00	0.1926917E-01
0.5367783E-03	-0.3148238E-04	-0.5540129E-01
0.4549951E+01	-0.1457487E-01	-0.3193076E-02
0.5311570E-03	-0.2026544E-04	-0.3675121E-01
0.7613561E+02	0.1112755E+01	0.1483221E-01
0.7448629E+02	-0.2910926E+00	-0.3892790E-02
0.7386514E+02	-0.1257283E+01	-0.1673645E-01

0.1899977E+02	0.1316783E+00	0.6978890E-02
0.2795901E-02	-0.1738444E-03	-0.5853850E-01
0.1313946E+02	-0.1224553E-01	-0.9310979E-03
0.5790695E-03	0.1080881E-04	0.1902086E-01
0.4533521E+01	-0.3100478E-01	-0.6792552E-02
0.5603894E-03	0.8967028E-05	0.1626163E-01
0.7412605E+02	-0.8968124E+00	-0.1195386E-01
0.7431868E+02	-0.4587028E+00	-0.6134245E-02
0.7685626E+02	0.1733844E+01	0.2308025E-01

0.1845476E+02	-0.4133275E+00	-0.2190617E-01
0.2803926E-02	-0.1658192E-03	-0.5583618E-01
0.1313054E+02	-0.2116906E-01	-0.1609605E-02
0.5609239E-03	-0.7336790E-05	-0.1291096E-01
0.4475001E+01	-0.8952414E-01	-0.1961302E-01
0.5596126E-03	0.8190247E-05	0.1485294E-01
0.7562456E+02	0.6016998E+00	0.8020220E-02
0.7467970E+02	-0.9768746E-01	-0.1306377E-02
0.7571716E+02	0.5947377E+00	0.7916913E-02

0.1895757E+02	0.8948205E-01	0.4742507E-02
0.2950601E-02	-0.1914408E-04	-0.6446370E-02
0.1290614E+02	-0.2455648E+00	-0.1867171E-01
0.5032141E-03	-0.6504654E-04	-0.1144660E+00
0.4580147E+01	0.1562132E-01	0.3422331E-02
0.5481148E-03	-0.3307629E-05	-0.5998358E-02
0.7544382E+02	0.4209671E+00	0.5611185E-02
0.7464386E+02	-0.1335227E+00	-0.1785603E-02
0.7480563E+02	-0.3167919E+00	-0.4217009E-02

0.1907677E+02	0.2086856E+00	0.1106024E-01
0.3135680E-02	0.1659344E-03	0.5587496E-01
0.1311959E+02	-0.3211819E-01	-0.2442131E-02
0.5856979E-03	0.1743726E-04	0.3068533E-01
0.4524822E+01	-0.3970372E-01	-0.8698323E-02
0.5658669E-03	0.1444454E-04	0.2619506E-01
0.7372791E+02	-0.1294952E+01	-0.1726077E-01
0.7350036E+02	-0.1277024E+01	-0.1707768E-01
0.7546471E+02	0.3422963E+00	0.4556513E-02

0.1904937E+02	0.1812770E+00	0.9607598E-02
0.3017769E-02	0.4802361E-04	0.1617095E-01
0.1308291E+02	-0.6879746E-01	-0.5231068E-02
0.5659783E-03	-0.2282328E-05	-0.4016339E-02
0.4599957E+01	0.3543104E-01	0.7762261E-02
0.5367087E-03	-0.1471365E-04	-0.2668309E-01
0.7503989E+02	0.1702881E-01	0.2269816E-03
0.7461190E+02	-0.1654823E+00	-0.2212999E-02
0.7413230E+02	-0.9901165E+00	-0.1318004E-01

0.1857218E+02	-0.2959054E+00	-0.1568285E-01
0.3166860E-02	0.1971151E-03	0.6637441E-01
0.1296198E+02	-0.1897205E+00	-0.1442554E-01
0.6142163E-03	0.4595564E-04	0.8087070E-01
0.4590923E+01	0.2639736E-01	0.5783155E-02

0.5450331E-03	-0.6389259E-05	-0.1158687E-01
0.7537366E+02	0.3508072E+00	0.4676004E-02
0.7495595E+02	0.1785653E+00	0.2387958E-02
0.7527285E+02	0.1504322E+00	0.2002495E-02

0.1907744E+02	0.2093513E+00	0.1109552E-01
0.3009796E-02	0.4005079E-04	0.1348627E-01
0.1320701E+02	0.5530418E-01	0.4205096E-02
0.5940024E-03	0.2574169E-04	0.4529909E-01
0.4557609E+01	-0.6916393E-02	-0.1515249E-02
0.5506821E-03	-0.7402639E-06	-0.1342462E-02
0.7534995E+02	0.3270950E+00	0.4359938E-02
0.7550769E+02	0.7303078E+00	0.9766427E-02
0.7478685E+02	-0.3355678E+00	-0.4466946E-02

BASE POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1682850E+02	0.1682849E+02
0.1698780E+02	0.1698780E+02
0.1266971E+02	0.1266971E+02
0.1633964E+02	0.1633964E+02
0.1564811E+02	0.1564810E+02
0.1687191E+02	0.1687192E+02
0.1870902E+02	0.1870903E+02
0.1539803E+02	0.1539803E+02
0.1475303E+02	0.1475304E+02
0.1334545E+02	0.1334545E+02
0.1465963E+02	0.1465965E+02

RANDOM POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1660087E+02	0.1664453E+02
0.1801246E+02	0.1841751E+02
0.1748724E+02	0.1715001E+02
0.1727151E+02	0.1715209E+02
0.1609066E+02	0.1613204E+02

FREQUENCY = 18.0327 RADIANS
OUTPUT NUMBER = 3
NUMBER OF SAMPLES USED = 15SAMPLES

```
0THE POLYNOMIAL COEFFICIENTS ARE:
 -0.6461257013891219E+01
  0.3981222489831385E+01
  0.5141978487239042E+01
  0.4349808639211382E+01
 -0.1891187178051686E+01
 -0.1530200970567875E+01
 -0.1642870531856995E+01
  0.5988362321484615E+01
  0.7830424207347979E+00
  0.8015379979134930E+00
  0.7038931542457340E+00
  0.5081811838345134E+00
 -0.4153991919403872E+00
  0.1392629147629395E+01
 -0.1020482641797406E+00
  0.1592807591429130E+01
 -0.3822585527249579E+00
  0.3947660827764128E+00
 -0.8957293805385355E-01
 -0.4780943845415205E+00
  0.4630932796693774E-01
 -0.1965365009622868E+01
 -0.6981890520221126E+00
 -0.1097921918510146E+01
 -0.1503202019045541E+00
 -0.8736819435772847E+00
  0.5737407927370116E-01
 -0.1471236915279613E+01
  0.5551411349062848E+00
 -0.1343899505292179E+01
 -0.1729501940927943E+01
 -0.1292612362400210E+01
  0.8566787174238004E-01
 -0.3009276807597518E+00
 -0.3868159113993894E+00
 -0.3596766537629615E+00
 -0.3432155419586501E+00
 -0.3103404249699607E+00
 -0.3284661799436262E-01
 -0.7263806357764684E+00
  0.7159467802318756E+00
  0.1118687297651672E+01
 -0.2919079269642750E+01
 -0.8388411443006014E+00
  0.1085208209906658E+01
 -0.1167123237498155E+00
  0.1878971239251798E+00
  0.3350669716896122E+00
```

-0.1167863490967382E+00
-0.9339855056085901E+00
0.1792437277177892E+01
0.6725729516077508E+00
0.2069532941807875E+00
-0.4624337699216449E+00
-0.3599342686429319E-01

ISCAS4 OUTPUT#3 FREQUENCY= 18.0327 RAD

PRESENT NUMBER OF SAMPLES USED = 15
PREVIOUS NUMBER OF SAMPLES USES = 0

NOMINAL POINT	AVERAGE POINT
0.1903000E+02	0.1892789E+02
0.3000000E-02	0.2987659E-02
0.1311000E+02	0.1313500E+02
0.5600000E-03	0.5632808E-03
0.4570000E+01	0.4554529E+01
0.5600000E-03	0.5561105E-03
0.7500000E+02	0.7492477E+02
0.7500000E+02	0.7474450E+02
0.7500000E+02	0.7488153E+02

BASE POINTS	DEVIATION FROM AVG	RELATIVE DEVIATION
0.1863364E+02	-0.2942434E+00	-0.1554549E-01
0.2986408E-02	-0.1250627E-05	-0.4185975E-03
0.1314856E+02	0.1356360E-01	0.1032631E-02
0.5613285E-03	-0.1952281E-05	-0.3465911E-02
0.4578350E+01	0.2382097E-01	0.5230173E-02
0.5801924E-03	0.2408191E-04	0.4330419E-01
0.7487767E+02	-0.4710185E-01	-0.6286552E-03
0.7545962E+02	0.7151225E+00	0.9567560E-02
0.7493042E+02	0.4888509E-01	0.6528324E-03
═══	═══	═══
0.1876527E+02	-0.1626134E+00	-0.8591208E-02
0.3045191E-02	0.5753236E-04	0.1925667E-01
0.1335950E+02	0.2245040E+00	0.1709204E-01
0.5422755E-03	-0.2100522E-04	-0.3729085E-01
0.4614855E+01	0.6032667E-01	0.1324543E-01
0.5708377E-03	0.1472724E-04	0.2648259E-01
0.7475592E+02	-0.1688517E+00	-0.2253617E-02
0.7476848E+02	0.2398326E-01	0.3208699E-03
0.7605817E+02	0.1176639E+01	0.1571334E-01
═══	═══	═══
0.1898621E+02	0.5832049E-01	0.3081194E-02
0.2965448E-02	-0.2221097E-04	-0.7434240E-02
0.1320794E+02	0.7294318E-01	0.5553345E-02
0.6073826E-03	0.4410185E-04	0.7829462E-01
0.4604646E+01	0.5011711E-01	0.1100380E-01
0.5170516E-03	-0.3905890E-04	-0.7023587E-01
0.7479639E+02	-0.1283778E+00	-0.1713422E-02
0.7561869E+02	0.8741954E+00	0.1169578E-01
0.7445711E+02	-0.4244273E+00	-0.5667983E-02
═══	═══	═══
0.1897599E+02	0.4810092E-01	0.2541272E-02
0.2789617E-02	-0.1980421E-03	-0.6628672E-01
0.1340513E+02	0.2701296E+00	0.2056564E-01
0.5367783E-03	-0.2650247E-04	-0.4705019E-01
0.4549951E+01	-0.4578018E-02	-0.1005157E-02
0.5311570E-03	-0.2495350E-04	-0.4487148E-01
0.7613561E+02	0.1210840E+01	0.1616075E-01
0.7448629E+02	-0.2582052E+00	-0.3454504E-02
0.7386514E+02	-0.1016400E+01	-0.1357343E-01

0.1899977E+02	0.7187983E-01	0.3797562E-02
0.2795901E-02	-0.1917582E-03	-0.6418345E-01
0.1313946E+02	0.4461734E-02	0.3396828E-03
0.5790695E-03	0.1578872E-04	0.2802992E-01
0.4533521E+01	-0.2100792E-01	-0.4612534E-02
0.5603894E-03	0.4278969E-05	0.7694459E-02
0.7412605E+02	-0.7987269E+00	-0.1066038E-01
0.7431868E+02	-0.4258153E+00	-0.5696946E-02
0.7685626E+02	0.1974727E+01	0.2637135E-01

0.1845476E+02	-0.4731260E+00	-0.2499624E-01
0.2803926E-02	-0.1837330E-03	-0.6149733E-01
0.1313054E+02	-0.4461797E-02	-0.3396877E-03
0.5609239E-03	-0.2356883E-05	-0.4184206E-02
0.4475001E+01	-0.7952728E-01	-0.1746114E-01
0.5596126E-03	0.3502188E-05	0.6297648E-02
0.7562456E+02	0.6997854E+00	0.9339840E-02
0.7467970E+02	-0.6480001E-01	-0.8669536E-03
0.7571716E+02	0.8356206E+00	0.1115923E-01

0.1895757E+02	0.2968356E-01	0.1568245E-02
0.2950601E-02	-0.3705788E-04	-0.1240365E-01
0.1290614E+02	-0.2288575E+00	-0.1742349E-01
0.5032141E-03	-0.6006664E-04	-0.1066371E+00
0.4580147E+01	0.2561817E-01	0.5624769E-02
0.5481148E-03	-0.7995688E-05	-0.1437788E-01
0.7544382E+02	0.5190526E+00	0.6927650E-02
0.7464386E+02	-0.1006353E+00	-0.1346390E-02
0.7480563E+02	-0.7590892E-01	-0.1013720E-02

0.1907677E+02	0.1488871E+00	0.7866019E-02
0.3135680E-02	0.1480206E-03	0.4954401E-01
0.1311959E+02	-0.1541093E-01	-0.1173273E-02
0.5856979E-03	0.2241717E-04	0.3979751E-01
0.4524822E+01	-0.2970686E-01	-0.6522488E-02
0.5658669E-03	0.9756485E-05	0.1754415E-01
0.7372791E+02	-0.1196867E+01	-0.1597425E-01
0.7350036E+02	-0.1244137E+01	-0.1664519E-01
0.7546471E+02	0.5831792E+00	0.7788024E-02

0.1904937E+02	0.1214785E+00	0.6417965E-02
0.3017769E-02	0.3010980E-04	0.1007806E-01
0.1308291E+02	-0.5209020E-01	-0.3965756E-02
0.5659783E-03	0.2697580E-05	0.4789050E-02
0.4599957E+01	0.4542789E-01	0.9974225E-02
0.5367087E-03	-0.1940171E-04	-0.3488823E-01
0.7503989E+02	0.1151143E+00	0.1536399E-02
0.7461190E+02	-0.1325948E+00	-0.1773974E-02
0.7413230E+02	-0.7492335E+00	-0.1000558E-01

0.1857218E+02	-0.3557039E+00	-0.1879258E-01
0.3166860E-02	0.1792013E-03	0.5998050E-01
0.1296198E+02	-0.1730132E+00	-0.1317193E-01

0.6142163E-03	0.5093555E-04	0.9042657E-01
0.4590923E+01	0.3639421E-01	0.7990775E-02
0.5450331E-03	-0.1107732E-04	-0.1991928E-01
0.7537366E+02	0.4488927E+00	0.5991246E-02
0.7495595E+02	0.2114527E+00	0.2829008E-02
0.7527285E+02	0.3913152E+00	0.5225790E-02

0.1907744E+02	0.1495528E+00	0.7901188E-02
0.3009796E-02	0.2213698E-04	0.7409473E-02
0.1320701E+02	0.7201144E-01	0.5482410E-02
0.5940024E-03	0.3072160E-04	0.5454047E-01
0.4557609E+01	0.3080463E-02	0.6763518E-03
0.5506821E-03	-0.5428322E-05	-0.9761230E-02
0.7534995E+02	0.4251806E+00	0.5674766E-02
0.7550769E+02	0.7631953E+00	0.1021072E-01
0.7478685E+02	-0.9468486E-01	-0.1264462E-02

0.1932299E+02	0.3951010E+00	0.2087402E-01
0.2993234E-02	0.5574571E-05	0.1865866E-02
0.1309305E+02	-0.4194883E-01	-0.3193668E-02
0.5494590E-03	-0.1382175E-04	-0.2453795E-01
0.4506594E+01	-0.4793444E-01	-0.1052457E-01
0.5466187E-03	-0.9491799E-05	-0.1706819E-01
0.7423624E+02	-0.6885356E+00	-0.9189692E-02
0.7393747E+02	-0.8070257E+00	-0.1079712E-01
0.7395318E+02	-0.9283564E+00	-0.1239767E-01

0.1938814E+02	0.4602503E+00	0.2431599E-01
0.3109088E-02	0.1214290E-03	0.4064353E-01
0.1325648E+02	0.1214852E+00	0.9248970E-02
0.5461388E-03	-0.1714198E-04	-0.3043239E-01
0.4494438E+01	-0.6009092E-01	-0.1319366E-01
0.5601500E-03	0.4039561E-05	0.7263955E-02
0.7513364E+02	0.2088720E+00	0.2787756E-02
0.7598312E+02	0.1238621E+01	0.1657140E-01
0.7459329E+02	-0.2882426E+00	-0.3849315E-02

0.1873581E+02	-0.1920801E+00	-0.1014799E-01
0.3130701E-02	0.1430420E-03	0.4787761E-01
0.1304114E+02	-0.9385351E-01	-0.7145300E-02
0.5617166E-03	-0.1564152E-05	-0.2776861E-02
0.4564159E+01	0.9630775E-02	0.2114549E-02
0.6148461E-03	0.5873567E-04	0.1056187E+00
0.7565073E+02	0.7259618E+00	0.9689209E-02
0.7388667E+02	-0.8578222E+00	-0.1147673E-01
0.7421711E+02	-0.6644252E+00	-0.8873017E-02

0.1892240E+02	-0.5487951E-02	-0.2899399E-03
0.2914665E-02	-0.7299366E-04	-0.2443173E-01
0.1296554E+02	-0.1694627E+00	-0.1290162E-01
0.5410297E-03	-0.2225110E-04	-0.3950267E-01
0.4542958E+01	-0.1157084E-01	-0.2540512E-02
0.5543957E-03	-0.1714790E-05	-0.3083542E-02
0.7359953E+02	-0.1325239E+01	-0.1768760E-01

0.7480896E+02	0.6446482E-01	0.8624692E-03
0.7411285E+02	-0.7686885E+00	-0.1026539E-01

BASE POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1682847E+02	0.1682849E+02
0.1698782E+02	0.1698780E+02
0.1266971E+02	0.1266971E+02
0.1633964E+02	0.1633964E+02
0.1564811E+02	0.1564810E+02
0.1687189E+02	0.1687192E+02
0.1870904E+02	0.1870903E+02
0.1539804E+02	0.1539803E+02
0.1475304E+02	0.1475304E+02
0.1334544E+02	0.1334545E+02
0.1465967E+02	0.1465965E+02
0.1664453E+02	0.1664453E+02
0.1841749E+02	0.1841751E+02
0.1715003E+02	0.1715001E+02
0.1715210E+02	0.1715209E+02

RANDOM POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1616760E+02	0.1613204E+02
0.1561548E+02	0.1566544E+02
0.1433121E+02	0.1383851E+02
0.1744162E+02	0.1729080E+02
0.1572512E+02	0.1586370E+02

FREQUENCY = 18.0327 RADIANS
OUTPUT NUMBER = 3
NUMBER OF SAMPLES USED = 20SAMPLES

THE POLYNOMIAL COEFFICIENTS ARE:

-0.6942189249561107E+01
0.1197806264771237E+01
0.3860950794570857E+01
0.1357805418088463E+01
0.6112931401673117E+01
0.4528271281973687E+01
-0.8363060658277107E+01
0.8167542703748585E+01
0.2155254876330136E+01
-0.1387890864798867E+01
0.1351648160815000E+01
0.1071125232068562E+01
0.6500036187246283E-01
0.2918051803614988E+01
0.3728234632207067E+00
-0.6516229147410028E+00
-0.1082771933434538E+01
0.1608542517405906E+00
-0.4127460566206703E+00
0.1528949355093746E+00
0.4445240664741794E+00
-0.3269883677933624E+01
-0.1865373181692102E+01
-0.4660024544862896E+00
0.2468824292864001E+00
0.3034767607993270E+00
0.5994674357305526E+00
0.1581612023270395E-01
-0.3203673969739217E+01
-0.3977551447269279E+01
0.1338832298049088E+01
-0.1715481158457447E+01
0.3807014394528787E+00
0.1157040603322047E+01
-0.3206508864113564E+01
-0.1280278193610220E+01
0.3132834614387619E+01
-0.6020293145204069E+00
0.2177630814053360E+00
-0.2937374953051638E+00
0.1458029908734869E+01
0.1119189720368898E+01
-0.3060536376346365E+01
-0.3655803057591994E+01
0.1926153103996209E+00
-0.3883475968382310E+00
-0.3187500987419596E+00

-0.1520769156605685E+01
-0.3946824273804383E+00
0.2162803475146299E+00
0.4343731752628837E+01
0.1334904247816152E+01
-0.8710451850707842E+00
-0.2964500342466962E+00
0.2477735266265455E+00

ISCAS4 OUTPUT#3 FREQUENCY= 18.0327 RAD

PRESENT NUMBER OF SAMPLES USED = 20
PREVIOUS NUMBER OF SAMPLES USES = 0

NOMINAL POINT	AVERAGE POINT
0.1903000E+02	0.1893901E+02
0.3000000E-02	0.3008737E-02
0.1311000E+02	0.1312772E+02
0.5600000E-03	0.5628793E-03
0.4570000E+01	0.4564410E+01
0.5600000E-03	0.5569131E-03
0.7500000E+02	0.7497114E+02
0.7500000E+02	0.7473571E+02
0.7500000E+02	0.7492192E+02

BASE POINTS	. DEVIATION FROM AVG	RELATIVE DEVIATION
0.1863364E+02	-0.3053650E+00	-0.1612360E-01
0.2986408E-02	-0.2232835E-04	-0.7421173E-02
0.1314856E+02	0.2084284E-01	0.1587697E-02
0.5613285E-03	-0.1550774E-05	-0.2755075E-02
0.4578350E+01	0.1393938E-01	0.3053928E-02
0.5801924E-03	0.2327925E-04	0.4180050E-01
0.7487767E+02	-0.9346695E-01	-0.1246706E-02
0.7545962E+02	0.7239044E+00	0.9686191E-02
0.7493042E+02	0.8499527E-02	0.1134451E-03
0.1876527E+02	-0.1737350E+00	-0.9173397E-02
0.3045191E-02	0.3645463E-04	0.1211626E-01
0.1335950E+02	0.2317832E+00	0.1765602E-01
0.5422755E-03	-0.2060371E-04	-0.3660414E-01
0.4614855E+01	0.5044508E-01	0.1105183E-01
0.5708377E-03	0.1392458E-04	0.2500314E-01
0.7475592E+02	-0.2152168E+00	-0.2870662E-02
0.7476848E+02	0.3276520E-01	0.4384142E-03
0.7605817E+02	0.1136254E+01	0.1516584E-01
0.1898621E+02	0.4719887E-01	0.2492151E-02
0.2965448E-02	-0.4328870E-04	-0.1438767E-01
0.1320794E+02	0.8022242E-01	0.6110918E-02
0.6073826E-03	0.4450336E-04	0.7906378E-01
0.4604646E+01	0.4023552E-01	0.8815053E-02
0.5170516E-03	-0.3986157E-04	-0.7157592E-01
0.7479639E+02	-0.1747429E+00	-0.2330802E-02
0.7561869E+02	0.8829773E+00	0.1181466E-01
0.7445711E+02	-0.4648129E+00	-0.6203963E-02
0.1897599E+02	0.3697929E-01	0.1952546E-02
0.2789617E-02	-0.2191198E-03	-0.7282786E-01
0.1340513E+02	0.2774089E+00	0.2113154E-01
0.5367783E-03	-0.2610096E-04	-0.4637044E-01
0.4549951E+01	-0.1445961E-01	-0.3167903E-02
0.5311570E-03	-0.2575617E-04	-0.4624808E-01
0.7613561E+02	0.1164475E+01	0.1553231E-01
0.7448629E+02	-0.2494232E+00	-0.3337403E-02
0.7386514E+02	-0.1056785E+01	-0.1410515E-01

0.1899977E+02	0.6075821E-01	0.3208099E-02
0.2795901E-02	-0.2128360E-03	-0.7073932E-01
0.1313946E+02	0.1174097E-01	0.8943649E-03
0.5790695E-03	0.1619022E-04	0.2876322E-01
0.4533521E+01	-0.3088951E-01	-0.6767470E-02
0.5603894E-03	0.3476304E-05	0.6242094E-02
0.7412605E+02	-0.8450920E+00	-0.1127223E-01
0.7431868E+02	-0.4170334E+00	-0.5580108E-02
0.7685626E+02	0.1934342E+01	0.2581810E-01

0.1845476E+02	-0.4842476E+00	-0.2556879E-01
0.2803926E-02	-0.2048108E-03	-0.6807202E-01
0.1313054E+02	0.2817440E-02	0.2146176E-03
0.5609239E-03	-0.1955376E-05	-0.3473881E-02
0.4475001E+01	-0.8940887E-01	-0.1958826E-01
0.5596126E-03	0.2699523E-05	0.4847296E-02
0.7562456E+02	0.6534203E+00	0.8715624E-02
0.7467970E+02	-0.5601807E-01	-0.7495488E-03
0.7571716E+02	0.7952351E+00	0.1061418E-01

0.1895757E+02	0.1856194E-01	0.9800901E-03
0.2950601E-02	-0.5813561E-04	-0.1932227E-01
0.1290614E+02	-0.2215783E+00	-0.1687866E-01
0.5032141E-03	-0.5966513E-04	-0.1059999E+00
0.4580147E+01	0.1573658E-01	0.3447670E-02
0.5481148E-03	-0.8798353E-05	-0.1579843E-01
0.7544382E+02	0.4726875E+00	0.6304927E-02
0.7464386E+02	-0.9185333E-01	-0.1229042E-02
0.7480563E+02	-0.1162945E+00	-0.1552209E-02

0.1907677E+02	0.1377655E+00	0.7274166E-02
0.3135680E-02	0.1269429E-03	0.4219142E-01
0.1311959E+02	-0.8131695E-02	-0.6194294E-03
0.5856979E-03	0.2281868E-04	0.4053921E-01
0.4524822E+01	-0.3958845E-01	-0.8673290E-02
0.5658669E-03	0.8953820E-05	0.1607759E-01
0.7372791E+02	-0.1243232E+01	-0.1658281E-01
0.7350036E+02	-0.1235355E+01	-0.1652964E-01
0.7546471E+02	0.5427937E+00	0.7244791E-02

0.1904937E+02	0.1103569E+00	0.5826963E-02
0.3017769E-02	0.9032071E-05	0.3001948E-02
0.1308291E+02	-0.4481096E-01	-0.3413462E-02
0.5659783E-03	0.3099086E-05	0.5505775E-02
0.4599957E+01	0.3554630E-01	0.7787710E-02
0.5367087E-03	-0.2020438E-04	-0.3627923E-01
0.7503989E+02	0.6874924E-01	0.9170094E-03
0.7461190E+02	-0.1238129E+00	-0.1656676E-02
0.7413230E+02	-0.7896191E+00	-0.1053923E-01

0.1857218E+02	-0.3668255E+00	-0.1936878E-01
0.3166860E-02	0.1581235E-03	0.5255480E-01
0.1296198E+02	-0.1657340E+00	-0.1262474E-01

0.6142163E-03	0.5133706E-04	0.9120438E-01
0.4590923E+01	0.2651262E-01	0.5808554E-02
0.5450331E-03	-0.1187998E-04	-0.2133184E-01
0.7537366E+02	0.4025276E+00	0.5369101E-02
0.7495595E+02	0.2202347E+00	0.2946847E-02
0.7527285E+02	0.3509296E+00	0.4683938E-02

0.1907744E+02	0.1384312E+00	0.7309314E-02
0.3009796E-02	0.1059251E-05	0.3520585E-03
0.1320701E+02	0.7929068E-01	0.6039943E-02
0.5940024E-03	0.3112311E-04	0.5529269E-01
0.4557609E+01	-0.6801128E-02	-0.1490034E-02
0.5506821E-03	-0.6230987E-05	-0.1118844E-01
0.7534995E+02	0.3788155E+00	0.5052817E-02
0.7550769E+02	0.7719772E+00	0.1032943E-01
0.7478685E+02	-0.1350704E+00	-0.1802816E-02

0.1932299E+02	0.3839794E+00	0.2027453E-01
0.2993234E-02	-0.1550316E-04	-0.5152713E-02
0.1309305E+02	-0.3466959E-01	-0.2640946E-02
0.5494590E-03	-0.1342025E-04	-0.2384214E-01
0.4506594E+01	-0.5781603E-01	-0.1266670E-01
0.5466187E-03	-0.1029446E-04	-0.1848487E-01
0.7423624E+02	-0.7349007E+00	-0.9802448E-02
0.7393747E+02	-0.7982437E+00	-0.1068089E-01
0.7395318E+02	-0.9687420E+00	-0.1293002E-01

0.1938814E+02	0.4491287E+00	0.2371448E-01
0.3109088E-02	0.1003513E-03	0.3335329E-01
0.1325648E+02	0.1287644E+00	0.9808592E-02
0.5461388E-03	-0.1674047E-04	-0.2974078E-01
0.4494438E+01	-0.6997252E-01	-0.1533002E-01
0.5601500E-03	0.3236896E-05	0.5812210E-02
0.7513364E+02	0.1625069E+00	0.2167592E-02
0.7598312E+02	0.1247403E+01	0.1669086E-01
0.7459329E+02	-0.3286282E+00	-0.4386275E-02

0.1873581E+02	-0.2032017E+00	-0.1072927E-01
0.3130701E-02	0.1219642E-03	0.4053670E-01
0.1304114E+02	-0.8657427E-01	-0.6594769E-02
0.5617166E-03	-0.1162646E-05	-0.2065533E-02
0.4564159E+01	-0.2508163E-03	-0.5495044E-04
0.6148461E-03	0.5793301E-04	0.1040252E+00
0.7565073E+02	0.6795967E+00	0.9064778E-02
0.7388667E+02	-0.8490402E+00	-0.1136057E-01
0.7421711E+02	-0.7048107E+00	-0.9407270E-02

0.1892240E+02	-0.1660957E-01	-0.8770033E-03
0.2914665E-02	-0.9407139E-04	-0.3126608E-01
0.1296554E+02	-0.1621835E+00	-0.1235428E-01
0.5410297E-03	-0.2184959E-04	-0.3881754E-01
0.4542958E+01	-0.2145243E-01	-0.4699934E-02
0.5543957E-03	-0.2517455E-05	-0.4520373E-02
0.7359953E+02	-0.1371604E+01	-0.1829510E-01

0.7480896E+02	0.7324677E-01	0.9800772E-03
0.7411285E+02	-0.8090740E+00	-0.1079890E-01

0.1909122E+02	0.1522118E+00	0.8036944E-02
0.3086908E-02	0.7817136E-04	0.2598146E-01
0.1317043E+02	0.4271631E-01	0.3253902E-02
0.5668069E-03	0.3927614E-05	0.6977721E-02
0.4567466E+01	0.3056049E-02	0.6695387E-03
0.5602133E-03	0.3300226E-05	0.5925926E-02
0.7543095E+02	0.4598167E+00	0.6133250E-02
0.7556896E+02	0.8332489E+00	0.1114927E-01
0.7437079E+02	-0.5511318E+00	-0.7356082E-02

0.1903901E+02	0.1000019E+00	0.5280208E-02
0.2827295E-02	-0.1814416E-03	-0.6030490E-01
0.1292475E+02	-0.2029683E+00	-0.1546105E-01
0.5491955E-03	-0.1368381E-04	-0.2431039E-01
0.4557155E+01	-0.7255077E-02	-0.1589488E-02
0.5351895E-03	-0.2172360E-04	-0.3900716E-01
0.7523461E+02	0.2634743E+00	0.3514343E-02
0.7374108E+02	-0.9946320E+00	-0.1330866E-01
0.7533955E+02	0.4176334E+00	0.5574249E-02

0.1912835E+02	0.1893383E+00	0.9997266E-02
0.3011499E-02	0.2762408E-05	0.9181287E-03
0.1328353E+02	0.1558078E+00	0.1186861E-01
0.5721714E-03	0.9292091E-05	0.1650814E-01
0.4616206E+01	0.5179596E-01	0.1134779E-01
0.5142334E-03	-0.4267969E-04	-0.7663618E-01
0.7541606E+02	0.4449242E+00	0.5934606E-02
0.7563942E+02	0.9037064E+00	0.1209203E-01
0.7485645E+02	-0.6547508E-01	-0.8739109E-03

0.1877086E+02	-0.1681446E+00	-0.8878216E-02
0.3063721E-02	0.5498446E-04	0.1827493E-01
0.1300301E+02	-0.1247041E+00	-0.9499295E-02
0.5422322E-03	-0.2064708E-04	-0.3668118E-01
0.4595560E+01	0.3114986E-01	0.6824510E-02
0.5969215E-03	0.4000842E-04	0.7183961E-01
0.7358849E+02	-0.1382644E+01	-0.1844235E-01
0.7349523E+02	-0.1240482E+01	-0.1659824E-01
0.7582794E+02	0.9060215E+00	0.1209288E-01

0.1883243E+02	-0.1065830E+00	-0.5627698E-02
0.3370426E-02	0.3616892E-03	0.1202130E+00
0.1314768E+02	0.1995974E-01	0.1520427E-02
0.5779678E-03	0.1508858E-04	0.2680608E-01
0.4633887E+01	0.6947708E-01	0.1522148E-01
0.5900477E-03	0.3313462E-04	0.5949693E-01
0.7588104E+02	0.9099052E+00	0.1213674E-01
0.7510214E+02	0.3664291E+00	0.4902999E-02
0.7482066E+02	-0.1012646E+00	-0.1351601E-02

BASE POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1682850E+02	0.1682849E+02
0.1698782E+02	0.1698780E+02
0.1266969E+02	0.1266971E+02
0.1633969E+02	0.1633964E+02
0.1564813E+02	0.1564810E+02
0.1687191E+02	0.1687192E+02
0.1870907E+02	0.1870903E+02
0.1539802E+02	0.1539803E+02
0.1475299E+02	0.1475304E+02
0.1334547E+02	0.1334545E+02
0.1465965E+02	0.1465965E+02
0.1664457E+02	0.1664453E+02
0.1841741E+02	0.1841751E+02
0.1715000E+02	0.1715001E+02
0.1715213E+02	0.1715209E+02
0.1613195E+02	0.1613204E+02
0.1566547E+02	0.1566544E+02
0.1383849E+02	0.1383851E+02
0.1729079E+02	0.1729080E+02
0.1586372E+02	0.1586370E+02

RANDOM POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1722661E+02	0.1733571E+02
0.1968150E+02	0.2020018E+02
0.1609655E+02	0.1626776E+02
0.1999169E+02	0.1964002E+02
0.1672752E+02	0.1674227E+02

FREQUENCY = 18.0237 RADIANS
OUTPUT NUMBER = 3
NUMBER OF SAMPLES USED = 35SAMPLES

THE POLYNOMIAL COEFFICIENTS ARE:

```
 0.4798648155406875E+01
 0.8910677456934270E+01
 0.1231087339110229E+02
-0.1575539982052280E+02
-0.1041142467914788E+02
-0.1262685361587043E+02
-0.3027759602050323E+02
-0.2329526168704561E+01
 0.2379257879962846E+02
 0.1982233934096393E+02
-0.6554117474383441E+01
-0.1327292939479843E+01
 0.1266172613770085E+02
 0.1651617199458171E+01
-0.7054588483469510E+00
-0.1767041938363605E+01
-0.1835916132630369E+02
-0.9875541649804882E+00
 0.9193146910089351E+00
 0.5853577228965828E+01
 0.9450512434313062E+01
-0.1468404432212452E+02
-0.9640820242940047E+01
 0.8458548709523317E+01
 0.2813295698881426E+01
 0.1500259388688442E+01
 0.3089813841890811E+00
-0.1234893731623193E+01
 0.5856902841786272E+00
-0.1168826811532510E+02
 0.3261555772538975E+01
 0.6641925843274787E+01
-0.7889430363108410E+01
-0.5182644470158194E+01
 0.9428945077363995E+01
 0.9742550726282601E+00
-0.5385458403532813E+01
-0.7600251827892812E+01
-0.1692992549304748E+01
-0.1311496473338777E+02
 0.2280961458030975E+02
 0.6064553055373955E+01
 0.3775973443553058E+01
-0.7454631889308468E+01
-0.1505607463894615E+02
-0.2394898207412343E+01
 0.8446436296499262E+01
```

0.4279854968237476E+01
-0.1596978276289094E+01
0.7349352963085874E+01
0.1030447906013310E+01
0.1652596889992320E+02
0.5003683235267746E+01
0.1245085637588656E+02
-0.1516304626005585E+02

ISCAS4 OUTPUT#3 FREQUENCY= 18.0327 RAD PRESENT NUMBER OF SAMPLES USED = 35
PREVIOUS NUMBER OF SAMPLES USES = 0

NOMINAL POINT	AVERAGE POINT
0.1903000E+02	0.1897169E+02
0.3000000E-02	0.3016148E-02
0.1311000E+02	0.1310996E+02
0.5600000E-03	0.5599974E-03
0.4570000E+01	0.4568696E+01
0.5600000E-03	0.5624649E-03
0.7500000E+02	0.7483724E+02
0.7500000E+02	0.7481617E+02
0.7500000E+02	0.7494780E+02

BASE POINTS	DEVIATION FROM AVG	RELATIVE DEVIATION
0.1863364E+02	-0.3380487E+00	-0.1781858E-01
0.2986408E-02	-0.2974014E-04	-0.9860305E-02
0.1314856E+02	0.3860190E-01	0.2944472E-02
0.5613285E-03	0.1331132E-05	0.2377032E-02
0.4578350E+01	0.9653405E-02	0.2112945E-02
0.5801924E-03	0.1772743E-04	0.3151740E-01
0.7487767E+02	0.4042642E-01	0.5401912E-03
0.7545962E+02	0.6434450E+00	0.8600347E-02
0.7493042E+02	-0.1738434E-01	-0.2319526E-03
0.1876527E+02	-0.2064187E+00	-0.1088035E-01
0.3045191E-02	0.2904284E-04	0.9629116E-02
0.1335950E+02	0.2495423E+00	0.1903456E-01
0.5422755E-03	-0.1772181E-04	-0.3164623E-01
0.4614855E+01	0.4615910E-01	0.1010334E-01
0.5708377E-03	0.8372761E-05	0.1488584E-01
0.7475592E+02	-0.8132346E-01	-0.1086671E-02
0.7476848E+02	-0.4769418E-01	-0.6374849E-03
0.7605817E+02	0.1110370E+01	0.1481524E-01
0.1898621E+02	0.1451520E-01	0.7650975E-03
0.2965448E-02	-0.5070049E-04	-0.1680968E-01
0.1320794E+02	0.9798148E-01	0.7473820E-02
0.6073826E-03	0.4738527E-04	0.8461695E-01
0.4604646E+01	0.3594954E-01	0.7868666E-02
0.5170516E-03	-0.4541339E-04	-0.8073994E-01
0.7479639E+02	-0.4084952E-01	-0.5458448E-03
0.7561869E+02	0.8025179E+00	0.1072653E-01
0.7445711E+02	-0.4906967E+00	-0.6547179E-02
0.1897599E+02	0.4295622E-02	0.2264227E-03
0.2789617E-02	-0.2265316E-03	-0.7510626E-01
0.1340513E+02	0.2951680E+00	0.2251479E-01
0.5367783E-03	-0.2321906E-04	-0.4146279E-01
0.4549951E+01	-0.1874559E-01	-0.4103049E-02
0.5311570E-03	-0.3130798E-04	-0.5566211E-01
0.7613561E+02	0.1298369E+01	0.1734923E-01
0.7448629E+02	-0.3298826E+00	-0.4409242E-02
0.7386514E+02	-0.1082669E+01	-0.1444564E-01

0.1899977E+02	0.2807454E-01	0.1479812E-02
0.2795901E-02	-0.2202478E-03	-0.7302285E-01
0.1313946E+02	0.2950003E-01	0.2250200E-02
0.5790695E-03	0.1907213E-04	0.3405753E-01
0.4533521E+01	-0.3517549E-01	-0.7699240E-02
0.5603894E-03	-0.2075514E-05	-0.3690032E-02
0.7412605E+02	-0.7111986E+00	-0.9503271E-02
0.7431868E+02	-0.4974928E+00	-0.6649535E-02
0.7685626E+02	0.1908458E+01	0.2546383E-01

0.1845476E+02	-0.5169313E+00	-0.2724750E-01
0.2803926E-02	-0.2122226E-03	-0.7036211E-01
0.1313054E+02	0.2057650E-01	0.1569532E-02
0.5609239E-03	0.9265304E-06	0.1654526E-02
0.4475001E+01	-0.9369485E-01	-0.2050801E-01
0.5596126E-03	-0.2852295E-05	-0.5071063E-02
0.7562456E+02	0.7873136E+00	0.1052035E-01
0.7467970E+02	-0.1364774E+00	-0.1824170E-02
0.7571716E+02	0.7693512E+00	0.1026516E-01

0.1895757E+02	-0.1412174E-01	-0.7443583E-03
0.2950601E-02	-0.6554740E-04	-0.2173215E-01
0.1290614E+02	-0.2038192E+00	-0.1554690E-01
0.5032141E-03	-0.5678322E-04	-0.1013991E+00
0.4580147E+01	0.1145060E-01	0.2506318E-02
0.5481148E-03	-0.1435017E-04	-0.2551301E-01
0.7544382E+02	0.6065809E+00	0.8105335E-02
0.7464386E+02	-0.1723127E+00	-0.2303148E-02
0.7480563E+02	-0.1421783E+00	-0.1897031E-02

0.1907677E+02	0.1050818E+00	0.5538875E-02
0.3135680E-02	0.1195311E-03	0.3963037E-01
0.1311959E+02	0.9627369E-02	0.7343554E-03
0.5856979E-03	0.2570058E-04	0.4589412E-01
0.4524822E+01	-0.4387443E-01	-0.9603271E-02
0.5658669E-03	0.3402002E-05	0.6048380E-02
0.7372791E+02	-0.1109339E+01	-0.1482335E-01
0.7350036E+02	-0.1315814E+01	-0.1758729E-01
0.7546471E+02	0.5169098E+00	0.6896930E-02

0.1904937E+02	0.7767323E-01	0.4094164E-02
0.3017769E-02	0.1620282E-05	0.5372023E-03
0.1308291E+02	-0.2705190E-01	-0.2063462E-02
0.5659783E-03	0.5980993E-05	0.1068039E-01
0.4599957E+01	0.3126033E-01	0.6842286E-02
0.5367087E-03	-0.2575619E-04	-0.4579165E-01
0.7503989E+02	0.2026426E+00	0.2707777E-02
0.7461190E+02	-0.2042722E+00	-0.2730322E-02
0.7413230E+02	-0.8155029E+00	-0.1088094E-01

0.1857218E+02	-0.3995092E+00	-0.2105817E-01
0.3166860E-02	0.1507118E-03	0.4996828E-01
0.1296198E+02	-0.1479749E+00	-0.1128722E-01

0.6142163E-03	0.5421896E-04	0.9682003E-01
0.4590923E+01	0.2222665E-01	0.4864987E-02
0.5450331E-03	-0.1743180E-04	-0.3099180E-01
0.7537366E+02	0.5364210E+00	0.7167835E-02
0.7495595E+02	0.1397753E+00	0.1868250E-02
0.7527285E+02	0.3250458E+00	0.4336962E-02

0.1907744E+02	0.1057475E+00	0.5573962E-02
0.3009796E-02	-0.6352538E-05	-0.2106176E-02
0.1320701E+02	0.9704974E-01	0.7402749E-02
0.5940024E-03	0.3400501E-04	0.6072352E-01
0.4557609E+01	-0.1108710E-01	-0.2426755E-02
0.5506821E-03	-0.1178281E-04	-0.2094852E-01
0.7534995E+02	0.5127088E+00	0.6850985E-02
0.7550769E+02	0.6915179E+00	0.9242893E-02
0.7478685E+02	-0.1609543E+00	-0.2147552E-02

0.1932299E+02	0.3512957E+00	0.1851684E-01
0.2993234E-02	-0.2291495E-04	-0.7597420E-02
0.1309305E+02	-0.1691053E-01	-0.1289899E-02
0.5494590E-03	-0.1053834E-04	-0.1881856E-01
0.4506594E+01	-0.6210200E-01	-0.1359294E-01
0.5466187E-03	-0.1584628E-04	-0.2817292E-01
0.7423624E+02	-0.6010073E+00	-0.8030858E-02
0.7393747E+02	-0.8787031E+00	-0.1174483E-01
0.7395318E+02	-0.9946259E+00	-0.1327091E-01

0.1938814E+02	0.4164451E+00	0.2195086E-01
0.3109088E-02	0.9293949E-04	0.3081396E-01
0.1325648E+02	0.1465235E+00	0.1117650E-01
0.5461388E-03	-0.1385856E-04	-0.2474755E-01
0.4494438E+01	-0.7425849E-01	-0.1625376E-01
0.5601500E-03	-0.2314922E-05	-0.4115673E-02
0.7513364E+02	0.2964002E+00	0.3960598E-02
0.7598312E+02	0.1166944E+01	0.1559748E-01
0.7459329E+02	-0.3545120E+00	-0.4730119E-02

0.1873581E+02	-0.2358853E+00	-0.1243354E-01
0.3130701E-02	0.1145525E-03	0.3797971E-01
0.1304114E+02	-0.6881520E-01	-0.5249078E-02
0.5617166E-03	0.1719261E-05	0.3070123E-02
0.4564159E+01	-0.4536792E-02	-0.9930168E-03
0.6148461E-03	0.5238119E-04	0.9312792E-01
0.7565073E+02	0.8134901E+00	0.1087012E-01
0.7388667E+02	-0.9294996E+00	-0.1242378E-01
0.7421711E+02	-0.7306946E+00	-0.9749379E-02

0.1892240E+02	-0.4929325E-01	-0.2598252E-02
0.2914665E-02	-0.1014832E-03	-0.3364661E-01
0.1296554E+02	-0.1444244E+00	-0.1101639E-01
0.5410297E-03	-0.1896768E-04	-0.3387102E-01
0.4542958E+01	-0.2573840E-01	-0.5633643E-02
0.5543957E-03	-0.8069273E-05	-0.1434627E-01
0.7359953E+02	-0.1237711E+01	-0.1653870E-01

0.7480896E+02	-0.7212612E-02	-0.9640445E-04
0.7411285E+02	-0.8349579E+00	-0.1114052E-01
0.1909122E+02	0.1195281E+00	0.6300339E-02
0.3086908E-02	0.7075957E-04	0.2346024E-01
0.1317043E+02	0.6047538E-01	0.4612934E-02
0.5668069E-03	0.6809521E-05	0.1215992E-01
0.4567466E+01	-0.1229927E-02	-0.2692073E-03
0.5602133E-03	-0.2251592E-05	-0.4003080E-02
0.7543095E+02	0.5937101E+00	0.7933351E-02
0.7556896E+02	0.7527895E+00	0.1006186E-01
0.7437079E+02	-0.5770157E+00	-0.7698900E-02
0.1903901E+02	0.6731824E-01	0.3548352E-02
0.2827295E-02	-0.1888533E-03	-0.6261408E-01
0.1292475E+02	-0.1852092E+00	-0.1412737E-01
0.5491955E-03	-0.1080191E-04	-0.1928921E-01
0.4557155E+01	-0.1154105E-01	-0.2526115E-02
0.5351895E-03	-0.2727541E-04	-0.4849265E-01
0.7523461E+02	0.3973676E+00	0.5309758E-02
0.7374108E+02	-0.1075091E+01	-0.1436977E-01
0.7533955E+02	0.3917496E+00	0.5226965E-02
0.1912835E+02	0.1566546E+00	0.8257283E-02
0.3011499E-02	-0.4649382E-05	-0.1541496E-02
0.1328353E+02	0.1735668E+00	0.1323931E-01
0.5721714E-03	0.1217400E-04	0.2173938E-01
0.4616206E+01	0.4750998E-01	0.1039902E-01
0.5142334E-03	-0.4823151E-04	-0.8575025E-01
0.7541606E+02	0.5788175E+00	0.7734351E-02
0.7563942E+02	0.8232470E+00	0.1100360E-01
0.7485645E+02	-0.9135895E-01	-0.1218968E-02
0.1877086E+02	-0.2008283E+00	-0.1058568E-01
0.3063721E-02	0.4757267E-04	0.1577266E-01
0.1300301E+02	-0.1069450E+00	-0.8157539E-02
0.5422322E-03	-0.1776517E-04	-0.3172367E-01
0.4595560E+01	0.2686389E-01	0.5879990E-02
0.5969215E-03	0.3445660E-04	0.6126000E-01
0.7358849E+02	-0.1248751E+01	-0.1668622E-01
0.7349523E+02	-0.1320941E+01	-0.1765582E-01
0.7582794E+02	0.8801376E+00	0.1174334E-01
0.1883243E+02	-0.1392667E+00	-0.7340763E-02
0.3370426E-02	0.3542774E-03	0.1174602E+00
0.1314768E+02	0.3771880E-01	0.2877110E-02
0.5779678E-03	0.1797049E-04	0.3209031E-01
0.4633887E+01	0.6519111E-01	0.1426908E-01
0.5900477E-03	0.2758280E-04	0.4903915E-01
0.7588104E+02	0.1043799E+01	0.1394758E-01
0.7510214E+02	0.2859698E+00	0.3822299E-02
0.7482066E+02	-0.1271484E+00	-0.1696493E-02
0.1865289E+02	-0.3188073E+00	-0.1680437E-01

0.2832858E-02	-0.1832901E-03	-0.6076959E-01
0.1313704E+02	0.2707579E-01	0.2065284E-02
0.5450626E-03	-0.1493471E-04	-0.2666925E-01
0.4575273E+01	0.6576851E-02	0.1439547E-02
0.5673754E-03	0.4910511E-05	0.8730342E-02
0.7541489E+02	0.5776426E+00	0.7718651E-02
0.7569989E+02	0.8837176E+00	0.1181185E-01
0.7511031E+02	0.1625015E+00	0.2168196E-02
0.1905310E+02	0.8140400E-01	0.4290814E-02
0.3219306E-02	0.2031576E-03	0.6735662E-01
0.1315493E+02	0.4496863E-01	0.3430112E-02
0.5565780E-03	-0.3419370E-05	-0.6106047E-02
0.4433434E+01	-0.1352622E+00	-0.2960630E-01
0.6050374E-03	0.4257250E-04	0.7568916E-01
0.7531792E+02	0.4806806E+00	0.6423014E-02
0.7494328E+02	0.1271029E+00	0.1698869E-02
0.7575062E+02	0.8028137E+00	0.1071164E-01
0.1895639E+02	-0.1530239E-01	-0.8065904E-03
0.3083762E-02	0.6761403E-04	0.2241734E-01
0.1295933E+02	-0.1506300E+00	-0.1148974E-01
0.5489024E-03	-0.1109492E-04	-0.1981246E-01
0.4644527E+01	0.7583125E-01	0.1659801E-01
0.5654952E-03	0.3030229E-05	0.5387410E-02
0.7610689E+02	0.1269644E+01	0.1696540E-01
0.7477065E+02	-0.4552743E-01	-0.6085239E-03
0.7600219E+02	0.1054385E+01	0.1406826E-01
0.1868895E+02	-0.2827413E+00	-0.1490332E-01
0.3183575E-02	0.1674262E-03	0.5550993E-01
0.1279361E+02	-0.3163538E+00	-0.2413080E-01
0.5116333E-03	-0.4836407E-04	-0.8636481E-01
0.4528373E+01	-0.4032342E-01	-0.8826024E-02
0.5753827E-03	0.1291779E-04	0.2296639E-01
0.7523125E+02	0.3940107E+00	0.5264901E-02
0.7470874E+02	-0.1074323E+00	-0.1435951E-02
0.7500664E+02	0.5883331E-01	0.7849905E-03
0.1916173E+02	0.1900351E+00	0.1001677E-01
0.3038972E-02	0.2282370E-04	0.7567168E-02
0.1322736E+02	0.1174002E+00	0.8955039E-02
0.5575734E-03	-0.2423961E-05	-0.4328522E-02
0.4565520E+01	-0.3175899E-02	-0.6951434E-03
0.5654028E-03	0.2937854E-05	0.5223177E-02
0.7462897E+02	-0.2082766E+00	-0.2783061E-02
0.7533808E+02	0.5219088E+00	0.6975882E-02
0.7449386E+02	-0.4539459E+00	-0.6056828E-02
0.1895930E+02	-0.1238796E-01	-0.6529706E-03
0.2886493E-02	-0.1296559E-03	-0.4298725E-01
0.1297495E+02	-0.1350078E+00	-0.1029811E-01
0.5939502E-03	0.3395280E-04	0.6063029E-01
0.4650370E+01	0.8167394E-01	0.1787686E-01

0.5814381E-03	0.1897313E-04	0.3373212E-01
0.7349292E+02	-0.1344324E+01	-0.1796330E-01
0.7415867E+02	-0.6575041E+00	-0.8788261E-02
0.7516232E+02	0.2145187E+00	0.2862242E-02
0.1915242E+02	0.1807254E+00	0.9526054E-02
0.2970994E-02	-0.4515493E-04	-0.1497106E-01
0.1326418E+02	0.1542158E+00	0.1176326E-01
0.5579768E-03	-0.2020582E-05	-0.3608199E-02
0.4629999E+01	0.6130250E-01	0.1341794E-01
0.5355765E-03	-0.2688845E-04	-0.4780467E-01
0.7261052E+02	-0.2226724E+01	-0.2975423E-01
0.7456401E+02	-0.2521620E+00	-0.3370420E-02
0.7534367E+02	0.3958618E+00	0.5281833E-02
0.1880671E+02	-0.1649873E+00	-0.8696498E-02
0.3198731E-02	0.1825825E-03	0.6053500E-01
0.1306350E+02	-0.4645822E-01	-0.3543735E-02
0.4794564E-03	-0.8054094E-04	-0.1438238E+00
0.4602175E+01	0.3347905E-01	0.7327922E-02
0.5626490E-03	0.1840493E-06	0.3272191E-03
0.7479528E+02	-0.4196341E-01	-0.5607290E-03
0.7502915E+02	0.2129793E+00	0.2846702E-02
0.7511514E+02	0.1673386E+00	0.2232735E-02
0.1893562E+02	-0.3606960E-01	-0.1901232E-02
0.2930391E-02	-0.8575733E-04	-0.2843273E-01
0.1296723E+02	-0.1427269E+00	-0.1088690E-01
0.6027853E-03	0.4278797E-04	0.7640744E-01
0.4555765E+01	-0.1293151E-01	-0.2830460E-02
0.5873390E-03	0.2487405E-04	0.4422329E-01
0.7521722E+02	0.3799726E+00	0.5077320E-02
0.7426526E+02	-0.5509138E+00	-0.7363565E-02
0.7491773E+02	-0.3007202E-01	-0.4012395E-03
0.1902264E+02	0.5094746E-01	0.2685446E-02
0.3007584E-02	-0.8564895E-05	-0.2839679E-02
0.1320251E+02	0.9254935E-01	0.7059469E-02
0.5925257E-03	0.3252834E-04	0.5808660E-01
0.4561439E+01	-0.7257625E-02	-0.1588555E-02
0.5714319E-03	0.8967003E-05	0.1594233E-01
0.7411697E+02	-0.7202776E+00	-0.9624588E-02
0.7560741E+02	0.7912340E+00	0.1057571E-01
0.7533031E+02	0.3825104E+00	0.5103690E-02
0.1897257E+02	0.8776528E-03	0.4626118E-04
0.2856227E-02	-0.1599211E-03	-0.5302163E-01
0.1313674E+02	0.2678111E-01	0.2042806E-02
0.6106051E-03	0.5060776E-04	0.9037142E-01
0.4576960E+01	0.8263424E-02	0.1808705E-02
0.5732005E-03	0.1073553E-04	0.1908657E-01
0.7371529E+02	-0.1121958E+01	-0.1499197E-01
0.7414867E+02	-0.6674986E+00	-0.8921849E-02
0.7476842E+02	-0.1793793E+00	-0.2393389E-02

0.1893291E+02	-0.3878375E-01	-0.2044296E-02
0.3105535E-02	0.8938672E-04	0.2963605E-01
0.1322522E+02	0.1152563E+00	0.8791510E-02
0.5069921E-03	-0.5300525E-04	-0.9465269E-01
0.4624336E+01	0.5564006E-01	0.1217854E-01
0.5089321E-03	-0.5353283E-04	-0.9517541E-01
0.7451653E+02	-0.3207110E+00	-0.4285446E-02
0.7449583E+02	-0.3203382E+00	-0.4281671E-02
0.7427850E+02	-0.6693008E+00	-0.8930226E-02
0.1957300E+02	0.6013091E+00	0.3169507E-01
0.2885086E-02	-0.1310620E-03	-0.4345343E-01
0.1305960E+02	-0.5035588E-01	-0.3841040E-02
0.5250146E-03	-0.3498277E-04	-0.6246952E-01
0.4567868E+01	-0.8284296E-03	-0.1813274E-03
0.6027611E-03	0.4029617E-04	0.7164210E-01
0.7457903E+02	-0.2582186E+00	-0.3450402E-02
0.7459381E+02	-0.2223615E+00	-0.2972105E-02
0.7430959E+02	-0.6382111E+00	-0.8515407E-02
0.1933319E+02	0.3614962E+00	0.1905451E-01
0.3148091E-02	0.1319430E-03	0.4374554E-01
0.1311969E+02	0.9732274E-02	0.7423573E-03
0.5694864E-03	0.9489052E-05	0.1694482E-01
0.4484386E+01	-0.8431022E-01	-0.1845389E-01
0.5592633E-03	-0.3201658E-05	-0.5692190E-02
0.7531438E+02	0.4771406E+00	0.6375710E-02
0.7513071E+02	0.3145418E+00	0.4204196E-02
0.7491142E+02	-0.3638916E-01	-0.4855267E-03
0.1902765E+02	0.5595807E-01	0.2949556E-02
0.3042857E-02	0.2670825E-04	0.8855084E-02
0.1300833E+02	-0.1016283E+00	-0.7751990E-02
0.5837799E-03	0.2378253E-04	0.4246900E-01
0.4615738E+01	0.4704173E-01	0.1029653E-01
0.5867254E-03	0.2426048E-04	0.4313244E-01
0.7482274E+02	-0.1450522E-01	-0.1938236E-03
0.7639761E+02	0.1581441E+01	0.2113769E-01
0.7423402E+02	-0.7137878E+00	-0.9523799E-02

BASE POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1682853E+02	0.1682849E+02
0.1698784E+02	0.1698780E+02
0.1266973E+02	0.1266971E+02
0.1633961E+02	0.1633964E+02
0.1564819E+02	0.1564810E+02
0.1687206E+02	0.1687192E+02
0.1870901E+02	0.1870903E+02
0.1539800E+02	0.1539803E+02
0.1475303E+02	0.1475304E+02

0.1334539E+02	0.1334545E+02
0.1465986E+02	0.1465965E+02
0.1664463E+02	0.1664453E+02
0.1841827E+02	0.1841751E+02
0.1715014E+02	0.1715001E+02
0.1715207E+02	0.1715209E+02
0.1613165E+02	0.1613204E+02
0.1566541E+02	0.1566544E+02
0.1383845E+02	0.1383851E+02
0.1729069E+02	0.1729080E+02
0.1586472E+02	0.1586370E+02
0.1733572E+02	0.1733571E+02
0.2020103E+02	0.2020018E+02
0.1626797E+02	0.1626776E+02
0.1964070E+02	0.1964002E+02
0.1674197E+02	0.1674227E+02
0.1473486E+02	0.1473490E+02
0.1493870E+02	0.1493928E+02
0.1946834E+02	0.1946895E+02
0.1533845E+02	0.1533805E+02
0.1553659E+02	0.1553647E+02
0.1439997E+02	0.1440000E+02
0.1660705E+02	0.1660699E+02
0.1846909E+02	0.1846885E+02
0.1674427E+02	0.1674447E+02
0.1604411E+02	0.1604411E+02

RANDOM POINTS

PREDICTED RESPONSE	ACTUAL RESPONSE
0.1653985E+02	0.1709623E+02
0.1559449E+02	0.1652122E+02
0.1594141E+02	0.1577145E+02
0.1703423E+02	0.1708289E+02
0.1696043E+02	0.1763427E+02