An algorithm for minimal partial realization with inaccurate data.

Marc Van Barel
Adhemar Bultheel

Report TW 80, July 1986

Department of Computer Science, K.U.Leuven

ABSTRACT

In this paper an algorithm is presented to compute a minimal partial realization (mpr) of the form $C(z) \cdot A(z)^{-1}$ given N $(m \times n)$-Markov parameters which are not exactly known ($C(z)$ and $A(z)$ are polynomial matrices, respectively $(m \times n)$ and $(n \times n)$). The algorithm is based on the singular value decomposition of certain block-Hankel matrices containing the Markov parameters. Insight is given in the errors between the given Markov parameters and those of the computed mpr $C(z) \cdot A(z)^{-1}$. It is shown that the method is a special case of the total linear least squares technique for identification and parameter estimation and that it can also be used for model reduction. Finally a numerical example is presented.

1. Introduction

The notation used will be for discrete stable linear systems but everything can also be applied to continuous linear systems.

The problem considered here is very important in linear systems theory. Given a finite sequence of N matrices $H_i \in \mathbb{R}^{m\times n}$, $i = 1, \ldots, N$, called Markov parameters, find a $(m\times n)$ polynomial matrix $C(z)$ and a nonsingular $(n\times n)$ polynomial matrix $A(z)$ such that :

$$C(z)\cdot A(z)^{-1} = H_1 z^{-1} + H_2 z^{-2} + \cdots + H_N z^{-N} + O_-(z^{-N-1}) \qquad (1)$$

with $C(z)$ and $A(z)$ right coprime and the degree of the determinant of $A(z)$ (deg det $A(z)$) as small as possible. (An analoguous definition can be given for the left polynomial matrix fraction representation $B(z)^{-1}\cdot D(z)$). $C(z)\cdot A(z)^{-1}$ is the transfer function of a discrete linear system. The coefficients of the power series of $C(z)\cdot A(z)^{-1}$ around $z = \infty$ form the impulse response of the system. The deg det $A(z)$ is equal to the minimal number of states to realize the system. So (1) means that we are looking for a transfer function representation of a system with a minimal number of states such that the first N impulse response coefficients are H_i, $i = 1, \ldots, N$. These Markov parameters H_i are supposed to be measured with a certain relative precision. (See [de Jong] and [Gragg, Lindquist] for the single input-single output case with absolute errors on the Markov parameters).

In Section 2 the problem is transformed and solved under the assumption that the Markov parameters are exactly known. Also the connection is given with the total linear least squares technique used in [Van Huffel, Vandewalle]. Section 3 contains an algorithm to solve the inexact problem. In Section 4 insight is given into the errors between the Markov parameters of the computed system and the given Markov parameters. Section 5 shows a numerical example. Section 6 gives some notes and conclusions on the effectiveness, the numerical stability and the adaptation to the model reduction problem of the algorithm of section 3.

2. Solution of the problem when the Markov parameters are exactly known

We denote the j-th column of $A(z)$ by $a_j(z) = a_{0,j} + a_{1,j} z + \cdots + a_{d_j,j} z^{d_j}$ with $a_{i,j} \in \mathbb{R}^n$ and $a_{d_j,j} \neq 0$. d_j is called the degree of the polynomial vector $a_j(z)$ (we can define the degree of the zero polynomial vector as $-\infty$). The matrix $HD_A = [a_{d_1,1} \;\; a_{d_2,2} \;\cdots\; a_{d_n,n}]$ is called the highest degree coefficient matrix of the polynomial matrix $A(z)$. When HD_A is nonsingular we say that $A(z)$ is column reduced. In this case deg det $A(z) = \sum_1^n d_j$. To transform the problem we need the definition of a unimodular matrix. A square polynomial matrix $U(z)$ is unimodular when det $U(z)$ is a nonzero constant. Every polynomial matrix $A(z)$ can be transformed into a column reduced polynomial matrix by right multiplication with a unimodular matrix : $A'(z) = A(z)\cdot U(z)$ and $HD_{A'}$ is nonsingular. This means that we can look for $C(z)\cdot A(z)^{-1}$ with $A(z)$ column reduced and $\sum_1^n d_j$ minimal satisfying (1).

The denominator columns $a_j(z)$, called solution vectors of level N given N Markov parameters, can be found as solutions of the following set of homogeneous linear equations with a block-Hankel coefficient matrix [Van Barel,Bultheel] :

$$\begin{bmatrix} H_1 & \cdots & H_{d+1} \\ \vdots & & \vdots \\ H_{N-d} & \cdots & H_N \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{or} \quad H_{N,d}\cdot a(z) = 0. \qquad (2)$$

Note that we use $a(z)$ and $[a_0^T \cdots a_d^T]^T$ as equivalent notations. The corresponding column of the numerator polynomial matrix can be computed as :

$$\begin{bmatrix} 0 & 0 & \cdots & H_1 \\ \vdots & \vdots & & \vdots \\ 0 & H_1 & \cdots & H_d \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} c_{d-1} \\ \vdots \\ c_0 \end{bmatrix} \quad \text{or } H'_{N,d} \cdot a(z) = c^*(z). \tag{3}$$

We can rewrite this as

$$\begin{bmatrix} H'_{N,d} & I \\ H_{N,d} & 0 \end{bmatrix} \cdot \begin{bmatrix} a(z) \\ -c^*(z) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Note the similarity between this problem for the single input- single output case and the parameter estimation problem [Van Huffel, Vandewalle]. I.e. given an input $u(z)$ and an output $y(z)$, find the parameters of a transfer function of degree d that matches the given data as close as possible. Their algorithm uses the total linear least squares technique but their article gives no insight about the errors between the given output $y(z)$ and the output $\tilde{y}(z)$ of the computed system when the same input $u(z)$ is applied. In our problem the input $u(z) = 1$ so the coefficients of the output power series $y(z)$ are the Markov parameters.
So (2) indicates that we have to study the structure of the nullspaces of the block-Hankel matrices $H_{N,d}$. We put the solution vectors of level p, $a_j^p(z)$, given p Markov parameters with $p = 1, \ldots, N$ in a two-dimensional table called the solution table.

level \ degree	0	1	2	⋯	
0	$a_1^0(z) \ldots a_n^0(z)$				
1		$a_1^1(z)$	⋯	$a_n^1(z)$	
2					
3					
⋮					
p			$a_1^p(z)$	⋯ $a_n^p(z)$	
⋮					
N				$a_1^N(z)$ ⋯	$a_n^N(z)$

Table 1: Solution table

A basis $BH_{p,d}$ of the nullspace of $H_{p,d}$ can be constructed as follows [Van Barel,Bultheel] :

$$BH_{p,d} = \bigcup_{\substack{\sigma=0,\ldots,d \\ i \leq p}} z^{d-\sigma} a_j^i(z).$$

σ is called the potential degree of $a_j^i(z)$ with $\sigma = d_j^i - i + p$ and d_j^i is the degree of the vector polynomial $a_j^i(z)$. This means that $BH_{p,d}$ is constructed with the solution vectors inside the triangle of table 2. We assume in the following that the value of N is big enough such that the solution vectors can be choosen to make the table look as table 3. This means that the solution vectors have reached a final value.
Note that the nested structure of these bases can be used to construct a recursive algorithm to

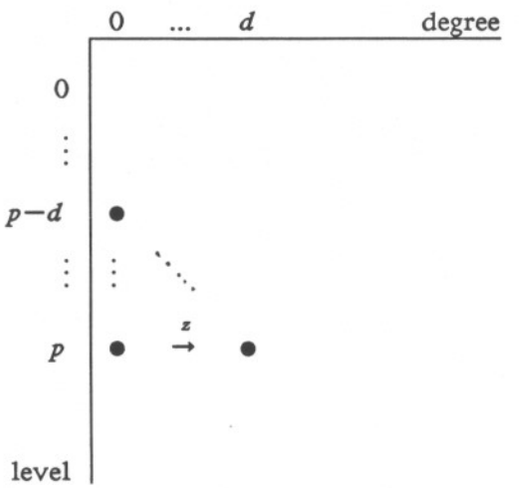


Table 2

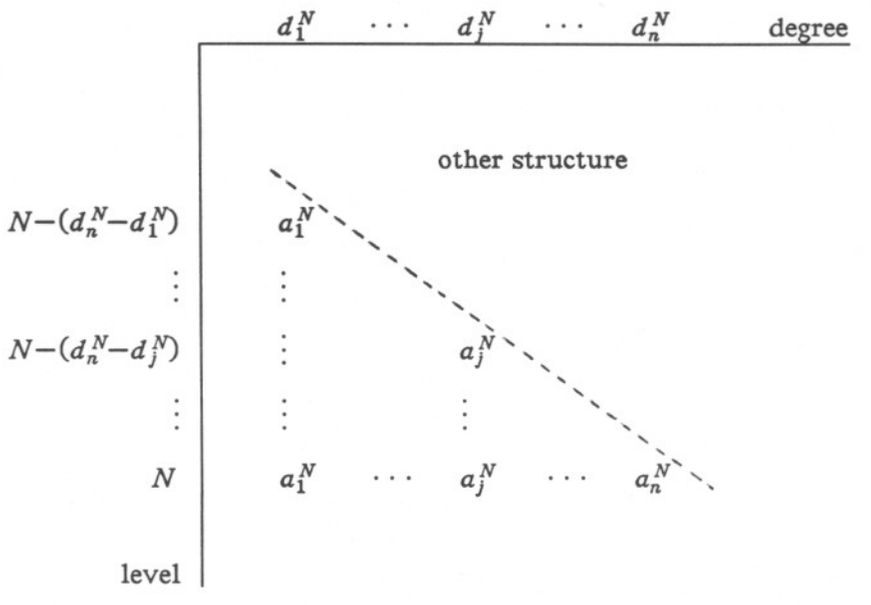


Table 3

compute the solution vectors ([Dickinson,Morf,Kailath]). This algorithm can not be changed into a robust one when the Markov parameters are inaccurate.
Here we propose a nonrecursive algorithm to compute the solution vectors of level N based on the singular value decomposition. Recall that every matrix $A \in \mathbb{R}^{p\times q}$ has a singular value decomposition (SVD) : $A = U \cdot \Sigma \cdot V^T$ with $U \in \mathbb{R}^{p\times p}$ and $V \in \mathbb{R}^{q\times q}$ orthogonal matrices and Σ a diagonal $(p\times q)$ matrix with diagonal elements (singular values) $\sigma_1 \geqslant \sigma_2 \geqslant \sigma_s > \sigma_{s+1} = \cdots = \sigma_{\min(p,q)} = 0$. The last $q-s$ columns of V form an orthonormal basis for the nullspace of A. The algorithm to compute the solution vectors is as follows :

1) Take $H_{N,0}$, $H_{N,1}$, $\cdots$ until the nullspace is different from $\{0\}$. Suppose this occurs for H_{N,d_1} :

$$H_{N,d_1} = \begin{bmatrix} U_1^{d_1} \\ U_2^{d_1} \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1^{d_1} & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1^{d_1 T} \\ V_2^{d_1 T} \end{bmatrix} \text{ with } \begin{cases} U_1^{d_1} \in \mathbb{R}^{m \times (N-d_1)m} \\ \Sigma_1^{d_1} \text{ nonsingular} \in \mathbb{R}^{s \times s} \end{cases}$$

Thus the nullspace of H_{N,d_1} has dimension $(d_1+1)n-s$ and therefore there are $(d_1+1)n-s$ solution vectors of degree d_1 [Van Barel,Bultheel]. We can take the columns of V_2 as these solution vectors.

2) Suppose we have already made a choice for the first k solution vectors of degree smaller than d ; if $k=n$ the algorithm ends. $H_{N,d-1}$ can be written as :

$$H_{N,d-1} = \begin{bmatrix} U_1^{d-1} \\ U_2^{d-1} \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1^{d-1} & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1^{d-1 T} \\ V_2^{d-1 T} \end{bmatrix} \text{ with } \begin{cases} U_1^{d-1} \in \mathbb{R}^{m \times (N-d+1)m} \\ \Sigma_1^{d-1} \text{ nonsingular} \in \mathbb{R}^{r \times r} \end{cases}$$

The number of columns of V_2^{d-1} is equal to the number of solution vectors in the triangle with vertices $(N,0)$, $(N,d-1)$, $(N-d+1,0)$ of the solution table. Suppose this number is l. Then we can write $H_{N,d}$ as follows :

$$H_{N,d} = \underbrace{\left[\begin{bmatrix} H_1 \\ \vdots \\ H_{N-d} \end{bmatrix} U_2^{d-1} \cdot \begin{bmatrix} \Sigma_1^{d-1} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right]}_{(4)} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} .$$

Suppose the SVD of (4) is

$$U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix} .$$

Then

$$H_{N,d} = \left[U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right] \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix}$$

$$= U' \cdot \begin{bmatrix} \Sigma_1' & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} V_1'^T & 0 \\ V_2'^T & 0 \\ 0 & I \end{bmatrix}}_{(5)} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} .$$

(5) is orthogonal, hence the singular value decomposition of $H_{N,d}$ can be written as :

$$H_{N,d} = U' \cdot \begin{bmatrix} \Sigma_1' & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \end{bmatrix} \\ 0 \qquad\qquad V_2^T \end{bmatrix}$$

or

$$H_{N,d} = \begin{bmatrix} U_1^d \\ U_2^d \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1^d & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1^{dT} \\ V_2^{dT} \end{bmatrix}$$

with

$$\begin{cases} U_1^d \text{ the first } m \text{ rows of } U' \\ U_2^d \text{ the last } (N-d-1)m \text{ rows of } U' \\ \Sigma_1^d = \Sigma_1' \\ V_1^{dT} = V_1'^T \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \end{bmatrix} \\ V_2^{dT} = \begin{bmatrix} V_2'^T \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \end{bmatrix} \\ 0 \qquad\qquad V_2^T \end{bmatrix} . \end{cases}$$

When the number of columns of

$$\begin{bmatrix} I & 0 \\ 0 & V_1 \end{bmatrix} \cdot [V_2']$$

is equal to $k+j$, there are j solution vectors of degree d (see table 3). Choose j columns of these $k+j$ columns such that the highest degree coefficient matrix of the solution vectors computed so far has full column rank.
Repeat 2) until there are n solution vectors. The numerator polynomial matrix can then be computed by using (3).

3. The solution of the inexact problem

Suppose now that the Markov parameters H_i, $i = 1, \ldots, N$ are not known exactly but are only given within a relative accuracy ϵ which we assume to be the same for all the parameters. We have to adapt the algorithm of section 2 because the relative errors on the Markov parameters will move certain singular values σ_i from zero to a value which is not zero but which is to be considered "numerically zero", given the errors on the Markov parameters.
Before we state the algorithm to solve the inexact problem we note the following property of the singular value decomposition : suppose the SVD of the $(p \times q)$ matrix A is : $A = U \cdot \Sigma \cdot V^T$ or $A = \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i^T$ with σ_i the i-th singular value and u_i, v_i the i-th column of U respectively V. Suppose $k < \min(m,n)$ then

$$\min_{\text{rank } \tilde{A} \leq k} \|A - \tilde{A}\| = \|\text{diag}(\sigma_{k+1}, \ldots, \sigma_{\min(m,n)})\|$$

with $\|\cdot\|$ an orthogonally invariant matrix norm.
A candidate for the matrix $\tilde{A}$ is the matrix : $\tilde{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. The absolute difference between the corresponding elements of A and $\tilde{A}$ will be of the order of magnitude of σ_{k+1}. Instead of having the same order of magnitude for the absolute difference between every element of A

and $\tilde{A}$ we want to have the same order of magnitude ϵ for the relative difference between the elements of A and $\tilde{A}$. Therefore the matrix A has to be scaled such that all its elements have the same order of magnitude and then the singular value decomposition can be applied to this scaled matrix. By using the inverse scaling we get a matrix $\tilde{A}$ of rank k approximating the matrix A with the same order of magnitude for the relative error between the corresponding elements.
This leads to the following algorithm for the inexact problem (with the same assumption concerning the structure of the solution vectors as in section 2)) :

1) First of all we search for solution vectors of degree 0 :

$$\begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix} \cdot [a_0] = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{or} \quad H_{N,0} \cdot a_0 = 0.$$

The nullspace of $H_{N,0}$ is constructed by SVD, but before we apply the SVD to $H_{N,0}$ we scale the matrix $H_{N,0}$ such that the elements have the same order of magnitude α. To this end we use the diagonal matrices D_1 and D_2 so that the matrix $D_1 H_{N,0} D_2$ is a scaled matrix. To compute D_1 and D_2 we use the following algorithm :

repeat 5 times

- compute the Euclidean norm of each row of $H_{N,0}$ and divide this row by the norm $\Rightarrow$ the length of this row is one
- do the same thing with each of the n columns of $H_{N,0}$.

Then we compute the SVD of this scaled matrix to determine the dimensions of the nullspace of $H_{N,0}$:

$$D_1 \cdot H_{N,0} \cdot D_2 = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} . \qquad (6)$$

Suppose the diagonal elements of Σ_2 are numerically zero, i.e. of the order of magnitude ϵ (the relative error on the data). Then we have the following solution vectors of degree 0 :

$$D_2 \cdot V_2.$$

If there are solution vectors of degree 0 we go to step 2). If not then we could construct $H_{N,1}$, scale it and compute its SVD and this procedure could be repeated until we get some singular values numerically equal to zero. If we don't want to loose the information we have already gathered about our system from the computed SVD, we shouldn't throw it away. So we use the computed SVD decomposition (6) in our search for the solution vectors of degree 1. Also numerical results were more accurate using this procedure.

2) So the method to compute the solution vectors of degree 1 when we know the SVD decomposition (6) is as follows :
Because $H_{N,1}$ is a block-Hankel matrix, it can be written as :

$$H_{N,1} = \left[\begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{N-1} \end{bmatrix} D_{1,2}^{-1} \cdot U_2 \cdot \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \cdot D_2^{-1} \right] .$$

$D_{1,2}$ is equal to D_1 without its first m rows and columns. Note that the matrix V_2^T could be empty.

$$H_{N,1} = D_{1,2}^{-1} \cdot \overbrace{\left[\underbrace{D_{1,2} \cdot \begin{bmatrix} H_1 \\ \vdots \\ H_{N-1} \end{bmatrix} \cdot D_2' \quad U_2 \cdot \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix}}_{(8)} U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix} \right]}^{(7)} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} \cdot \begin{bmatrix} D_2'^{-1} & 0 \\ 0 & D_2^{-1} \end{bmatrix}$$

D_2' is computed such that the columns of (8) have length one.
Note that the order of magnitude of the corresponding elements of D_2' and D_2 will be the same because the elements of the Markov parameter H_N will be small compared with the previous Markov parameters (we consider only discrete stable systems). When the order of magnitude of the elements of the Markov parameters are varying slowly for a specific input and output and when the scaling of the previous step can be done successfully then the scaling of this step will guarantee that the elements of (8) have the same order of magnitude α. When the previous assumption is not valid we should look for other more complicated scaling schemes. See also section 6.
We compute the SVD of (7) and so we can rewrite $H_{N,1}$ as :

$$H_{N,1} = D_{1,2}^{-1} \cdot \left[U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & \Sigma_2' \end{bmatrix} \cdot \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix} U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix} \right] \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} \cdot \begin{bmatrix} D_2'^{-1} & 0 \\ 0 & D_2^{-1} \end{bmatrix}$$

$$= D_{1,2}^{-1} \cdot \left[U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & \Sigma_2' \end{bmatrix} U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix} \right] \cdot \underbrace{\begin{bmatrix} V_1'^T & 0 \\ V_2'^T & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix}}_{(9)} \cdot \begin{bmatrix} D_2'^{-1} & 0 \\ 0 & D_2^{-1} \end{bmatrix} .$$

Note that (9) is an orthogonal matrix. The diagonal elements of Σ_2' are numerically zero. Suppose the number of these elements is $k+j$ where k is the number of solution vectors of degree 0. Then there are j solution vectors of degree 1.
We choose these solution vectors as column vectors of

$$\begin{bmatrix} D_2' & 0 \\ 0 & D_2 \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1 \end{bmatrix} \cdot V_2' \qquad (10)$$

such that the matrix of highest degree coefficients of all the solution vectors found so far has full column rank. To this end we use the QR decomposition with partial column pivoting. We construct a matrix HD whose first k columns are the highest degree coefficients (hdc) of the solution vectors of degree 0. These columns can not be pivoted. The following $k+j$ columns are the hdc of the solution vectors of degree 1. These columns can be pivoted. So we get :

$$HD = [\underbrace{HD_1}_{k \text{ columns}} \mid \underbrace{HD_2}_{k+j \text{ columns}}].$$

Afther the QR decomposition with partial column pivoting :

$$Q^T \cdot HD \cdot \begin{bmatrix} I & 0 \\ \underbrace{0}_{k} & \underbrace{J}_{k+j} \end{bmatrix} = \begin{bmatrix} R & \mid & X' & X'' \\ 0 & \mid & R' & Y'' \\ \underbrace{0}_{k} & \mid & \underbrace{0}_{j} & \underbrace{R''}_{k} \end{bmatrix}.$$

Suppose the $(k+i)$-th column of $HD \cdot \begin{bmatrix} I & 0 \\ 0 & J \end{bmatrix}$ is the $(k+j_i)$-th column of HD then the j solution vectors of degree 1 are the columns j_i, $i = 1, \ldots, j$ of (10). Note that we don't need to compute the whole QR decomposition. It is enough to compute the j_i. Note that we could use $\begin{bmatrix} R & \mid & X' \\ 0 & \mid & R' \end{bmatrix}$ and Q^T in the next step to compute the solution vectors of degree 2.

3) The same procedure can be repeated to obtain solution vectors of degree 2,3,... until we have n solution vectors.
Note that in each step D_1 is loosing its first m diagonal elements and D_2 is extended with n new first diagonal elements.

4. The correspondence between the given Markov parameters and these of the computed transfer function

For each solution vector $a_j(z)$ of degree d_j we can write :

$$\begin{bmatrix} & & & H_1 \\ & & & \vdots \\ & H_1 & \cdots & H_{d_j} \\ H_1 & H_2 & \cdots & H_{d_j+1} \\ \vdots & & & \vdots \\ H_{N-d_j} & H_{N-d_j+1} & \cdots & H_N \end{bmatrix} \cdot \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ \vdots \\ a_{d_j,j} \end{bmatrix} = \begin{bmatrix} c_{d_j-1,j} \\ \vdots \\ c_{0,j} \\ D_{1,d_j+1}^{-1} \cdot \sigma_j \cdot u_{j,d_j+1} \\ \vdots \\ D_{1,N}^{-1} \cdot \sigma_j \cdot u_{j,N} \end{bmatrix} \begin{matrix} \leftarrow z^{d_j-1} \\ \vdots \\ \leftarrow z^0 \\ \leftarrow z^{-1} \\ \vdots \\ \leftarrow z^{-N+d_j} \end{matrix} \qquad (11)$$

with

σ_j (numerically zero) the singular value corresponding to $a_j(z)$

$[u_{j,d_j+1}^T \cdots u_{j,N}^T]^T$ the corresponding left singular vector with each $u_{j,i} \in \mathbb{R}^m$

$$\begin{bmatrix} D_{1,1} & & \\ & \ddots & \\ & & D_{1,N} \end{bmatrix} = D_1.$$

Note that (11) is in fact the matrix notation of a power series multiplication. The solution vector can also be written as :

$$\begin{bmatrix} a_{0,j} \\ \vdots \\ a_{d_j,j} \end{bmatrix} = \begin{bmatrix} D_{2,h-d_j} \cdot v_{0,j} \\ \vdots \\ D_{2,h} \cdot v_{d_j,j} \end{bmatrix} \text{ where } \begin{bmatrix} D_{2,h-d_j} & & 0 \\ & \ddots & \\ 0 & & D_{2,h} \end{bmatrix} = D_2 \qquad (12)$$

is used to compute the solution vectors of degree d_j.

(11) together with (12) gives :

$$H(z^{-1}) - C(z)\cdot A(z)^{-1} = O_{-}(z^{-N-1}) +$$

$$\underbrace{(D_{1,d_1+1}^{-1}\cdot z^{-d_1-1}\cdot[\sigma_1 u_{1,d_1+1}\cdots\sigma_n u_{n,d_1+1}] + \cdots + D_{1,N}^{-1}\cdot z^{-N}\cdot[\sigma_1 u_{1,N}\cdots\sigma_n u_{n,N}])\cdot}_{(13)}$$

$$\underbrace{(D_{2,h}\cdot[v_{d_1,1}\cdots v_{d_n,n}] + \cdots + D_{2,0}\cdot[v_{d_1-h,1}\cdots v_{d_n-h,n}]\cdot z^{-h})^{-1}}_{(14)}$$

with

$$\begin{cases} H(z^{-1}) = H_1 z^{-1} + H_2 z^{-2} + \cdots + H_N z^{-N} \\ u_{j,i} = 0 \text{ when } i \leqslant d_j \\ v_{i,j} = 0 \text{ when } i < 0 \\ d_1 \leqslant d_2 \leqslant \cdots \leqslant d_n = h \end{cases}$$

The last factor (14) can also be written as :

$$(V_h + D_{2,h}^{-1}\cdot D_{2,h-1}\cdot V_{h-1} z^{-1} + \cdots + D_{2,h}^{-1}\cdot D_{2,0}\cdot V_0\cdot z^{-h})^{-1}\cdot D_{2,h}^{-1},$$

where

$$V_{h-j} = [v_{d_1-j,1}\cdots v_{d_n-j,n}]\ ,\ j = 0,\ldots,h.$$

Because the corresponding elements of $D_{2,h}, D_{2,h-1}, \ldots, D_{2,0}$ have the same order of magnitude we have $D_{2,h}^{-1}\cdot D_{2,i} \approx I_{n,n}$. When there are no (almost) multiple poles and when the condition number of V_h is reasonably small all elements of the coefficients of the power series around $z = \infty$ of the factor

$$(V_h + D_{2,h}^{-1} D_{2,h-1} V_{h-1}\cdot z^{-1} + \cdots + D_{2,h}^{-1}\cdot D_{2,0} V_0 z^{-h})^{-1} \qquad (15)$$

will have order of magnitude 1 and will ultimately decrease (see section 6 note 3). Now the error power series is the product of the two power series (13) and (14) or the absolute error sequence is the convolution of the two Markov parameter sequences related with (13) and (14). The Markov parameter sequence of (14) is decreasing as we have just explained. The Markov parameter sequence of (13) reflects the behaviour of the given Markov parameter sequence because the factors

$$[\sigma_1 u_{1,j}\ \cdots\ \sigma_n u_{n,j}]\ ,\ j = d_1+1,\ldots,N$$

have the same order of magnitude because of the scaling, so that the factors $D_{1,j}^{-1}$, $j = d_1+1,\ldots,N$ dictate the behaviour of the Markov parameter sequence of (13). Now these scaling matrices are related with the magnitude of the given Markov parameters, so that (13) reflects this behaviour. As a consequence it turns out that the convolution of (13) and (14) is decreasing almost like the given Markov parameters are decreasing and therefore the relative error will be almost constant. We shall now investigate these results by using an example.

5. Example

All computations were done in double precision on a Microvax II computer. To check the performance of the algorithm described in section 4 we compute the first 50 Markov parameters of the following system with 3 inputs and 2 outputs and denominator column degrees 2, 2 and 4 respectively :

$$\begin{bmatrix} 3.7z-5.4 & -6.3z+4.2 & -2.12z^3+3.5z^2+1.6z+6.7 \\ 5.8z+4 & -2.3z+3.7 & 5.3z^3-8.4z^2+8.4z-2.3 \end{bmatrix}.$$

$$\begin{bmatrix} z^2-0.25 & 0 & 0 \\ 0 & z^2+0.1z-0.12 & 0 \\ 0 & 0 & z^4+0.21z^2-0.01 \end{bmatrix}^{-1} .$$

The first column has poles : -0.5 $+0.5$
The second column has poles : $+0.3$ -0.4
The third column has poles : $+0.2$ -0.2 $+i.0.5$ $-i.0.5$.

The 50 Markov parameters are corrupted with relative uniform random errors over the interval $[-10^{-8},10^{-8}]$. These corrupted Markov parameters are given as input to the algorithm of section 4. In the following figures the absolute values of the variables in function of an index are represented using a logarithmic scale. Figures 1, 2, 3 and 4 give the elements of the first 50 Markov parameters. Note that the modulus of the dominant poles of the first and third column is 0.5 and of the second column 0.4. Figure 5 represents the diagonal elements of the scaling matrix D_1 corresponding with the first block-Hankel matrix constructed by the algorithm. Figure 6 shows the diagonal elements of the scaling matrix D_2 corresponding with the last block Hankel matrix. Note that for the second column the scaling is not very good. The order of magnitude of the element (100,2) of $D_1 \cdot H_{N,0} \cdot D_2$ is 10^{-4}. The singular values corresponding with the solution vectors are :

$$\begin{cases} \sigma_1 = 6.5 \cdot 10^{-9} \\ \sigma_2 = 5.0 \cdot 10^{-9} \\ \sigma_3 = 5.3 \cdot 10^{-9} \end{cases}$$

The corresponding left singular vectors are respresented in figures 7, 8 and 9. They have length one and their elements have the same order of magnitude. Figures 10, 11, 12, 13, 14 and 15 give the elements of factor (15). Figures 16, 17, 18, 19, 20 show n_1, n_2, d_1, d_2, d_3 respectively such that the elements (i,j) of (13) $\cdot$ (14) are the convolutions of n_i and d_j. The relative errors are represented in figures 21, 22, 23, 24 and 25. Figure 26 shows for each index value the maximum of these relative errors.

We consider the same problem but we apply no scaling; all coefficients of the elements of factor (15) have the same order of magnitude as in the previous example. The singular values corresponding with the solution vectors are :

$$\begin{cases} \sigma_1 = 2.4 \cdot 10^{-9} \\ \sigma_2 = 2.9 \cdot 10^{-9} \\ \sigma_3 = 1.8 \cdot 10^{-9} \end{cases}$$

The corresponding left singular vectors are given in figures 27, 28 and 29. Note that their elements follow the behaviour of the most slowly decreasing entry of the given Markov parameters. Figures 30, 31, 32, 33 and 34 show the relative errors. Figure 35 gives the maximum of these relative errors. Convolution with the left singular vectors for the first and third column of factor (15) gives for Markov parameter 50 an error matrix with elements of order of magnitude $\approx 10^{-12} \cdot 10^{-9} = 10^{-21}$; when we compare this with the elements of the first and third column of the given Markov parameters of the order of magnitude 10^{-14} we get a relative error $\approx 10^{-7}$; doing the same thing for the second column gives : $10^{-12} \cdot 10^{-9} = 10^{-21}$ to be compared with 10^{-18} gives a relative error of 10^{-3}.

6. Some notes and conclusions :

Note 1 : When the block-Hankel matrix can not be scaled properly then we can loose α digits when $\beta.10^{-\alpha}$ is the smallest element in absolute value in the scaled matrix $D_1 \cdot H_{N,0} \cdot D_2$.

This problem occurs when different entries in the transfer function matrix have different dominant poles in modulus.

Note 2 : When we consider the single-input single-output case and assume that the Markov parameters are not varying rapidly the scaling can be done very well (in fact the column scaling is then not necessary).

Note 3 : More digits can be lost when there are (almost) multiple poles because then the Markov parameter sequence of (14) will not decrease sufficiently fast or even increase initially and decay only at a later stage. Another bad situation occurs when the highest degree coefficient matrix V_h has a large condition number. This increases the magnitude of the Markov parameter sequence of (14). so that they are not of the order of magnitude 1 but are much larger. In both cases the convolution of (13) and (14) has not the desired behaviour as described at the end of section 4.

Note 4 : It is clear that the algorithm of section 4 could also be used for model reduction; instead of defining the concept 'numerically zero' based on the relative error ϵ, it is based on the relative error between the given Markov parameters and the Markov parameters of the approximation to be searched for.

Note 5 : It is clear that each step of the algorithm can be implemented in a numerically stable way (The routines DSVDC and DQRDC of LINPACK can be used).

Conclusion : An algorithm is proposed that computes a right matrix fraction description whose Markov parameters approximate the given Markov parameters with a certain relative precision.
Insight is given in the approximating properties of the computed transfer function by using a numerical example.

References

[de Jong] L.S. de Jong, 'Numerical aspects of realization algorithms in linear systems theory', Ph.D. thesis, Technische Hogeschool Eindhoven, 1975.

[Dickinson,Morf,Kailath] B.W. Dickinson, M. Morf and T. Kailath, 'A minimal realization algorithm for matrix sequences', IEEE Trans. Automat. control AC-19 (1974), 31-38.

[Gragg,Lindquist] W.B. Gragg and A. Lindquist, 'On the partial realization problem', Linear Algebra Appl. 50 (1983), 277-319.

[Van Barel,Bultheel] M. Van Barel and A. Bultheel, 'A minimal partial realization algorithm for MIMO systems', Internal report TW79, K.U.leuven, Department of Computer Science, 1986.

[Van Huffel,Vandewalle] S. Van Huffel and J. Vandewalle, 'The use of total linear least squares techniques for identification and parameter estimation', IFAC Identification and System Parameter Estimation 1985, York, U.K., 1167-1172.

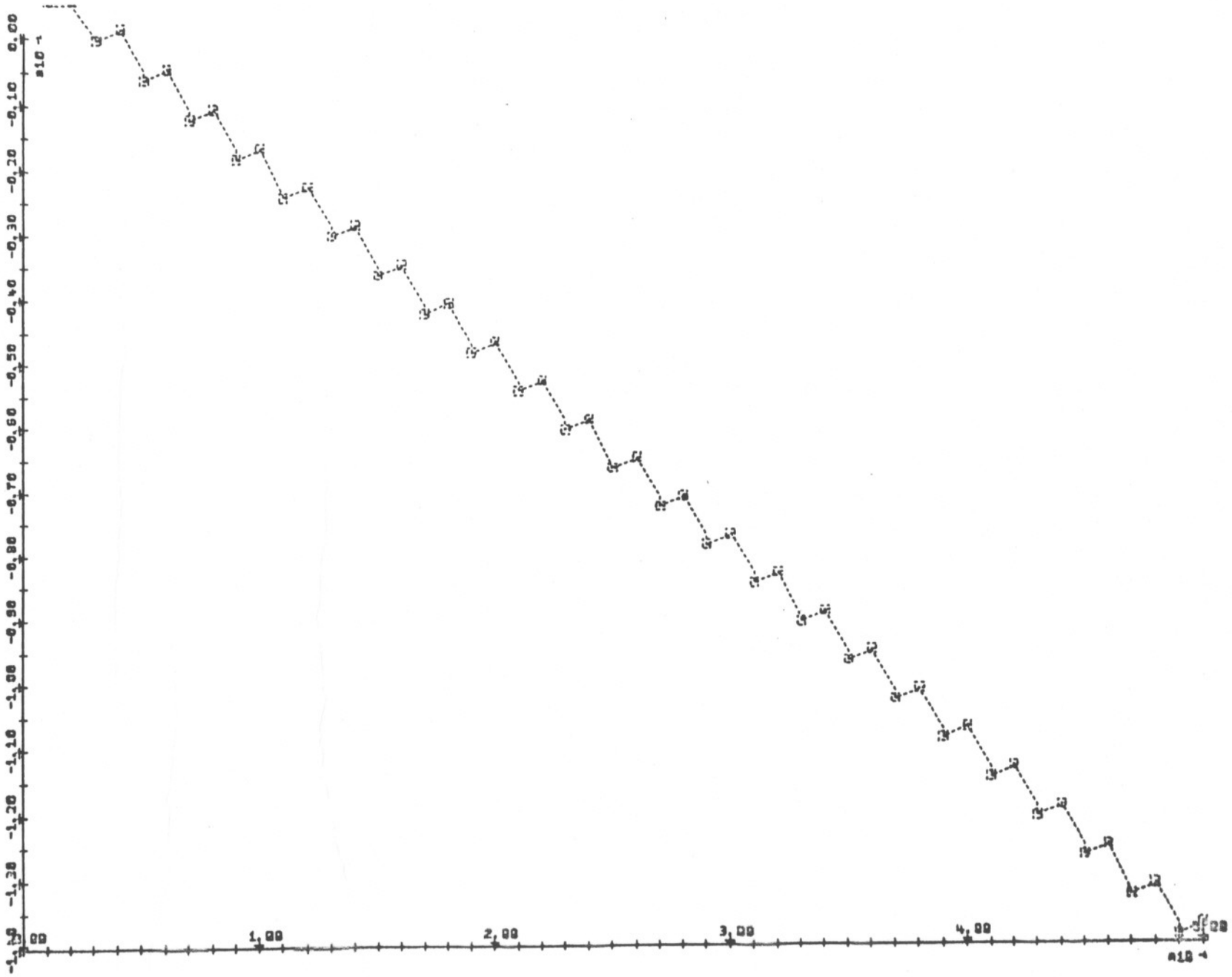


Figure 1 : Markov parameter elements (1,1), (2,1) and (1,3).

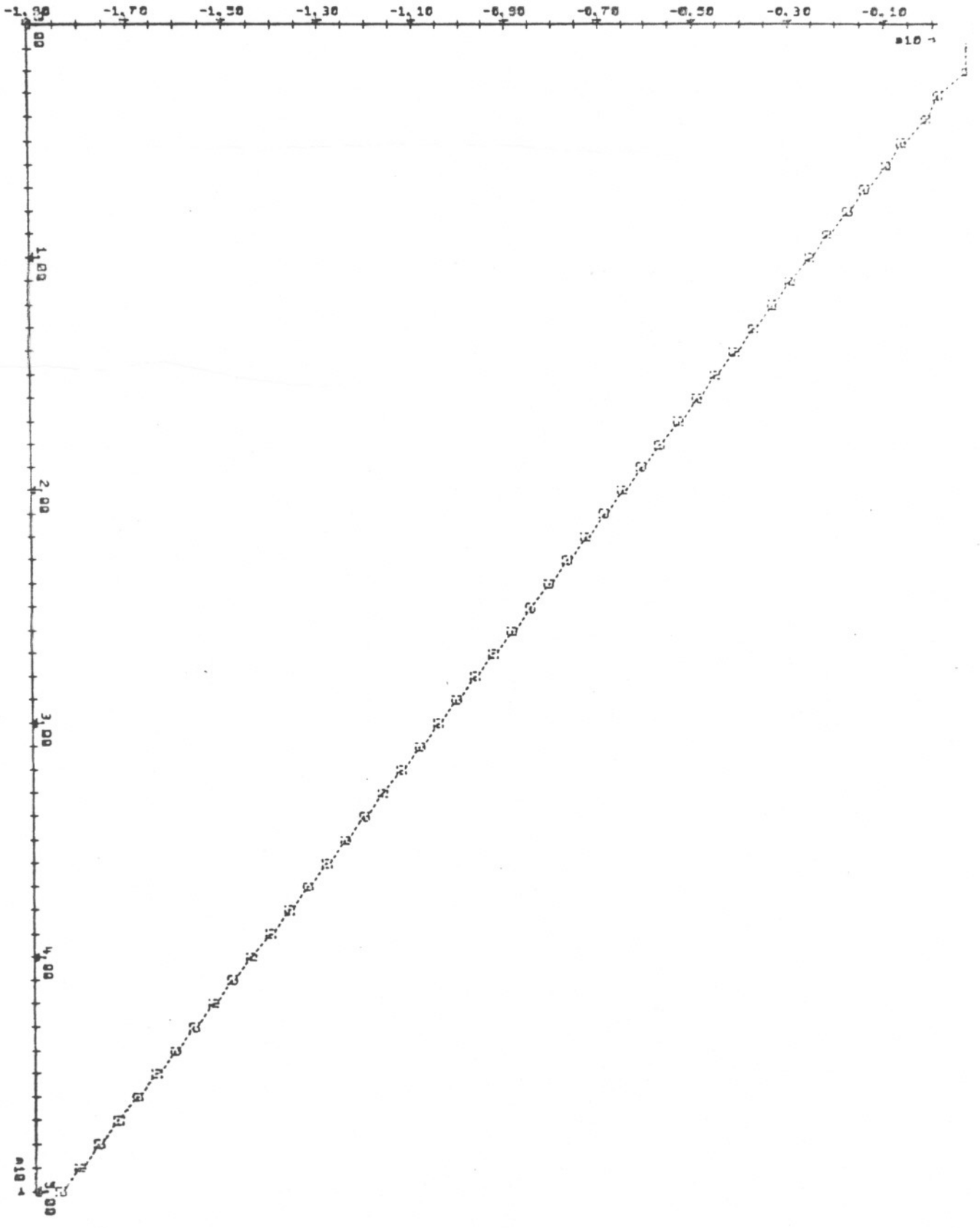


Figure 2 : Markov parameter element (1,2).

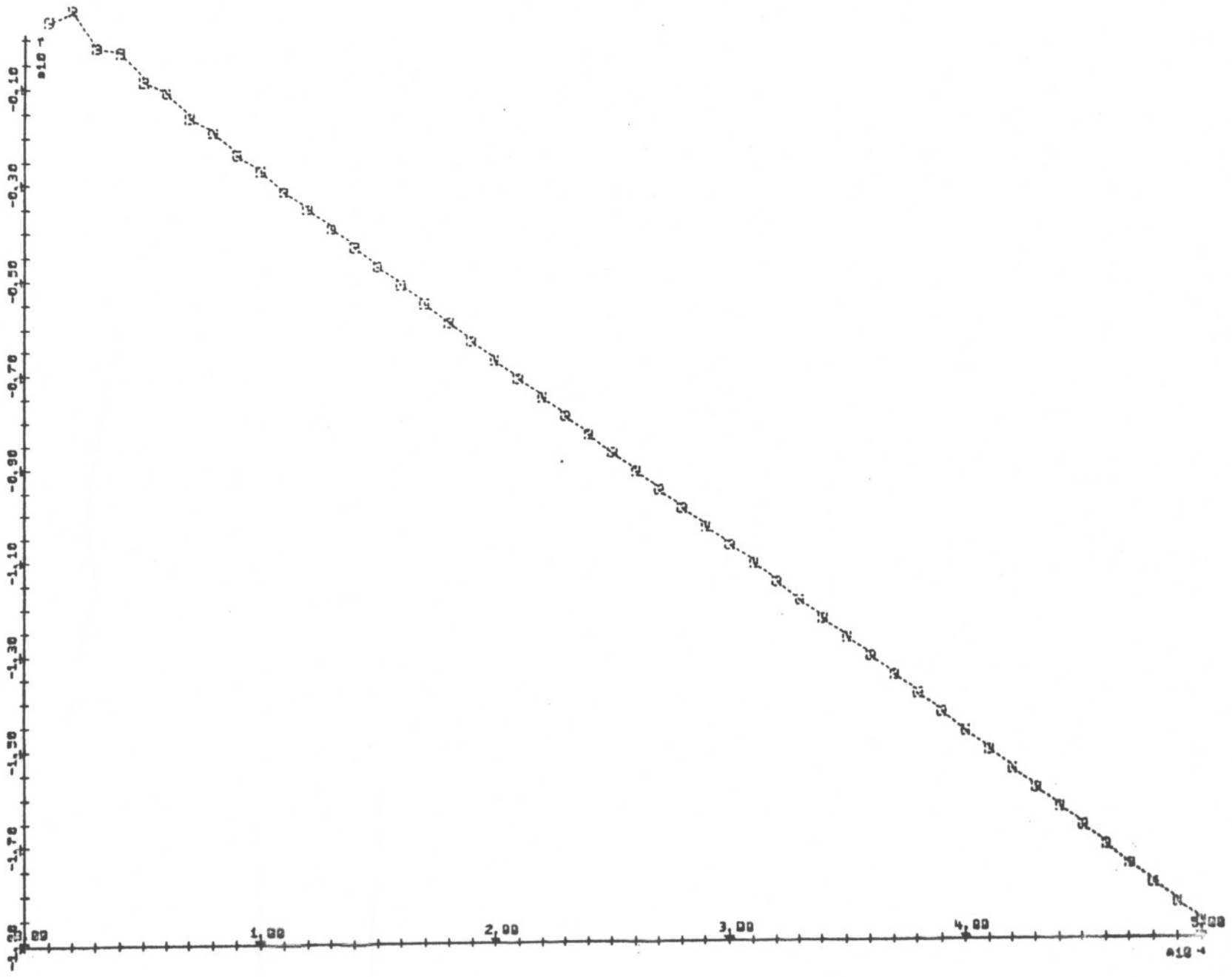


Figure 3 : Markov parameter element (2,2).

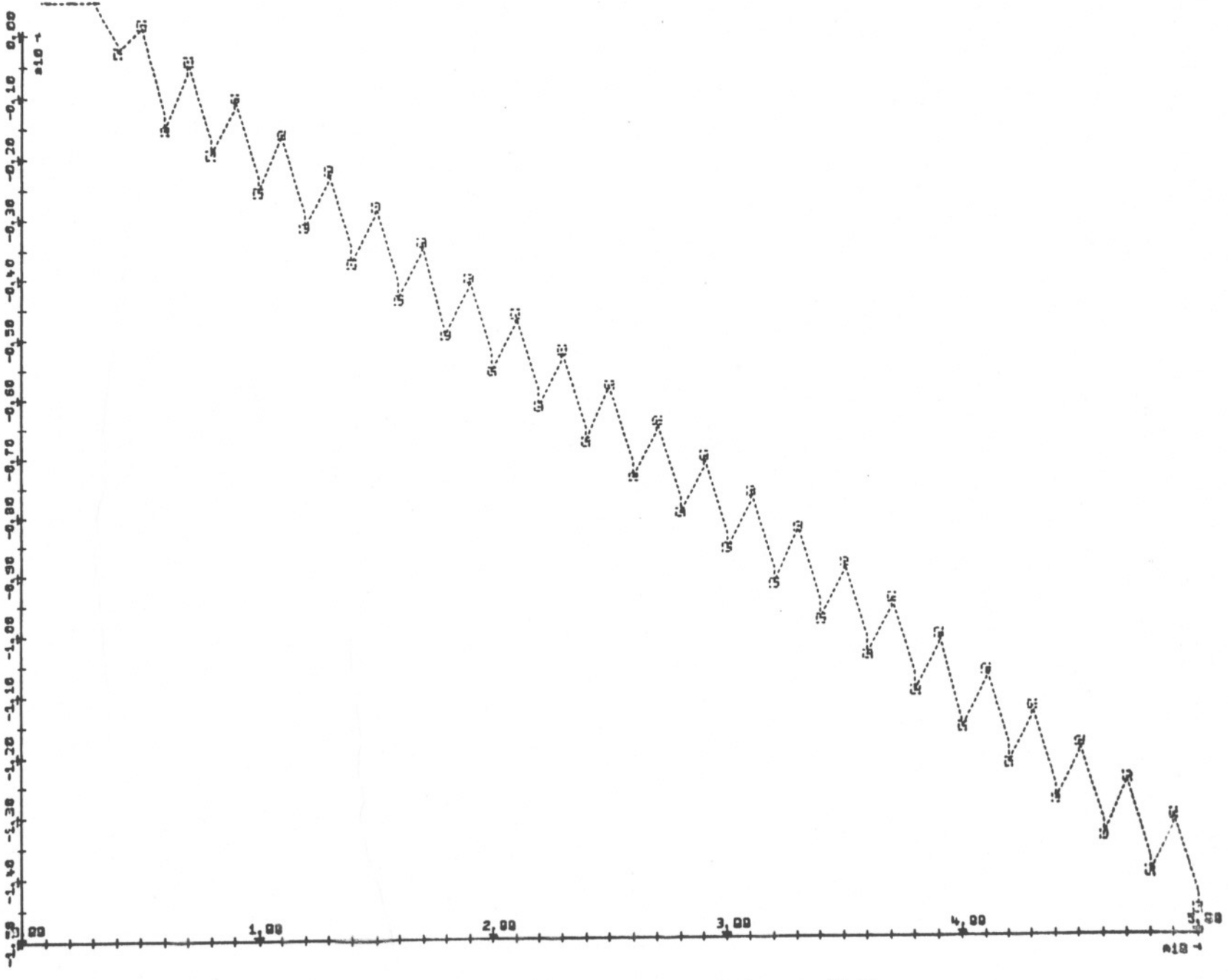


Figure 4 : Markov parameter element (2,3).

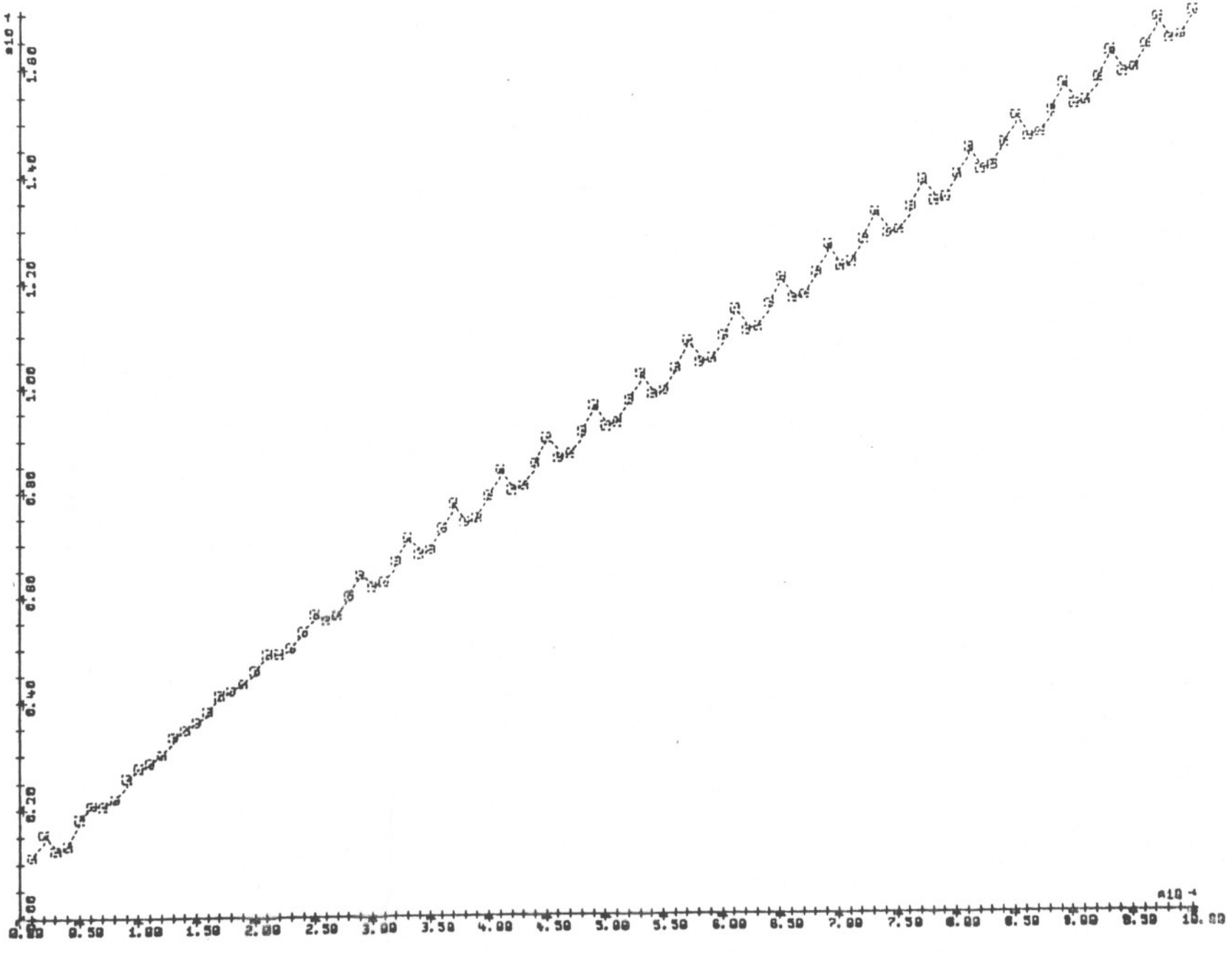


Figure 5 : Diagonal elements of the scaling matrix D_1.

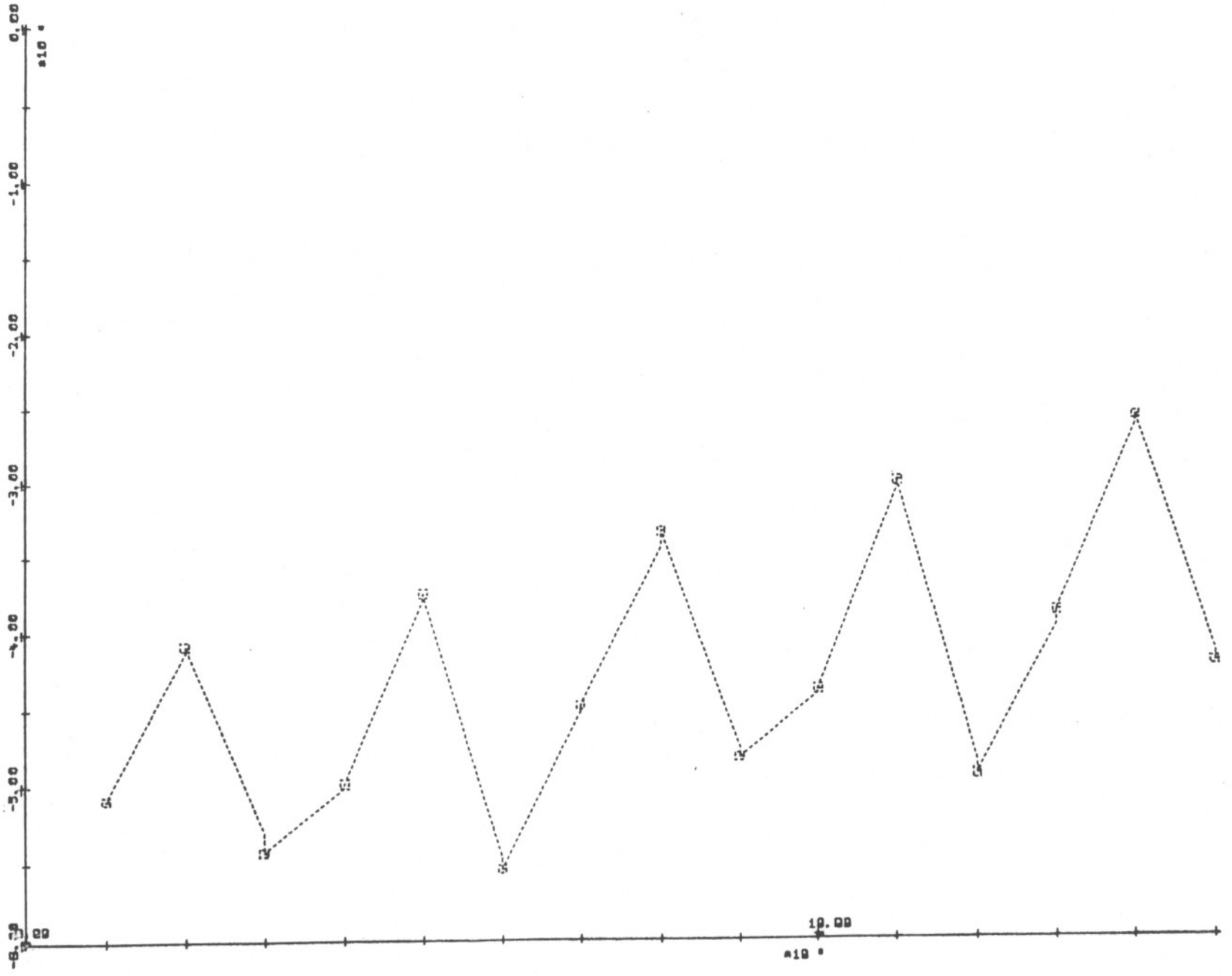


Figure 6 : Diagonal elements of the scaling matrix D_2.

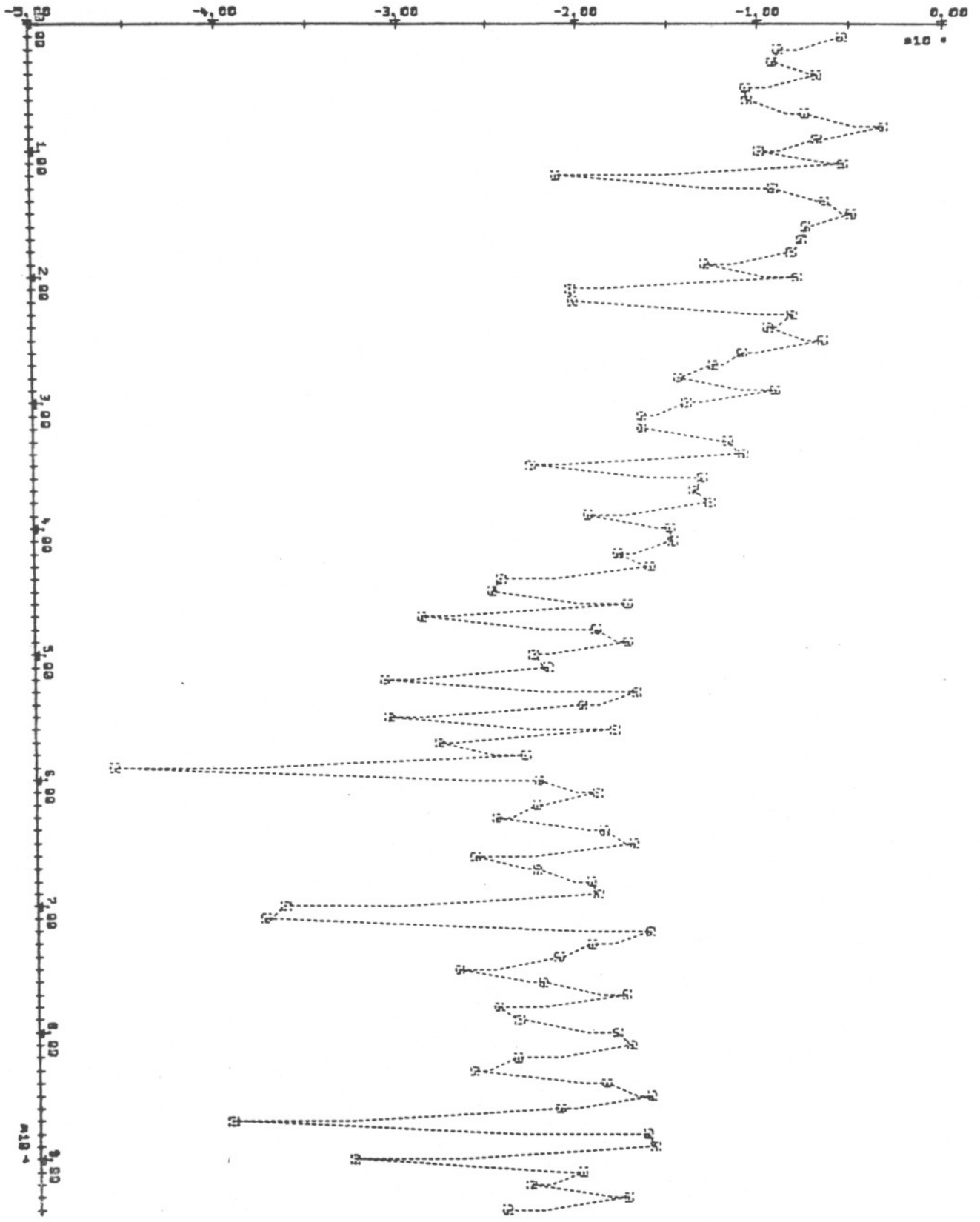


Figure 7 : Left singular vector corresponding to σ_1.

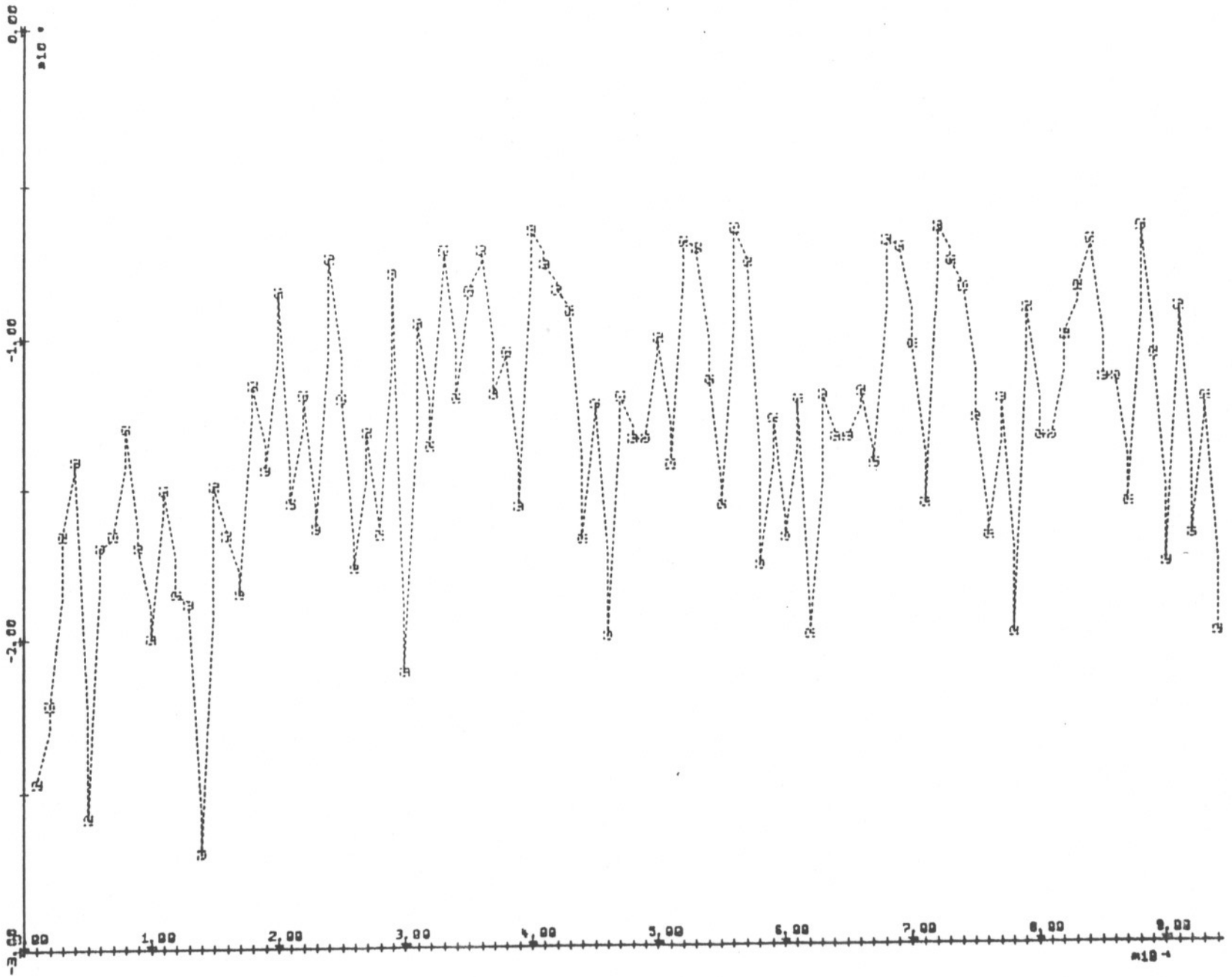


Figure 8 : Left singular vector corresponding to σ_2.

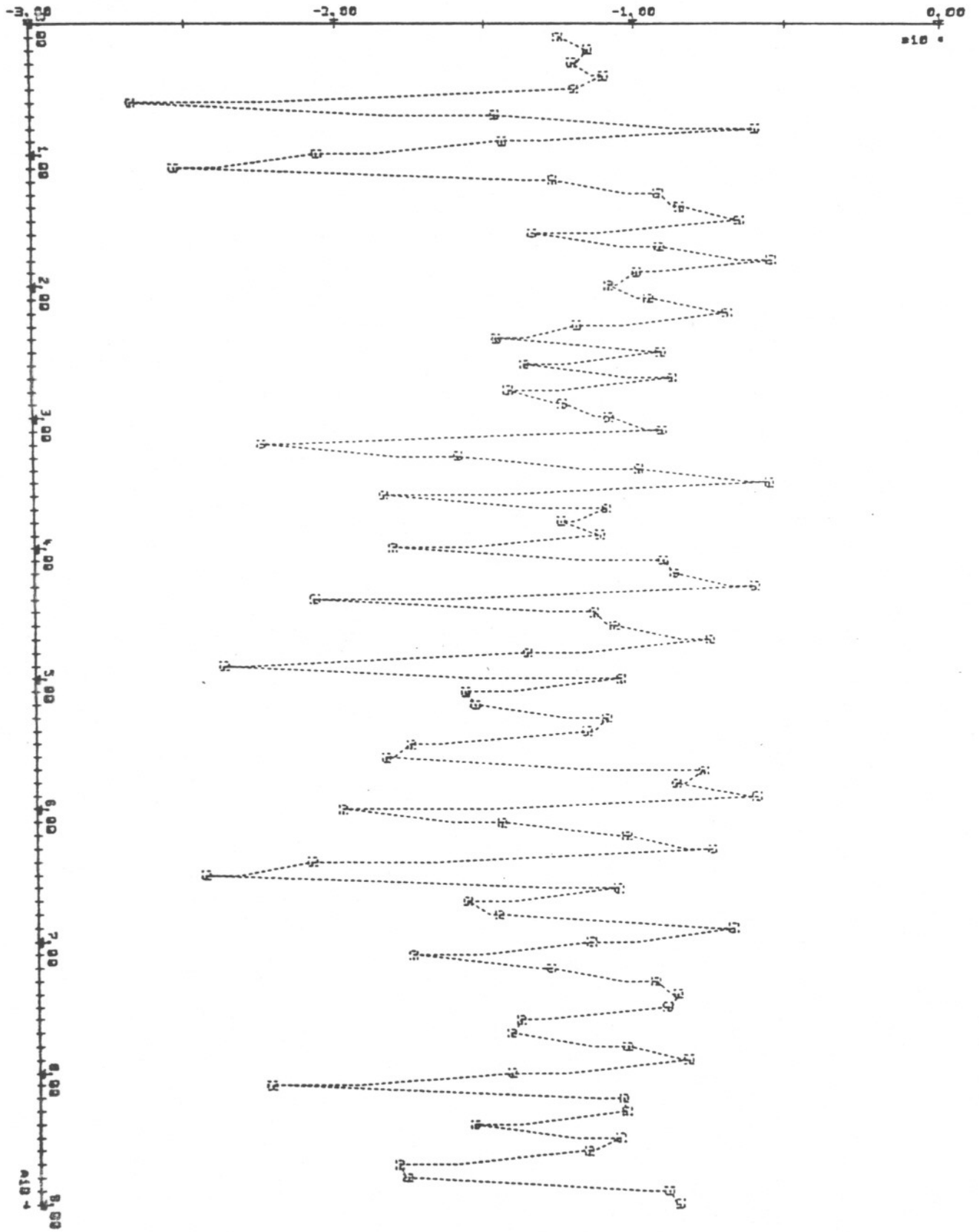


Figure 9 : Left singular vector corresponding to σ_3.

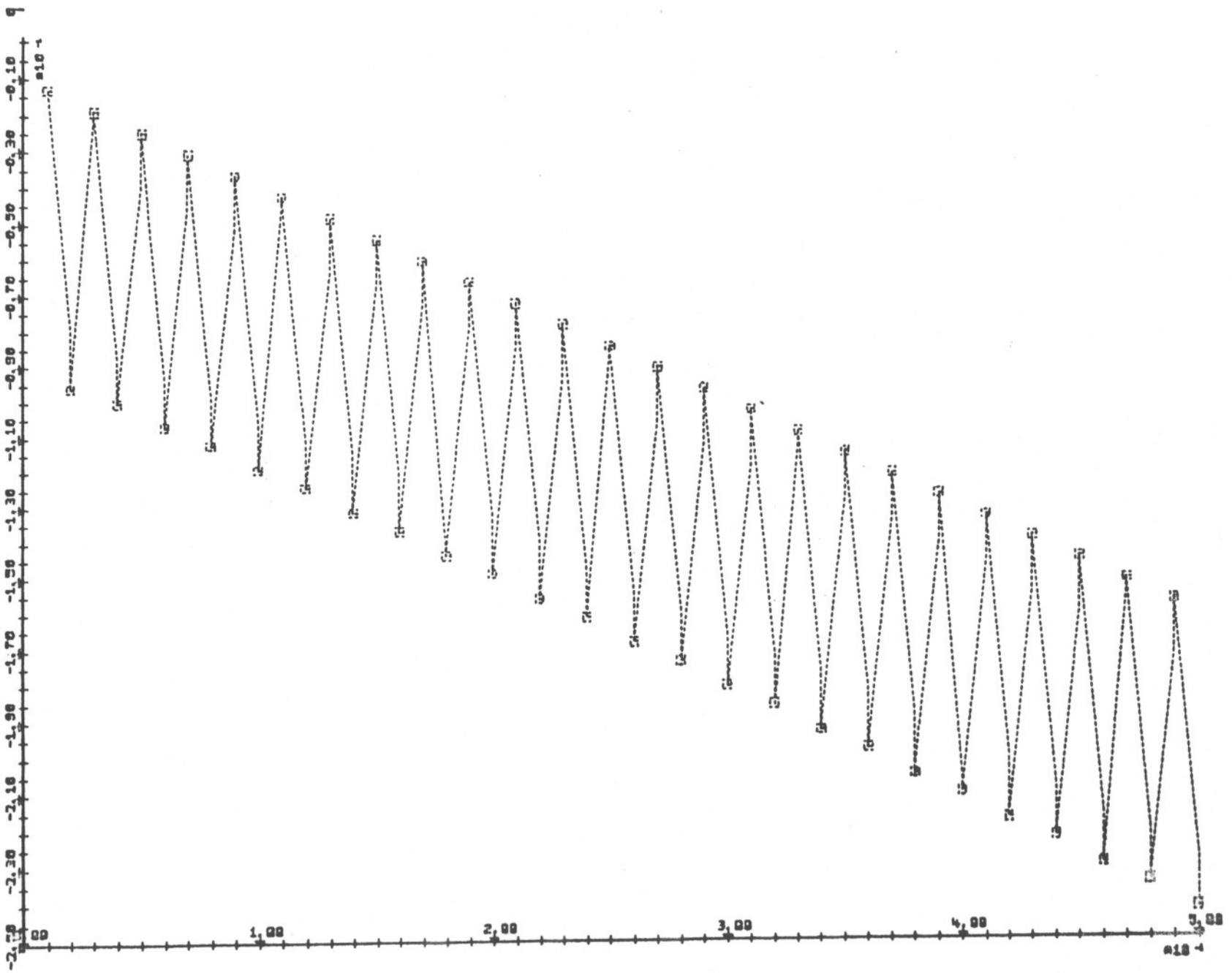


Figure 10 : Elements (1,1) and (2,1) of factor (15).

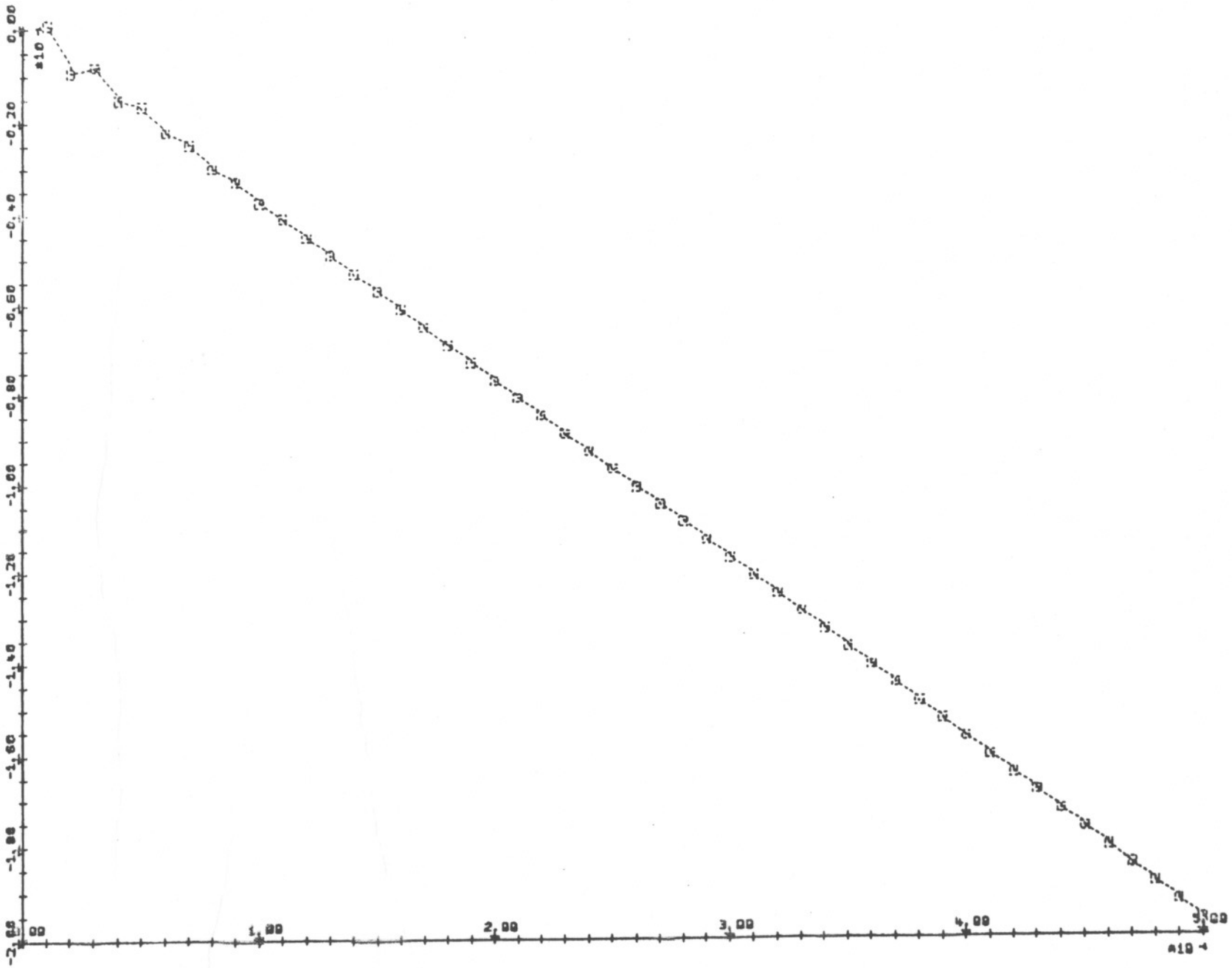


Figure 11 : Elements (1,2) and (2,2) of factor (15).

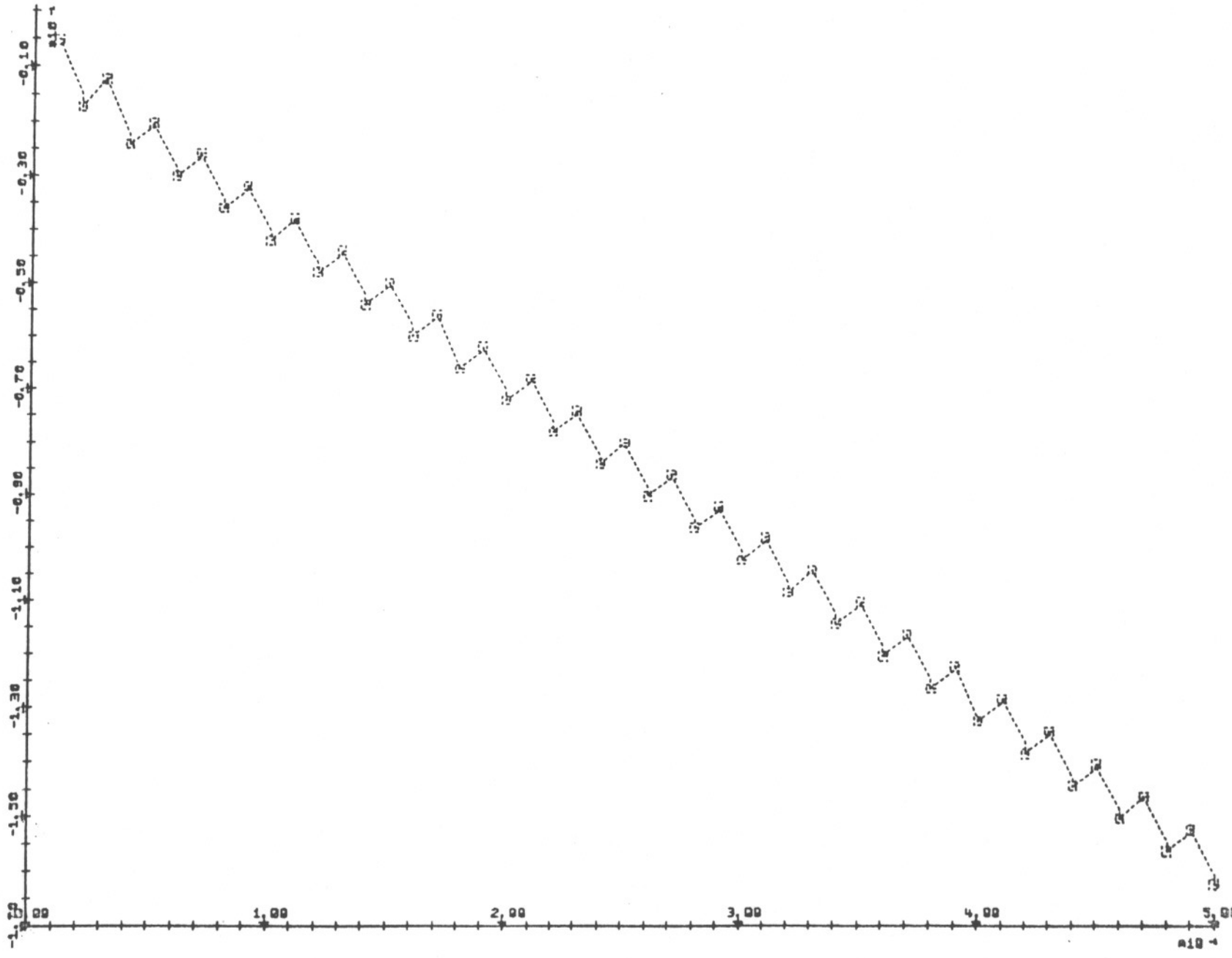


Figure 12 : Elements (1,3) and (2,3) of factor (15).

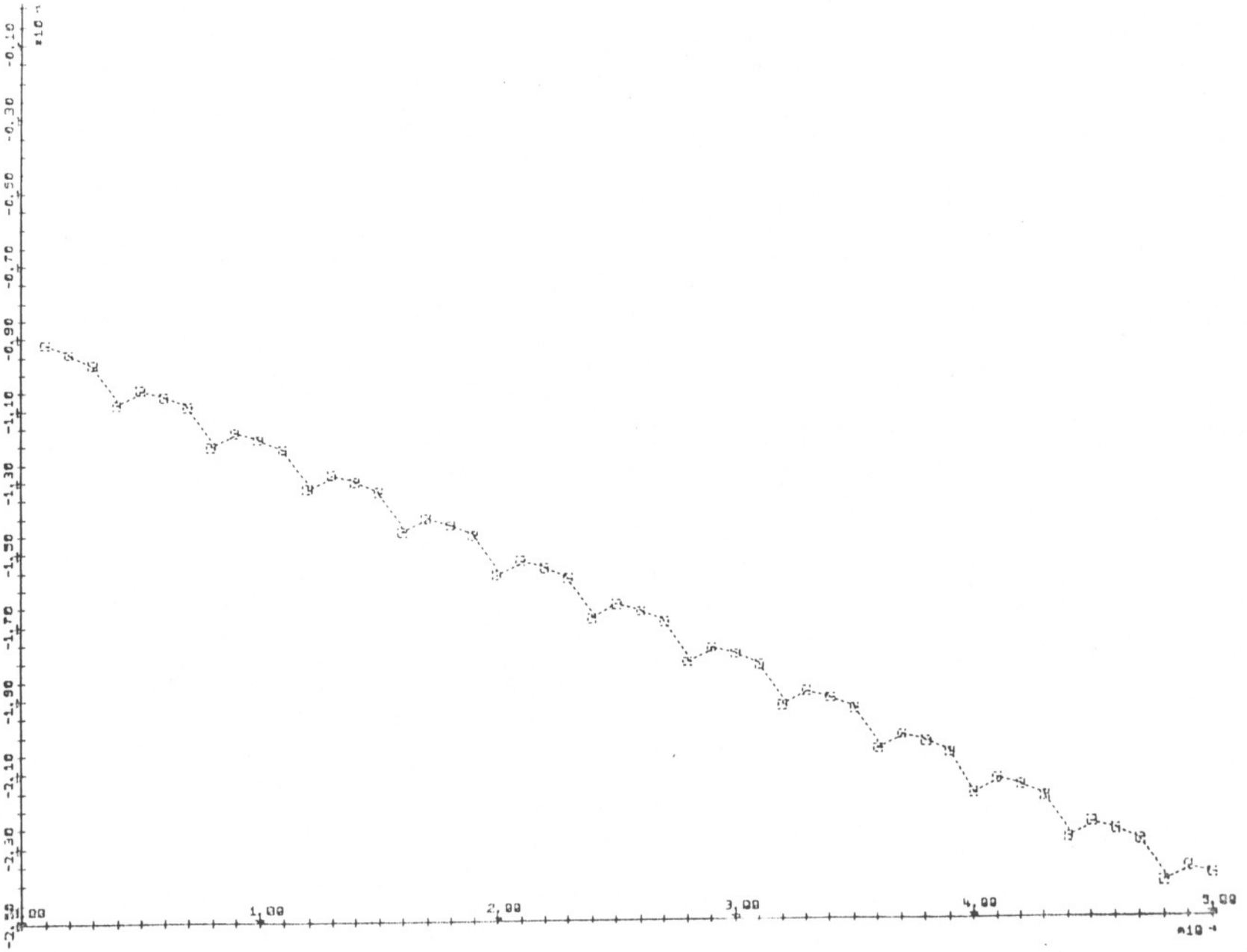


Figure 13 : Element (3,1) of factor (15).

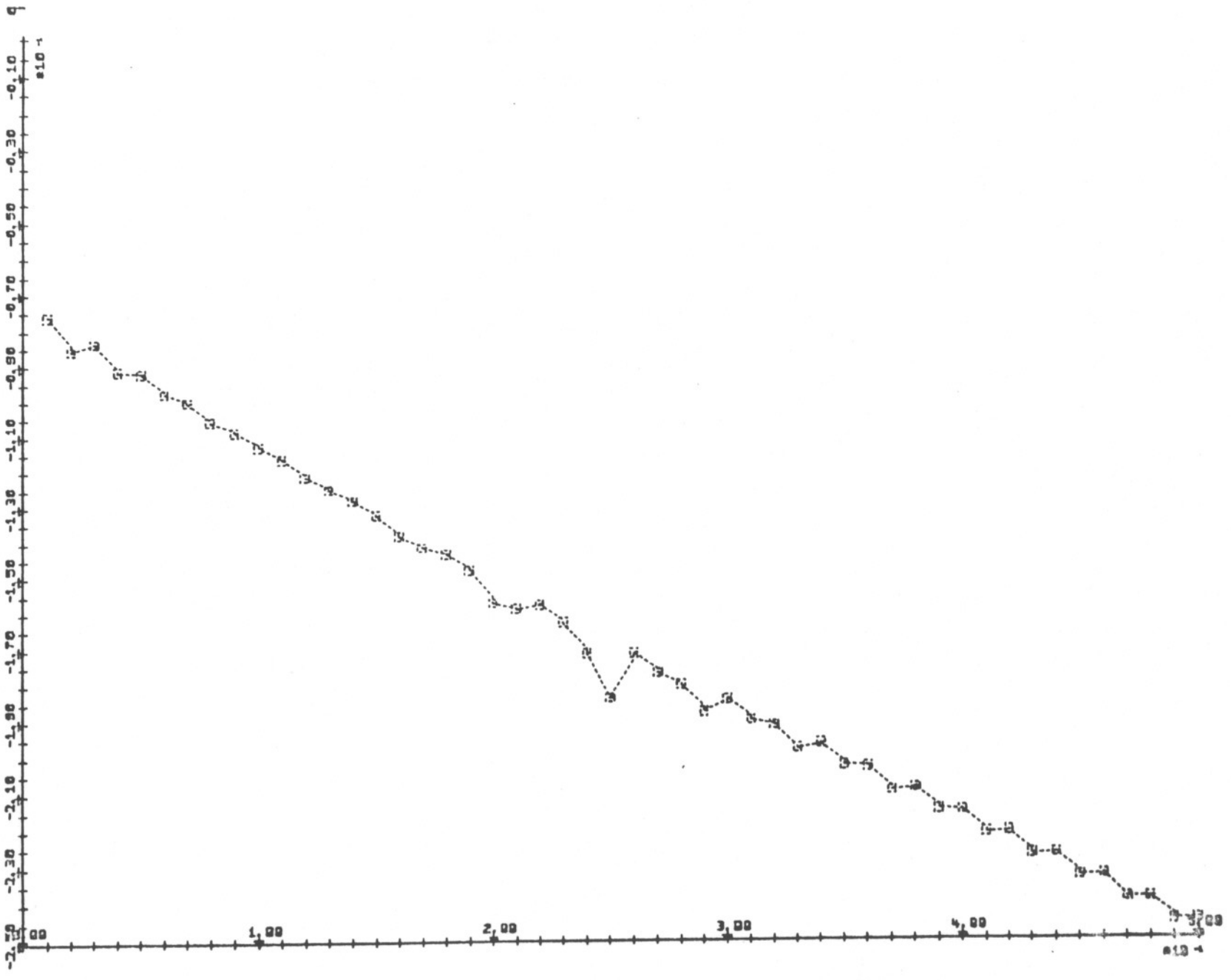


Figure 14 : Element (3,2) of factor (15).

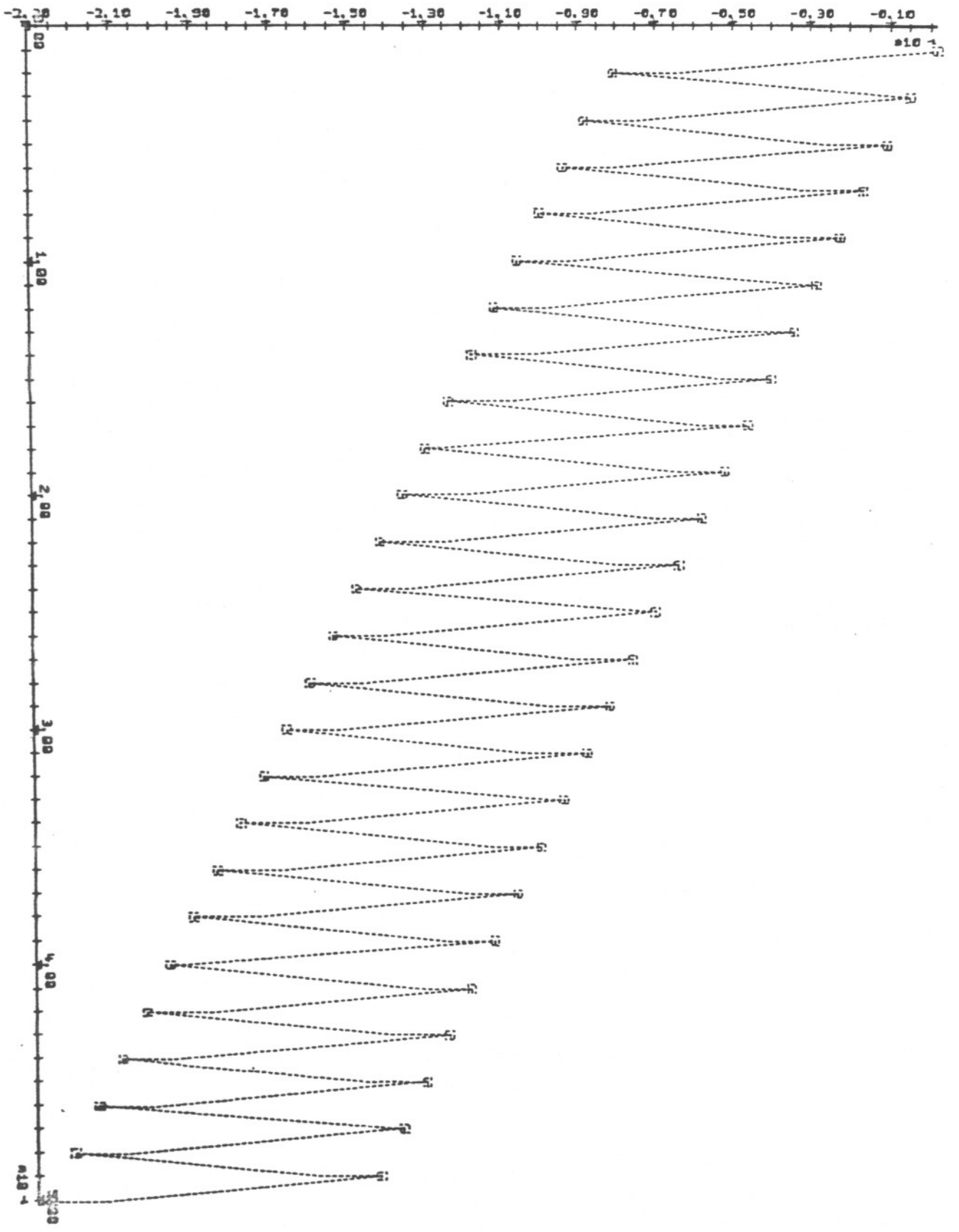


Figure 15 : Element (3,3) of factor (15).

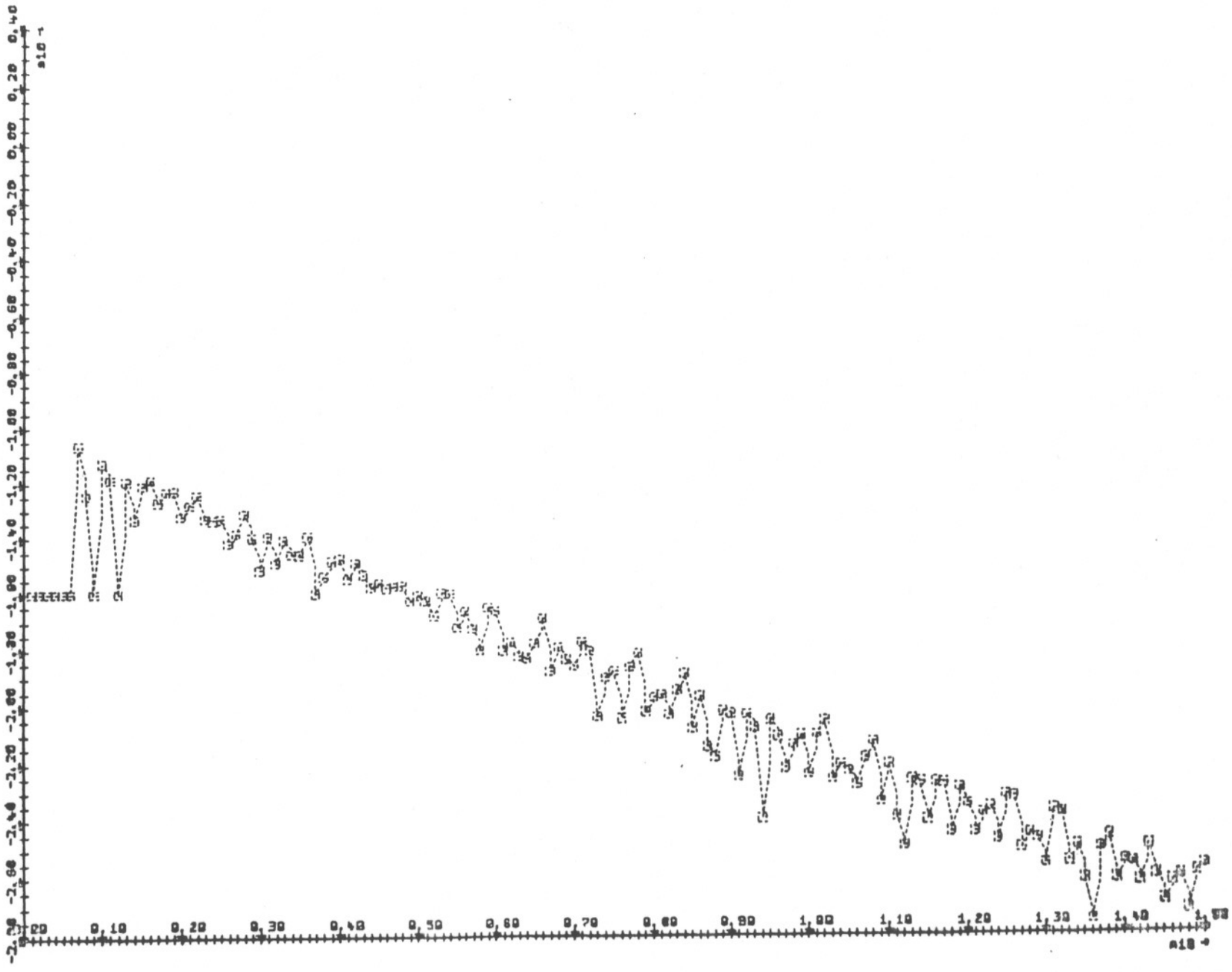


Figure 16 : n_1.

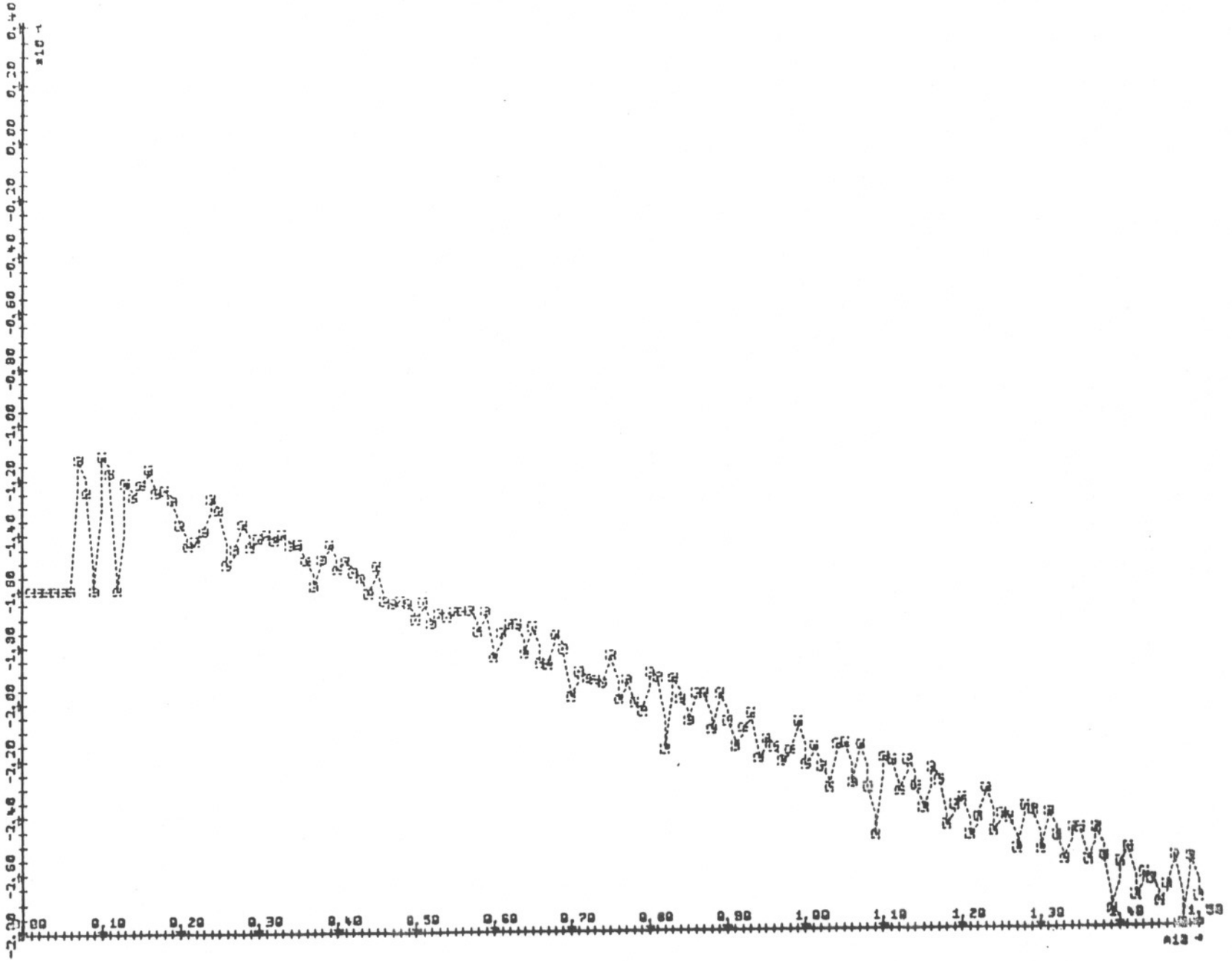

x10 -1
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
-1.20
-1.40
-1.60
-1.80
-2.00
-2.20
-2.40
-2.60
-2.80
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
1.10
1.20
1.30
1.40
1.50
x10 -4

Figure 17 : n_2.

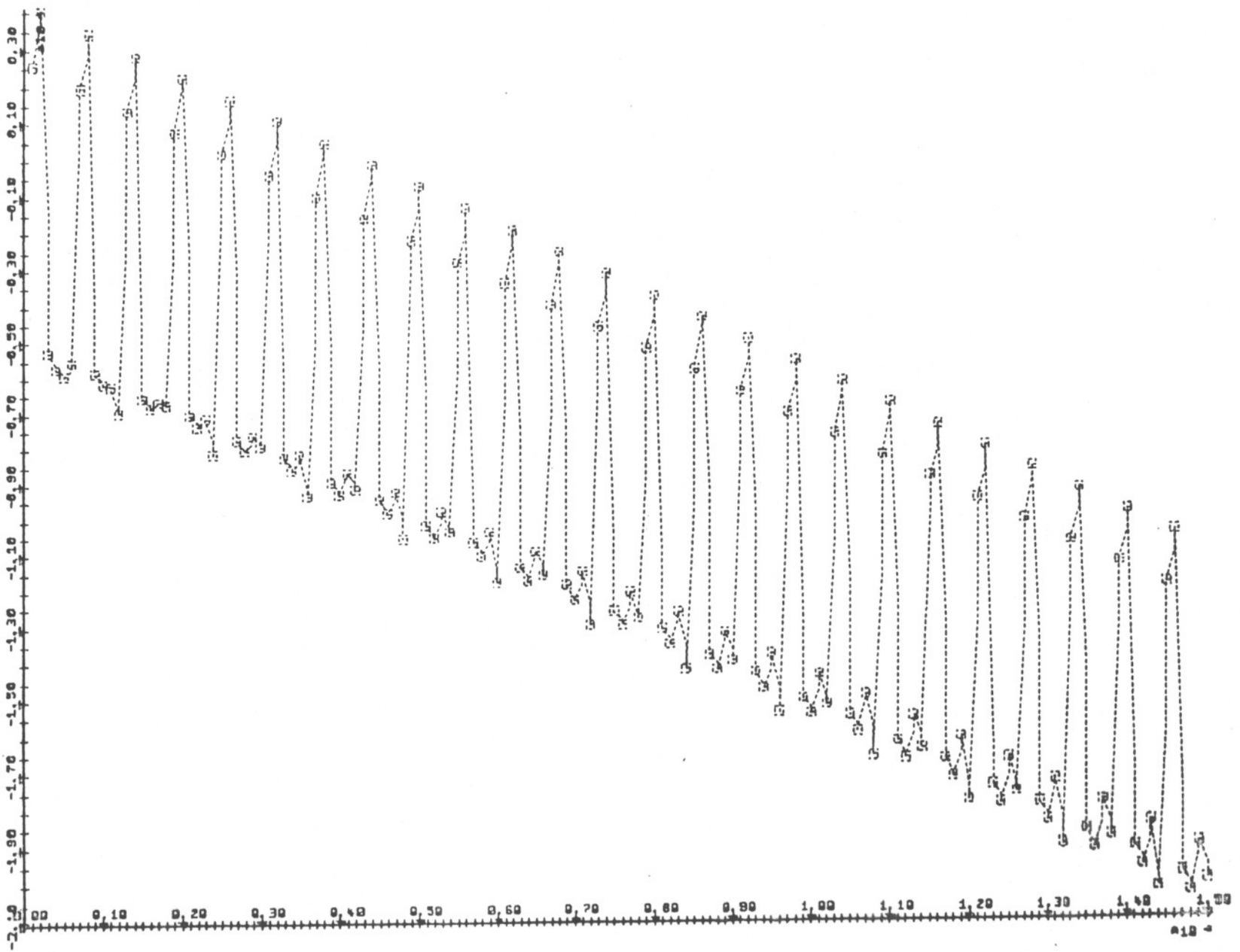

0,30
0,10
-0,10
-0,30
-0,50
-0,70
-0,90
-1,10
-1,30
-1,50
-1,70
-1,90
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
0,90
1,00
1,10
1,20
1,30
1,40
1,50

Figure 18 : d_1.

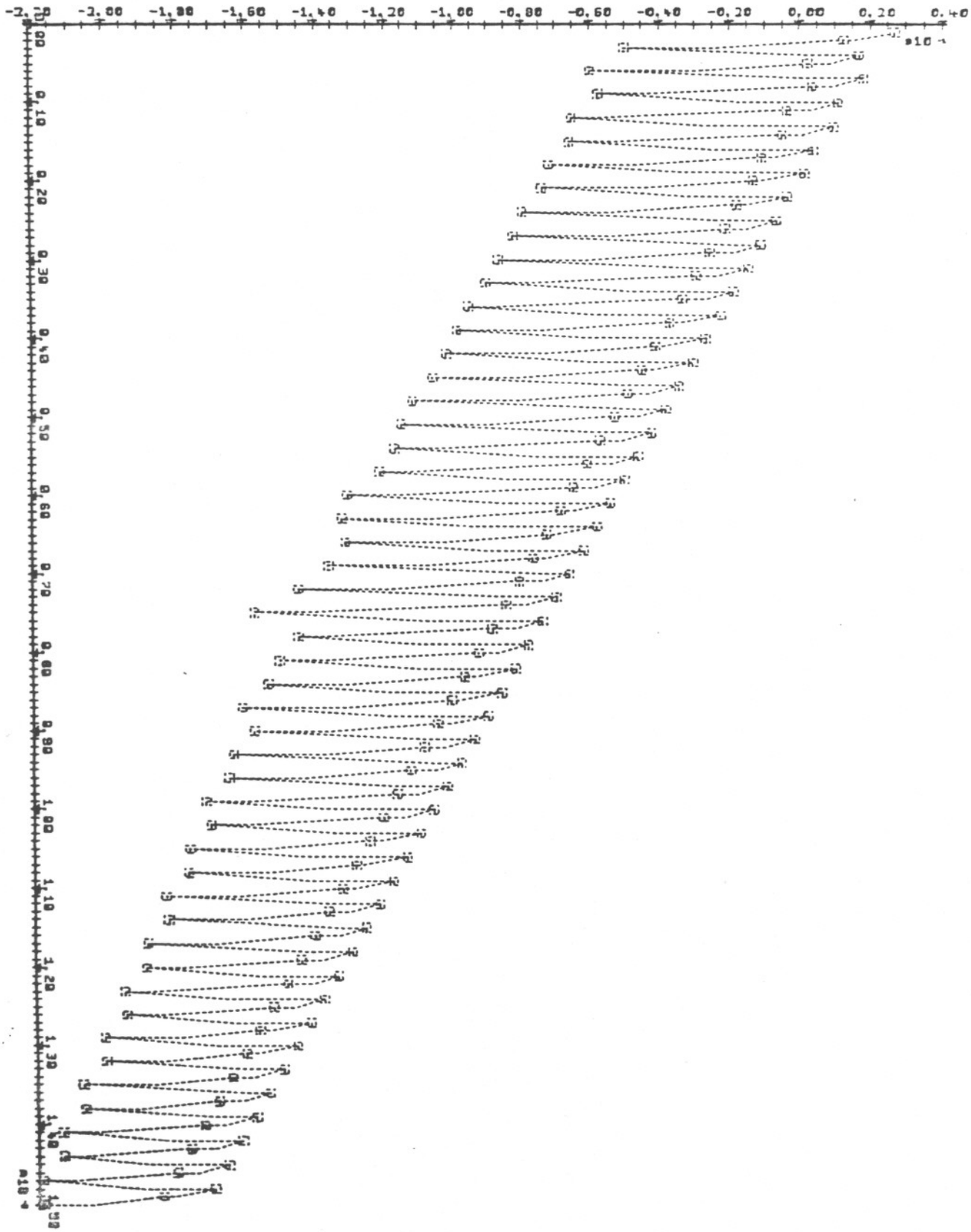

-2.00
-1.80
-1.60
-1.40
-1.20
-1.00
-0.80
-0.60
-0.40
-0.20
0.00
0.20
0.40
*10 -1
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
1.10
1.20
1.30
*10 -1

Figure 19 : d_2.

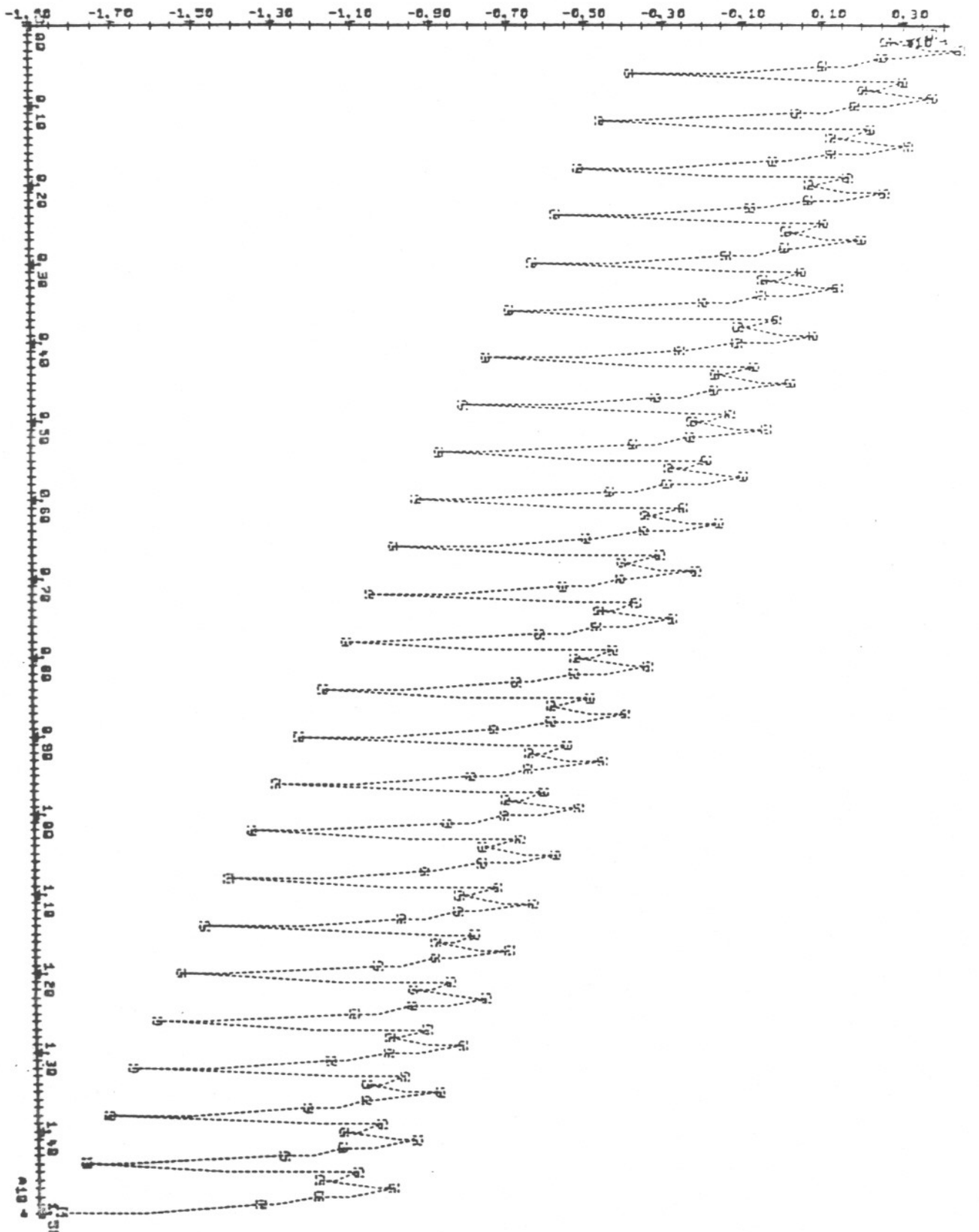

-1.90
-1.70
-1.50
-1.30
-1.10
-0.90
-0.70
-0.50
-0.30
-0.10
0.10
0.30
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
1.10
1.20
1.30
1.40
1.50
e10 +

Figure 20 : d_3.

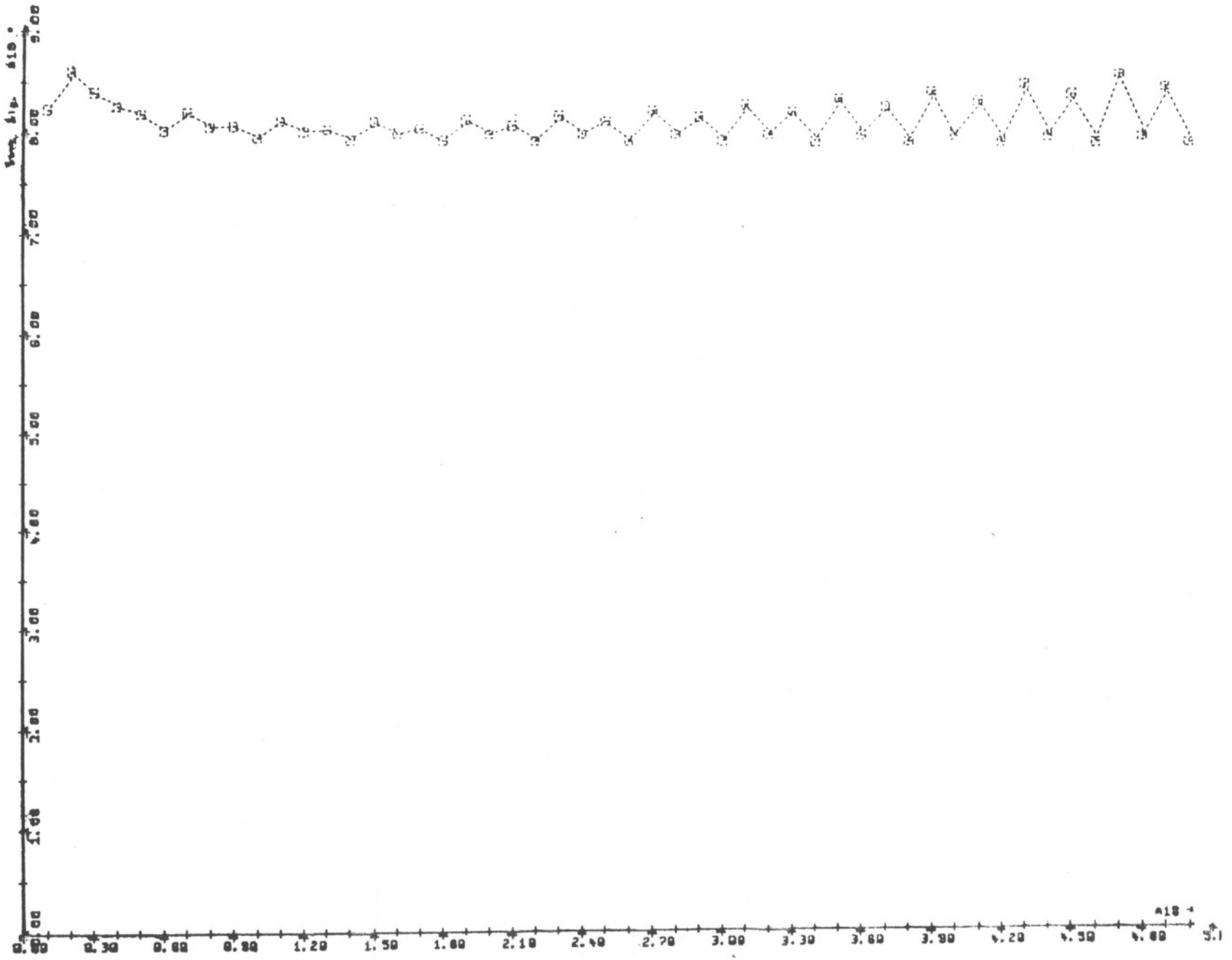


Figure 21 : Relative errors for elements (1,1) and (2,1).

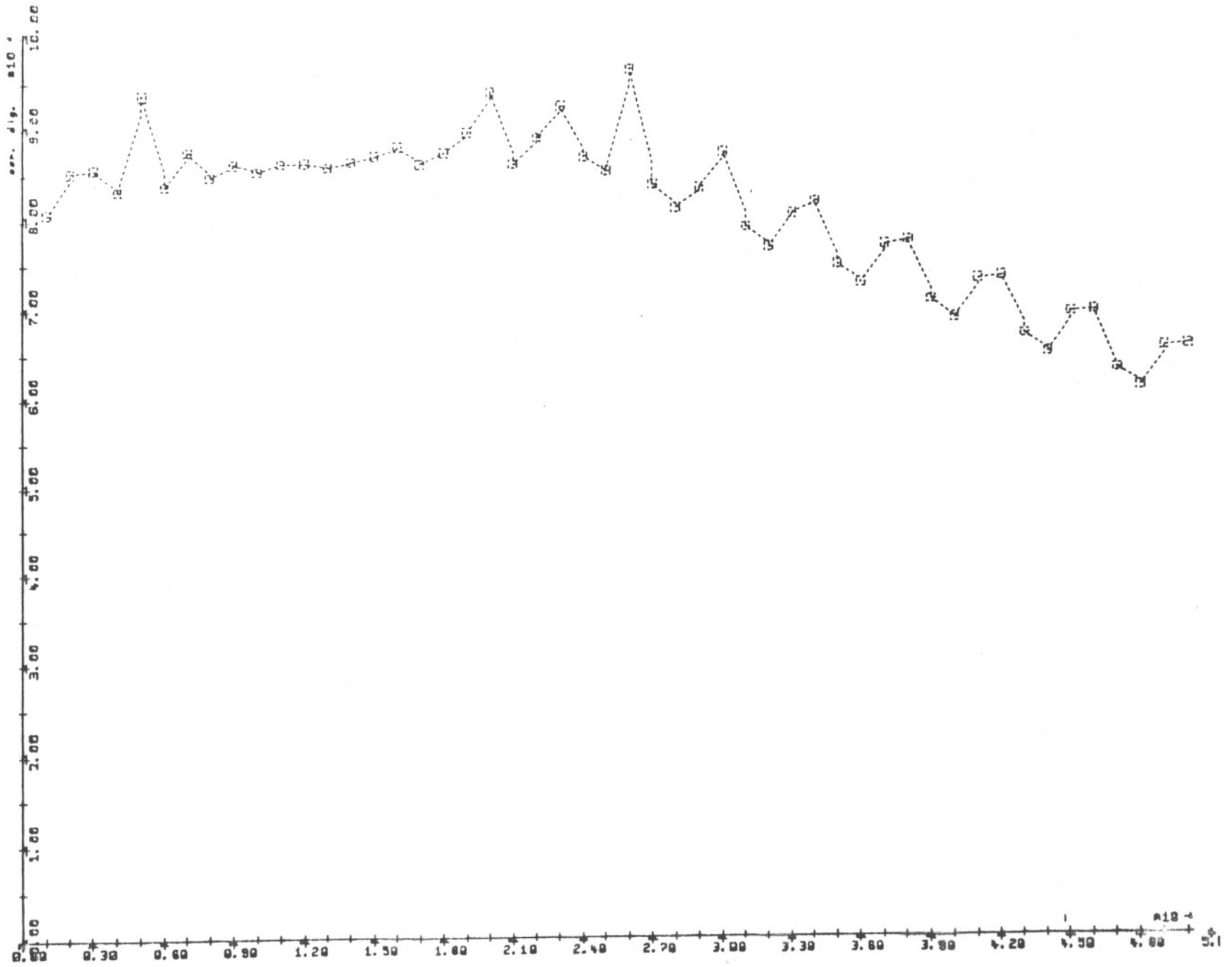


Figure 22 : Relative errors for element (1,2).

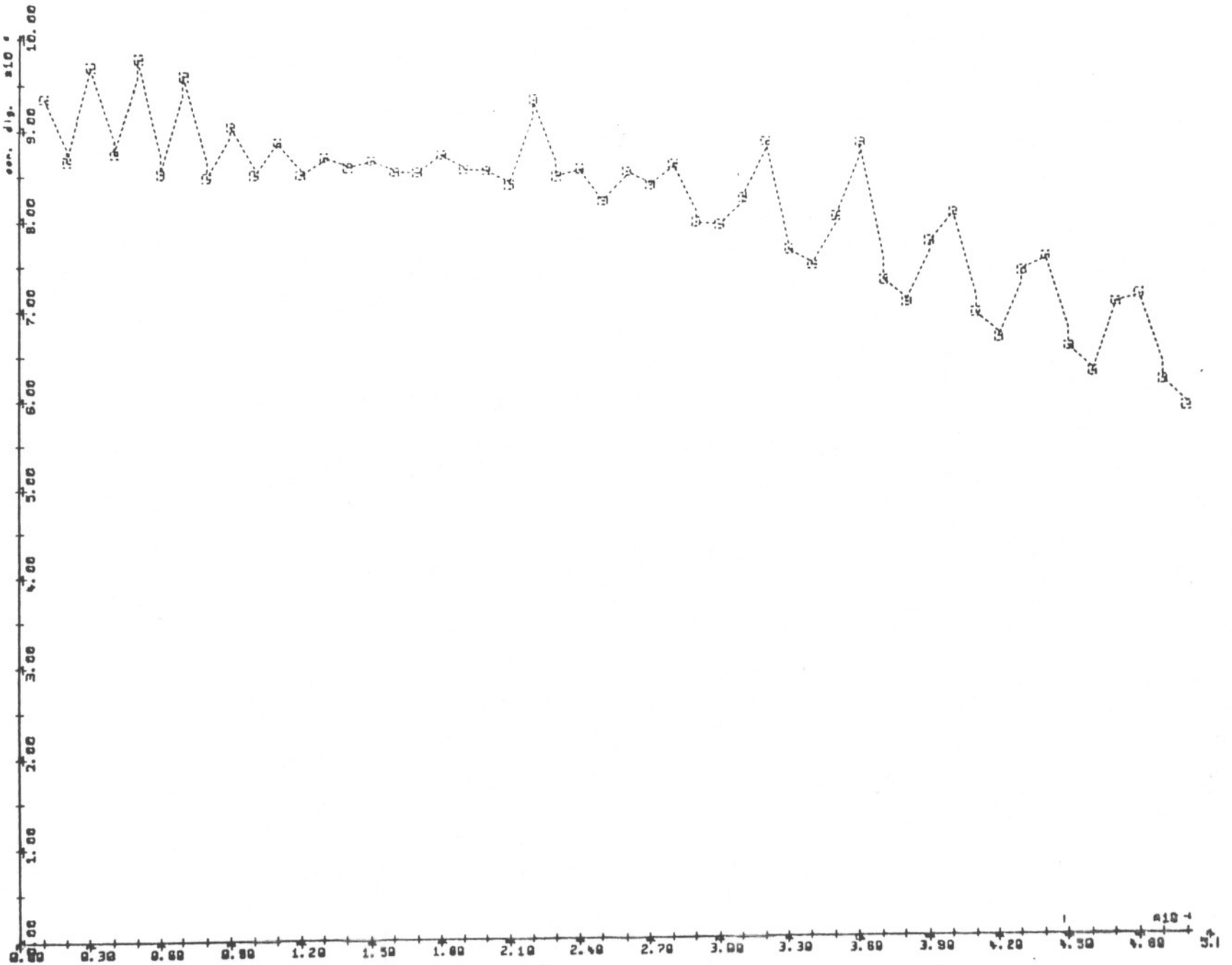


Figure 23 : Relative errors for element (2,2).

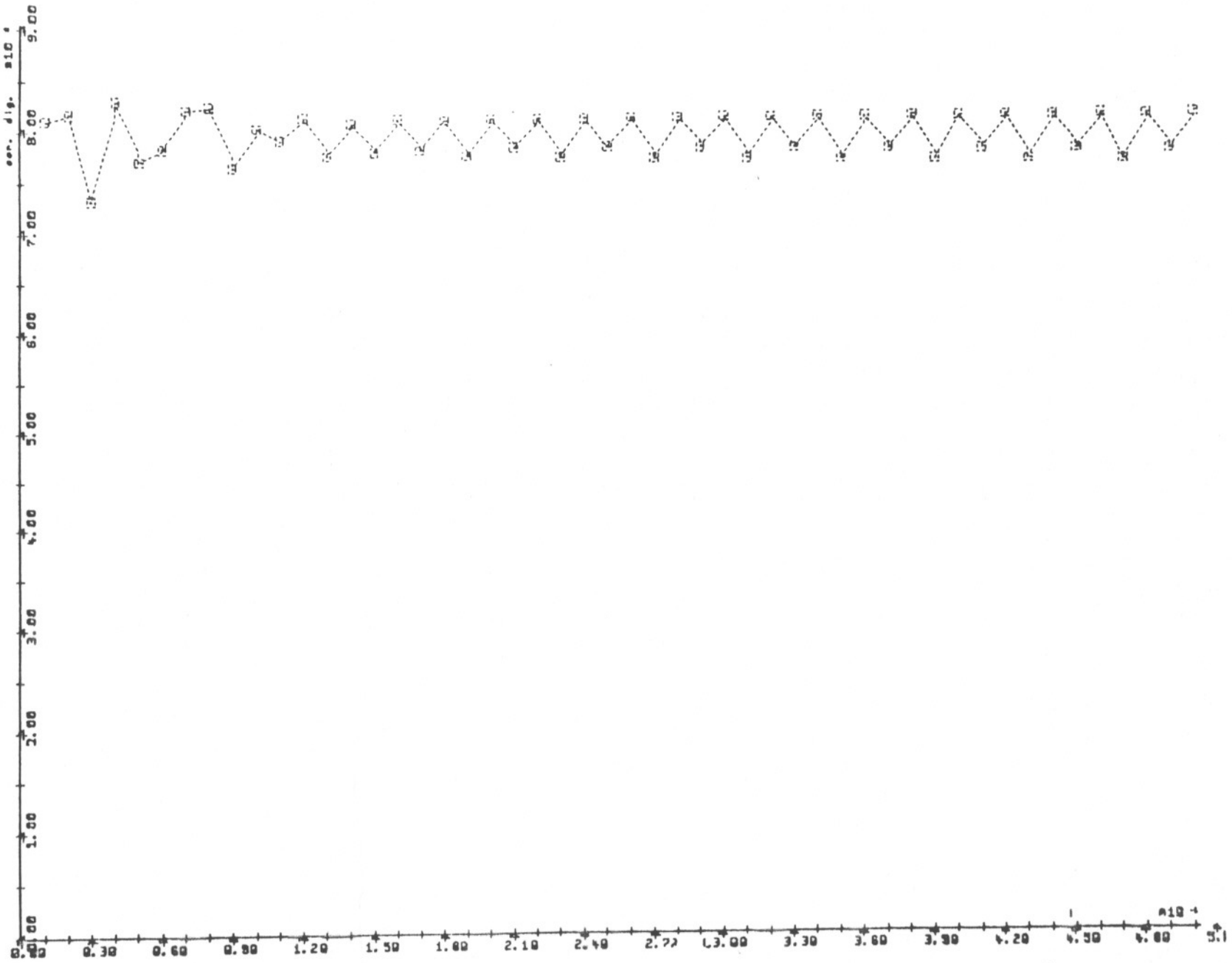


Figure 24 ; Relative errors for element (1,3).

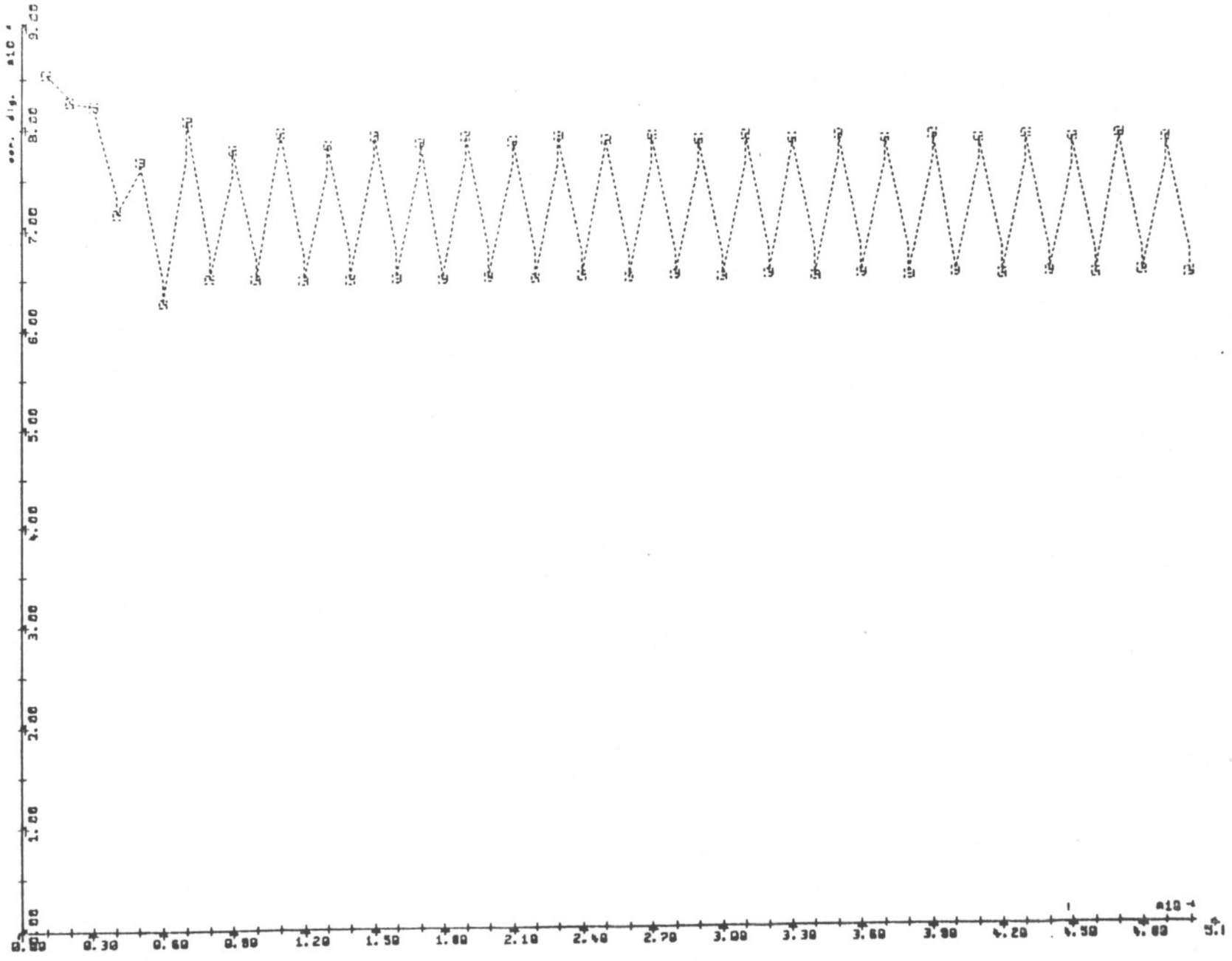


Figure 25 : Relative errors for element (2,3).

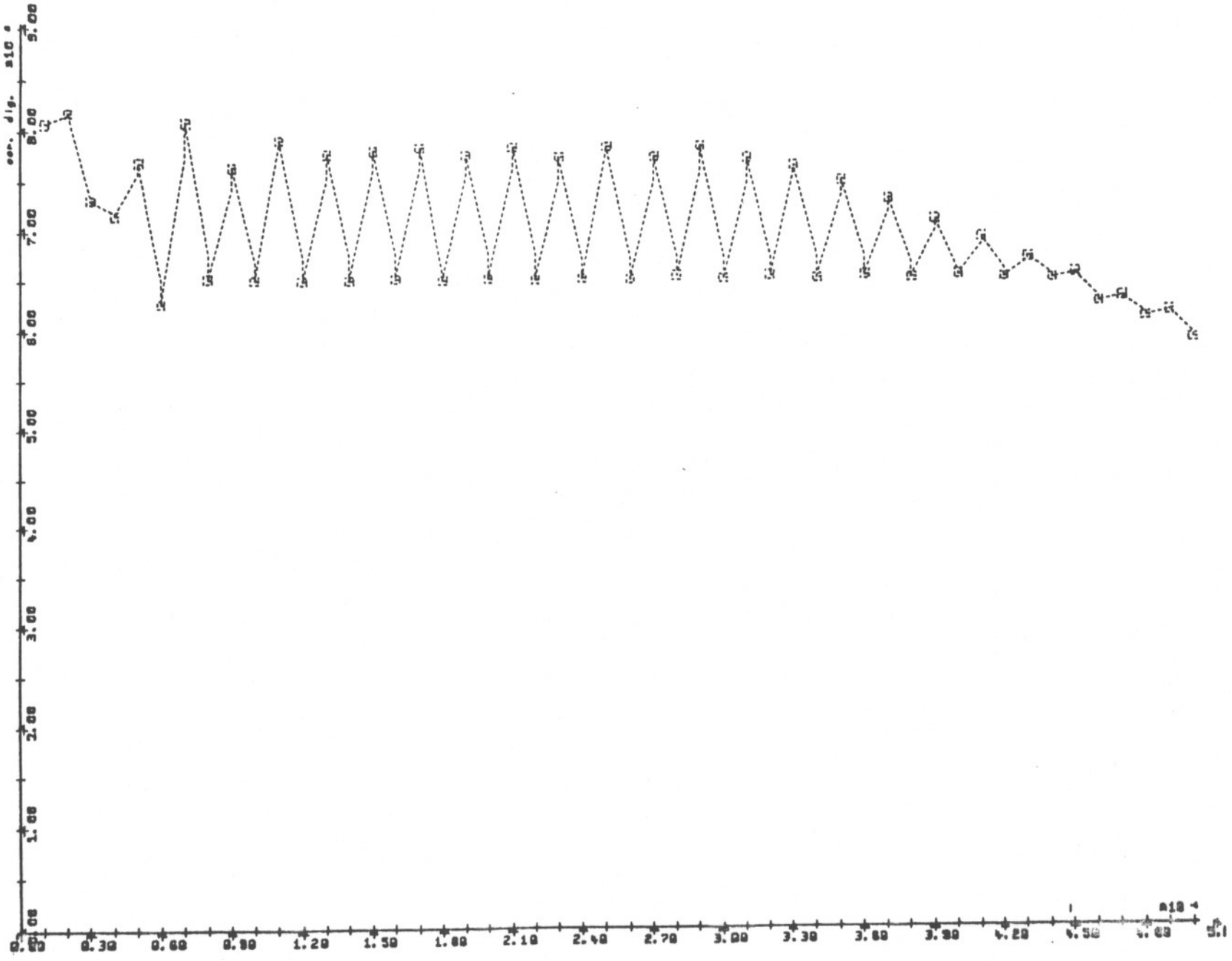


Figure 26 : Maximum of the relative errors.

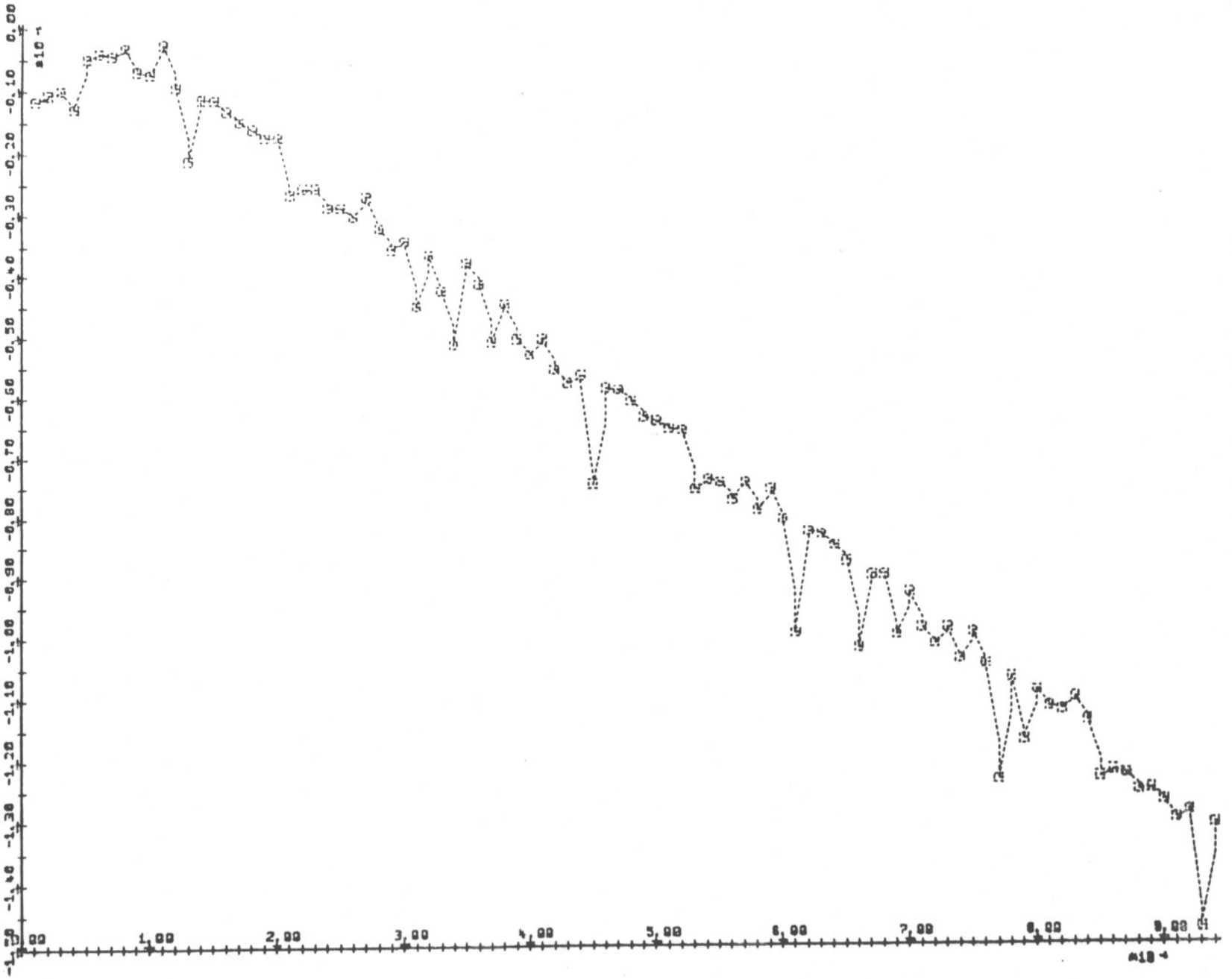


Figure 27 : Left singular vector corresponding to σ_1.

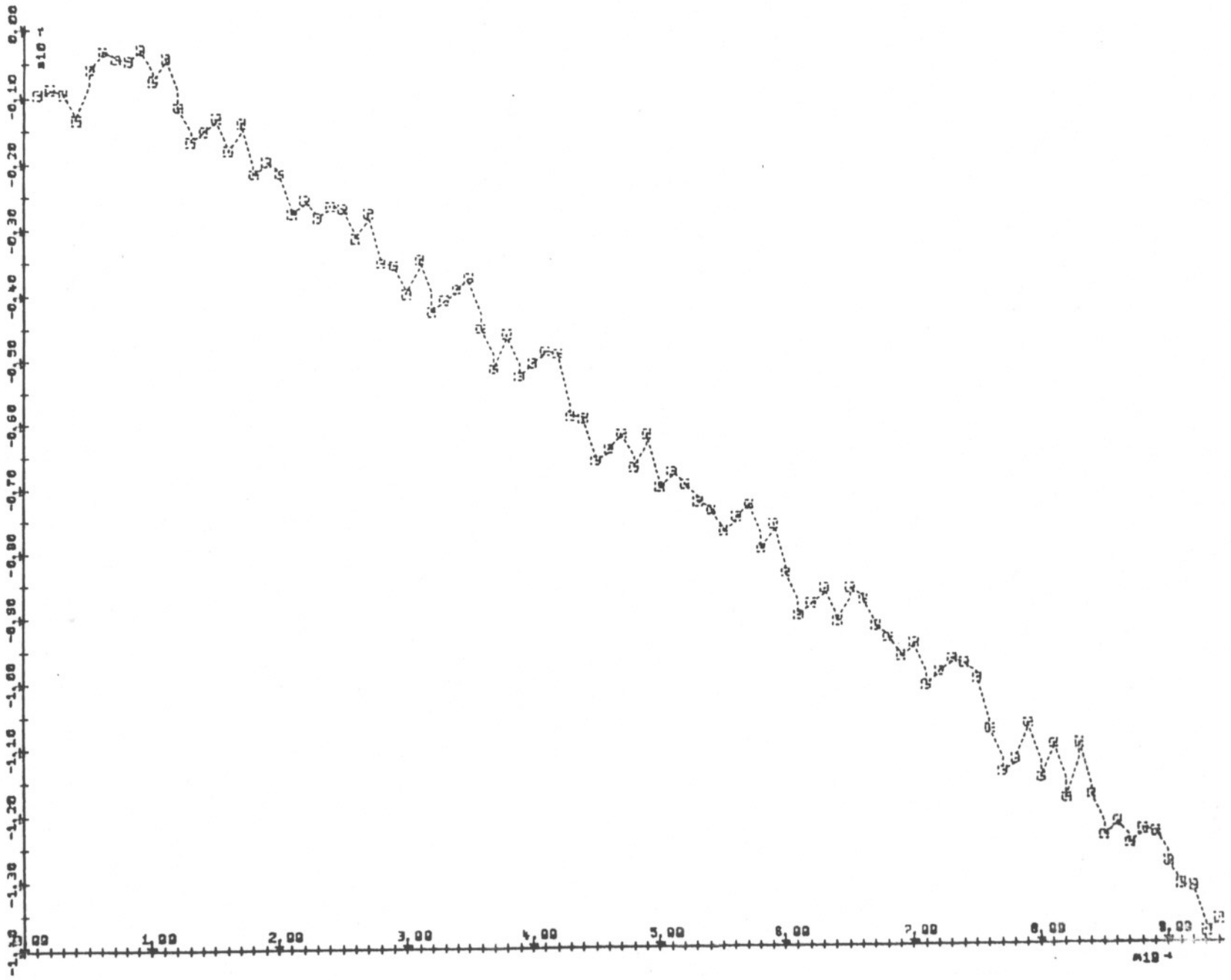


Figure 28 : Left singular vector corresponding to σ_2.

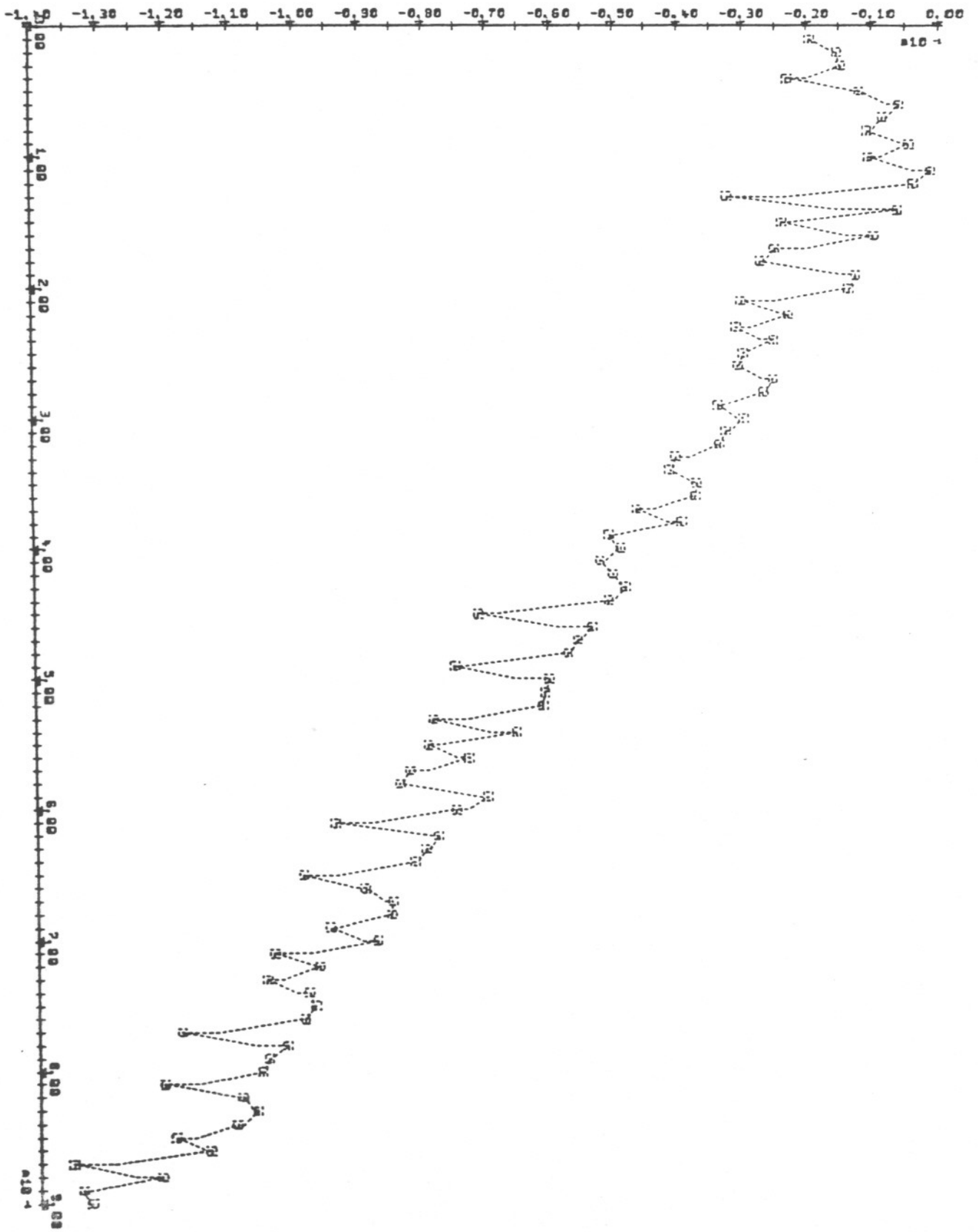


Figure 29 : Left singular vector corresponding to σ_2.

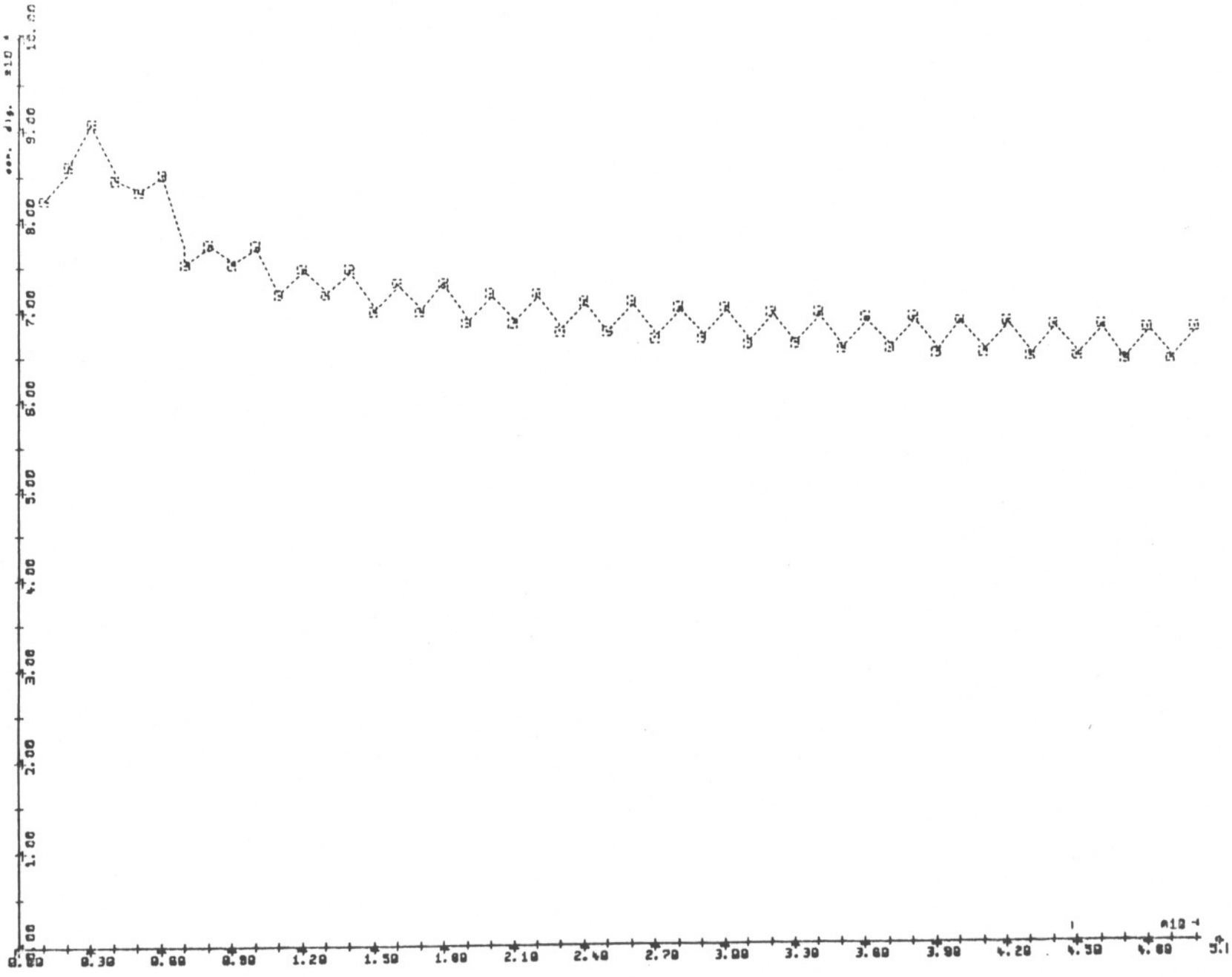


Figure 30 : Relative errors for the elements (1,1) and (1,3).

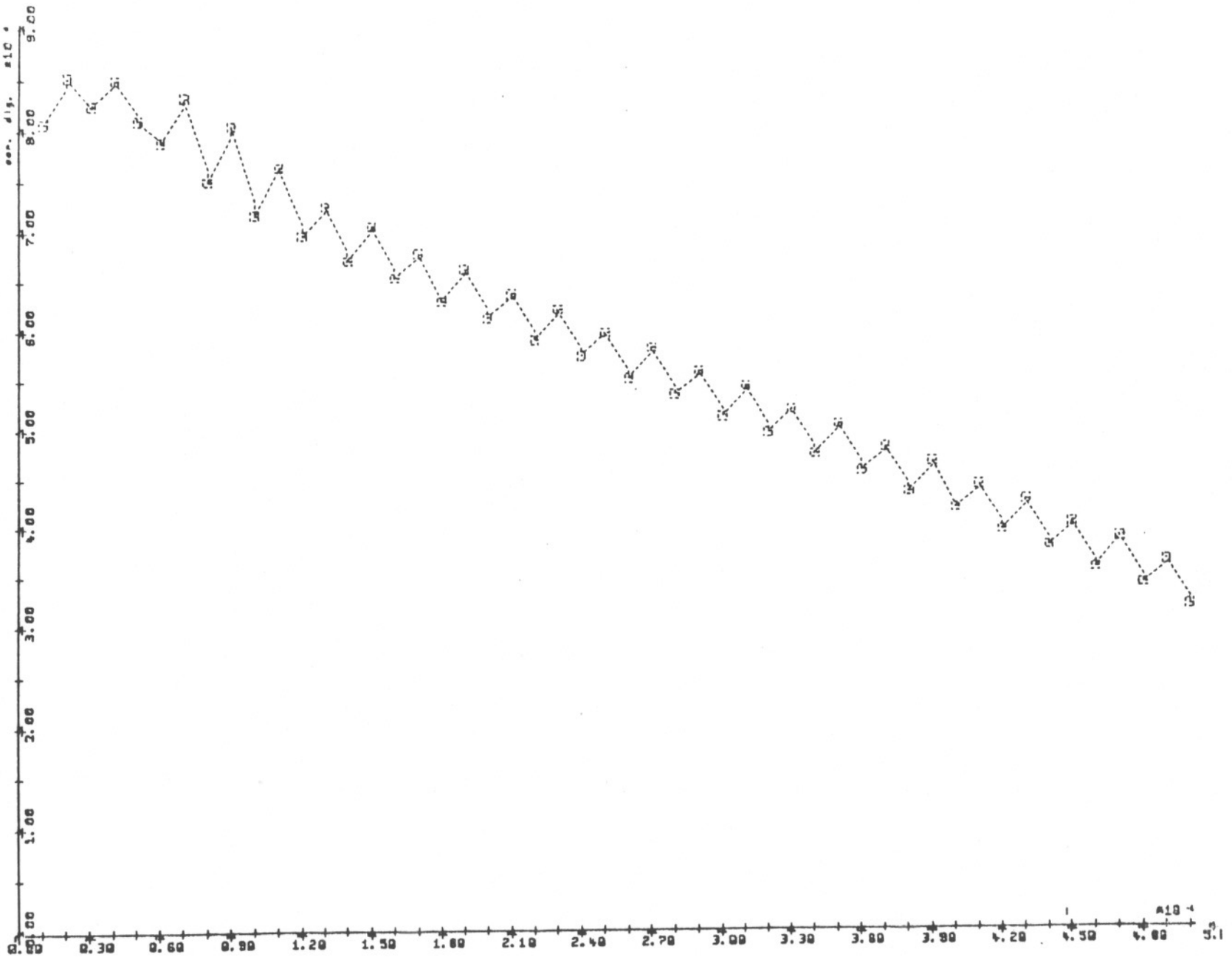


Figure 31 : Relative errors for the element (1,2).

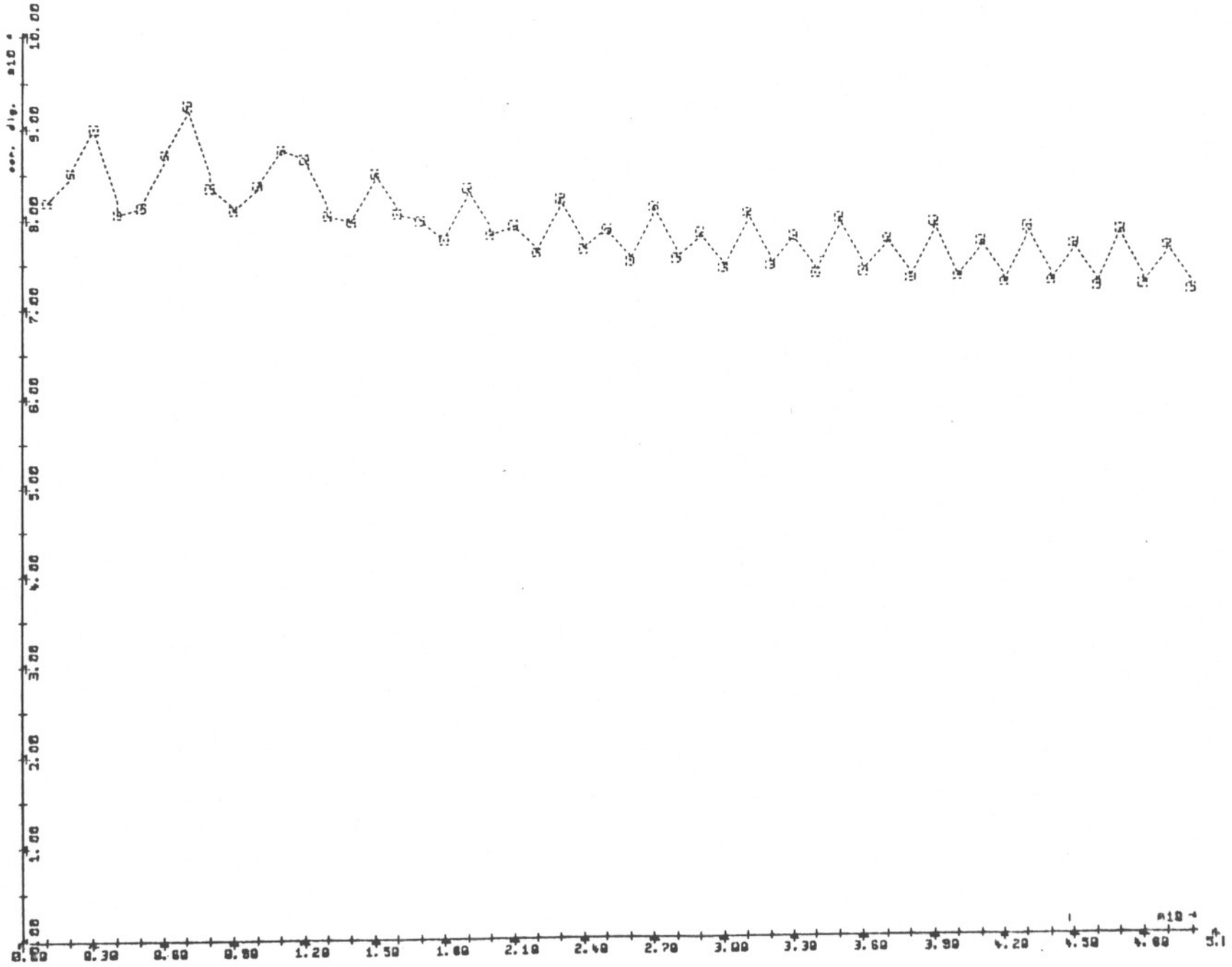


Figure 32 : Relative errors for the element (2,1).

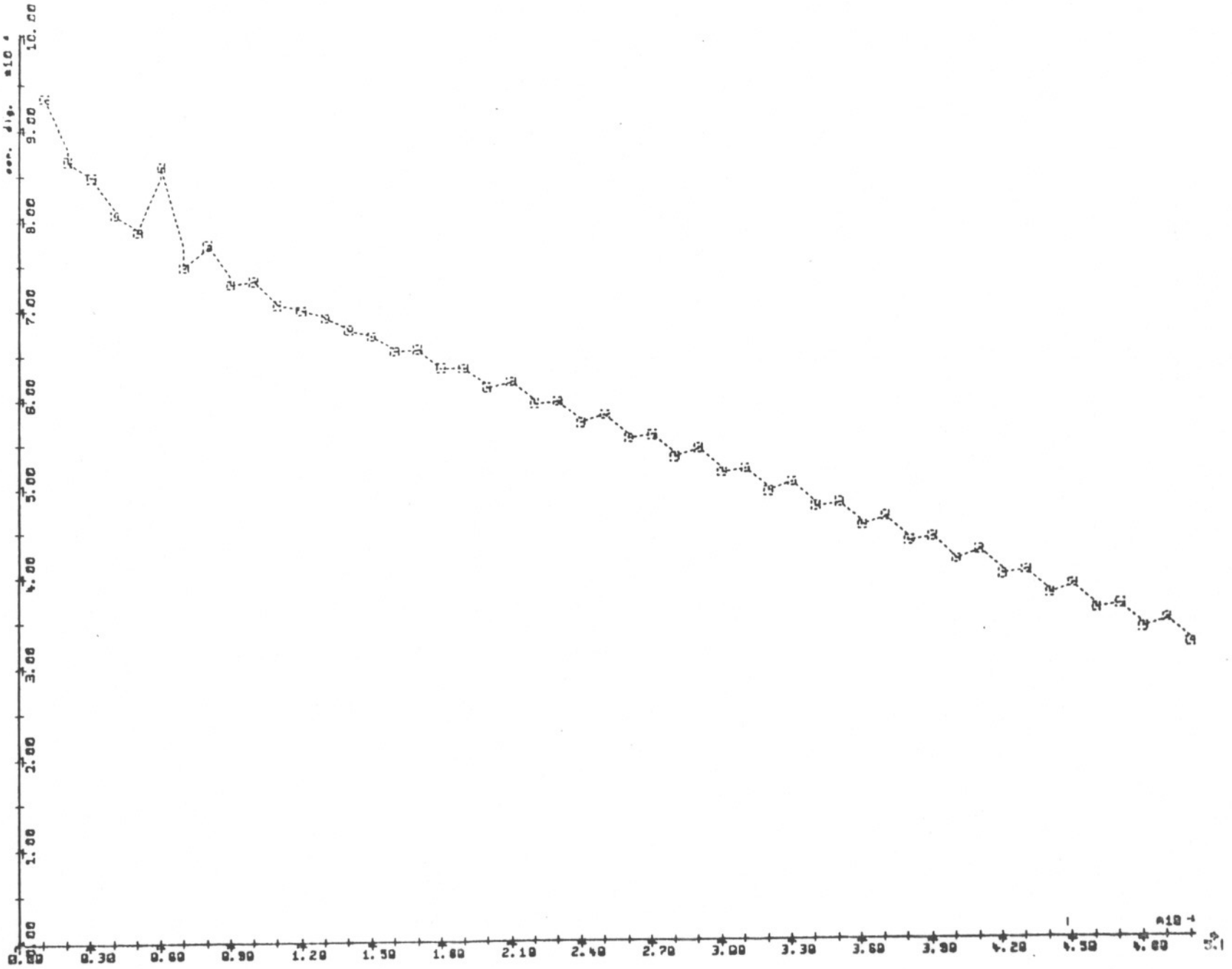


Figure 33 : Relative errors for the element (2,2).

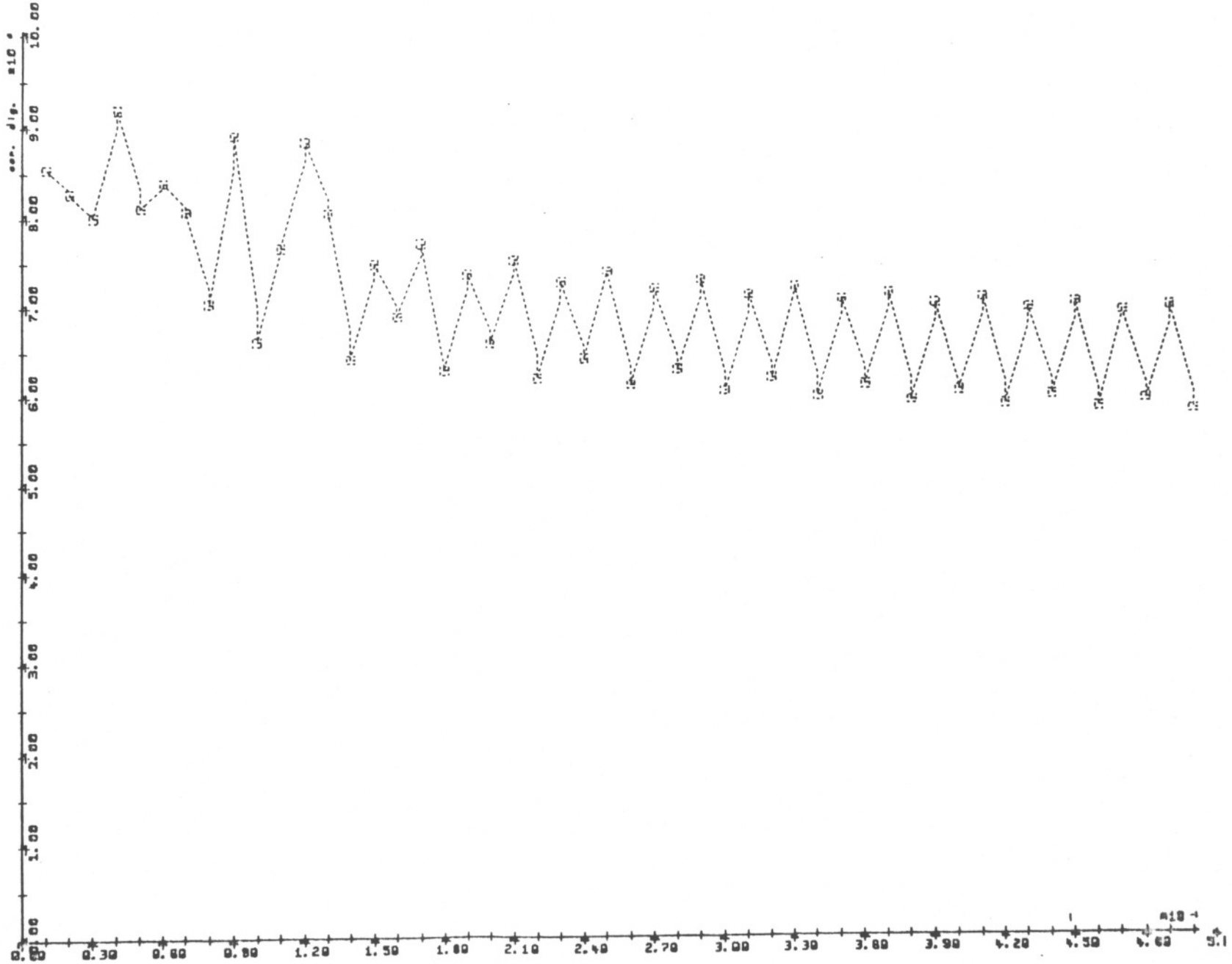


Figure 34 : Relative errors for the element (2,3).

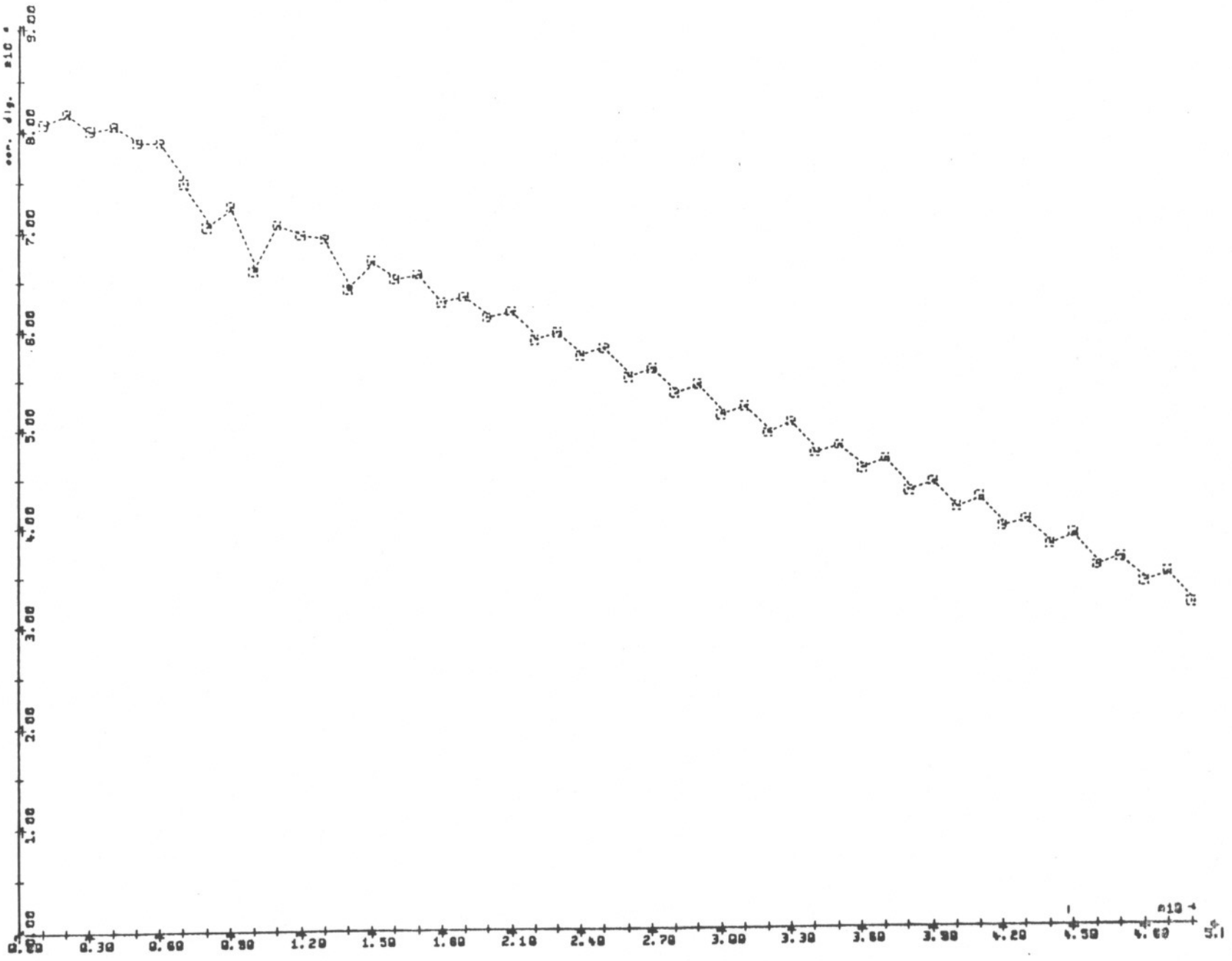


Figure 35 : Maximum of the relative errors.