Yugoslav Journal of Operations Research
4 (1994), Number 1, 79–103

COMPUTER MODELLING OF LANDSCAPES

Jelena PETRIĆ

ABACUS, Department of Architecture & Bldg. Science, University of Strathclyde, Glasgow, UK

Abstract: This paper presents two modelling techniques which help to increase the realism of the computer generated images of terrain and vegetation. The issues of realism, as related to modelling of natural objects in the landscape, are concerned with the texture of natural terrain and the richness of vegetation growth patterns.

The methods for modelling ground surfaces are basically divided into deterministic (i.e. polynomial) and stochastic methods. As opposed to polynomial methods, which provide for modelling of relatively smooth terrain surfaces, the stochastic methods, based on different classes of stochastic primitives, allow for the modelling of irregular (rough) terrain surfaces. This paper is particularly concerned with one such class of modelling primitives based on Iterated Function Systems (IFS).

The paper also lays out the background theory and rule specification, together with software means, for the modelling of various vegetation growth patterns.

Keywords: Fractal interpolation, iterated functions, affine maps, attractors, inhibitors.

1 INTRODUCTION

Industrialization of the countryside is causing significant changes to the visual quality of our landscapes. Recently some political will, in the form of and EC Council Directive, requires professionals such as planners, architects and engineers to make explicit the visual impact that their developments will have on the countryside. A number of computer based tools have been developed which provide assistance in visual impact analysis.

The system for Computer Aided Visual Impact Analysis (CAVIA) is devoted to the modelling of landform, vegetation and man–made objects such as buildings, transmission towers, bridges, dams, and similar structures. Man–made objects are all comparatively straight forward to model on a computer, even when the geometry is complex. When the model of a particular geometry has been created it can be viewed from any position and it can become part of a database which extends over a large area.

A computer model of landform can be created by digitising the contours from Ordnance Survey maps. From the digital terrain model (DTM) data it is possible to generate perspective views of the terrain and to analyse distant and intermediate horizons, and to generate maps which indicate the degree of visibility in the landscape of an object within it. New fractal techniques have been tried and tested, facilitating the addition of small scale texturing to the terrain model.

The geometry of vegetation is more complex. Mathematical models of the formal characteristics of a particular species of vegetation have been developed and it is already possible, in the CAVIA system, to simulate the growth pattern and seasonal variation of individual trees and of complete forest plantations.

2. REALISTIC TERRAIN MODELLING USING ITERATED FUNCTION SYSTEMS (IFS)

In standard visual simulations it is required to mix computer models of man-made objects with computer models of terrain and vegetation. When modelling a landform the first step is to input the terrain data into the modelling system. Generally, this involves placing and Ordnance Survey map on a digitising table, establishing the origin point and defining a small number of technical details. The operator then traces the contour lines with the 'mouse' unit; this samples the contours and reproduces them on an adjacent display screen. If an aerial survey of the terrain already exists or if data is available from LANDSAT satellite survey, the map digitising stage is not required, as the necessary data can be input directly. It is possible to combine terrain data from various sources into the final terrain model, although care is necessary in establishing consistent control points between the various survey. The raw contour data is then interpolated into a 'grid' format.

In order to achieve realism in the graphical representation of a terrain, different methods can be used depending on the shape and the complexity of the terrain surface. For terrains with pronounced irregularities, realism in graphical representation can be achieved by using new methods of fractal interpolation. Fractals are a mathematical concept which has significant potential for realistic modelling of natural terrain surfaces. Fractals have the desirable characteristic that only a little information is needed for their specification and a small increase in the specification increases the visual complexity significantly. Natural terrain exhibits the property that as one views its surface at greater magnification more and more structure is revealed. Fractal sets exhibit exactly this property; they manifest a high degree of visual complexity since there is no limit to their structure.

In this research a fractal based method of Iterated Function Systems (IFS) was used in order to achieve realism in fine surface texturing. In effect, the IFS based fractal interpolation is extended from planar definition [2] to 3D, allowing one to generate apparently different surface textures and to control their roughness by minor variations of input parameters.

2.1. Fractal Interpolation Functions in 3D

IFS are determined by a set of linear maps ω_n (affine transformations of d-dimensional space into itself) and an associated set of probabilities P_i. For a given set of co-planar points:

$$f(x_i) = y_i, \qquad i = 1, 2, \ldots, N; \tag{1}$$

and in the interval $I = [x_0, x_n]$, for which the function $F(x_i)$ is given, a set of affine maps ω_n can be derived from the following equations:

$$L_n : I - I_n,$$

where

$$I_n = [x_{n-1}, x_n]$$

is such that

$$L_n(x_0) = x_{n-1} \; ; \; L_n(x_N) = x_n \tag{2}$$

and mapping F_n is such that

$$F_n(x_0, y_0) = y_{n-1} \; ; \; F_n(x_N, y_N) = y_n \tag{3}$$

The above conditions can be satisfied by the system of linear fractal interpolation functions, derived from affine maps:

$$\begin{aligned} L_n(x) &= x_{n-1} + (x_n - x_{n-1})(x - x_0) / (x_N - x_0) \\ F_n(x, y) &= b_n x + \alpha_n y + k_n , \end{aligned} \tag{4}$$

where constants b_n and k_n are selected so that they satisfy the condition (3):

$$F_n(x_0, y_0) = y_{n-1} \text{ and } F_n(x_N, y_N) = y_n,$$

and $\alpha_n \in [-1, 1]$ is a parameter which controls dispersion of points around a smooth curve defined by $f(x)$. From the above conditions we get:

$$\begin{aligned} b_n &= [y_n - y_{n-1} - \alpha_n (y_N - y_0)] / (x_N - x_0) \\ k_n &= y_{n-1} - \alpha_n y_0 - b_n x_0 . \end{aligned} \tag{5}$$

If a system of mapping functions is defined so that ω_n is following:

$$\omega_n(x, y) = [L_n(x), F_n(x, y)], \tag{6}$$

then ω_n is a set of IFS maps for which it was proved [2] that it has one unique attractor G which in itself represents a continuous fractal function $f(x)$.

The method used to generate a set of mapping functions ω_n in 2D was in this research extended to 3D surface modelling using fractal interpolation functions.

A set of affine maps (4) in 2D is similar to set of maps in 3D space. The 3D set of maps is given for the surface points $z = z(x_n, y_m)$, $n = 1, 2, \ldots, N$; $m = 1, 2, \ldots, M$, and can be defined as following:

$$L_n(x_1) = x_{n-1} \; ; \; L_n(x_N) = x_n \qquad (7)$$
$$V_m(y_1) = y_{m-1} \; ; \; V_m(y_M) = y_m;$$

The following are set of affine mappings associated with linear fractal interpolation functions:

$$L_n(x) = x_{n-1} + (x_n - x_{n-1})\,(x - x_1) \,/\, (x_N - x_1) \qquad (8)$$
$$V_m(y) = y_{m-1} + (y_m - y_{m-1})\,(y - y_1) \,/\, (y_M - y_1).$$

By analogy with 2D mapping function F_n, a mapping function in 3D is defined as:

$$F_{n,m}(x, y, z) = a_{n,m}\,x + b_{n,m}\,y + \alpha_{n,m}\,z + c_{n,m} \qquad (9)$$

so that the following conditions are satisfied:

$$\begin{aligned} F_{n,m}(x_1, y_1, z(x_1, y_1)) &= \\ &a_{n,m}\,x_1 + b_{n,m}\,y_1 + \alpha z(x_1, y_1) + c_{n,m} = z(x_{n-1}, y_{m-1}) \\ F_{n,m}(x_N, y_1, z(x_N, y_1)) &= \\ &a_{n,m}\,x_N + b_{n,m}\,y_1 + \alpha z(x_N, y_1) + c_{n,m} = z(x_n, y_{m-1}) \qquad (10) \\ F_{n,m}(x_N, y_M, z(x_N, y_M)) &= \\ &a_{n,m}\,x_N + b_{n,m}\,y_M + \alpha z(x_N, y_M) + c_{n,m} = z(x_{n-1}, y_m) \end{aligned}$$

By solving a system of linear Equations (1) for each mapping ($n = 1, 2, \ldots, N$; $m = 1, 2, \ldots, M$), and for a given parameter α, we get the unknown mapping parameters $a_{n,m}$, $b_{n,m}$, $c_{n,m}$.

The mapping parameters obtained from the system of Equations (1) are:

$$\begin{aligned} a_{n,m} &= D_1 / D \\ b_{n,m} &= D_2 / D \qquad (11) \\ c_{n,m} &= D_3 / D, \end{aligned}$$

where matrices D_1, D_2, D_3, D :

$$D_1 = \begin{bmatrix} P & y_1 & 1 \\ Q & y_1 & 1 \\ S & y_M & 1 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} x_1 & P & 1 \\ x_N & Q & 1 \\ x_N & S & 1 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} x_1 & y_1 & P \\ x_N & y_1 & Q \\ x_N & y_M & S \end{bmatrix}$$

$$D = \begin{bmatrix} x_1 & y_1 & 1 \\ x_N & y_1 & 1 \\ x_N & y_M & 1 \end{bmatrix}$$

and

$$P = z(x_{n-1}, y_{m-1}) - \alpha_{n,m} Z(x_1, y_1)$$
$$Q = z(x_n, y_{m-1}) - \alpha_{n,m} Z(x_N, y_1)$$
$$S = z(x_n, y_m) - \alpha_{n,m} Z(x_N, y_M) .$$

A set of 3D mapping functions $\omega_{n,m}(x, y, z)$, $n = 1, 2, \ldots ,N$; $m = 1, 2, \ldots, M$, formed by analogy with a set of maps (6), is defined as following:

$$\omega_{n,m}(x, y, z) = [\, L_n(x), V_m(y), F_{n,m}(x, y, z) \,] . \tag{12}$$

According to IFS method a geometry of an attractor is defined by the mapping function (12), while the configuration (i.e. density) of the interpolated surface points is controlled by a set of probabilities $R_{n,m}$. Parameter α controls the dispersion of the interpolated points, and in effect defines surface roughness.

2.2. IFS Algorithm for Fractal Surface Interpolation

Terrain surface points are interpolated using a set of maps $\omega_{n,m}$ (12), which are essentially derived from the IFS method. The algorithm for fractal interpolation is shown in Figure 1.

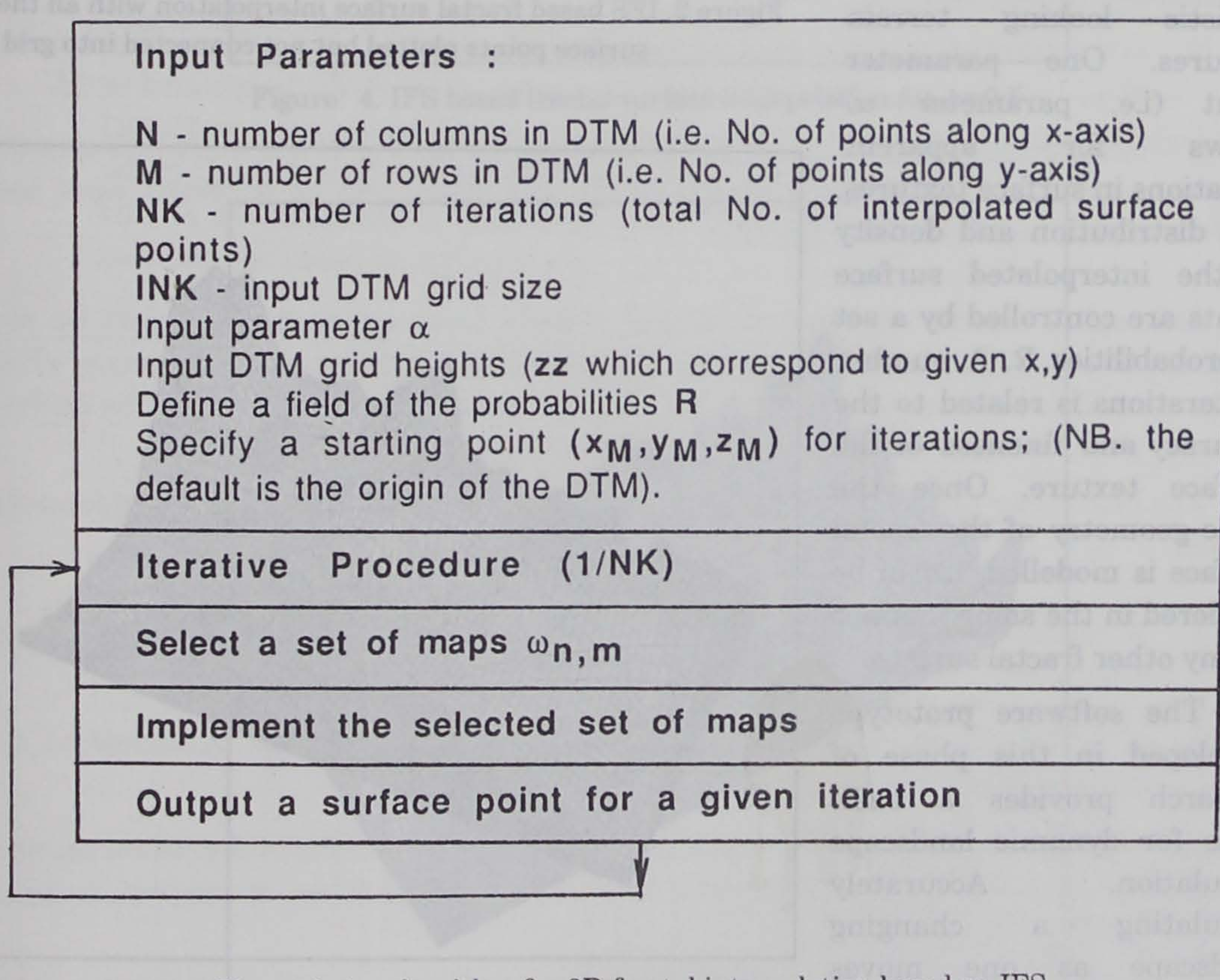

Figure 1. An algorithm for 3D fractal interpolation based on IFS

From the Figure 1 follows that each *NK* iteration interpolates one surface point (x, y, z). These interpolated points can either be plotted directly (see Figure 2) or connected into a surface grid (see Figures 3, 4 and 5). An example of a segment of realistic terrain data is given showing fractal interpolation based on the IFS's which was used to generate different surface textures.

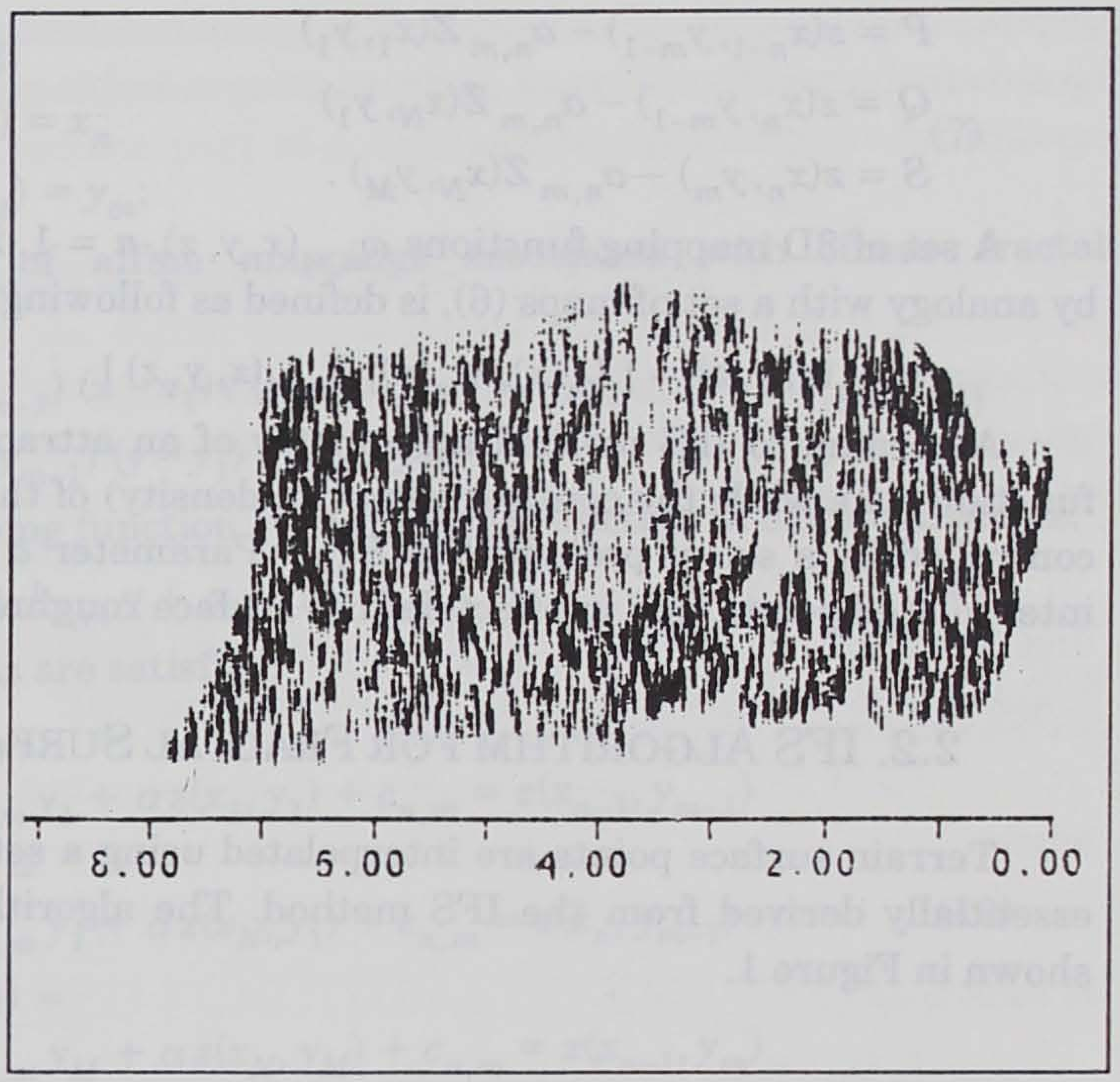

Figure 2. IFS based fractal surface interpolation with all the surface points plotted but not connected into grid

The method of fractal interpolation based on Iterated Function Systems (IFS) was used to produce realistic looking terrain textures. One parameter input (i.e. parameter α) allows for apparent variations in surface textures. The distribution and density of the interpolated surface points are controlled by a set of probabilities R. A number of iterations is related to the accuracy and 'fineness' of the surface texture. Once the basic geometry of the fractal surface is modelled, it can be rendered in the same manner as any other fractal surface.

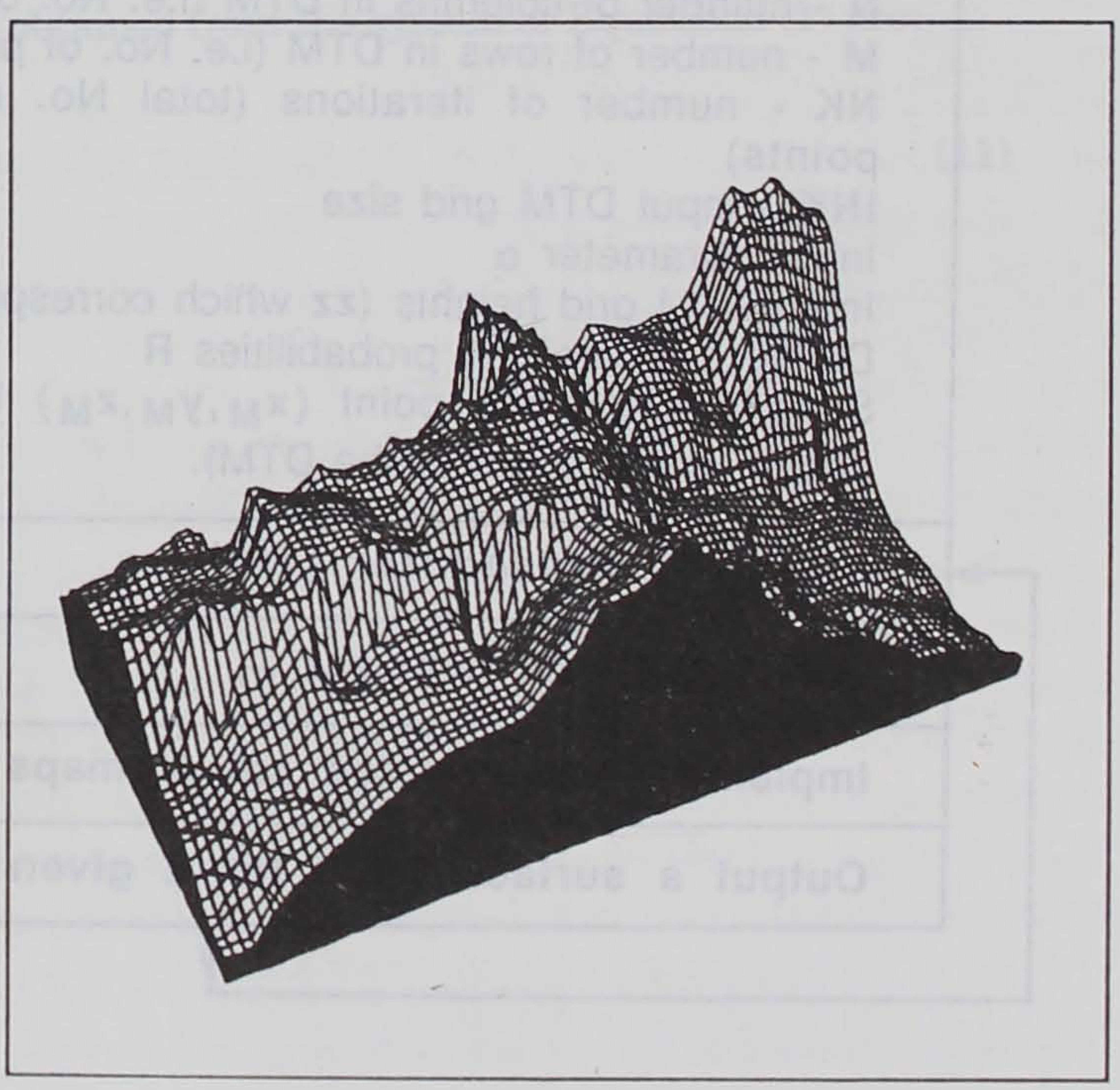

Figure 3. IFS based fractal surface interpolation for α=0.1

The software prototype developed in this phase of research provides a basic tools for dynamic landscape simulation. Accurately simulating a changing landscape as one moves through it in real time, has major implications in landscape planning and management and it has been used as useful technique for the visual analysis of man-made objects in rural landscapes.

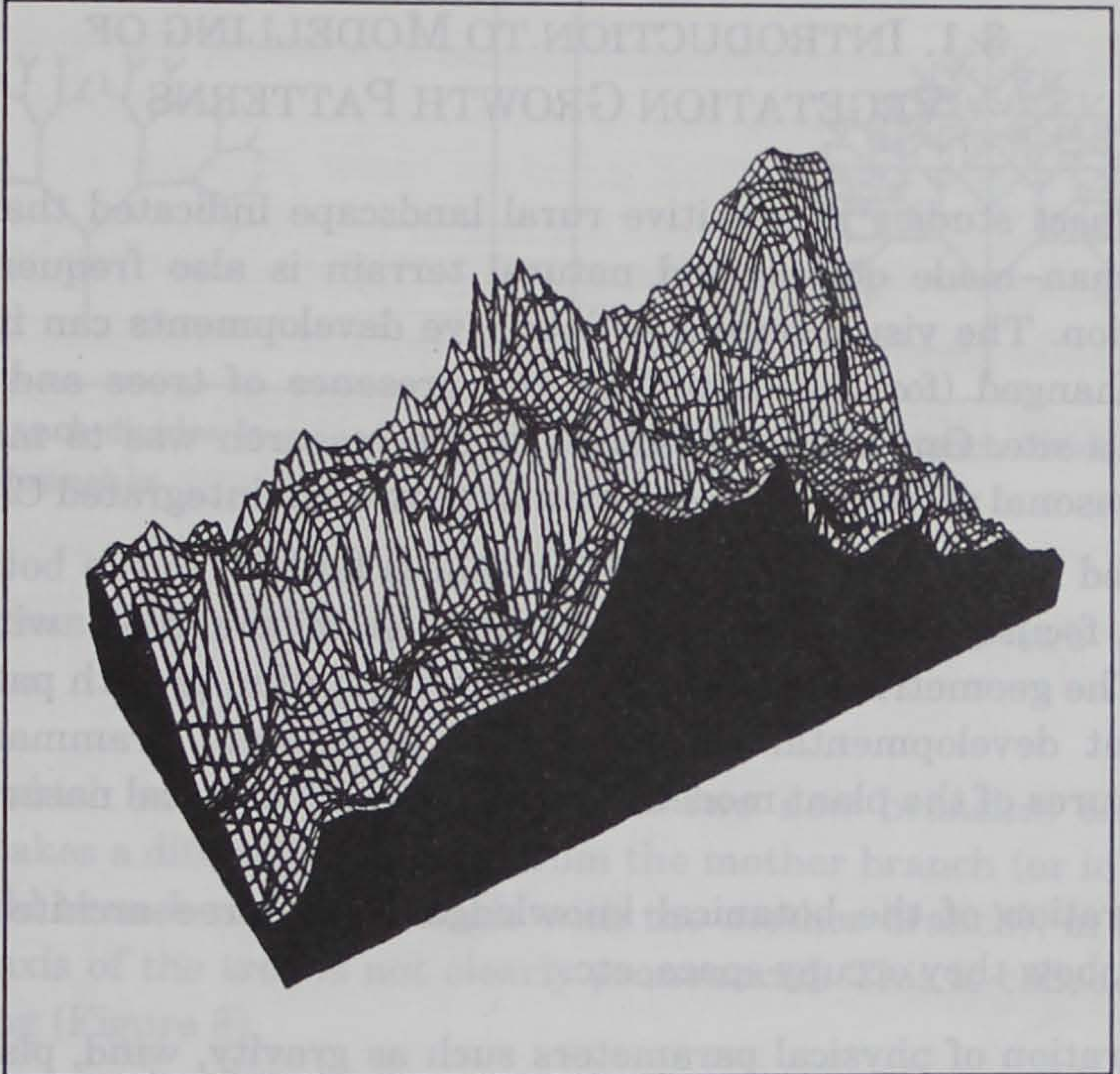

Figure 4. IFS based fractal surface interpolation for α=0.5

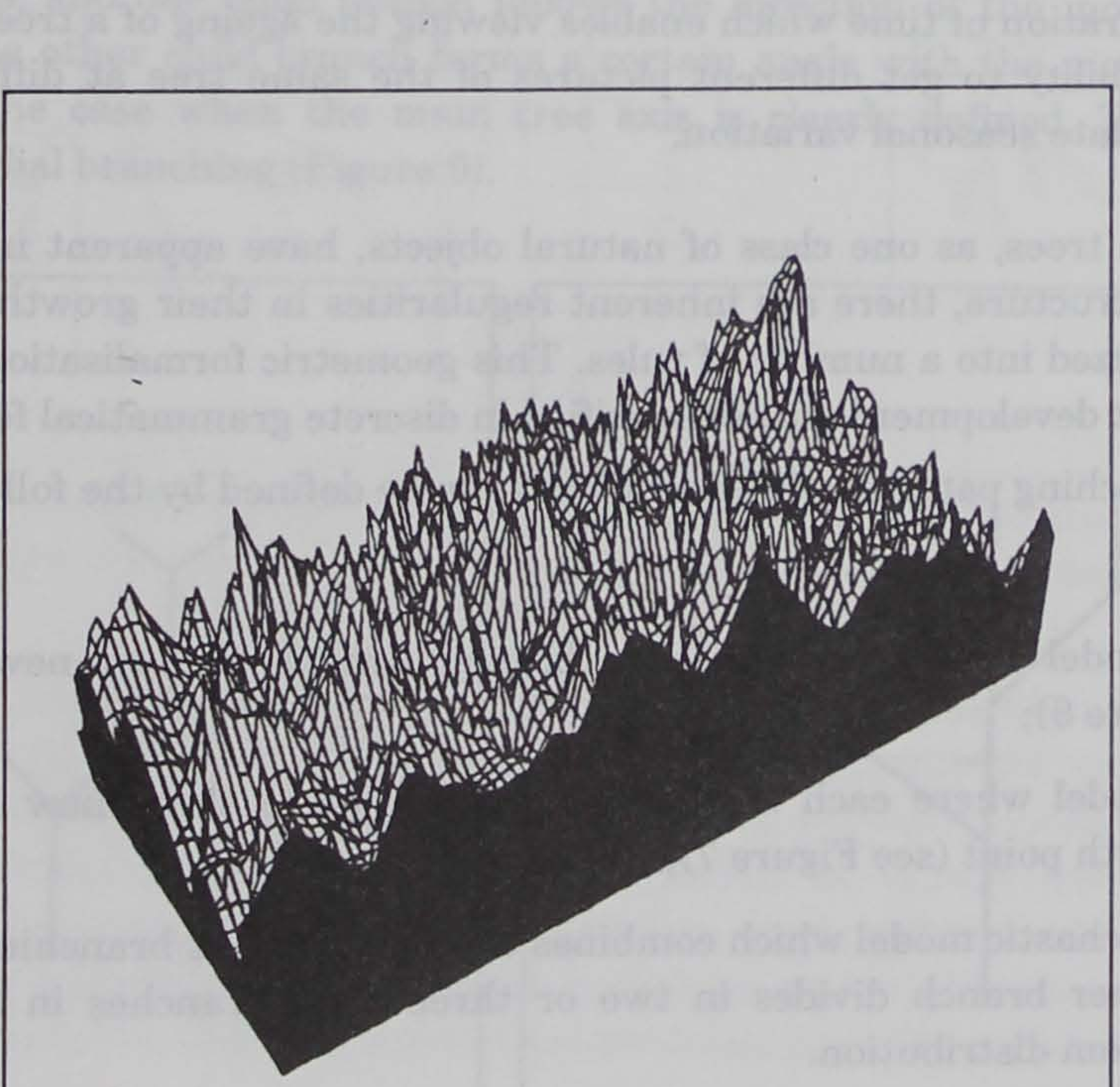

Figure 5. IFS based fractal surface interpolation for α=0.9

3. COMPUTER MODELLING OF VEGETATION

3.1. INTRODUCTION TO MODELLING OF VEGETATION GROWTH PATTERNS

Visual impact studies in sensitive rural landscape indicated that in addition to modelling of man-made objects and natural terrain is also frequently required to model vegetation. The visual impact of intrusive developments can in some cases be significantly changed (for the better) by the presence of trees and other forms of vegetation on a site. One of the objectives of the research was to model the growth pattern and seasonal variation of vegetation as part of an integrated CAVIA system.

An adopted mathematical model is one that is faithful to the botanical nature of trees [1] and focuses on the branching patterns of the trees with emphasis on morphology. The geometric formalization of the vegetation growth patterns is derived from the plant developmental rules specified in discrete grammatical form. The important features of the plant models faithful to their botanical nature are:

a) integration of the botanical knowledge of the tree-architecture: how they grow, how they occupy space, etc.

b) integration of physical parameters such as gravity, wind, plantation density, etc.

c) integration of time which enables viewing the ageing of a tree. It includes the possibility to get different pictures of the same tree at different ages, and simulate seasonal variation.

Although trees, as one class of natural objects, have apparent irregularities and fuzziness of structure, there are inherent regularities in their growth patterns which can be formalized into a number of rules. This geometric formalisation can be derived from the plant developmental rules specified in discrete grammatical form.

The branching patterns of higher plants can be defined by the following geometric models:

1. a model where each mother branch divides in two new branches (see Figure 6);
2. a model where each mother branch divides in three new branches at the growth point (see Figure 7);
3. a stochastic model which combines the two previous branching patterns; each mother branch divides in two or three child branches in respect to some random distribution.

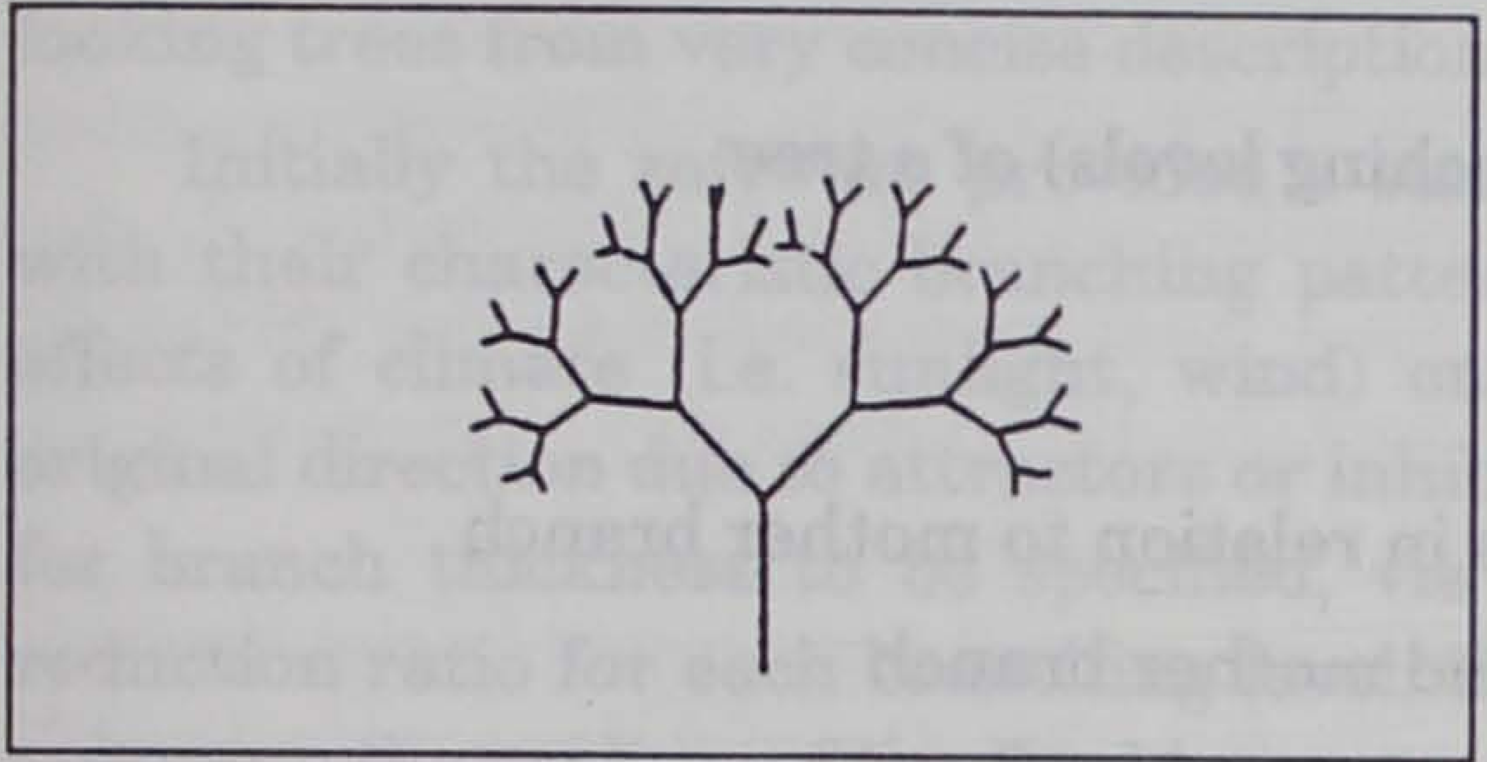

Figure 6. Mother branch divides in two child branches

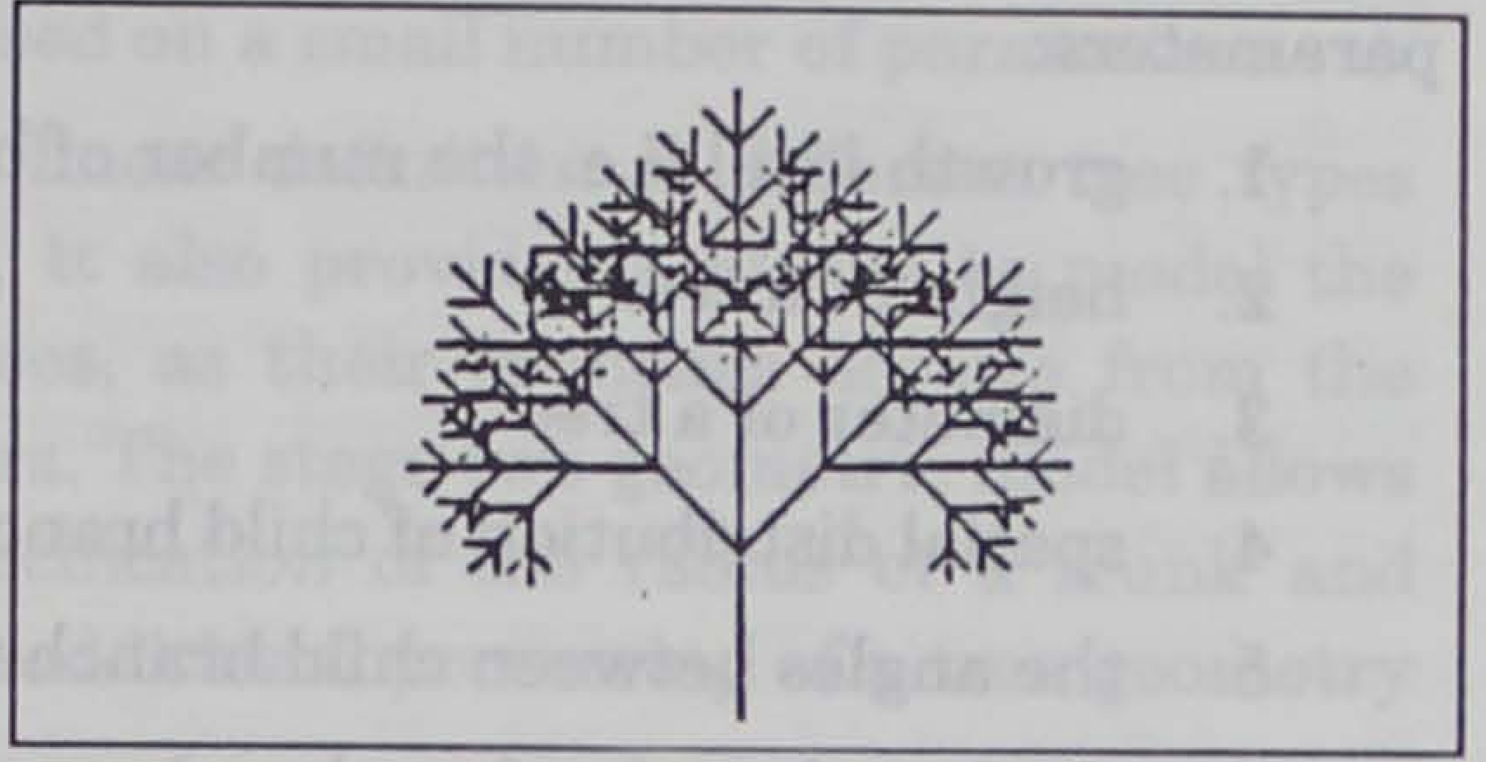

Figure 7. Mother branch divides in three child branches

The first geometric model can be subdivided further in two categories of branching patterns:

1a) one in which mother branch divides in two new branches and each child branch takes a different direction from the mother branch (or in other words, each child branch makes an angle with the mother branch); in this case, the central axis of the tree is not clearly pronounced. This is called dichotomous branching (Figure 8),

1b) a branching pattern in which each mother branch divides in two new branches, and one child branch follows the direction of the mother branch, while the other child branch forms a certain angle with the mother branch; this is the case when the main tree axis is clearly defined. This is called monopodial branching (Figure 9).

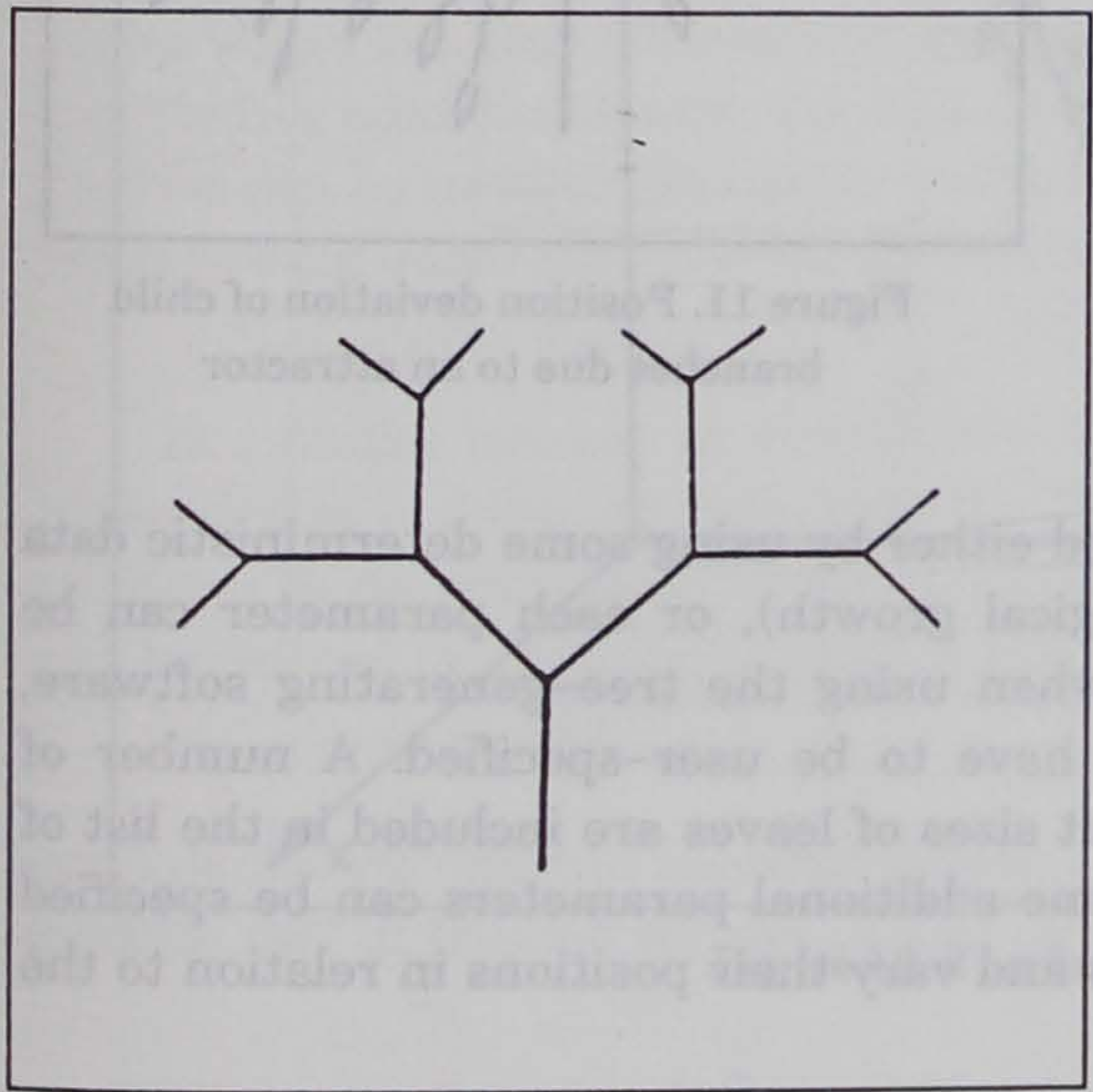

Figure 8. Dichotomous branching

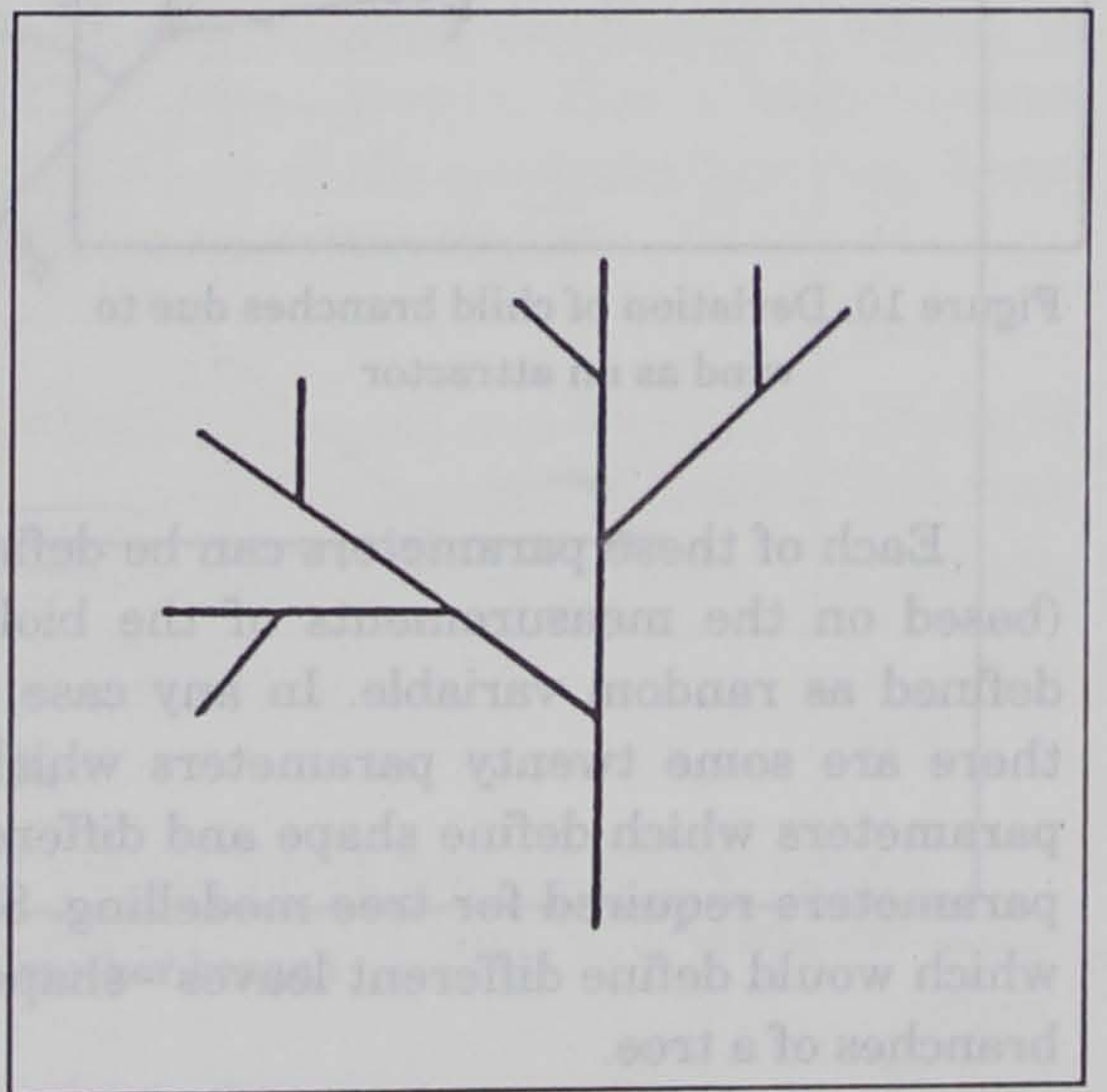

Figure 9. Monopodial branching

Mathematical modelling of different tree types is based on the following set of parameters:

1. growth level (i.e. the number of branching levels) of a tree
2. height of a tree
3. diameter of a tree
4. spatial distribution of child branches in relation to mother branch
5. the angles between child branches and mother branch
6. the length ratio of mother–branch to child–branch
7. divergence angle (which defines the arrangement of branches around the main axis)
8. position–deviations of child branches due to some attractors, such as sunlight or wind (see Figures 10 and 11).

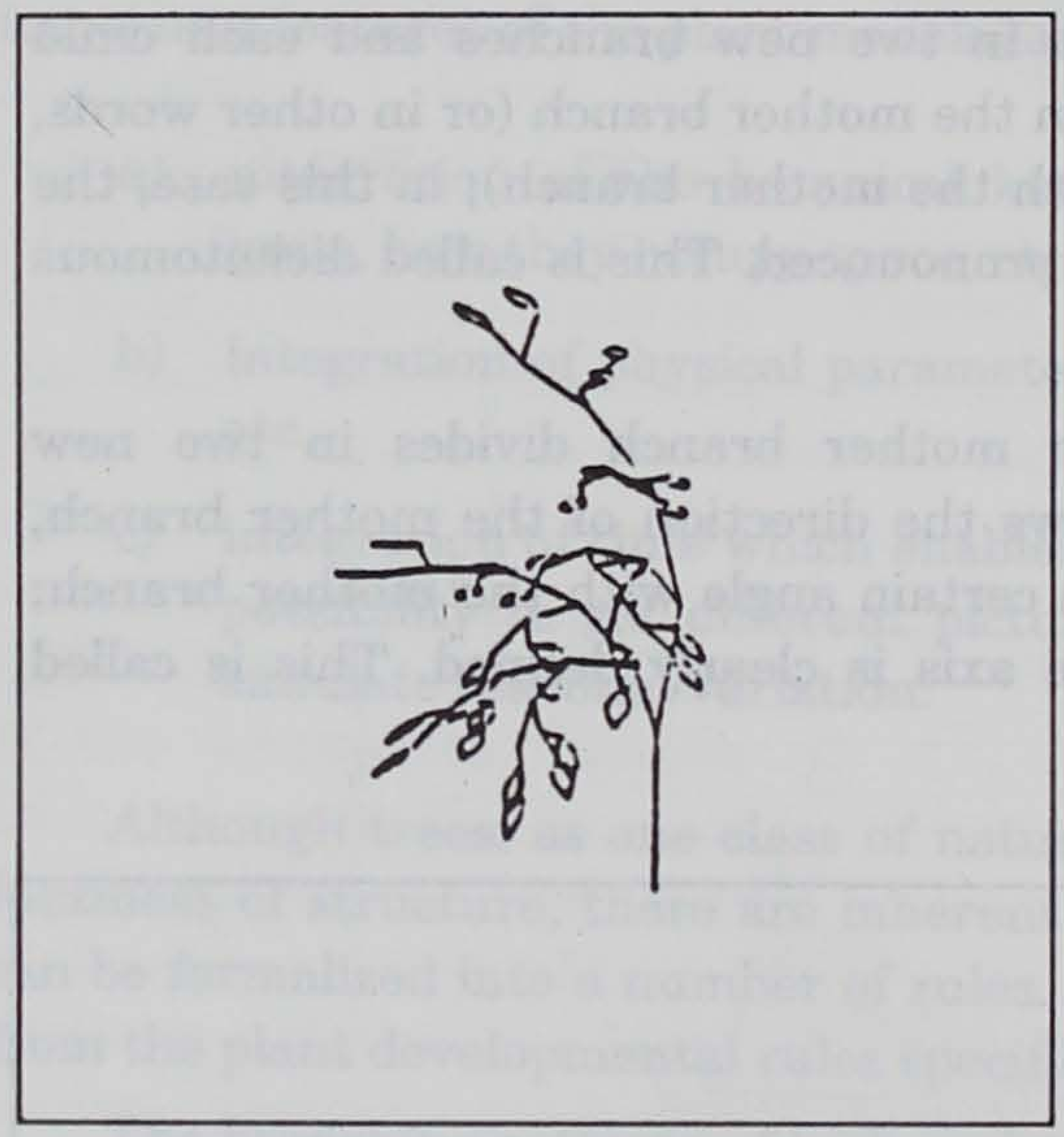

Figure 10. Deviation of child branches due to wind as an attractor

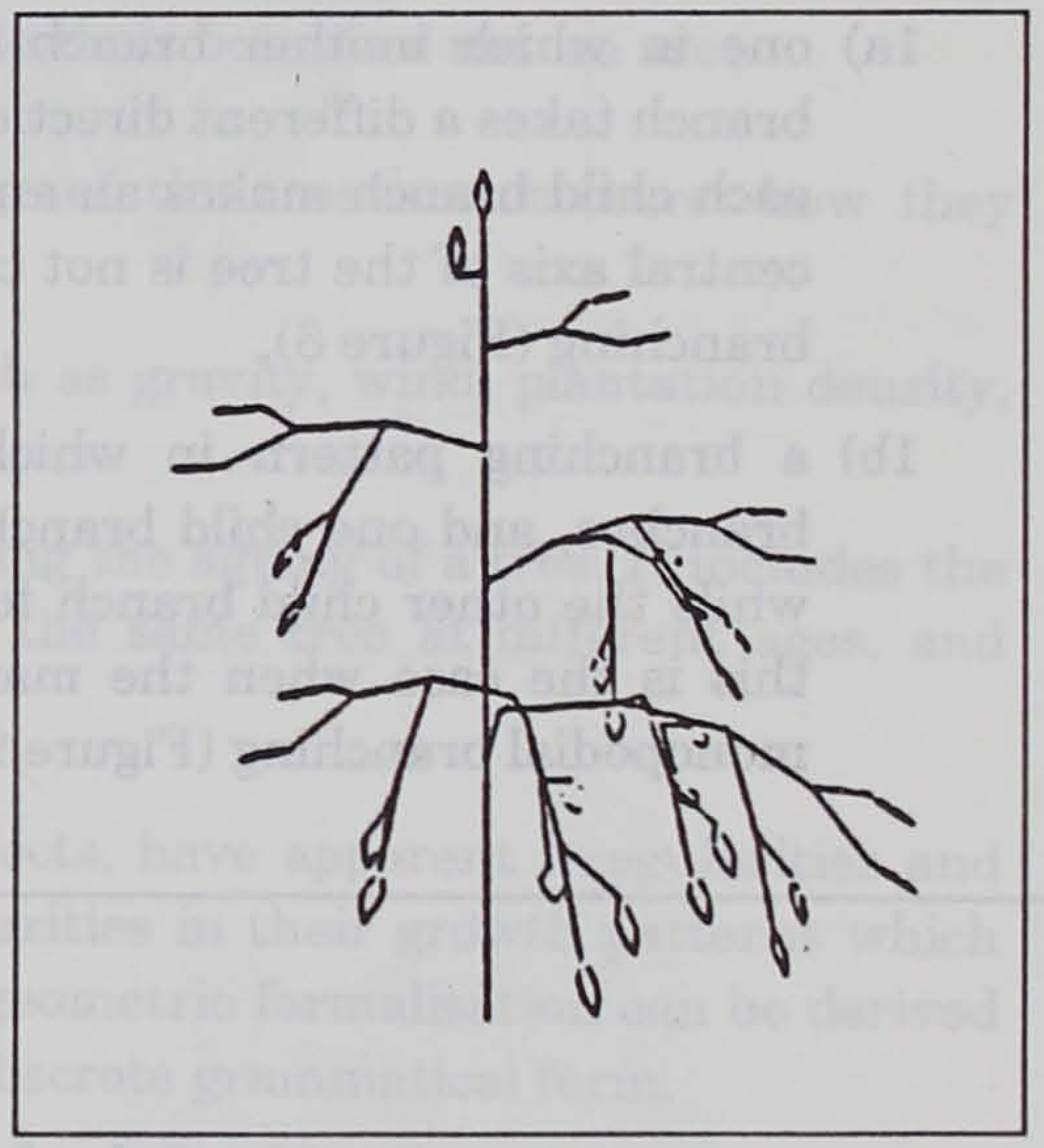

Figure 11. Position deviation of child branches due to an attractor

Each of these parameters can be defined either by using some deterministic data (based on the measurements of the biological growth), or each parameter can be defined as random variable. In any case, when using the tree–generating software, there are some twenty parameters which have to be user–specified. A number of parameters which define shape and different sizes of leaves are included in the list of parameters required for tree modelling. Some additional parameters can be specified which would define different leaves' –shapes and vary their positions in relation to the branches of a tree.

A software prototype has been developed which can model different tree–growth patterns. One of the most distinctive features of this approach to realistic tree modelling is the data base amplification. This term refers to the generation of complex looking trees from very concise description based on a small number of parameters.

Initially the software provided a simple linear definition of different tree types with their characteristic branching patterns. It also provides a means to model the effects of climate (i.e. sunlight, wind) on trees, as their branches deviate from the original direction due to attractors or inhibitors. The stage two geometric model allows for branch thickness to be specified, via specification of the radius of a trunk and reduction ratio for each branching level. This additional processing of a tree geometry enhances the realism of the final image.

3.2. Mathematical Basis for Geometric Modelling of a Tree

A tree branching pattern is defined geometrically as a set of linear segments (in 3D space) which represent branches of a tree. The first mother branch, or the tree trunk, is defined by vertices P_a (root) and P_b (tip of a mother branch) - see Figure 12. The coordinates of these vertices are defined in the static orthogonal coordinate system x, y, z (Figure 12), where the starting point and the tip of the branch are represented as: $P_a(x_a, y_a, z_a)$ and $P_b(x_b, y_b, z_b)$. The coordinate system x, y, z is defined in such a way that axis O_z is parallel with a line segment $P_a P_b$ which represents a tree trunk (Figure 12); the axes O_x and O_y form the left-hand coordinate system.

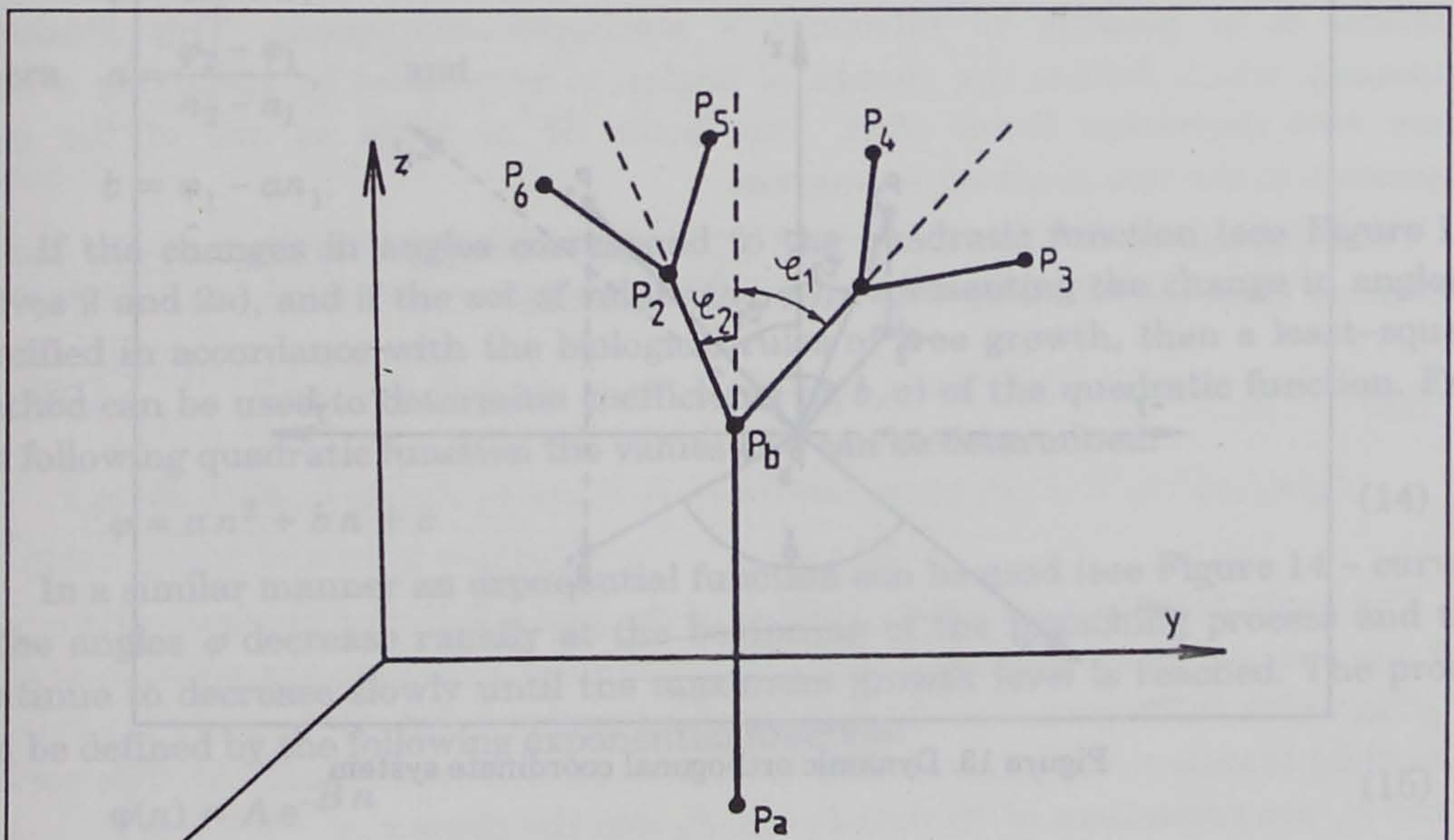

Figure 12. The first mother branch

The branching on the first level of a tree growth is defined by the length of child branches ($P_b P_1$ and $P_b P_2$) and by the angles (φ_1 φ_2) between child branches and the mother branch. These parameters determine the coordinates of the vertices P_1 and P_2.

On any subsequent growth level (P_3, P_4, P_5, P_6) the branching of a tree is defined in the same manner as on the first branching level. A branching pattern of a tree is defined as an iterative procedure, where the tip of a branch from the lower branching level (P_b) is a starting point for the next higher branching level. The vertex P_1, for example, which appears in the second branching level corresponds to the vertex P_b from the first branching level ($P_1 \Rightarrow P_b$); and in the same way the vertex P_b corresponds to the vertex P_a ($P_b \Rightarrow P_a$).

The vertices P_3 and P_4 – on the second branching level correspond to the vertices P_1 and P_2 on the first level of branching ($P_3 \Rightarrow P_1, P_4 \Rightarrow P_2$). On each branching level it is necessary to determine the location of vertices P_1 and P_2 which represent the tips of child branches in relation to the mother branch $P_a P_b$. The iterative procedure is repeated *n* number of times, depending on the desired number of branching levels (i.e. the age of a tree).

The position of vertices P_1 and P_2 in relation to vertex P_b is defined in a dynamic orthogonal coordinate system *x*', *y*', *z*' (see Figure 13); this coordinate system is related to each mother branch. The dynamc orthogonal coordinate system connected to the mother branch is defined so that axis $P_b z'$ is in the direction of a mother branch, and axis $P_b y'$ is parallel to the plane O_x, O_y. The axis $P_b x'$ forms the lefthand orthogonal coordinate system with axes $P_b z'$ and $P_b y'$.

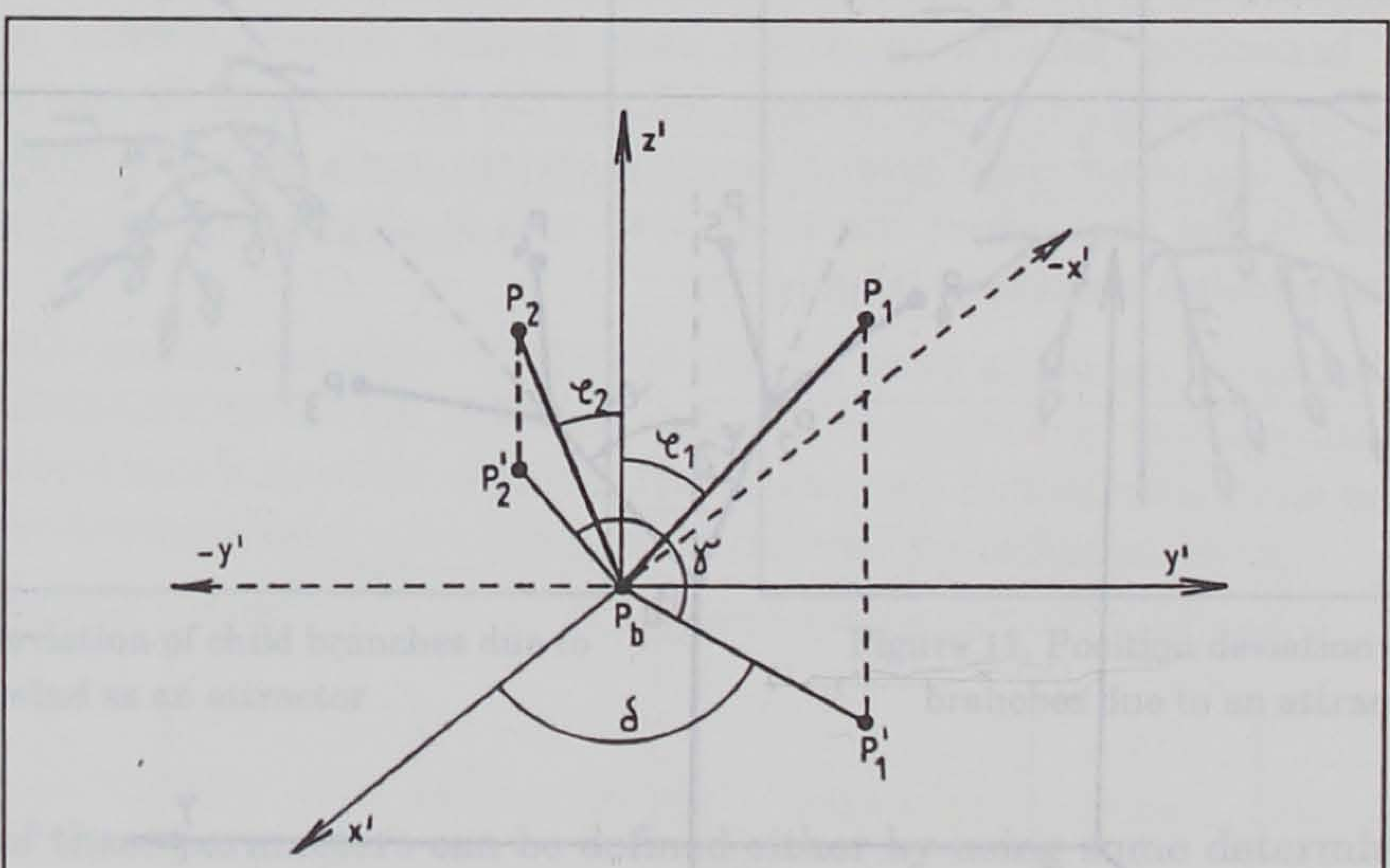

Figure 13. Dynamic orthogonal coordinate system

Angles $\varphi_1(FI(1))$ and $\varphi_2(FI(2))$ are the angles between mother branch $P_b z'$ and child branches $P_b P_1$ and $P_b P_2$. The angles φ_1 and φ_2 can be defined in two different ways:

1) the angles φ_1 and φ_2 (in degrees) can be specified as constant values of the parameters (*FI*);

2) the angles φ_1 and φ_2 can be variables whose value depends on the height of a tree and the level on which the particular branching occurs. The following instances of branching angles are possible:
   a) The angles decrease gradually until they reach the middle level, then they decrease sharply.
   b) The angles decrease sharply until they reach the middle level, then they decrease gradually.
   c) The angles decrease gradually until they reach the middle level where they decrease sharply for a time, then they decrease gradually again.
   d) The angles increase gradually until they reach the middle, then they decrease.

The second method, in which the angles φ_1 and φ_2 are defined as variables, gives a more natural representation of a tree form. The method is derived from biological rules which apply to the majority of tree species. The changes in the branching angles related to the growth level (n) are defined by corresponding mathematical equations. For example, the constant increase or decrease of angles can be defined by linear equations with coefficients which correspond to different tree types. If we, for example, want to determine the angles on two subsequent growth levels (n_1, φ_1) and (n_2, φ_2) and if we attribute a linear decrease (or increase) to angles, then the value of φ on any branching level can be calculated form the following equations:

$$\varphi = an + b, \tag{13}$$

where $a = \dfrac{\varphi_2 - \varphi_1}{n_2 - n_1}$, and

$$b = \varphi_1 - an_1.$$

If the changes in angles correspond to the quadratic function (see Figure 14 – curves 2 and 2a), and if the set of values (n_i, φ_i), representing the change in angles, is specified in accordance with the biological rules of tree growth, then a least-squares method can be used to determine coefficients (a, b, c) of the quadratic function. From the following quadratic function the values of φ can be determined:

$$\varphi = a\,n^2 + b\,n + c \tag{14}$$

In a similar manner an exponential function can be used (see Figure 14 – curve 3) if the angles φ decrease rapidly at the beginning of the branching process and then continue to decrease slowly until the maximum growth level is reached. The process can be defined by the following exponential function:

$$\varphi(n) = A\,e^{-B\,n} \tag{15}$$

$$A = \varphi(0)$$

$$B = -\frac{1}{n_2}\,\lg_e\left(\frac{\varphi(n_2)}{A}\right),$$

and the two vertices are defined as (n_1, φ_1) and (n_2, φ_2):

$$n_1 = 0,\ \varphi_1 = \varphi(0)$$

$n_2, \ \varphi_2 = \varphi(n_2).$

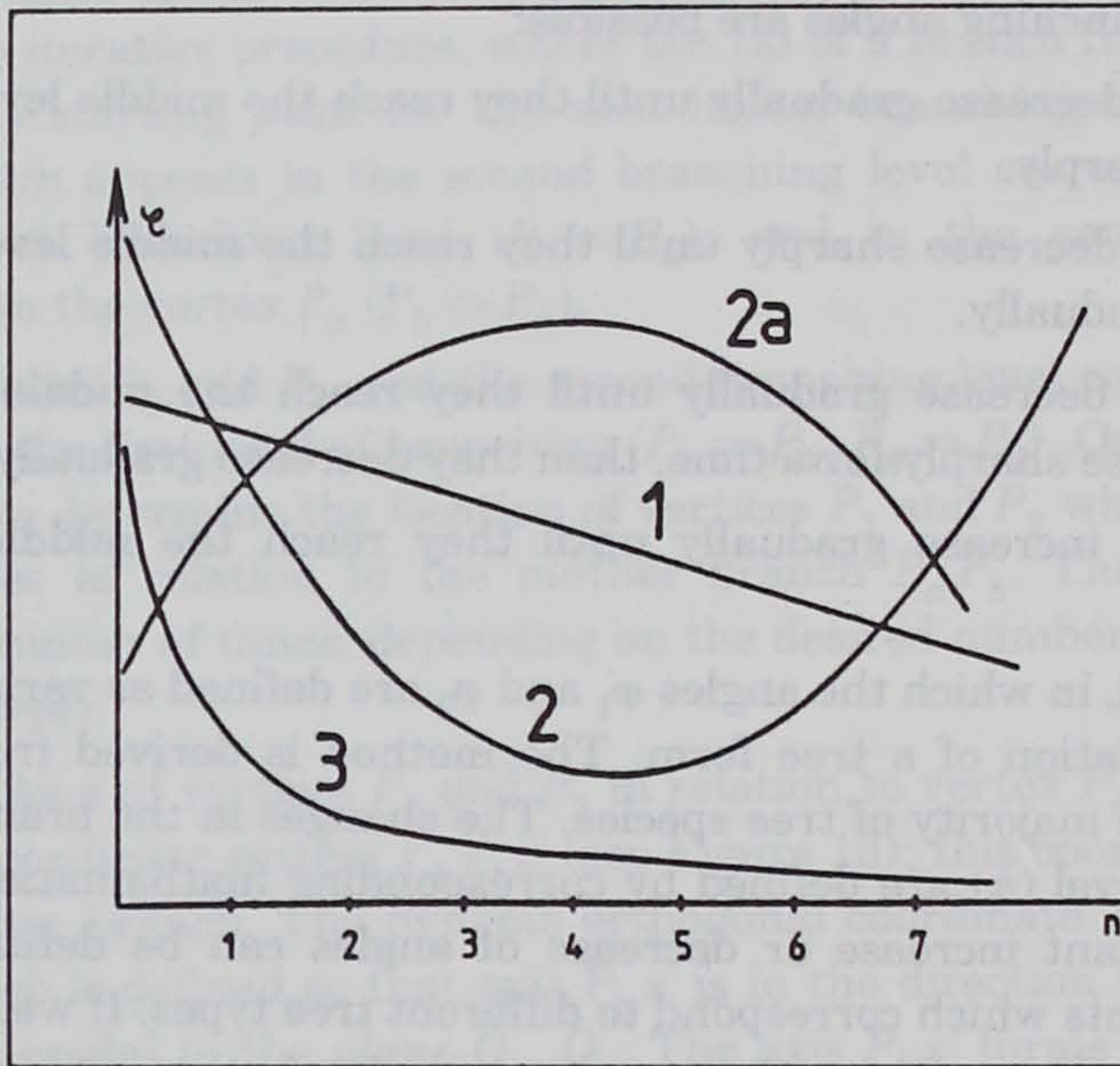

Figure 14. Changes in angles

In both methods (1 – when angles are constants and 2 – when angles are variables) it is possible to introduce a stochastic component. This stochastic component, which defines the change of angles, is introduced by specifying a mean square root derivation factor (*SF*). The factor *SF* is given as one of the input parameters in the tree generation program.

The coordinates of vertices P_1 and P_2 in the coordinate system x', y', z' can be determined from the following mathematical equations:

$$P_1(x_1', y_1', z_1') : (\overrightarrow{P_1 P_b} \sin\varphi_1 \cos\delta, \overrightarrow{P_1 P_b} \sin\varphi_1 \sin\delta, \overrightarrow{P_1 P_b} \cos\varphi_1) \qquad (16)$$

$$P_2(x_2', y_2', z_2') : (\underbrace{\overrightarrow{P_2 P_b} \sin\varphi_2 \cos(\delta+\gamma)}_{x_2'}, \underbrace{\overrightarrow{P_2 P_b} \sin\varphi_2 \sin(\delta+\gamma)}_{y_2'}, \underbrace{\overrightarrow{P_2 P_b} \cos\varphi_2}_{z_2'}).$$

The angle δ (DEL) defines the position of the plane P_1P_bz' of the first child branch (P_1P_b) in the coordinate system x', y', z'. The angle $(\delta + \gamma)$ defines the position of plane P_2P_bz' in which the second child branch lies. The angles δ and γ determine the rotation of the child branches along the mother branch as an axis (see Figure 15). The vertices P_1' and P_2' are projections of vertices P_1 and P_2 into the plane x', y'.

This method solves the problem of spatial distribution of the child branches, avoiding the distribution where all branches are in the same plane.

The coordinates (x_i, y_i, z_i) of the vertices P_1 and P_2, in the static (fixed) coordinate system x, y, z can be determined from the transformations (translations and rotations) of the coordinates (x_i', y_i', z_i') – as given in the coordinate system x', y', z':

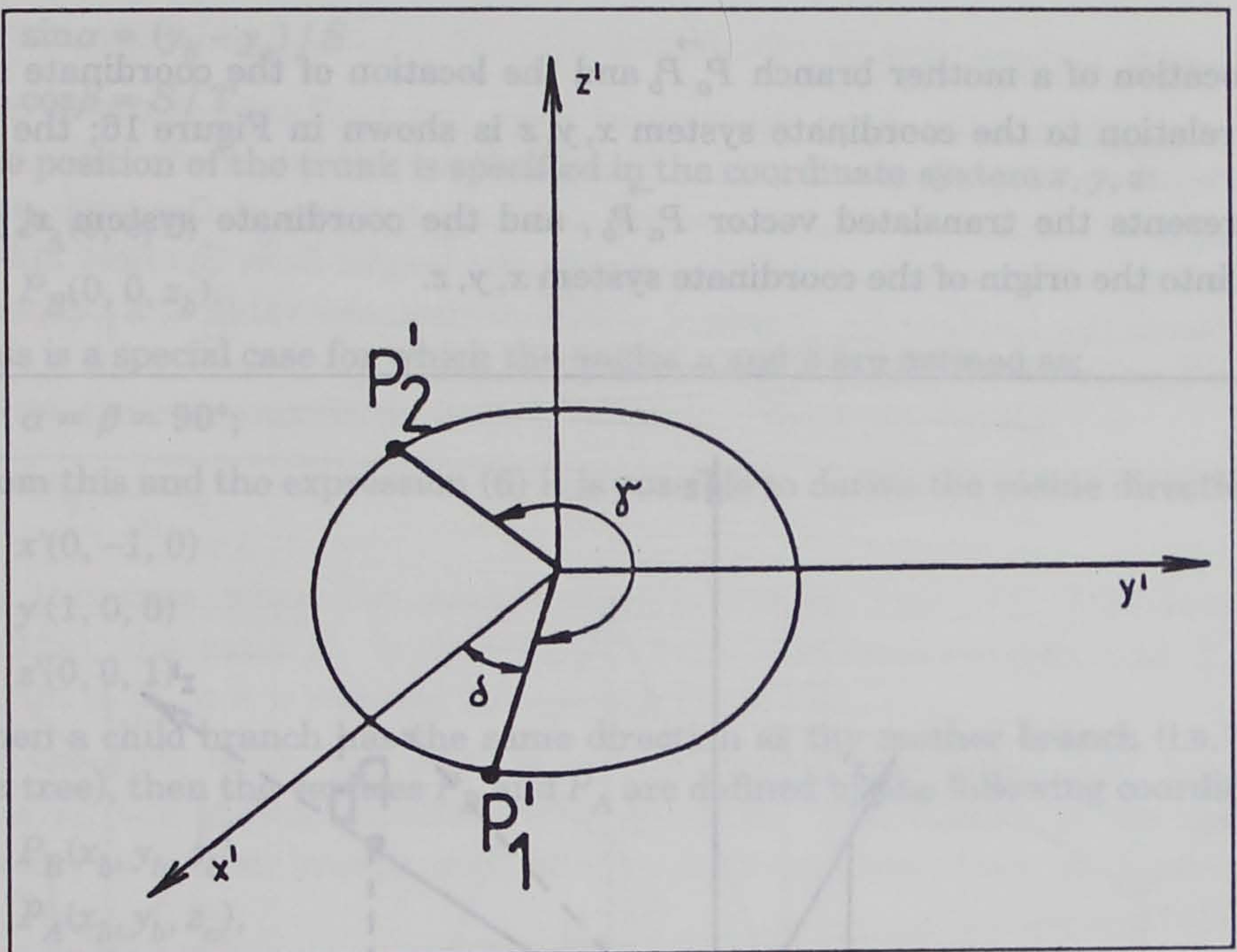

Figure 15. Rotation of the child branches

$$
\begin{aligned}
x_i &= T_{11} x_i' + T_{12} y_i' + T_{13} z_i' + x_b \\
y_i &= T_{21} x_i' + T_{22} y_i' + T_{23} z_i' + y_b \\
z_i &= T_{31} x_i' + T_{32} y_i' + T_{33} z_i' + z_b \,,
\end{aligned}
\tag{17}
$$

where

x_b, y_b, z_b are the coordinates of the vertex P_b in a coordinate system x, y, z ; the cosines of angles between the corresponding axes of the coordinate systems

$$
\begin{aligned}
&x'(T_{11}, T_{21}, T_{31}) \\
&y'(T_{12}, T_{22}, T_{32}) \\
&z'(T_{13}, T_{23}, T_{33})
\end{aligned}
\tag{18}
$$

are given by the following expressions:

$$
\begin{aligned}
T_{11} &= -\sin\beta \cos\alpha \\
T_{21} &= -\sin\beta \sin\alpha \\
T_{31} &= \cos\beta \\
T_{12} &= \sin\alpha \\
T_{22} &= -\cos\alpha \\
T_{32} &= 0 \\
T_{13} &= \cos\beta \cos\alpha \\
T_{23} &= \cos\beta \cos\alpha \\
T_{33} &= \sin\beta \,.
\end{aligned}
$$

The location of a mother branch $\overrightarrow{P_a P_b}$ and the location of the coordinate system x', y', z' in relation to the coordinate system x, y, z is shown in Figure 16; the vector $P_a' P_b'$ represents the translated vector $\overrightarrow{P_a P_b}$, and the coordinate system x', y', z' is translated into the origin of the coordinate system x, y, z.

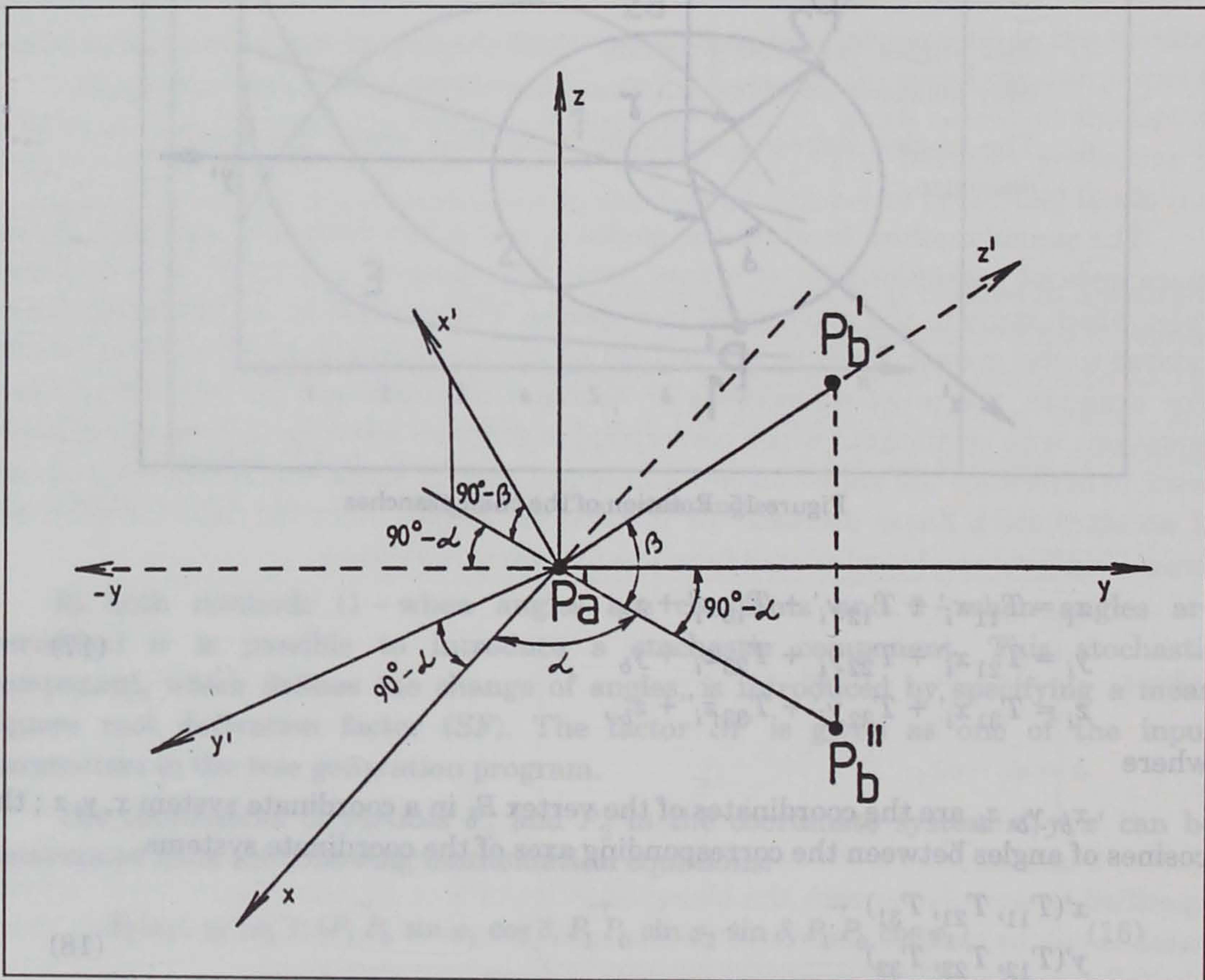

Figure 16. Translated location of a mother branch

The angles α and β are determined on the basis of the given coordinates of vertices P_a and P_b, which represent the root and the tip of a mother branch; the coordinates of these two vertices are $P_a(x_a, y_a, z_a)$ and $P_b(x_b, y_b, z_b)$. From this we get:

$$\sin\beta = (z_b - z_a) / T$$

where

$$T = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2}\,,$$

$$\cos\alpha = (x_b - x_a) / S\,, \tag{19}$$

where

$$S = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}\,,$$

$\sin\alpha = (y_b - y_a) / S$

$\cos\beta = S / T$.

The position of the trunk is specified in the coordinate system x, y, z:

$P_A(0, 0, 0)$

$P_B(0, 0, z_b)$.

This is a special case for which the angles α and β are defined as:

$\alpha = \beta = 90°$;

From this and the expression (6) it is possible to derive the cosine directions:

$x'(0, -1, 0)$

$y'(1, 0, 0)$

$z'(0, 0, 1)$

When a child branch has the same direction as the mother branch (i.e. the main axis of a tree), then the vertices P_B and P_A are defined by the following coordinates:

$P_B(x_b, y_b, z_b)$

$P_A(x_b, y_b, z_a)$,

and from the Equations (19) it is not possible to calculate cosines of the directions $T_{i,j}$, because S is equivalent to zero; this has to be considered as a special case, where cosine directions are given by the following equations:

$$\cos\alpha = \frac{x_b}{\sqrt{{x_b}^2 + {y_b}^2}}$$

$$\sin\alpha = \frac{y_b}{\sqrt{{x_b}^2 + {y_b}^2}}$$

$\beta = 90°$

$\cos\beta = 0$

$\sin\beta = 1$,

from this we derive

$$x'\left(-\frac{x_b}{\sqrt{{x_b}^2 + {y_b}^2}}, -\frac{y_b}{\sqrt{{x_b}^2 + {y_b}^2}}, 0\right)$$

$$y'\left(\frac{x_b}{\sqrt{{x_b}^2 + {y_b}^2}}, -\frac{y_b}{\sqrt{{x_b}^2 + {y_b}^2}}, 0\right)$$

$z'(0, 0, 1)$.

A length of a child branch $\rho_1(\overrightarrow{P_1 P_b})$ can be determined in two ways:

1) a contraction ratio R_i is given as a constant; R_i represents the contraction ratio of child–branch length (ρ_i) to its mother–branch length (ρ_{op})

$$\rho_i = \rho_{op}{}^{R_i},$$

where R_i is constant;

2) a contraction ratio R_i is determined from the biological characteristics of different tree species; in this case R_i depends on the height of a tree (H_0), the length (h) from the first branching point to the top of the tree, and the diameter of a tree (L) or, in other words, the length from the first branching point to the furthest tip of lateral child branches; the value of R_1 depends on the type of a tree (Figure 17).

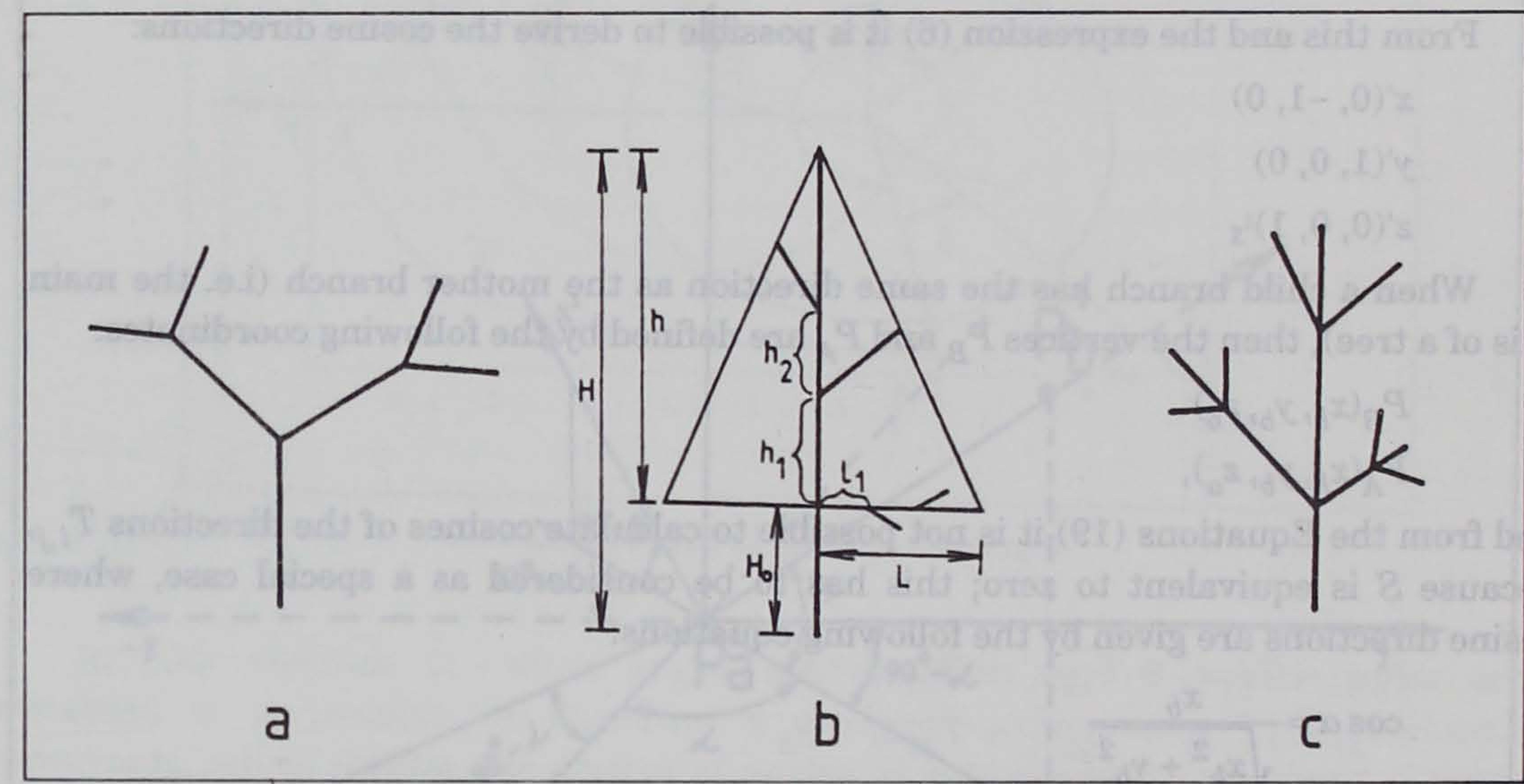

Figure 17. R_i depends on the type of a tree

When a branching process starts off one main axis (see Figure 17b), then the relations described can be approximated by the following equations:

$$H = H_0 + h$$

$$h = \sum_{i=1}^{n} h_i = R_1 H_0 + R_1 R_1 H_0 + R_1^3 H_0 + \ldots + R_1^n H_0 = H_0 R_1 \frac{1 - R_1^n}{1 - R_1} \tag{20}$$

$$L = \sum_{i=1}^{n} l_i = R_2 H_0 + R_1 R_2 H_0 + R_1^2 R_2 H_0 + \ldots + R_1^{n-1} R_2 H_0 = H_0 R^2 \frac{1 - R_1^n}{1 - R_1}$$

From the Equations (20) follows:

$$\frac{h}{L} = \frac{R_1}{R_2}, \tag{21}$$

and because $0 < R_1$, $R_2 < 1$, it is possible to approximate:

$$1 - R_1^n \cong 1,$$

and for big enough n, we get:

$$h \cong H_0 \frac{R_1}{1 - R_1} \tag{22}$$

If we substitute relation (19) in the relation (8), we get:

$$H = H_0 + H_0 \frac{R_1}{1-R_1} = H_0 \frac{1}{1-R_1},$$

and

$$R_1 = 1 - \frac{H_0}{H} \qquad (23)$$

From the relations (9) we get the following expression for R_2:

$$R_2 = \frac{L}{h} R_1 \qquad (24)$$

In this case when the mother branch divides into two child branches (see Figure 17a), the ratio R_1 is determined from the Equation (23), and R_2 is either equivalent to R_1 or it is reduced by a certain factor (RKO).

In the case when the mother branch divides into three child branches (see Figure 17c), the growth direction of the middle child-branch is the same as the direction of its mother branch, and the ratio R_3 is calculated from the Equation (23):

$$R_3 = 1 - \frac{H_0}{H},$$

and R_1 and R_2 are calculated from the Equation (24):

$$R_1 = R_2 = \frac{L}{H} R_3 .$$

3) A contraction ratio of the child branch to its mother branch can be defined as a stochastic parameter, so that R_i is computed as a random value equally spread within the interval [0, 1].

The following examples (Figures 18 and 19) shows tree growth pattern with branching angles FI specified as constants, and with the constant contraction ratios RI of the child branches to the mother branch. Figure 18 shows a monopodial branching pattern (specified by the parameter LVT = 1) where the mother branch divides into two child branches, and one child branch follows the direction of the mother branch. Figure 19 shows the case where the mother branch divides into three child branches, and for all three child branches the branching angles and contraction ratios are specified as constants.

3.3. Modelling of a Tree Taking into Account the Effects of Climate and Gravity

In order to model a realistic tree growth it is important to be able to simulate the effects of gravity, wind, and sunlight on the external of a tree.

The two characteristic types of deviations, which occur on child branches due to external factors, are the following:

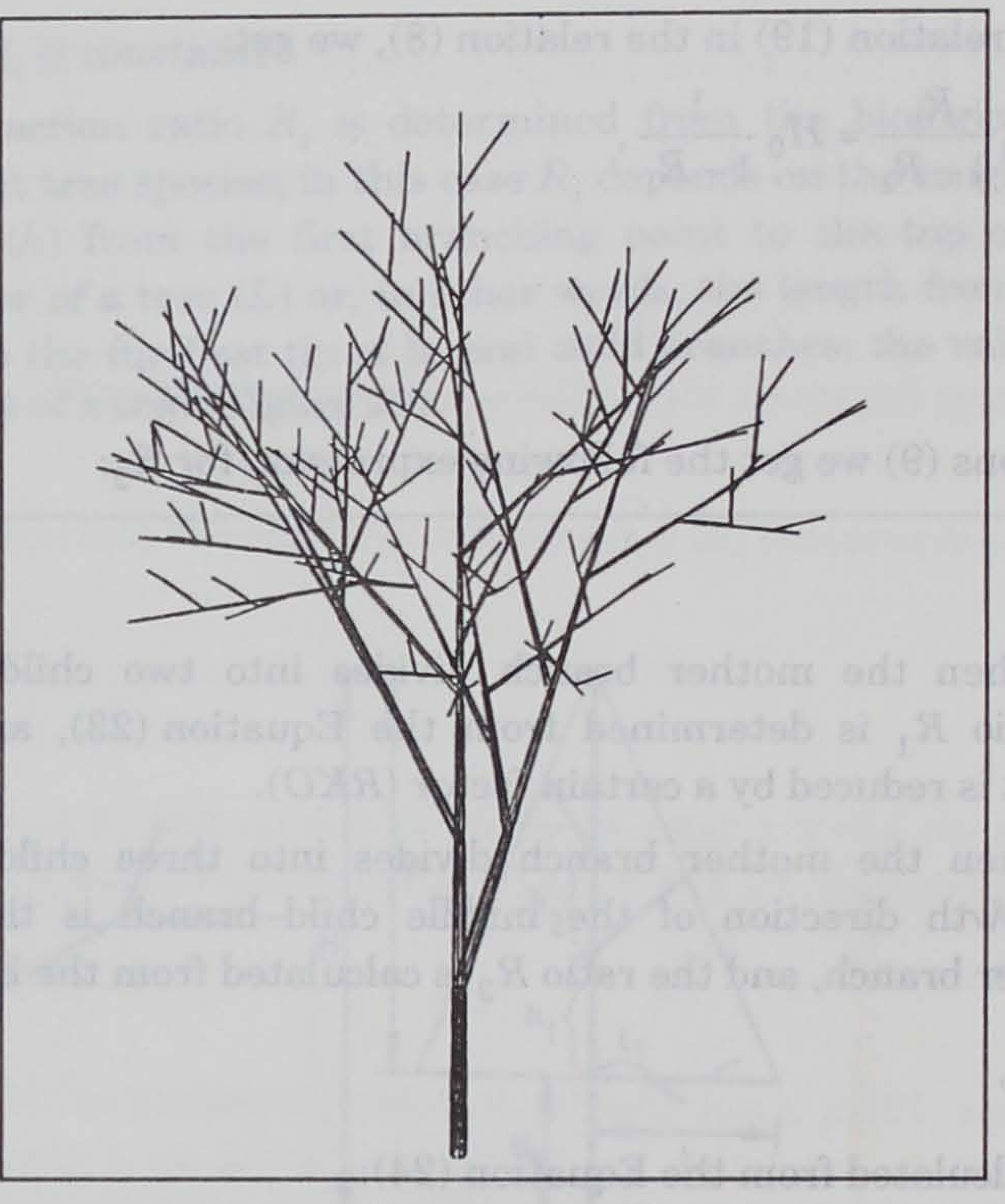

Figure 18. Monopodial branching pattern; FI=const, RI=const, $FI(1)=0$, $FI(2)=45$, $RI(1)=0.9$, $RI(2)=0.8$

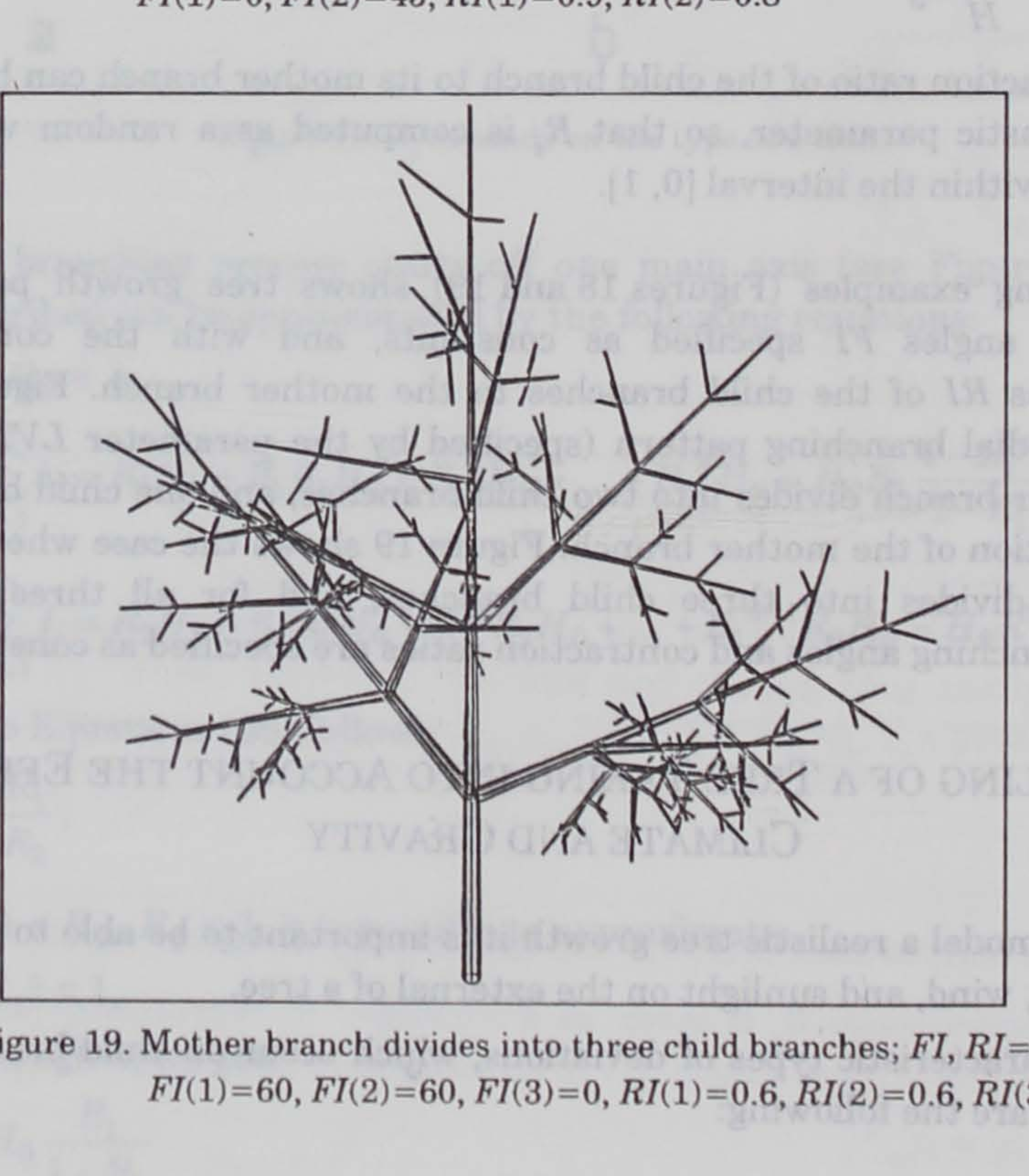

Figure 19. Mother branch divides into three child branches; FI, RI=const, $FI(1)=60$, $FI(2)=60$, $FI(3)=0$, $RI(1)=0.6$, $RI(2)=0.6$, $RI(3)=0.9$

1. Uniform deviations which affect equally every branch of a tree. The deviation is defined by three factors (D_x, D_y, D_z) which are simply added to the set of coordinates (x_i, y_i, z_i) defining the tips (P_i) of child branches.
2. Non–uniform deviations which are deviations of child branches towards some selected point controllers (eg. the effect of a strong wind, etc.). The direction of a child–branch vector $P_b P_1$ is corrected in accordance with point controllers (attractors and/or inhibitors). Each point controller can be defined by its three coordinates $A \mid (Q_x(i), Q_y(i), Q_z(i) \mid$ in the coordinate system x, y, z (see Figure 20).

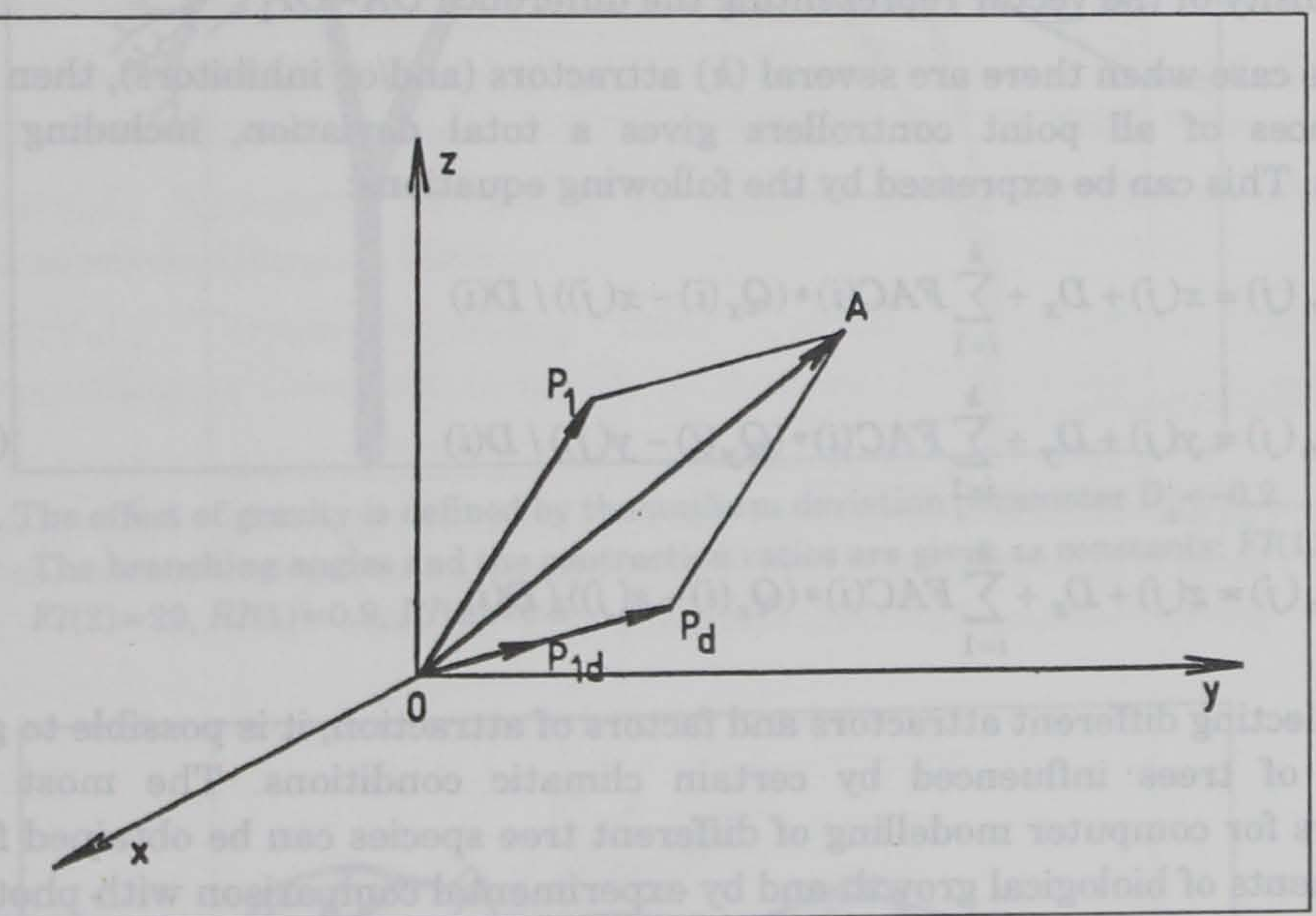

Figure 20.

The location of the vertex P_1 on the growth level j, in the case when there is no apparent deviation of child branches, is defined by the following coordinates:

$P_1(x(j), y(j), z(j))$.

When deviations are non–uniform, the vector $\overrightarrow{OP_1}$ (Figure 20) changes its direction in respect to point controller A, in such a way that it takes the direction of difference of two vectors:

$$\overrightarrow{OP_d} = \overrightarrow{OA} - \overrightarrow{OP_1},$$

and the intensity of a new vector is controlled by a factor ($FAC(i)$); this factor is positive or negative depending on the type of point controller (the controller can be either an attractor or inhibitor). If $FAC(i) > 0$, the point controller is an attractor, and if $FAC(i) < 0$, the point controller is an inhibitor. The vector $\overrightarrow{OP_{1d}}$ is defined by its components:

$$\overrightarrow{OP_{1d}}(\,FAC(1)\cdot(Q_x(i)-x(j))\,/\,D(i),\ FAC(i)\cdot(Q_y(i)-y(j))\,/\,D(i),\ FAC(i)\cdot(Q_z(i)-z(j))\,/\,D(i)\,)$$

where

$$D(i)=\sqrt{(Q_x(i)-x(j))^2+(Q_y(i)-y(j))^2+(Q_z(i)-z(j))^2}$$

is the intensity of the vector representing the difference $\overrightarrow{OA}-\overrightarrow{OP_1}$.

In the case when there are several (k) attractors (and/or inhibitors), then the sum of influences of all point controllers gives a total deviation, including uniform deviations. This can be expressed by the following equations:

$$\begin{aligned}
x_n(j) &= x(j)+D_x+\sum_{i=1}^{k} FAC(i)*(Q_x(i)-x(j))\,/\,D(i)\\
y_n(j) &= y(j)+D_y+\sum_{i=1}^{k} FAC(i)*(Q_y(i)-y(j))\,/\,D(i)\\
z_n(j) &= z(j)+D_z+\sum_{i=1}^{k} FAC(i)*(Q_z(i)-z(j))\,/\,D(i)
\end{aligned} \tag{25}$$

By selecting different attractors and factors of attraction, it is possible to generate a variety of trees influenced by certain climatic conditions. The most suitable parameters for computer modelling of different tree species can be obtained from the measurements of biological growth and by experimental comparison with photographs of existing trees.

In the examples given in Figures 21 and 22 the trees were modelled taking into account the effect of gravity (uniform – constant deviation specified by the parameter D_z). The branching angles and the contraction ratios are given as constants.

In Figures 23 and 24 the examples are of non–uniform deviation, with one point controller given. The coordinate of the point controller are specified the parameters $Q_x(I)$, $Q_y(I)$, and $Q_z(I)$. In Figure 23 the factor of attraction is $FAC(1) = 1$, while in Figure 24 the factor of attraction is $FAC(1) = -1$.

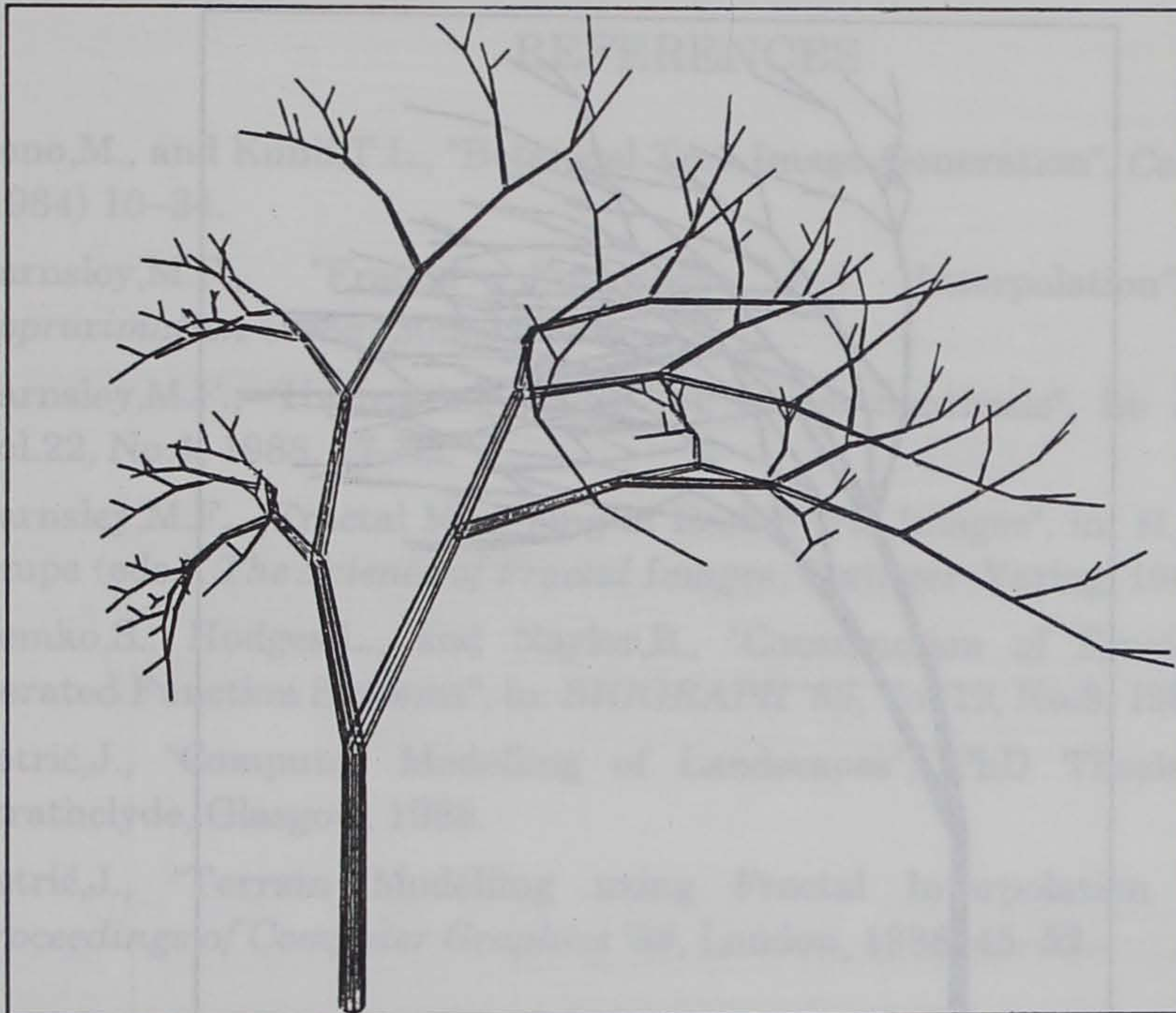

Figure 21. The effect of gravity is defined by the uniform deviation parameter D_z=−0.2. The branching angles and the contraction ratios are given as constants: $FI(1)=20$, $FI(2)=20$, $RI(1)=0.9$, $RI(2)=0.6$

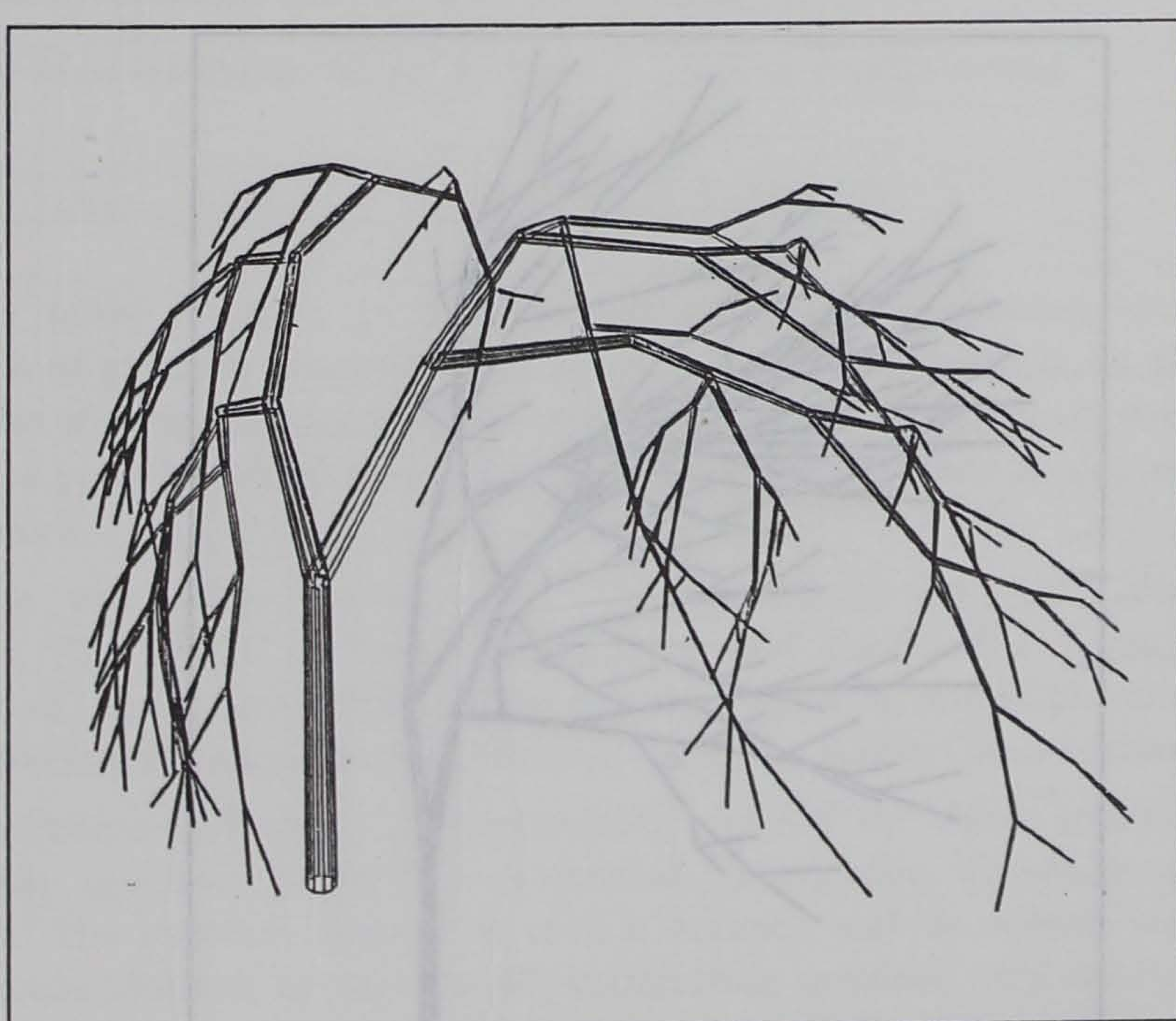

Figure 22. The dichotomous branching pattern with the uniform deviation parameter D_z=−0.6.The branching angles Fl and the contraction ratios Rl are given as constants: $FI(1)=20$, $FI(2)=50$, $RI(1)=0.9$, $RI(2)=0.6$

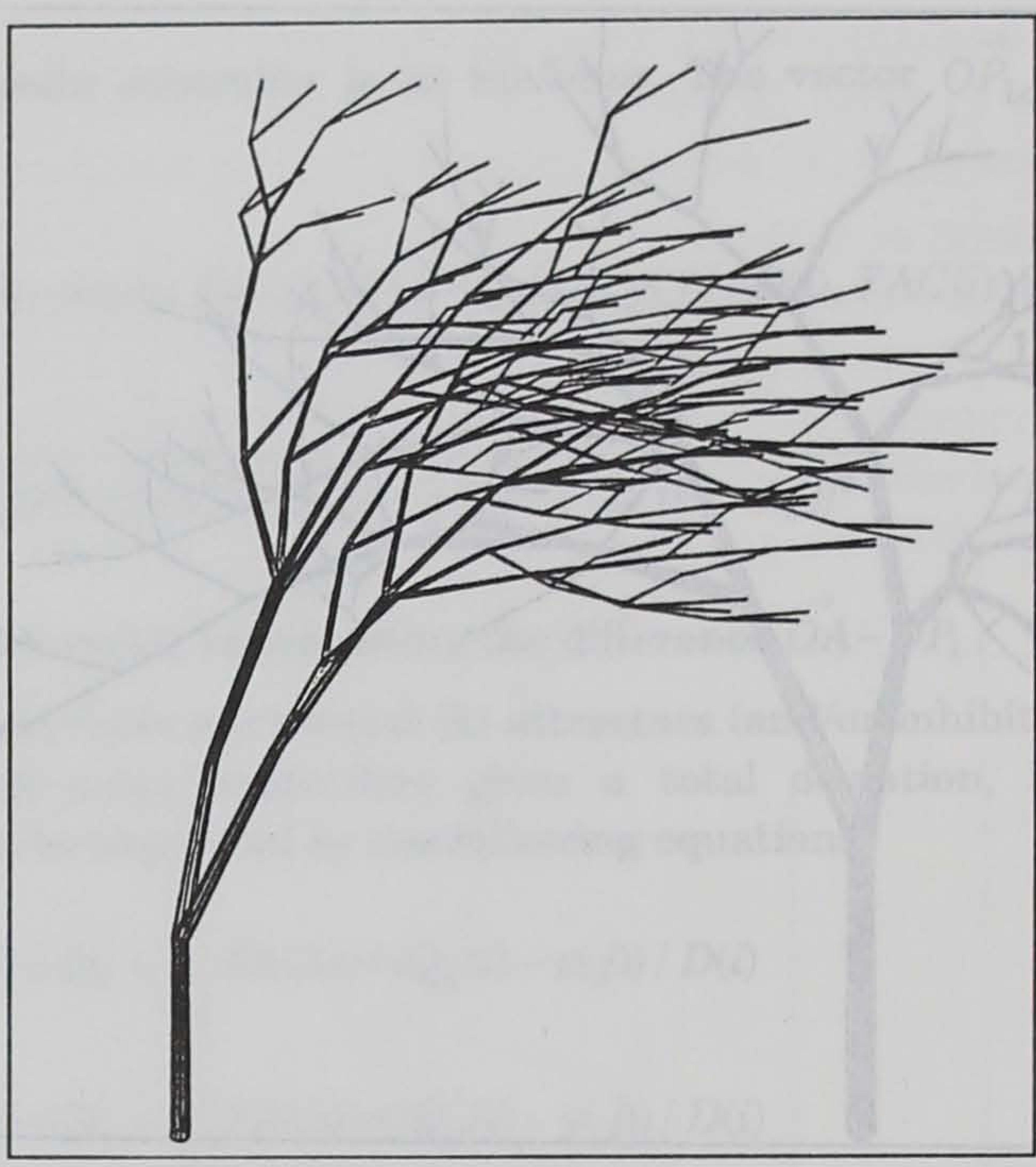

Figure 23. The non–uniform deviation with one point controller: $Q_x=30$, $Q_y=0$, $Q_z=0$, and the factor of attraction $FAC(1)=1$. The branching angles FI and the contraction ratios are given as constants: $FI(1)=0$, $FI(2)=45$, $RI(1)=0.9$, $RI(2)=0.8$

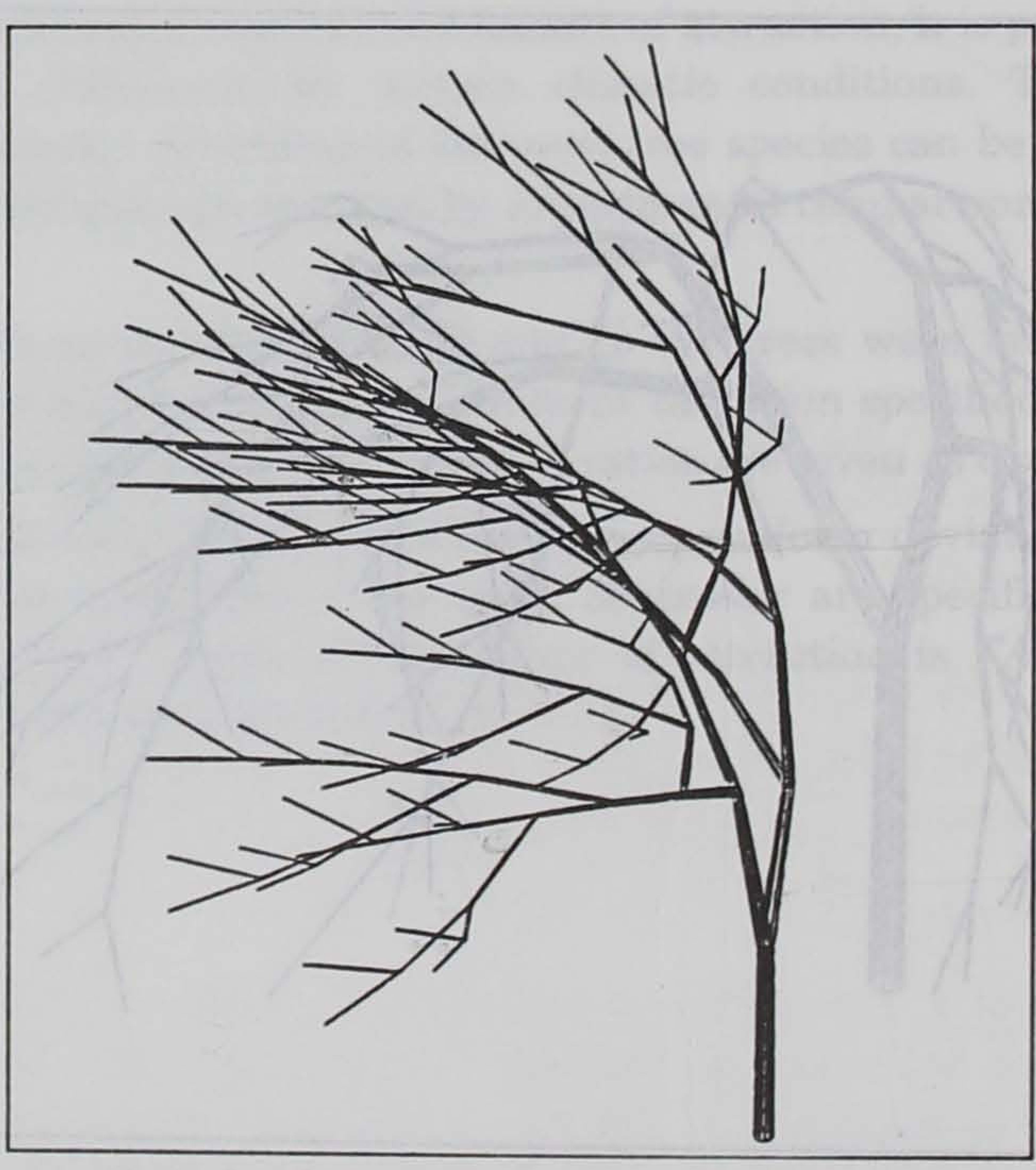

Figure 24. The non–uniform deviation with one point controller: $Q_x=30$, $Q_y=0$, $Q_z=0$, and the factor of attraction $FAC(1)=-1$. The branching angles FI and the contraction ratios are given as constants: $FI(1)=0$, $FI(2)=45$, $RI(1)=0.9$, $RI(2)=0.8$

REFERENCES

[1] Aono,M., and Kunii,T.L., "Botanical Tree Image Generation", *Computer Graphics* (1984) 10–34.

[2] Barnsley,M.F., "Fractal Functions and Interpolation", *Construction Approximation* (1986) 303–329.

[3] Barnsley,M.F., "Harnessing Chaos for Image Synthesis", in: *SIGGRAPH '88*, Vol.22, No.4, 1988, 27–32.

[4] Barnsley,M.F., "Fractal Modelling of Real World Images", in: H. Peitgen, and D. Saupe (eds.), *The Science of Fractal Images*, Springer–Verlag, 1988, 219–239.

[5] Demko,S., Hodges,L., and Naylor,B., "Construction of Fractal Objects with Iterated Function Systems", in: *SIGGRAPH '85*, Vol.19, No.3. 1985, 42–51.

[6] Petrić,J., "Computer Modelling of Landscapes", PhD Thesis, University of Strathclyde, Glasgow, 1988.

[7] Petrić,J., "Terrain Modelling using Fractal Interpolation Functions", in: *Proceedings of Computer Graphics '88*, London, 1988, 45–52.