Chapter 4

Leaf Shape Model and Growth Simulation

This chapter describes the definitions of some important words and parameters used in this thesis, the proposed algorithm for constructing the leaf shape model and simulating the leaf growth, the leaf information, the leaf modeling, approximating growth function, the leaf growth simulation technique, and the model evaluation. The details of each step of the algorithm are illustrated.

4.1 Definitions

There are many of important words and parameters that are used in this thesis. In leaf modeling, Figure 4.1 shows the basic structure of a leaf model. The leaf model is represented by two components: the outline of leaf shape and the leaf skeleton. The leaf skeleton is the leaf network. In this thesis, we use only two level of leaf vein, the midrib and the secondary vein as the leaf skeleton.

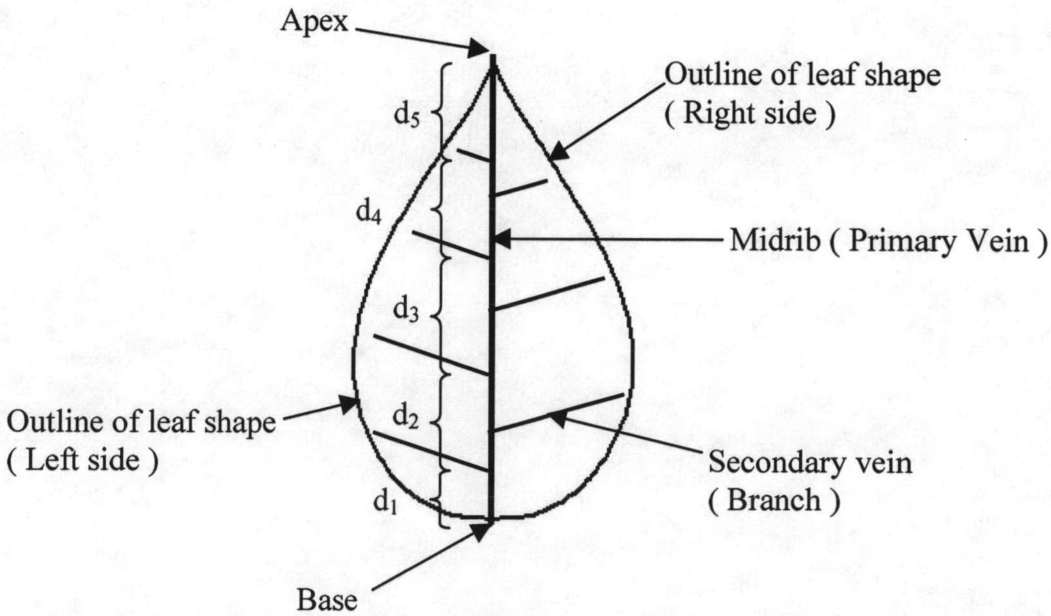


Figure 4.1: Definition of a basic structure of the leaf model.

The outline of leaf shape is divided into two sides: leaf and right. Each side is constructed from a shape function (see section 4.4) that is directly mapped to the length of the midrib. The structure of a leaf skeleton is controlled and defined by the length and the angle of each midrib and secondary vein. The middle line of a leaf model is called midrib while the other lines are called secondary vein. Denote d_i in distance between two secondary veins, or between the apex and the top secondary vein, or the base of the leaf and the bottom secondary vein, where i is the countable number.

4.2 An Algorithm for Computer Model Creation

We presents an algorithm for creating computer models that capture the development of leaf using proposed model of leaf and growth function incorporating biological data. The proposed model is represents the leaf topology and development. The process of constructing a leaf consists of the following steps.

1) Record raw data from a real leaf,
2) Construct a computer model from actual information of the leaf,
3) Approximate the growth function from the collected raw data,
4) Combine the qualitative model and the growth functions,
5) Visualize the simulation of leaf growth,
6) Evaluate the model. For simulating the other kinds of leaf, the parameterized components of the leaf can be adjusted accordingly by new observed data of interested leaves.

The algorithm of the computer model to simulate a leaf growth is described in Figure 4.2. To generate a leaf, several pieces of information are required. This information is discussed in section 4.3.

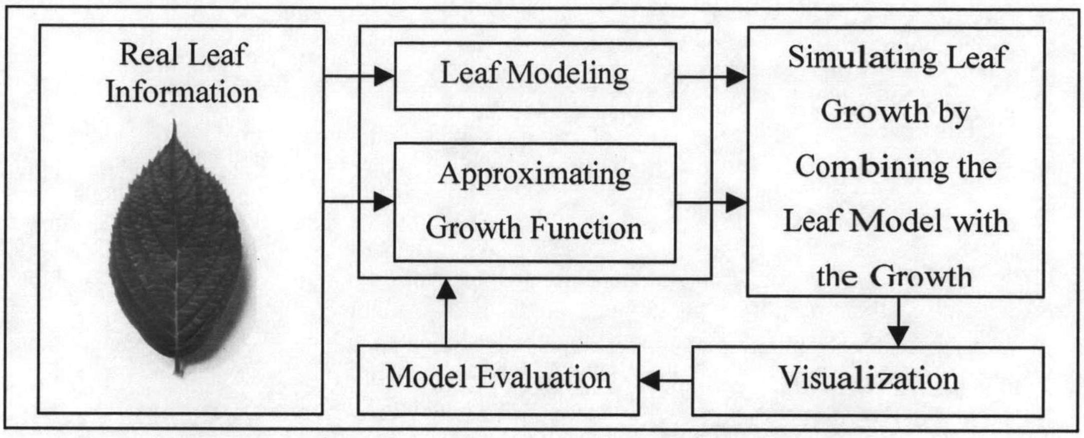


Figure 4.2: The diagram of the leaf growth simulation.

4.3 Leaf Information

The leaf information of a leaf which we want to model, are composed of two parts: the model information and the growth information. We can observe and collect the information from a real leaf.

The model information is the set of parameters which is important to the leaf model. The parameters are the shape of leaf, the length of the midrib, the number of secondary veins on the left side and the right side of the midrib. The name of parameters and their meanings are given as follows:

Midrib_Length - Length of the midrib measured when the leaf is full grow,

Shape_of_Leaf - a number represented a shape of a leaf. In this thesis, six leaf shapes are considered and numbered from 1 to 6.

No_of_Left_Branch - The number of secondary veins on the left side of the midrib,

No_of_Right_Branch - The number of secondary vein on the right side of the midrib,

Left$\{\theta_1, \theta_2, \ldots, \theta_n\}$ - Set of the starting angles, θ_i for $1 \le i \le n$, between midrib and secondary vein along left side of the midrib,

Right$\{\theta_1, \theta_2, \ldots, \theta_n\}$ - Set of the starting angles, θ_i for $1 \le i \le n$, between midrib and secondary vein along right side of the midrib.

The growth information is the set of raw data used to approximate the leaf growth function. There are two parameters in this part: length and width of leaf. Figure 4.3 shows the measurement of the length and width of leaf collected from a real leaf.

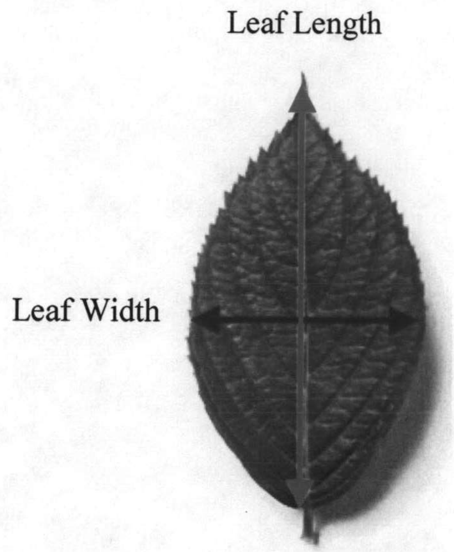


Figure 4.3: The leaf length and width measurement.

The parameters of the growth information are given as follows:

Growth_Length - length of the leaf computed from the growth function,
Growth_Width - width of the leaf computed from the growth function.

The growth function is constructed by using the collected data obtained from the actual growth experiment.

The leaf module is a set of essential parameters described above, which is composed of the name of a leaf, the set of model information, and set of growth information. The module of a leaf can be described in the following format:

Leaf_Name = {

Model Information Section

Midrib_Length

Shape_of_Leaf

No_of_Left_Branch

No_of_Right_Branch

Left{ $\theta_1, \theta_2, \ldots, \theta_n$ }

Right{ $\theta_1, \theta_2, \ldots, \theta_n$ }

Growth Information Section

Growth_Length

Growth_Width

}

4.4 Leaf Modeling

The most important component of a leaf growth simulation is the model of a leaf. This section describes the structure of leaf model, the shape function, the leaf shape and the leaf skeleton.

4.4.1 Structure of Leaf Model

Our primary goal is the modeling of a leaf shape, which can be changed by adjusting some parameters of the model. An experimental leaf model consisted of leaf network and leaf margin. Only the midrib and the secondary of the main level of leaf network are the interest in this study. The leaf margin is defined by the Bezier spline function. There are five steps to construct the model of leaf as follows:

Leaf Modeling Algorithm

1) Define the outline of leaf shape by adjusting the parameters of a Bezier spline function. (See 4.4.2 for The Shape Function)
2) Set the length of the midrib.
3) Trace the leaf shape by direct mapping the Bezier spline function along to the midrib.
4) Find the branching points of the secondary vein. (See 4.4.3 for Leaf Skeleton)
5) Draw the secondary vein by tagging into a number of unit intervals, and using the length function and the angle function.

4.4.2 The Shape Function

The shape function is the Bezier spline function. In general form of the Bezier spline function (see Appendix) which can be defined the number of the control points. The number of control point is effected to the complexity of the shape of curve. In this thesis, we use the six control points $P_0(x_0,y_0)$, $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$, $P_4(x_4,y_4)$, and $P_5(x_5,y_5)$ in the Bezier spline function for constructing the outline of six shapes of leaf. The shape function is given below:

$$X(u) = (1-u)^5 x_0 + 5(1-u)^4 u x_1 + 10(1-u)^3 u^2 x_2 + 10(1-u)^2 u^3 x_3 + 5(1-u) u^4 x_4 + u^5 x_5,$$

$$Y(u) = (1-u)^5 y_0 + 5(1-u)^4 u y_1 + 10(1-u)^3 u^2 y_2 + 10(1-u)^2 u^3 y_3 + 5(1-u) u^4 y_4 + u^5 y_5,$$

where u is the step of the spline function, $0 \le u \le 1$,

The value of $X(u)$ and $Y(u)$ are used to draw the outline of the leaf shape by mapping the value of the function directly along the length of the midrib. Figure 4.4 illustrates the values of the six control points, namely, P_0, P_1, P_2, P_3, P_4, and P_5. The left and right outline of the leaf shape model, which the length of the midrib is set to *Midrib_Length* are shown in this Figure.

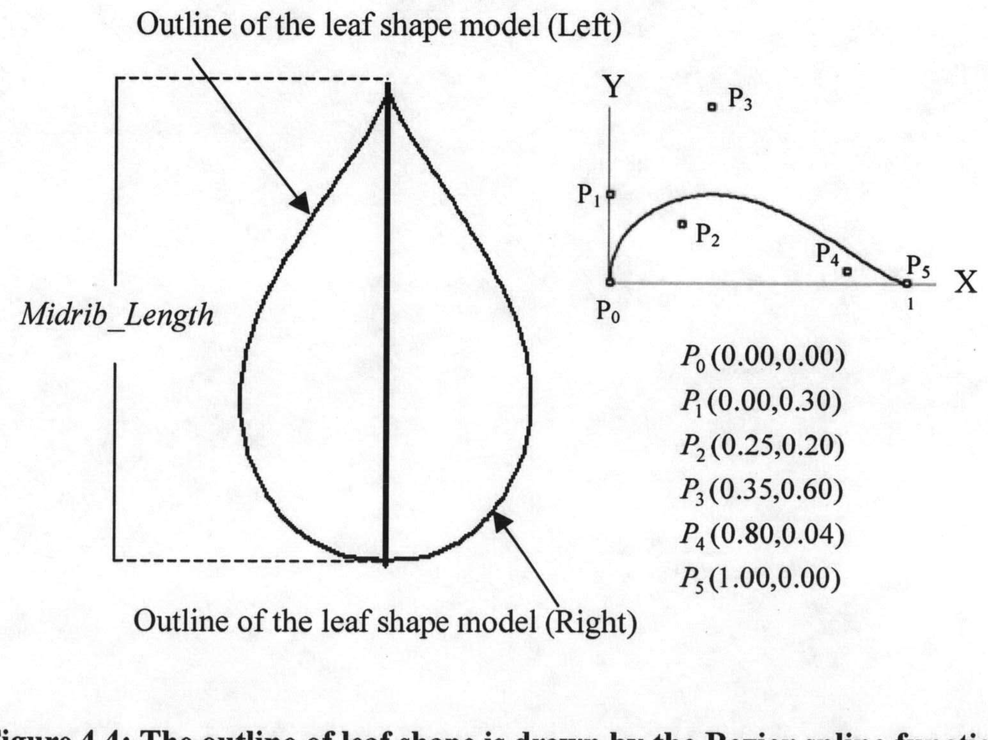


Figure 4.4: The outline of leaf shape is drawn by the Bezier spline function.

P_0 is set to the origin in the shape function, and is mapped to draw the base of the leaf model. Also P_5 is fixed at $X(u)=1$ on the shape function, and is mapped to the apex of the leaf model.

The distance between P_0 and P_5 is equal to 1 on the shape function. So, the length of midrib is set to the ratio use for scaling the shape function to the outline of the shape model.

Figure 4.5 shows a different shape of leaf obtained from a new set control points P_0, P_1, P_2, P_3, P_4, and P_5.

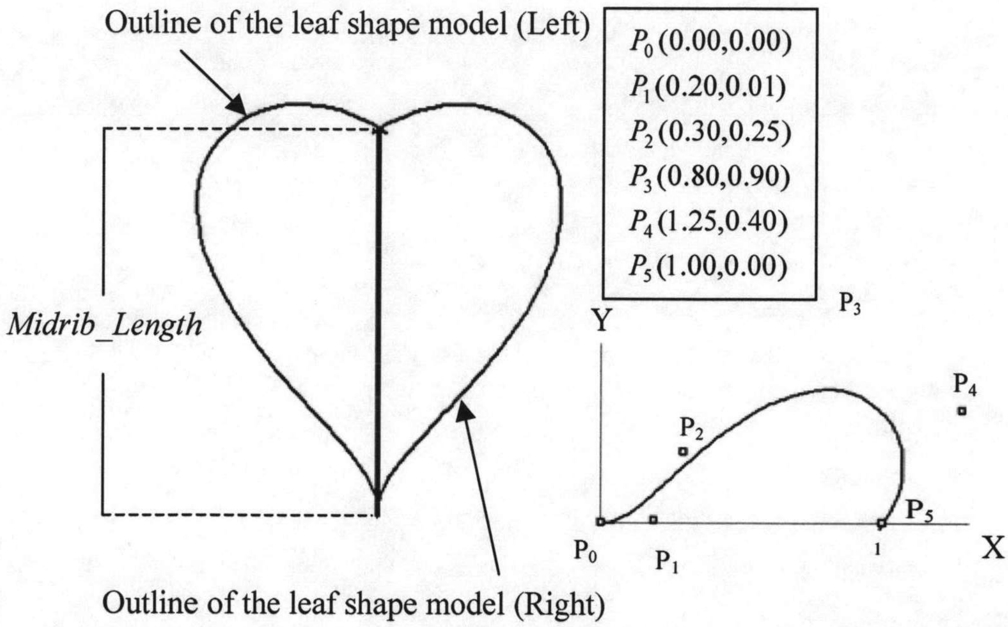


Figure 4.5: The changing of the leaf shape by the parameter adjusting.

4.4.3 Leaf Skeleton

There are three steps to generate the skeleton of a leaf model; constructing the midrib, finding the position of secondary vein branching, and drawing the branches of a model. The midrib construction and the position of the branch are described in Section a), and Section b) is the algorithm for constructing the branches.

a) Midrib (Primary Vein)

The midrib is a straight line, whose the length is set to *Midrib_Length* obtained from the leaf module. The positions of the branches on the left side and right side of the midrib are defined by the position function.

The position function is the Bezier spline function constructed from four control points. The equation is as follows:

$$X(u) = (1-u)^3 x_0 + 3(1-u)^2 u x_1 + 3(1-u)u^2 x_2 + u^3 x_3,$$

$$Y(u) = (1-u)^3 y_0 + 3(1-u)^2 u y_1 + 3(1-u)u^2 y_2 + u^3 y_3,$$

where u is the step for drawing the curve of the function, and $0 \le u \le 1$,

For each side of the midrib, the distance between branching points along the length of the midrib is denoted by d_i. The number of secondary veins on each side is n. The value of d_i is calculated from the position function as shown in equation below:

$$d_i = \frac{Midrib\_Length \cdot Y(u_i)}{\sum_{i=1}^{n} Y(u_i)},$$

where $Y(u_i)$ is the value on the position function, for $0 \le u \le 1$.

For example, to find the values of d_i on the left-side of the midrib, where $i = 1..4$ as shown in Figure 4.6(a) four parameters are needed. The interval u is divided into five steps. Figure 4.6(b) illustrates the values of $Y(u_i)$, which can be computed from the shape function.

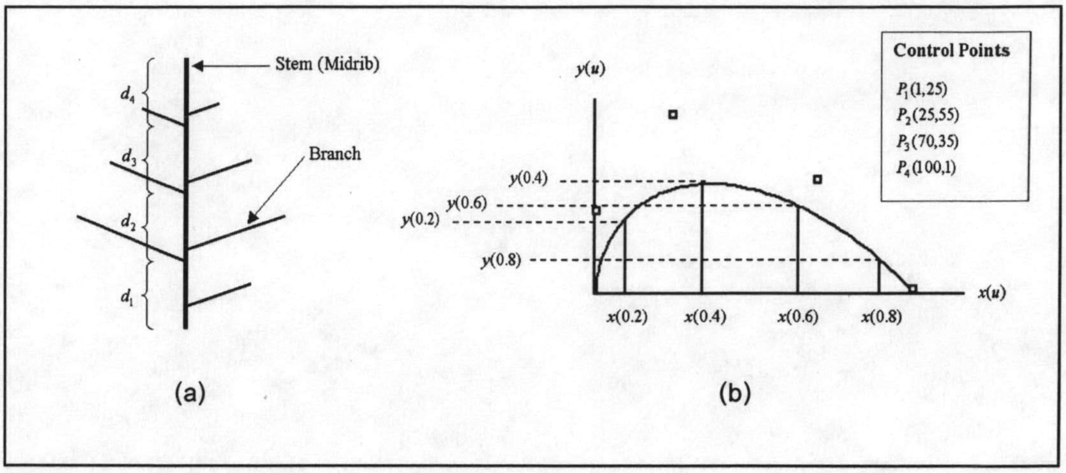


Figure 4.6: An example of the parameters received from shape function.

The midrib with the position of branching points is defined as follows. Set each position of the branching from the midrib to d_i. The value of $Y(u_i)$ is used to control the position of branching points in the model. Adjusting the coordinate of the control points will result in different branching position.

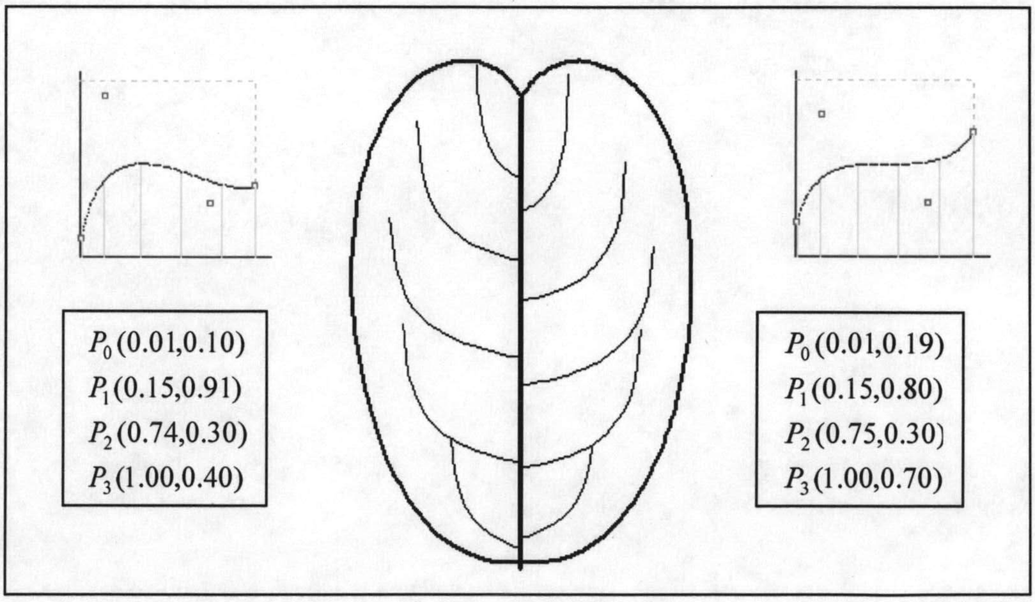


Figure 4.7: Different ratio of d_i for the left and right side branches of the midrib.

b) Secondary Vein (Branch)

Naturally, the secondary vein of a leaf is not a straight line but segments of straight lines making up a curvature. This curvature is a direct result of the leaf's genetic code and nutrients the plant absorbs from the soil and air. To simulate this branch curvature, a branch must be broken down into segments of unit length. Each segment is tagged with an angle. Thus, a branch is defined by a set of segments and their angles. The direction of a branch curvature is controlled by the angle of the first branch interval, which is connected to the stem. The remaining angles on the same branch are derived from the first branch angle. Generation of secondary veins will stop when either the branches intersect the margin or the angle becomes zero. The primitive leaf unit, hence, consists of a unit interval and its vertical angle as shown in Figure 4.8.

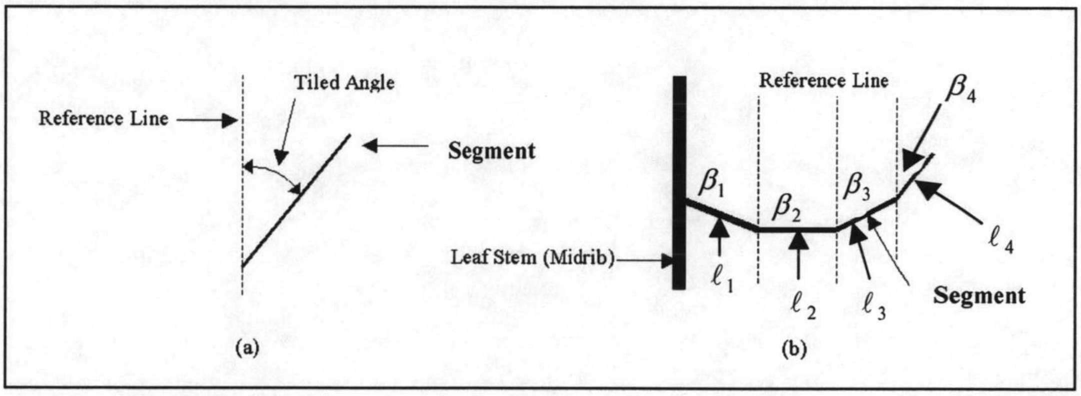


Figure 4.8: Primitive leaf unit (a) and the considered angle β_i of i^{th} internode (b).

The angle between the first segment of a branch and the midrib is set to β_1 as the starting angle of that branch. Subsequent angles of the segments are computed by the following angular function:

$$\beta_{i+1} = \beta_i \cdot a^{i+1},$$

where a is the adjusted parameter, $0 \le a \le 1$.

The end branch consists of a set of intervals $\{\ell_i\}$. The length of the first and i^{th} segments is set to ℓ_1 and ℓ_i, respectively. The length function used to control the length of the unit interval is given below:

$$\ell_{i+1} = \ell_i \cdot b^{i+1},$$

where b is the adjusted parameter, $0 \le b \le 1$.

The algorithm for the drawing of each secondary vein is the following:

Branch Drawing Algorithm

1) Start at the connected point of a branch on the midrib,
2) Set $i = 1$,
3) Set values of the initial parameters: β_1, ℓ_1, a, and b
4) **WHILE** the branch does not intersect the outline of the leaf shape **OR** the angle β_i is not zero **DO**

   BEGIN

   Draw the segment with length = ℓ_i, and in direction of angle = β_i

   Calculate $\beta_{i+1} = \beta_i \cdot a^{i+1}$

   Calculate $\ell_{i+1} = \ell_i \cdot b^{i+1}$

   $i = i + 1$

   END

4.5 Leaf Growth Simulation

In dicots leaves, a leaf grows in two directions: along the length and along the width. So, we suppose that two growth functions, namely, growth function of length $G_L(t)$, and growth function of width $G_W(t)$, will be used to describe the growth rate of a dicots leaf.

In this thesis, we proposed a leaf shape model which is constructed by mapping the shape function to the actual size of a leaf when it is fully grown. Since the leaves considered in this study is symmetric with respect to the midrib, the construction of the outline of a leaf is concerned only either the left outline or the right outline. The shape of a leaf outline is constructed and controlled by six controlling points P_0, P_1, P_2, P_3, P_4 and P_5. Each point P_i is defined by its x-coordinate and y-coordinate.

Naturally, a leaf must grow larger and, eventually, must stop its growth in times. Although the growth process is directed by a growth function, the shape of a leaf during this process never changes. To preserve the shape throughout the growth process, all the x and y coordinates of the six controlling points used to specify the outlines must be relocated and related to the growth function.

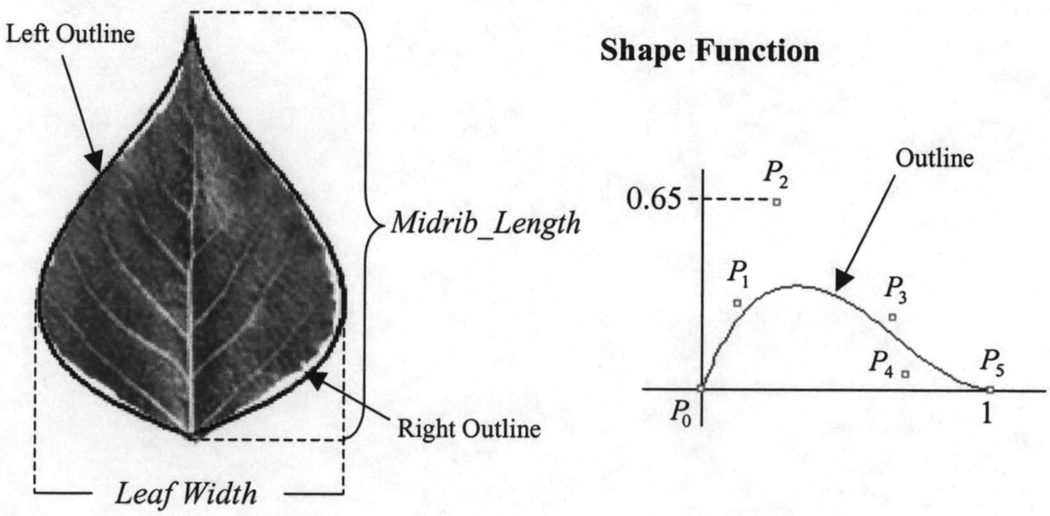


Figure 4.9: The scaling ratio and the shape function of soybean leaf.

Figure 4.9 shows the example of soybean leaf shape model construction with initialized six control points; namely, P_0 is at $(0.00, 0.00)$, P_1 is at $(0.12, 0.30)$, P_2 is at $(0.25, 0.65)$, P_3 is at $(0.65, 0.25)$, P_4 is at $(0.70, 0.05)$, and P_5 is at $(1.00, 0.00)$ are the position on the shape function. Initially, we assume that the midrib length is always set to one unit, which is the shortest length of the midrib. Therefore, the distance between P_0 and P_5 must be one unit apart. The leaf width is controlled by the value of the y-coordinate of P_2 which is initialized to 0.65. This means that the value of the y-coordinate of P_2 can be used to indirectly represent the leaf width during the simulation process.

The curvatures of both left and right outlines are drawn by using a Bezier spline function with all six controlling points. When a leaf is growing, its midrib length and width are non-linearly enlarged. This means that the x and y coordinates of the controlling points adjusted between two consecutive growing time steps are not equal. In this situation, a growth function must be defined prior to the adjustment of the coordinates of the controlling points. There are several growth functions proposed by several researchers. Among these functions are Gompertz growth curve [38], Logistic growth curve [39], Bertalanffy growth curve [40], Brody growth curve [41], Richards growth curve [42], Janoschek growth curve [43]. Depending upon the experimental results of different species, all growth functions have different shapes.

In general, the x and y coordinates of a controlling point P_i can be written in terms of a growth function as follows. Let $G_L(t)$ and $G_W(t)$ be the growth functions having a time variable t for the midrib length and the leaf width, respectively. Therefore, each P_i becomes

$$P_0 \text{ is at } (0.00 \times G_L(t), 0.00 \times G_W(t)),$$

$$P_1 \text{ is at } (0.12 \times G_L(t), 0.30 \times G_W(t)),$$

$$P_2 \text{ is at } (0.25 \times G_L(t), 0.65 \times G_W(t)),$$

$$P_3 \text{ is at } (0.65 \times G_L(t), 0.25 \times G_W(t)),$$

$$P_4 \text{ is at } (0.70 \times G_L(t), 0.05 \times G_W(t)),$$

$$P_5 \text{ is at } (1.00 \times G_L(t), 0.00 \times G_W(t)).$$

Note that the values of $G_L(t)$ and $G_W(t)$ act as the scaling parameters for the x and y coordinates to either expand or shrink the leaf shape. Based on the available experimental data and study reported in [2], soybean is our main interest and concern. The growth function for both midrib length and leaf width of a soybean leaf has a shape of a sigmoid function, namely,

$$G_L(t) = \frac{1}{1+e^{-m_L t}}, \text{ and } G_W(t) = \frac{1}{1+e^{-m_W t}},$$

where m_L is the slope parameter of the length growth function, m_W is the slope parameter of the width growth function. Both m_L and m_W are used to control the growth speed. However, these functions cannot be applied directly to our simulation. Some modifications with the considerations of the maximum midrib length, the leaf width, and growing time must be performed. Let ML be the maximum midrib length and MW the maximum value of the y-coordinate of P_2, where P_2 is the controlling point which yields the maximum width of a soybean leaf. To include ML to $G_L(t)$, the height of the sigmoid function must be changed to ML. This can be achieved by multiplying ML to the sigmoid function as the following

$$G_L(t) = \frac{ML}{1+e^{-m_L t}}.$$

Similarly, after including MW to $G_W(t)$, the growth function $G_W(t)$ becomes

$$G_W(t) = \frac{MW}{1+e^{-m_W t}}.$$

Since the value of P_2 is used to indirectly represent the leaf width, the value of MW obviously implies the half of the maximum leaf width. Until now, both functions are still not correct and cannot be used in our simulation. It can be seen that when t is zero, the value of $G_L(0)$ is equal to $\frac{ML}{2}$ and $G_W(0)$ is equal to $\frac{MW}{2}$. This contradicts with the natural fact that at the starting time of a leaf life, its midrib length and leaf width are minimum. Hence, the time t in both functions must be shifted so that when

t is equal to zero, the values of $G_L(0)$ must be one and $G_W(0)$ must be 0.65. The modifications of $G_L(t)$ and $G_W(t)$ to preserve the above constraints are the followings

$$G_L(t) = \frac{ML}{1+e^{-m_L(t+T_L)}}, \text{ and } G_W(t) = \frac{MW}{1+e^{-m_W(t+T_W)}},$$

where T_L is a constant that makes $G_L(0)$ equal to one, and T_W is a constant that makes $G_W(0)$ equal to 0.65. So, the value of T_L is

$$\frac{1}{m_L} \cdot \ln\left(\frac{1}{ML-1}\right),$$

and the value of T_W is

$$\frac{1}{m_W} \cdot \ln\left(\frac{0.65}{MW-0.65}\right).$$

It can be seen that the values of T_L and T_W are defined by ML and MW, respectively. In case of soybean, the values of ML and MW obtained from the experiments are set to 5.3 cm. and 2.2 cm. Therefore, T_L is equal to –0.05834 and T_W is equal to –0.02897. The growth functions $G_L(t)$ and $G_W(t)$ with T_L and T_W set according to the experiment are as follows

$$G_L(t) = \frac{5.3}{1+e^{-25(t+(-0.05834))}},$$

$$G_W(t) = \frac{2.2}{1+e^{-30(t+(-0.02897))}},$$

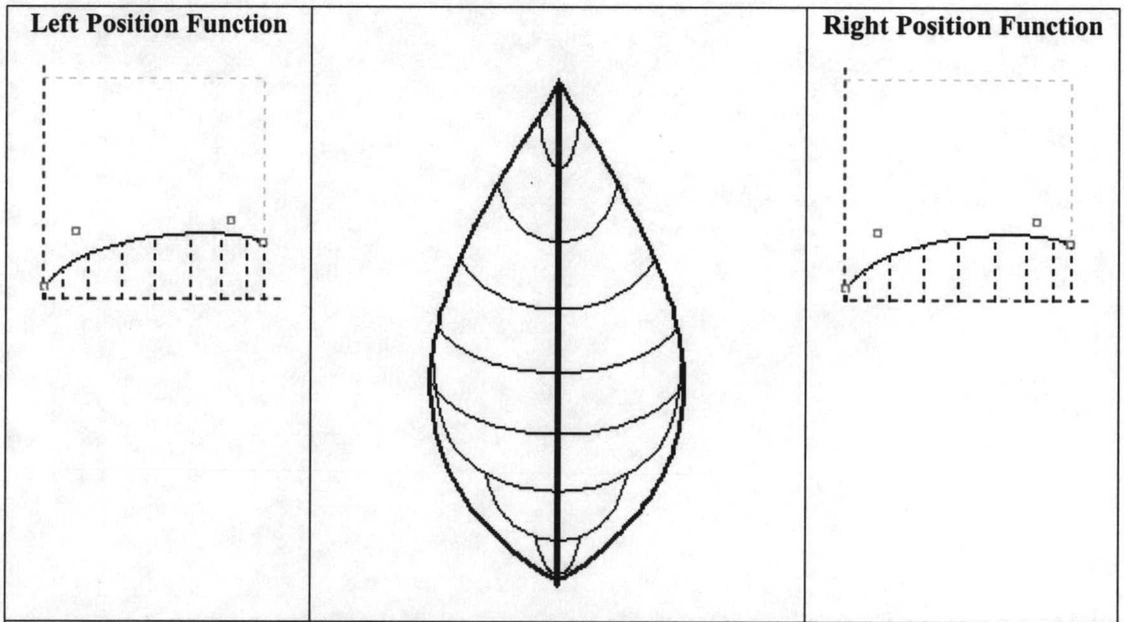
where m_L of the soybean leaf is equal to 25 and m_W is equal to 30. The results of the soybean leaf growth simulation are shown in Chapter 5.

4.6 Model Evaluation

The proposed model of the leaf shape in this thesis is consisted of the outline of the leaf shape and the leaf skeleton. A model has the information use to construct the shape and its skeleton represents as the following:

1) The position of six control points used for modeling the shape.
2) The number of left branches and right branches.
3) The left and right position function used to state the position of the secondary veins.
4) The set of the starting angle of the secondary veins on the left and right side of the midrib.

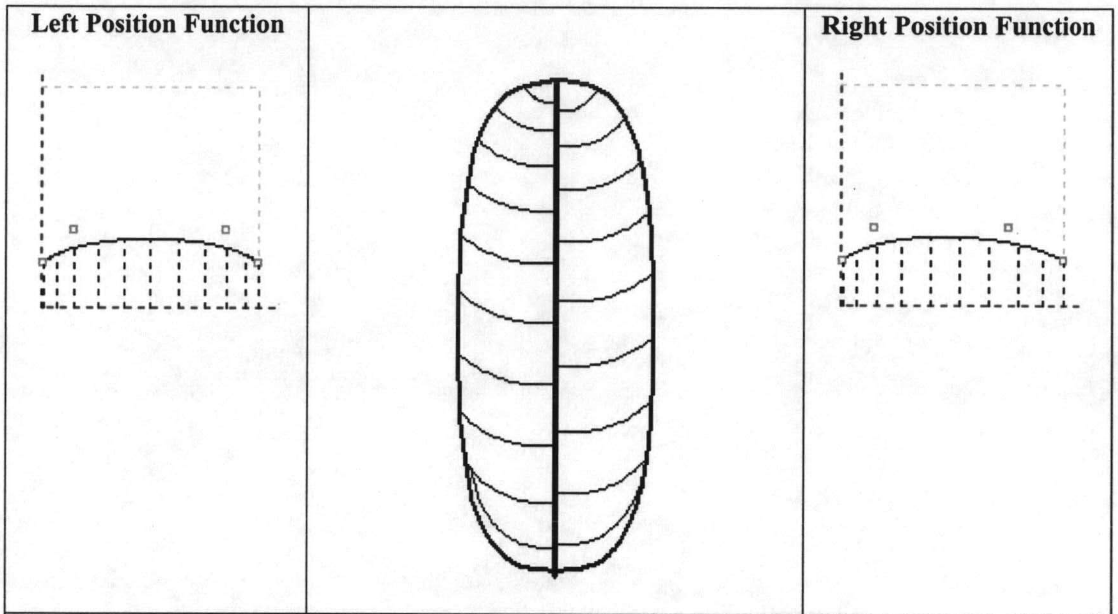
For example, to construct a Lanceolate shape model shown as in Figure 4.10, the information of this type is illustrated below:



Position of Control Points in Shape Function =
$P_0(0.00,0.00), P_1(0.01,0.01), P_2(0.25,0.45), P_3(0.50,0.35), P_4(0.95,0.01), P_5(0.00,1.00)$
Number of Left Branches = 8
Number of Right Branches = 8
Left Starting Angle = {50, 85, 95, 96, 96, 96, 90, 45}
Right Starting Angle = {50, 85, 95, 96, 96, 96, 90, 45}

Figure 4.10: The Lanceolate shape model and parameters.

The leaf model offers flexible parameter adjustment capability. This allows the designer to verify the parameters in the proposed model and to modify the appearance of the graphical image of the generated leaf in real time mode. Figure 4.11 shows the results of leaf shape model which is adjusted some parameters of the Lanceolate shape. The result is look like the Oblong shape.



Position of Control Points in Shape Function =
$P_0(0.00,0.00), P_1(0.01,0.25), P_2(0.25,0.20), P_3(0.75,0.20), P_4(1.00,0.25), P_5(0.00,1.00)$
Number of Left Branches = 10
Number of Right Branches = 9
Left Starting Angle = {90, 90, 90, 90, 90, 90, 90, 90, 90, 90}
Right Starting Angle = {90, 90, 90, 90, 90, 90, 90, 90, 90}

Figure 4.11: The Oblong shape caused by adjusting the parameters of Lanceolate shape model.

We utilize any leaf model that has shape as these six types: Lanceolate, Oblong, Elliptical, Ovate, Cordate, and Obcordate. The results will be shown in Chapter 5.