Chapter 11

Approximation

Xin Li, from the University of Science and Technology of China, contributed significantly to this chapter.

11.1 Introduction

Approximation theory plays an important role in computer aided geometry design and computer graphics. A common general problem in approximation theory can be stated as follows. Let $\mathbf{G}$ denote a set of given data. Let $\mathbf{A}$ denote a set (usually infinite) of candidate mathematical objects (such as functions, parametric surfaces, T-Splines, etc.) with which to approximate $\mathbf{G}$. Let $e(A_i, \mathbf{G})$ be an error measure that indicates how well $A_i \in \mathbf{A}$ approximates $\mathbf{G}$. We wish to find a "best" approximation $A_i \in \mathbf{A}$ for which $e(A_i, \mathbf{G}) \leq e(A_j, \mathbf{G}) \forall A_j \in \mathbf{A}$. Since finding a best approximation can involve expensive optimization algorithms, we might be content to find any approximation for which $e(A_i, \mathbf{G})$ is less than a threshold error. A few simple examples will illustrate.

Example 1. Let $\mathbf{G} = \{2, 3, 3, 5, 7\}$ and let $\mathbf{A}$ be the set of all real numbers. If we choose the error measure to be

$$e(A_i, \mathbf{G}) = \sum_{G \in \mathbf{G}} ((A_i - G)^2$$

the best approximation is $A_i = 4$ for which $e(4, \mathbf{G}) = 16$. This error measure is referred to as the L^2 error measure, and the best approximation is the mean of the values in $\mathbf{G}$.

Example 2. Again, let $\mathbf{G} = \{2, 3, 3, 5, 7\}$ and let $\mathbf{A}$ be the set of all real numbers. Let the error measure be

$$e(A_i, \mathbf{G}) = \max_{G \in \mathbf{G}} |A_i - G|.$$

The best approximation is now $A_i = 4.5$, for which $e(4, \mathbf{G}) = 2.5$. This error measure is referred to as the L^∞ error measure, and the best approximation is the average of the largest and smallest values in $\mathbf{G}$. Thus, different error functions can produce different best approximations.

Example 3. Let $\mathbf{G}$ be a Bézier curve of degree n (call it $\mathbf{P}(t)$) and let $\mathbf{A}$ be the set of all Bézier curves of degree $n - 1$, and $A(t)$ a Bézier curve in $\mathbf{A}$. Let the error measure be

$$\max ||\mathbf{P}(t) - A(t)||, t \in [0, 1].$$

This is the "degree reduction" problem.

Example 4. Let **G** be a B-Spline curve (call it $\mathbf{P}(t)$) of degree n, knot vector k, and domain $t \in [t_0, t_1]$. Let **A** be the set of all B-Spline curves of degree n and with a knot vector $\tilde{k}$ which is obtained by removing one or more knots from k. Let $A(t)$ be a curve in **A**. Let the error measure be

$$\max ||\mathbf{P}(t) - A(t)||, t \in [0, 1].$$

This is the "knot removal" problem.

Example 5. **G** is a large set of data points, measured on the surface of a physical object. **A** is a set of NURBS, T-Splines, or Subdivision Surfaces. This is the "reverse engineering," for which it is customary to use the L^2 error measure.

Example 6. In drawing a curve or surface, it is customary to approximate the curve with a polygon, or the surface with a polyhedron. These are examples of approximating a continuous object with a simpler continuous object. The question of how many line segment or triangles are needed to make the curve or surface appear smooth belongs to approximation theory.

11.2 L^2 Error

The L^2 error measure is widely used in approximation problems because in many cases it is particularly easy to identify A_i that minimizes the L^2 error measure. Approximation using the L^2 error measure is also called "least squares" approximation. In the case where **G** is a set of real numbers, the best L^2 approximation is simply the mean of those numbers. Specifically, let

$$f(x) = \sum_{i=1}^{n} (x - x_i)^2.$$

Then we have the following lemma.

Lemma 11.2.1 *Suppose* $\hat{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, *then*

$$f(x) \geq f(\hat{x}).$$

Proof 1

$$\begin{aligned} f(x) - f(\hat{x}) &= \sum_{i=1}^{n} [(x - x_i)^2 - (\hat{x} - x_i)^2] \\ &= \sum_{i=1}^{n} [(x - \hat{x})(x + \hat{x} - 2x_i)] \\ &= (x - \hat{x})(nx - \sum_{i=1}^{n} x_i) \\ &= n(x - \hat{x})^2 \geq 0 \end{aligned}$$

So the lemma holds.

11.3 Approximating a Set of Discrete Points with a B-Spline Curve

Let **G** be a set of data points $P_i = (x_i, y_i, z_i)$, $i = 1, \ldots, M$, with associated parameter values, t_i. Let **A** be the set of all B-Spline curves with a given knot vector and degree. We seek $A_i \in \mathbf{A}$ that minimizes the discrete L^2 error where $A_i \in \mathbf{A}$ is given by $B(t) = \sum_{j=1}^{n} T_j B_j(t)$. This amounts to identifying the B-Spline control points $T_1, T_2, \ldots, T_n$ that minimize the function

$$Lsq(T_1, T_2, \ldots, T_n) = \sum_{i=1}^{M} (P_i - \sum_{j=1}^{n} T_j B_j(t_i))^2. \tag{11.1}$$

This problem reduces to the solution of the following linear functions:

$$\frac{\partial Lsq(T_1, T_2, \ldots, T_n)}{\partial T_j} = 0. \tag{11.2}$$

We can solve for the control points T_i from the linear equation

$$M_{fit} * T = B \tag{11.3}$$

where M_{fit} is a $n \times n$ matrix with element $a_{ij} = \sum_{k=1}^{M} B_i(s_k) B_j(s_k)$. $T = [T_i]$ is the vector of control points for which we are solving and B is a vector whose elements are $b_i = \sum_{k=1}^{M} P_k B_i(t_k)$.

Algorithm 1 Setting Up the fitting matrix M_{fit}

Require: sample points P_i, associated parameter values (t_i), knot vector and degree for the B-Spline curve.
Ensure: $M_{fit} = [a_{ij}]$

```
for i = 1 to N do
  for j = i to N do
    a_ij ⇐ a_ji ⇐ 0
    Ω_ij = Support(B_i) ∩ Support(B_j)
    if Ω_ij ≠ ∅ then
      for k = 1 to M do
        if (s_k) ∈ Ω_ij then
          d ⇐ B_i(s_k) * B_j(s_k)
          a_ij += d
          a_ji += d
        end if
      end for
    end if
  end for
end for
```

The prudent choice of s_i for each sample point P_i, which determines the parametrization of the curve, is crucial in obtaining a good fit. The choice of knot vector or knot intervals can also impact the quality of the fit. The next two subsections elaborate on these topics.

11.3.1 Parametrization

The simplest parametrization assigns evenly spaced parameter values to the data points:

$$t_j = \frac{j-1}{M-1}. \tag{11.4}$$

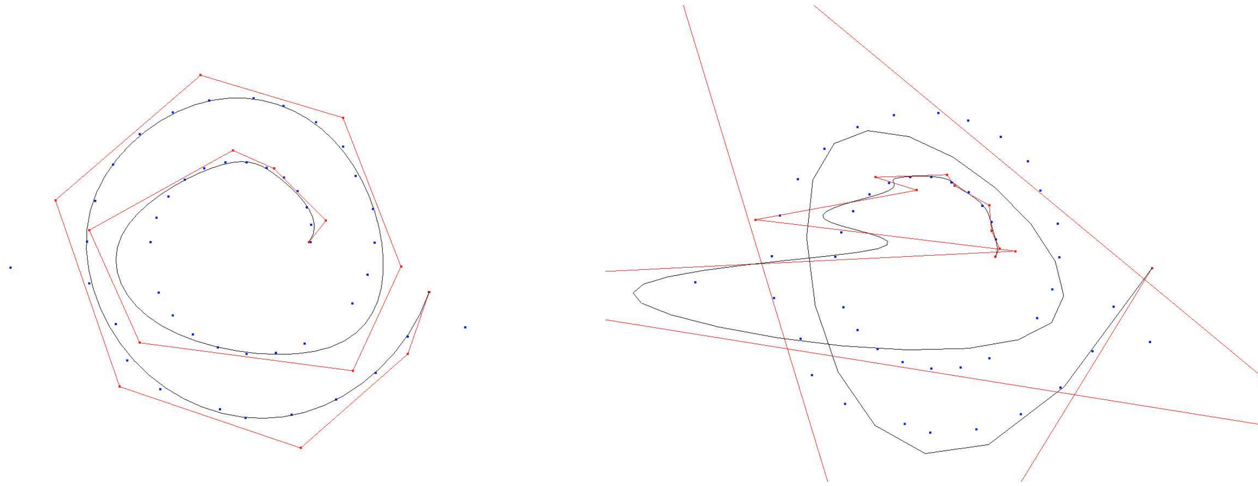

Figure 11.1: Uniform vs. bad parametrization.

This *uniform* parametrization works well if the data points are evenly spaced, as illustrated in Figurefig:1.

The second is arc length parametrization, which is defined as

$$s_j = \frac{\sum_{i=2}^{j} \|P_i - P_{i-1}\|}{\sum_{i=2}^{M} \|P_i - P_{i-1}\|}. \tag{11.5}$$

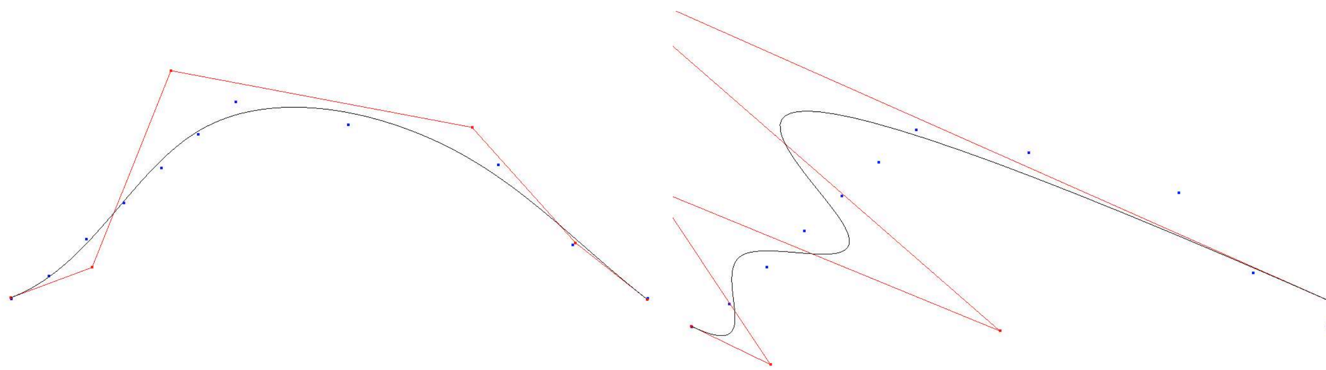

Figure 11.2: Arc length vs. bad parametrization.

The uniform parametrization only suits for regular sample points. The arc length parametrization is much more popular. Here we will give some examples to show the result of least square fitting with different parametrization. In the following examples, the red line is the control polygon and the blue points are the sample points. Figure 11.2,11.1 are two examples of using different parametrization.

11.3.2 Knot vector

Knot vectors are also very important for the curve or surface fitting. As we known, the degree three B-spline function which associated with knot vector $s = [s_0, s_1, s_2, s_3, s_4]$ is zero out of the interval $[s_0, s_4]$, which is called the support of the basis function. The knots and the parameters are the keys for the structure of the matrix M_{fit}. So if the knots around some parameter are very sparse but very dense around some others, then the curve won't be very good, to see 11.3 as an example. The second picture is fitting with a very bas knots which are very dense among the first six knots.

In general, the knots are selected as the subset of the parametrization. Given a integer m, then then knots are s_{j*m} respectively. Bad knots also will lead to very bad spline curve. Figure 11.3 gives some examples. The first picture is fitting with uniform knots.

11.4 Fairing

In order to get smooth curve or surface to fit the given samples points very well, we often add some energy to the equation 11.3. For curve, the item often used is$\int f_{ss}^2$. So the problem is changed to

$$\min \sum_{i=1}^{M}(P_i - \sum_{j=1}^{n} T_j B_j(s_i))^2 + c\int(\sum_{j=1}^{n} T_j B_j(s))^2) \tag{11.6}$$

Here the constant c is defined by user. But this constant is very important for the result curve and it is very difficult to choose the constant. Similarly, the problem also equates the solution of the following linear functions:

$$(M_{fit} + cM_{fair}) * T = B; \tag{11.7}$$

Here the algorithm for computing the fairing matrix is listed as follows:

Algorithm 2 Compute the fairing matrix M_{fair}

Require: all the blending functions B_i
Ensure: $M_{fair} = [b_{ij}]$
  for $i = 1$ to N **do**
    for $j = i$ to$N$ **do**
      $a_{ij} \Leftarrow a_{ji} \Leftarrow d \Leftarrow 0$
      $\Omega_{ij} = Support(B_{iss}) \cap Support(B_{jss}) = [a, b]$
      if $\Omega_{ij} \neq \varnothing$ **then**
        $k \Leftarrow 10, l \Leftarrow 10$
        for $i1 = 1$ to $2k - 1$, $i1+ = 2$ **do**
          $d+ = B_{iss}(a + \frac{i1(b-a)}{2k})B_{jss}(a + \frac{i1(b-a)}{2k})$
        end for
      end if
      $d\times = (b - a)/(k)$
      $a_{ij}+ = d$
      $a_{ji}+ = d$
    end for
  end for

Figure 11.4 illustrated two pictures which are least square fitting and after fairing. We can see that after fairing, the curve becomes much better.

But as we just said, it is very difficult to choose the constant. A larger constant will leads the curve to be shrank, to see Figure 11.5 as an examples.

We must mention here that it is very difficult to smooth a curve with bad parameters. Figure 11.6 gives the example. With a bigger constant, the curve will be shrank as the picture in Figure 11.5.

11.4.1 Interpolation

The interpolation means that we specify some points that the result curve must pass through. Suppose the interpolation points are $Q_i, i = 1, \ldots, I$, and the associated parameter are u_i respectively. Then the problem turns to

$$\min \sum_{i=1}^{M} (P_i - \sum_{j=1}^{n} T_j B_j(s_i))^2 + c \int (\sum_{j=1}^{n} T_j B_j(s))^2)$$

$$\text{subject to: } \sum_{j=1}^{n} T_j B_j(u_i) = Q_i, i = 1, \ldots, I.$$

Using **Lagrange Multiplier method**, we also can change the problem to

$$\min \sum_{i=1}^{M} (P_i - \sum_{j=1}^{n} T_j B_j(s_i))^2 + c \int (\sum_{j=1}^{n} T_j B_j(s))^2) + \sum_{k=1}^{I} \lambda_k (\sum_{j=1}^{n} T_j B_j(u_k) - Q_k)$$

The derivations of the function for all the T_j and λ_k are all zero. all the linear functions can be written as a matrix form:

$$\begin{pmatrix} M_{fit} + cM_{fair} & | & A \\ - - - - - - - - & + & -- \\ A^T & | & 0 \end{pmatrix} [T_1, \ldots, T_n, \lambda_1, \ldots, \lambda_I]^T = [B|C]^T$$

Here A is a matrix with n rows and I columns which element of $i - th$ rows and $j - th$ columns is

$$a_{ij} = B_i(u_j), 1 \le i \le n, 1 \le j \le I.$$

And the element of C is

$$R_j = Q_j, 0 < i \le I.$$

Solve the linear functions, we can compute all the control points for the curve. But the result curve is also associated with the constant. The curve is much less sensitive than that of without interpolation. Figure 11.7 shows an example of interpolation with different constants. In the next section, we will give the method for eliminating the constant.

11.4.2 Constrained fairing

In this section, we will discuss a new idea for constructing a fairing curve or surface.

Denote

$$L_\varepsilon^2 = \{f | f = \sum_{i=1}^{n} T_i B_i(s), Lsq(T_1, T_2, \ldots, T_n) < \varepsilon\}. \tag{11.8}$$

If ε is less than the least square error, then the set is null. The idea for fairing is that select the fairest curve from L^2_ε according to a given error ε. The problem is a convex problem. But it is very difficult to solve. So we want to reduce the condition to turn the problem to a linear condition.

Define a bound box BX_i for each sample point P_i, and we restrict real point belongs to the bound box. That is to say, we want to find the fairest curve from the space:

$$L = \{f | f(s) = \sum_{i=1}^{n} T_i B_i(s), f(u_i \in BX_i)\}. \quad (11.9)$$

Here we will define two kinds of bounding box, the first is very popular, which parallel to the coordinates.

$$BX_i = \{(x, y) | sx_i \le x \le lx_i, sy_i \le y \le ly_i\} \quad (11.10)$$

Then the problem can be divided into two or three independent problems such like that:

$$\begin{aligned} &\min X^T H X \\ &\text{subject to: } AX \le B \end{aligned} \quad (11.11)$$

Here

$$\begin{aligned} X &= [x_1, x_2, \ldots, x_M]; \\ H &= [h_{ij}]_{n\times n}, h_{ij} = \int B_i(s) B_j(s); \\ A &= [a_{ij}]_{2M\times n}, a_{ij} = a_{(i+M)j} = B_j(u_i); \\ B &= [b_i]_{2M}, b_i = lx_i, b_{i+M} = -sx_i. \end{aligned}$$

Here X is a vector of all the control points. H is called the Hessian matrix for the object function and A is the constrain matrix.

The results are listed in Figure 11.8. All the parameters are same as those of in Figure 11.7. If we set very big bound boxes, then we compute a line.

The second boxes are associated with the neighbors of the sample points. We estimate a tangent vector t_i and a normal vector n_i for each point, and the bound box for the point is

$$BX_i = \{P_i + s * t_i + t * n_i | -0.5 \le s \le 0.5, -0.5 \le t \le 0.5, \} \quad (11.12)$$

Then the problem become to a similar problem as 11.11.

$$\begin{aligned} X &= [x_1, x_2, \ldots, x_M, y_1, y_2, \ldots, y_M]; \\ H &= [h_{ij}]_{2n\times 2n}, h_{ij} = h_{(i+n)(j+n)} \int B_i(s) B_j(s); \\ A &= [a_{ij}]_{4M\times 2n}, (a_{ij}, a_{i(j+n)}) = -(a_{(i+M)j}, a_{(i+M)(j+n)}) = B_j(u_i) t_i, \\ &(a_{(i+2M)j}, a_{(i+2M)(j+n)}) = -(a_{(i+3M)j}, a_{(i+3M)(j+n)}) = B_j(u_i) n_i; \\ B &= [b_i]_{4M}, b_i = P_i \cdot t_i + 0.5 t_i \cdot t_i, b_{i+M} = -P_i \cdot t_i + 0.5 t_i \cdot t_i, \\ &b_{i+2M} = P_i \cdot n_i + 0.5 n_i \cdot n_i, b_{i+3M} = -P_i \cdot n_i + 0.5 n_i \cdot n_i, \end{aligned}$$

Solve the quadratic problem, we can get the solution. We can see that it is much better than both least square and fairing.

Here the tangent t_i and normal n_i for each point P_i are estimated with a very simple method. Let $P_0 = P_1$ and $P_{M+1} = P_M$, then the direction of t_i is the same as $P_{i+1} - P_{i-1}$. n_i is the perpendicularity direction of t_i.

11.4.3 Images

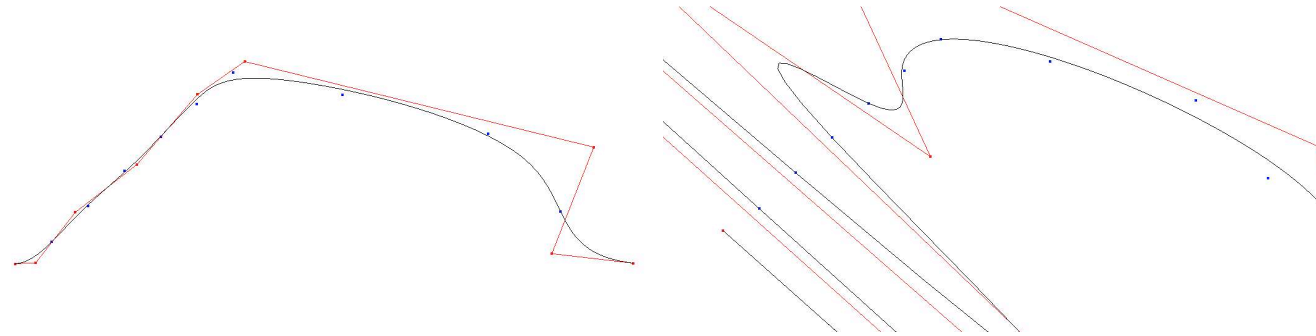

Figure 11.3: Uniform vs. bad knots.

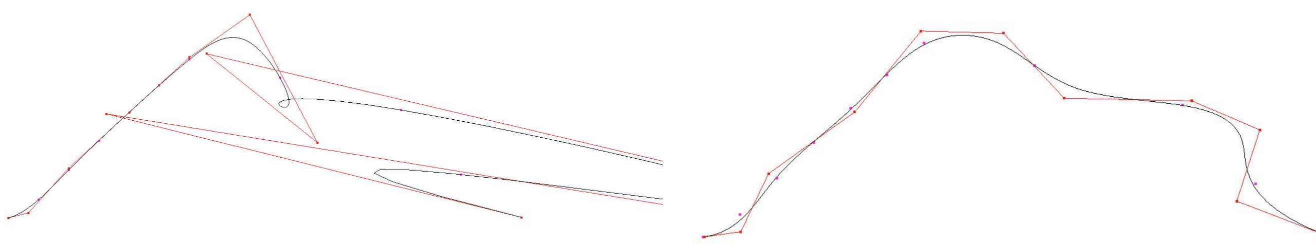

Figure 11.4: The fairing effect.

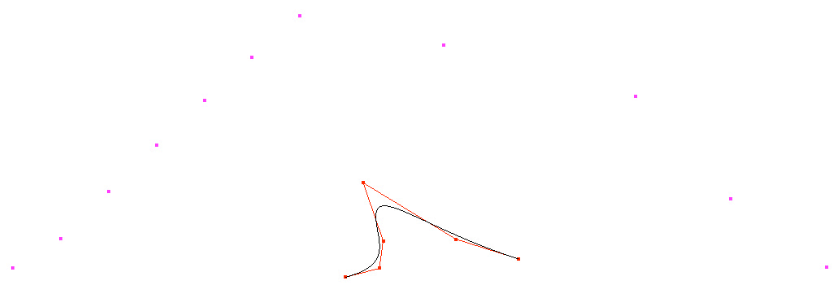

Figure 11.5: The shrank curve.

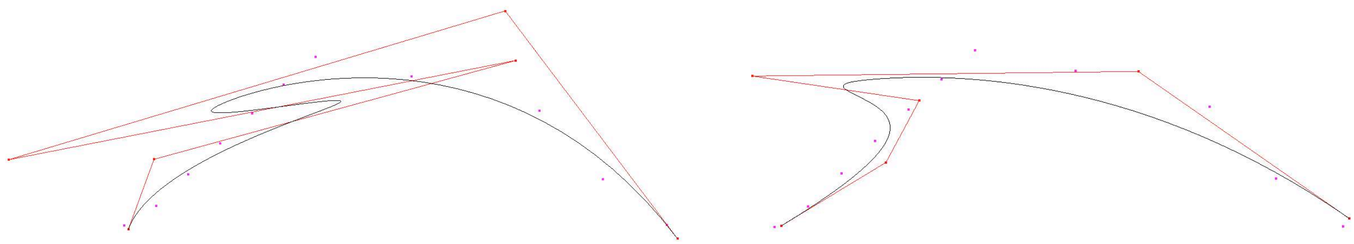

Figure 11.6: The fairing effect of bad parameter.

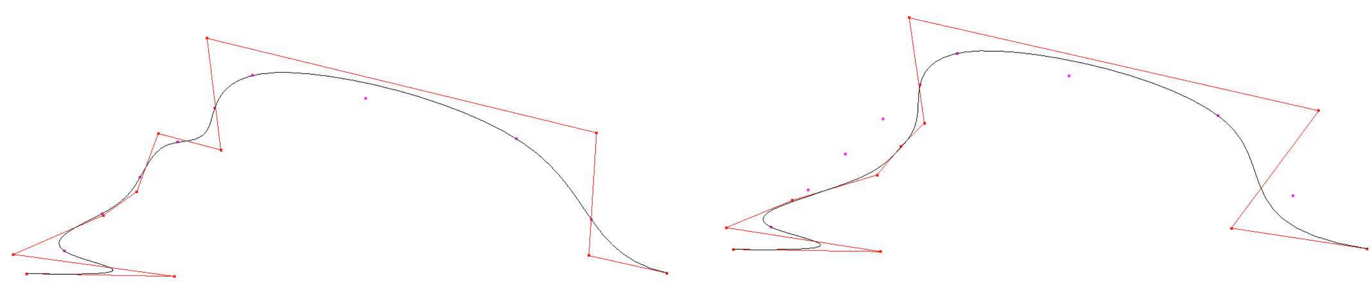

Figure 11.7: Fairing and interpolation with different constant.

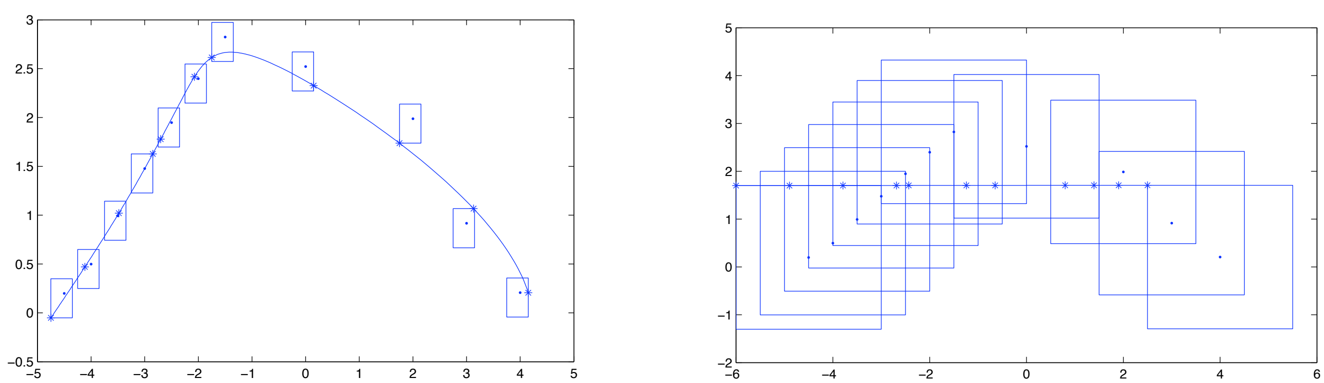

Figure 11.8: Constrained fairing.

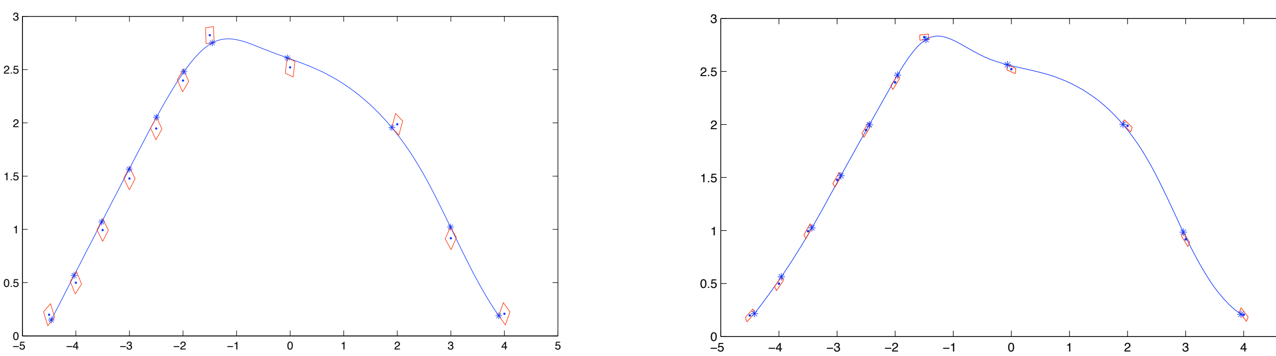

Figure 11.9: Constrained fairing.