A NOVEL MODELING AND SMOOTHING TECHNIQUE IN GLOBAL OPTIMIZATION

Ahmet Sahiner*, Nurullah Yılmaz and Gulden Kapusuz
Suleyman Demirel University, Department of Mathematics
Isparta, 32100, Turkey
(Communicated by Gerhard-Wilhelm Weber)

Abstract. In this paper, we introduce a new methodology for modeling of the given data and finding the global optimum value of the model function. First, a new surface blending technique is offered by using Bezier curves and a smooth objective function is obtained with the help of this technique. Second, a new global optimization method followed by an adapted algorithm is presented to reach the global minimizer of the objective function. As an application of this new methodology, we consider energy conformation problem in Physical Chemistry as a very important real-world problem.

1. Introduction. Surface modeling consists of finding the best continuous function for representing the scattered data points. It is an important issue in Computer-Aided Geometric Design (CAGD) and in many branches of engineering such as mechanic, geology, electric, industrial design and imaging sciences. Various methods for this aim have been investigated depending on the approximation or interpolation in decades [11, 17, 18, 21, 27].

Bezier surface method is one of the most important mathematical spline methods which is commonly used to model the data at hand smoothly. The Bezier surface is defined by its control points, as in the Bezier curve. Bezier curves, began to be developed in 1950. Paul de Casteljau, in fact, is the first scientist developed the Bezier curves in 1959. This approach is comprehensively handled in important books [6, 7]. Bezier surfaces have been used in geometric design for decades as well as being used in data modeling problems in recent years [20, 26].

The blending of surfaces, especially blending of Bezier surfaces is another important issue for CAGD. The blending of surfaces includes two main steps: first, smoothing the transition of each intersecting surfaces; second, obtaining one smooth-ed surface representing each of the surfaces. The common way in literature to blend the surfaces is cutting the region near the intersection curve and filling the gap by a suitable surface patch which contact with the given surfaces at connection points at which the resulting surface is at least first order differentiable [13, 31]. There are many important studies to solve the blending problem parametrically such as [1, 3]. But from the optimization point of view, we need further information about the rectangular form of surfaces and their continuous derivatives.

2010 Mathematics Subject Classification. Primary: 90C26; Secondary: 65D05, 65D17, 97M10.
Key words and phrases. Global optimization, blending surface, data modeling, smoothing technique.

* Corresponding author: ahmetsahiner@sdu.edu.tr.
The global optimization is one of the most important parts of the optimization. There are different types of global optimization methods and algorithms based on deterministic, stochastic and heuristic ideas. Deterministic methods are one of the most reliable approaches among the others and they also have some advantages over the others in terms of easy implementations [25, 34]. One of the important deterministic methods is the auxiliary function approach. This approach is usually based on finding a minimizer lower than the current one by making some suitable modification on the objective function. The outstanding auxiliary function approaches are Tunneling Algorithm, Filled Function Method and Global Descent Method [8, 9, 15, 19, 28, 30].

The global optimization methods have recently been applied in molecular conformation problems, significantly. The threonine molecule $C_4H_9NO_3$ used in this article is considered in many papers and the chemical knowledge about the molecule can be found in [4, 12]. Density Functional Theory (DFT) approach is an important way for stating all the ground-state properties of the electronic structure of atoms and molecules. It is used for calculating the structural energy values of the molecule depending on the torsion angles [14, 22].

In this study, we first propose a new surface construction method and present a new global optimization method. As an application, we consider the data obtained from DFT calculations [23]. Second, we take a partition of the obtained data and we construct Bezier surface patches by using the data for each element of the partitions. We design a new approach in order to blend these surface patches and we obtain a smooth model defined on $\Omega$. Finally, we apply our new global optimization method to find the global minimizer of this model function. Hence, we realize that besides giving successful results on many important test problems in the literature, this newly offered methodology also gives satisfactory results on the real-world problems.

2. Preliminaries. Throughout the paper, $\mathbb{R}_+$ denotes the non-negative real numbers. In order to obtain a continuous model from the data, we consider Bezier surface modeling approach. Bezier surface of degree $(n, m)$ is defined by a set of $(n+1)(m+1)$ control points $P_{i,j}$. A two dimensional Bezier surface can be defined as a parametric surface where the position of a point as a function of the parametric coordinates $u, v$ is given by:

$$S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B^n_i(u) B^m_j(v) P_{i,j},$$

where

$$B^n_i(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

is a Bernstein polynomial, and

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

is the binomial coefficient. The Bezier surfaces are smooth and the computing of derivatives of these surfaces are quite easy. Several properties of the Bezier functions make them very useful for approximating other functions. The piecewise smooth functions are commonly used in many surface constructions and they are studied in many important papers in terms of optimization such as [10, 29]. The piecewise smooth function is defined by the following:
Definition 2.1. [32] A function \( f : (\emptyset \neq \Omega) \subset \mathbb{R}^n \rightarrow \mathbb{R} \) is called piecewise smooth on \( \Omega \) if \( f \) is continuous on \( \Omega \), there exists a family of smooth functions \( f_i : \Omega \rightarrow \mathbb{R} \), and there exists a family of sets \( A_i \subset \Omega \) and \( i = 1, 2, \ldots, m \) ( \( \bigcup_{i=1}^{m} A_i = \Omega \) and \( A_i \cap A_j = \emptyset \) for \( i \neq j \)), such that

\[
f(x) = f_i(x), \quad \forall x \in A_i, \ i = 1, 2, \ldots, m.
\]

According to the above definition, the piecewise smooth function can be re-written as

\[
f(x) = \sum_{i=1}^{m} f_i(x) \chi_{A_i}(x) = \sum_{i=1}^{m} f(x) \chi_{A_i}(x),
\]

where \( \chi_{A_i}(x) \) is characteristic function of the set \( A_i \) which is defined by

\[
\chi_{A_i}(x) = \begin{cases} 
1, & x \in A_i, \\
0, & x \notin A_i.
\end{cases}
\]

The smoothing function is described by the following definition:

Definition 2.2. [2] Let \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) be a continuous function. The function \( \tilde{f} : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R} \) is called a smoothing function of \( f(x) \), if \( \tilde{f}(\cdot, \tau) \) is continuously differentiable in \( \mathbb{R}^n \) for any fixed \( \tau \), and for any \( x \in \mathbb{R}^n \),

\[
\lim_{z \rightarrow x, \tau \rightarrow 0} \tilde{f}(z, \tau) = f(x).
\]

We use \( C(\Omega) \) for the set of all the continuous functions on \( \Omega \subset \mathbb{R}^n \) and \( C^k(\Omega) \) for the set of all functions whose \( k \)-th order partial derivatives exist and continuous on \( \Omega \). We use \( C^k \) for \( C^k(\mathbb{R}^n) \) and define the following set

\[
C^{a,p} = \left\{ f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R} : \frac{\partial^{p+s}}{\partial x^p \partial y^s} f \in C(\Omega), 0 \leq r \leq p, 0 \leq s \leq q \right\}.
\]

We also recall the following definitions.

Definition 2.3. [8] The basin of \( f(x) \) at an isolated minimum \( x_0 \) is a connected domain \( B(x_0) \) containing \( x_0 \) only in which starting from any point the steepest descent trajectory of \( f(x) \) converges to \( x_0 \).

Definition 2.4. [8] The simple basin of \( f(x) \) at an isolated minimum \( x_0 \) is a connected domain \( S(x_0) \subset B(x_0) \) such that for any \( x \neq x_0 \), \( (x - x_0)^T \nabla f(x) > 0 \).

3. Smooth blending of surface patches. Assume \( \Omega \subset \mathbb{R}^2 \) is a rectangular bounded domain equipped with the rectangular partition. Assume that continuously differentiable bivariate functions are defined on each element of the partitions. In fact, let the partitions of intervals \( I = [a, b] \) and \( J = [c, d] \) defined as \( \Delta_n = \{a = a_0 < a_1 < \cdots < a_n = b\} \) and \( \Delta_m = \{e = e_0 < e_1 < \cdots < e_m = d\} \), respectively. The rectangular partitions of \( \Omega = I \times J \) is defined as

\[
\Delta_{n,m} = \{D_{i,j} = [a_i, a_{i+1}] \times [c_j, c_{j+1}], i = 0, 1, \ldots, n-1, \ j = 0, 1, \ldots, m-1\}.
\]

Let \( f_{i,j} \) be continuously differentiable functions on \( D_{i,j} \). We define the following piecewise smooth function:
Let \( f : \Omega \subset \mathbb{R}^2 \to \mathbb{R} \) be defined by

\[
f(x, y) = \begin{cases} 
    f_{0,0}(x, y), & (x, y) \in D_{0,0} = [a_0, a_1] \times [c_0, c_1], \\
    f_{1,0}(x, y), & (x, y) \in D_{1,0} = [a_1, a_2] \times [c_0, c_1], \\
    \vdots & \\
    f_{i,j}(x, y), & (x, y) \in D_{i,j} = [a_i, a_{i+1}] \times [c_j, c_{j+1}], \\
    \vdots & \\
    f_{n-1,m-1}(x, y), & (x, y) \in D_{n-1,m-1} = [a_{n-1}, b] \times [c_{n-1}, d],
\end{cases}
\]

where the functions \( f_{i,j} : \mathbb{R}^2 \to \mathbb{R} \) \( (0 \leq i \leq n-1 \) and \( 0 \leq j \leq m-1 \) \) are continuously differentiable. In general, \( f \) is not differentiable at the knot points. Therefore, the function \( f \) is not proper to apply minimization techniques including gradients.

Equivalently, we consider the following reformulation of \( f \) as

\[
f(x, y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} f_{i,j}(x, y) \chi_{D_{i,j}}(x, y),
\]

where \( \chi_{D_{i,j}}(x, y) \) is characteristic function of a set \( D_{i,j} \) which is defined by

\[
\chi_{D_{i,j}}(x, y) = \begin{cases} 
    1, & (x, y) \in D_{i,j}, \\
    0, & (x, y) \notin D_{i,j}.
\end{cases}
\]

Since \( D_{i,j} = [a_i, a_{i+1}] \times [c_j, c_{j+1}] \), we re-state (3) by

\[
\chi_{D_{i,j}}(x, y) = \chi_{[a_i, a_{i+1}]}(x) \chi_{[c_j, c_{j+1}]}(y).
\]

Hence, the function \( f \) can be defined by

\[
f(x, y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} f_{i,j}(x, y) \chi_{[a_i, a_{i+1}]}(x) \chi_{[c_j, c_{j+1}]}(y).
\]

Considering the above formulation, we can observe that the non-smoothness of \( f \) stems from the non-smoothness of characteristic functions. If we smooth out the characteristic functions, then \( f(x, y) \) will also be a smooth function. In order to smooth out the characteristic functions, we consider the following smoothing functions for characteristic functions of subintervals of \( I \)

\[
\tilde{\chi}_{[a_i, a_{i+1}]}(x, \varepsilon) = \begin{cases} 
    0, & x \leq a_i - \varepsilon, \\
    P_1(-(x - a_i)), & a_i - \varepsilon < x < a_i + \varepsilon, \\
    1, & a_i + \varepsilon \leq x \leq a_{i+1} - \varepsilon, \\
    P_1(x - a_{i+1}), & a_{i+1} - \varepsilon < x < a_{i+1} + \varepsilon, \\
    0, & x \geq a_{i+1} + \varepsilon,
\end{cases}
\]

for \( i = 1, 2, \ldots, n-2 \) and for border intervals \( [a_0, a_1] \) and \( [a_{n-1}, a_n] \) the smoothing functions of characteristic functions are given as the following, respectively.

\[
\tilde{\chi}_{[a_0, a_1]}(x, \varepsilon) = \begin{cases} 
    1, & a_0 \leq x \leq a_1 - \varepsilon, \\
    P_1(x - a_1), & a_1 - \varepsilon < x < a_1 + \varepsilon, \\
    0, & x \geq a_1 + \varepsilon,
\end{cases}
\]

and

\[
\tilde{\chi}_{[a_{n-1}, a_n]}(x, \varepsilon) = \begin{cases} 
    0, & x \leq a_{n-1} - \varepsilon, \\
    P_1(-(x - a_{n-1})), & a_{n-1} - \varepsilon < x < a_{n-1} + \varepsilon, \\
    1, & a_{n-1} + \varepsilon \leq x \leq a_n,
\end{cases}
\]
where
\[ P_1(x - a_i) = \frac{1}{4\varepsilon^3} (x - a_i)^3 - \frac{3}{4\varepsilon} (x - a_i) + \frac{1}{2}, \]
for \( i = 1, 2, \ldots, n - 1 \). The smoothing functions \( \tilde{\chi}_{[c_j, c_{j+1}]}(y, \varepsilon) \) for subintervals \([c_j, c_{j+1}]\) for \( j = 0, 1, \ldots, m - 1 \) can be defined in a similar way and any \( 0 < \varepsilon < \min_{i,j} \left\{ \frac{a_{i+1} - a_i}{2}, \frac{c_{j+1} - c_j}{2} \right\} \).

**Lemma 3.1.** Let \( \chi_{[a,b]}(x) \) be a characteristic function of the interval \([a, b] \subset \mathbb{R} \) and \( \tilde{\chi}_{[a,b]}(x, \varepsilon) \) be a smoothing function. Then, we have
\[ |\tilde{\chi}_{[a,b]}(x, \varepsilon) - \chi_{[a,b]}(x)| \leq \frac{1}{2} \]
for any \( \varepsilon > 0 \).

**Proof.** For any \( \varepsilon > 0 \), the function \( \tilde{\chi}_{[a,b]}(x, \varepsilon) \) and \( \chi_{[a,b]}(x) \) are equal on \([a + \varepsilon, b - \varepsilon], (-\infty, a - \varepsilon] \) and \([b + \varepsilon, +\infty) \). Therefore, we consider the cases \( x \in [a - \varepsilon, a + \varepsilon] \) and \( x \in [b - \varepsilon, b + \varepsilon] \). For \( x \in [a - \varepsilon, 0] \) we have
\[ |\tilde{\chi}_{[a,b]}(x, \varepsilon) - \chi_{[a,b]}(x)| = |P_1(x - a)| \leq \frac{1}{2}, \]
and for \( x \in [0, a + \varepsilon] \)
\[ |\tilde{\chi}_{[a,b]}(x, \varepsilon) - \chi_{[a,b]}(x)| = |P_1(x - a) - 1| \leq \frac{1}{2}. \]
Similarly, the same results can be obtained for \([b - \varepsilon, b + \varepsilon]\). \( \square \)

It can easily be observed that \( \tilde{\chi}_{[a,b]}(x, \varepsilon) \to \chi_{[a,b]}(x) \) almost everywhere as \( \varepsilon \to 0 \).

By considering the above smoothing functions of characteristic functions, the smoothing function of \( f \) is defined as
\[ \tilde{f}(x, y, \varepsilon) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} f_{i,j}(x, y) \tilde{\chi}_{[a_i, a_{i+1}]}(x, \varepsilon) \tilde{\chi}_{[c_j, c_{j+1}]}(y, \varepsilon). \quad (5) \]

By applying the above smoothing process to the function \( f \), it can be stated that the function \( f \) is of class \( C^{1,1} \). If the functions \( f_{i,j} \) are of class \( C^k \), \( k \geq 2 \), and if \( f \) is required to have second order continuous derivatives, this can be achieved by taking \( P_2 \) instead of \( P_1 \) where
\[ P_2(x - a_i) = \frac{3}{16\varepsilon^5} (x - a_i)^5 - \frac{10}{16\varepsilon^3} (x - a_i)^3 + \frac{15}{16\varepsilon} (x - a_i) + \frac{1}{2}. \]

**Theorem 3.2.** Let \( f \) be continuous function on \( \Omega \) and the function \( \tilde{f}(x, y, \varepsilon) \) is defined as in (5). Then, the function \( \tilde{f}(x, y, \varepsilon) \) is a smooth function and for any fixed \((x, y)\), we have
\[ \lim_{\varepsilon \to 0} \tilde{f}(x, y, \varepsilon) = f(x, y) \]
for \( \varepsilon > 0 \).

**Proof.** The smoothing functions of characteristic functions \( \tilde{\chi}_{[a_i, a_{i+1}]}(x, \varepsilon) \) and \( \tilde{\chi}_{[c_j, c_{j+1}]}(y, \varepsilon) \) are of class \( C^k \) for \( \varepsilon > 0 \) and \( f_i \) is of class \( C^k \) for each \( i \) then, \( f_i(x) \tilde{\chi}_{[a_i, a_{i+1}]}(x, \varepsilon) \tilde{\chi}_{[c_j, c_{j+1}]}(y, \varepsilon) \) are of class \( C^k \) for \( k \geq 1 \). According to equation (5), \( \tilde{f}(x, y, \varepsilon) \) is of class \( C^k \). \( \square \)
Theorem 3.3. Let $f$ be continuous function on $\Omega$ and $\hat{f}(x, y, \varepsilon)$ defined as in (5). Then, we have

$$|\hat{f}(x, y, \varepsilon) - f(x, y)| \leq \max_{i,j} \{A_{i,j}(x, y), B_{i,j}(x, y)\},$$

where

$$A_{i,j}(x, y) = \max_{i=0,1} \max_{k,l=0,1} |(f_{i-k,j-l} - f_{i-l,j-l}) (x, y)|$$

and

$$B_{i,j}(x, y) = \frac{1}{4} |(f_{i-1,j-1} + f_{i,j} - f_{i-1,j} - f_{i,j-1}) (x, y)|$$

$$+ \frac{1}{2} \max_{i=0,1} \max_{k,l=0,1} |(f_{i-0-k,j-0-l} - f_{i-0-t,j-0-t}) (x, y)|$$

$$+ \frac{1}{2} \max_{k,l=0,1} \max_{i=0,1} |(f_{i-0-j-0-k} - f_{i-0-r,j-0-r}) (x, y)|$$

for $\varepsilon > 0$ and $(x, y) \in \Omega$.

Proof. For any $\varepsilon$, the function $\hat{f}(x, y, \varepsilon)$ and $f(x, y)$ are equal outside of the plus shaped neighborhood of knot points $S_{i,j} = (a_i - \varepsilon, a_i + \varepsilon) \times (c_j, c_j) \cup (a_0, a_n) \times (c_j - \varepsilon, c_j + \varepsilon)$ for $i = 0, 1, \ldots, n-1$ and $j = 0, 1, \ldots, m-1$. Therefore, it is sufficient to prove that the inequality (6) holds on $S_{i,j}$. Let us choose any knot point $(a_{i_0}, c_{j_0})$. We present the proof by considering two important cases.

Case 1. Assume first that $(x, y) \in (a_{i_0} - \varepsilon, a_{i_0}) \times (c_{j_0} - 1, c_{j_0} - \varepsilon)$.

$$|\hat{f}(x, y, \varepsilon) - f(x, y)| = \left| \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} f_{i,j}(x, y) \tilde{\chi}_{[a_i, a_{i+1}]}(x, \varepsilon) \tilde{\chi}_{[c_j, c_{j+1}]}(y, \varepsilon) \right|$$

$$- \left| \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} f_{i,j}(x, y) \chi_{[a_i, a_{i+1}]}(x) \chi_{[c_j, c_{j+1}]}(y) \right|$$

$$= \left| f_{i_0-1,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0-1}, a_{i_0}]}(x, \varepsilon) \right. + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0+1}]}(x, \varepsilon)$$

$$\left. - f_{i_0-1,j_0-1}(x, y) \chi_{[a_{i_0-1}, a_{i_0}]}(x, \varepsilon) \right|$$

$$= \left| f_{i_0-1,j_0-1}(x, y) \left( \tilde{\chi}_{[a_{i_0-1}, a_{i_0}]}(x, \varepsilon) - 1 \right) \right. + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0+1}]}(x, \varepsilon)$$

Since $\tilde{\chi}_{[a_{i_0-1}, a_{i_0}]}(x, \varepsilon) - 1 = -\tilde{\chi}_{[a_{i_0}, a_{i_0+1}]}(x, \varepsilon)$ and $\tilde{\chi}_{[a_{i_0}, a_{i_0+1}]}$ takes the value at most $1/2$ on the segment $x \in (a_{i_0} - \varepsilon, a_{i_0})$, we have

$$|\hat{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{2} |(f_{i_0,j_0-1} - f_{i_0-1,j_0-1}) (x, y)|. \quad (7)$$

By the similar way, the same result is obtained for $(x, y) \in (a_{i_0}, a_{i_0} + \varepsilon) \times (c_{j_0} - 1, c_{j_0} - \varepsilon)$ as

$$|\hat{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{2} |(f_{i_0-1,j_0-1} - f_{i_0,j_0-1}) (x, y)|.$$
By considering the above approach, for \((x, y) \in (a_{i_0} - \varepsilon, a_i) \times (c_{j_0}, c_{j_0+1} - \varepsilon)\) we have

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{2} |(f_{i_0,j_0 - 1} - f_{i_0,j_0}) (x, y)|. \tag{8}
\]

Similarly, for \((x, y) \in (a_{i_0} - 1, a_{i_0} - \varepsilon) \times (c_{j_0} - \varepsilon, c_{j_0} + \varepsilon)\) we have

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{2} |(f_{i_0,j_0-1} - f_{i_0,j_0}) (x, y)| \tag{9}
\]

and for \((x, y) \in (a_{i_0}, a_{i_0} + 1 - \varepsilon) \times (c_{j_0} - \varepsilon, c_{j_0} + \varepsilon)\), we have

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{2} |(f_{i_0,j_0} - f_{i_0,j_0-1}) (x, y)|. \tag{10}
\]

From the above equations (7), (8), (9) and (10) we obtain

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| \leq \max_{t=0,1} \max_{k,l=0,1} |(f_{i_0-k,j-l} - f_{i_0-1,j_0-l}) (x, y)|. \tag{11}
\]

**Case 2.** Now, let us consider the case \((x, y) \in (a_{i_0} - \varepsilon, a_{i_0}) \times (c_{j_0} - \varepsilon, c_{j_0})\) then, we have

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| = |f_{i_0,j_0-1}(x, y) \left( \tilde{\chi}_{[a_{i_0} - 1, a_{i_0}]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0} - 1, c_{j_0}]}(y, \varepsilon) - 1 \right) + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0} - 1, a_{i_0}]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0} - 1, c_{j_0}]}(y, \varepsilon) + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0} - 1, a_{i_0}]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) - f_{i_0,j_0-1}(x, y) \right) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0} - 1, a_{i_0}]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0} - 1, a_{i_0}]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0-1}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon) + f_{i_0,j_0}(x, y) \tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon) \tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon)
\]

Since \(\tilde{\chi}_{[a_{i_0}, a_{i_0}+1]}(x, \varepsilon)\), \(\tilde{\chi}_{[c_{j_0}, c_{j_0}+1]}(y, \varepsilon)\), takes the value at most 1/2 on the segments \(x \in (a_{i_0} - \varepsilon, a_{i_0})\) and \(y \in (c_{j_0} - \varepsilon, c_{j_0})\) we have

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{4} |(f_{i_0,j_0-1} + f_{i_0,j_0 - f_{i_0,j_0-1}} - f_{i_0,j_0-1}) (x, y)| + \frac{1}{2} |(f_{i_0,j_0-1} - f_{i_0,j_0-1}) (x, y)|. \tag{12}
\]

By considering similar approach, the same result is obtained for \((x, y) \in (a_{i_0}, a_{i_0} + \varepsilon) \times (c_{j_0} - \varepsilon, c_{j_0})\) as

\[
|\tilde{f}(x, y, \varepsilon) - f(x, y)| \leq \frac{1}{4} |(f_{i_0,j_0} + f_{i_0,j_0} - f_{i_0,j_0-1} - f_{i_0,j_0}) (x, y)| + \frac{1}{2} |(f_{i_0,j_0} - f_{i_0,j_0-1}) (x, y)| + \frac{1}{2} |(f_{i_0,j_0} - f_{i_0,j_0-1}) (x, y)|. \tag{13}
\]
By considering similar approach, the same result is obtained for \((x, y) \in (a_i, a_i + \varepsilon) \times (c_j, c_j + \varepsilon)\) as

\[
|\tilde{f}(x, y) - f(x, y)| \leq \frac{1}{4} |(f_{i_0-1,j_0} + f_{i_0,j_0} - f_{i_0,j_0-1} - f_{i_0,j_0-1})(x, y)| + \frac{1}{2} |(f_{i_0-1,j_0} - f_{i_0,j_0})(x, y)| + \frac{1}{2} |(f_{i_0,j_0} - f_{i_0,j_0-1})(x, y)|.
\]

By considering similar approach, the same result is obtained for \((x, y) \in (a_i, a_i + \varepsilon) \times (c_j, c_j + \varepsilon)\) as

\[
|\tilde{f}(x, y) - f(x, y)| \leq \frac{1}{4} |(f_{i_0-1,j_0} + f_{i_0,j_0} - f_{i_0-1,j_0} - f_{i_0,j_0})(x, y)| + \frac{1}{2} |(f_{i_0-1,j_0} - f_{i_0,j_0})(x, y)| + \frac{1}{2} |(f_{i_0,j_0} - f_{i_0,j_0-1})(x, y)|.
\]

From the above inequalities, we obtain

\[
|\tilde{f}(x, y) - f(x, y)| \leq \frac{1}{4} |(f_{i_0-1,j_0} + f_{i_0,j_0} - f_{i_0-1,j_0} - f_{i_0,j_0})(x, y)| + \frac{1}{2} \max_{l=0,1} \max_{k \neq l} |(f_{i_0-k,j_0-l} - f_{i_0-l,j_0-l})(x, y)| + \frac{1}{2} \max_{k \neq l} \max_{r=0,1} |(f_{i_0-l,j_0-k} - f_{i_0-r,j_0-r})(x, y)|
\]

Taking all \(i = 0, 1, \ldots, n-1\) and \(j = 0, 1, \ldots, n-1\) into consideration, we obtain inequality (6).

Our surface blending technique can easily be extended to multi dimensional case with ideas very similar to those in two dimensional case. Let us define the following piecewise smooth function from \(\mathbb{R}^n\) to \(\mathbb{R}\) as

\[
f(x_1, \ldots, x_m) = \sum_{i_1=0}^{n_1-1} \cdots \sum_{i_m=0}^{n_m-1} f_{i_1, \ldots, i_m}(x_1, \ldots, x_m) \chi_{[a_{i_1}, a_{i_1+1}]}(x_1) \cdots \chi_{[a_{i_m}, a_{i_m+1}]}(x_m).
\]

By using the smoothing functions of each of the characteristic functions we obtain the smoothing function

\[
\tilde{f}(x_1, \ldots, x_m, \varepsilon) = \sum_{i_1=0}^{n_1-1} \cdots \sum_{i_m=0}^{n_m-1} f_{i_1, \ldots, i_m}(x_1, \ldots, x_m) \tilde{\chi}_{[a_{i_1}, a_{i_1+1}]}(x_1, \varepsilon) \cdots \tilde{\chi}_{[a_{i_m}, a_{i_m+1}]}(x_m, \varepsilon).
\]

In this smoothing approach, all the functions \(f_{i_1, \ldots, i_m}, i_j = 1, \ldots, n_j - 1, 1 < j < n\) must be smooth. The difference between \(f(x)\) and \(f(x, \varepsilon)\) can be controlled by the parameter \(\varepsilon\).
4. Global minimization. In this section, we introduce a new global optimization method based on the auxiliary function approach. We consider the following problem:

\[(P) \quad \min_{\Omega \subset \mathbb{R}^n} f(x)\]

where \(\Omega\) is a box-shaped subset of \(\mathbb{R}^n\) and \(f\) is continuously differentiable function on \(\mathbb{R}^n\). It is assumed that the function \(f\) has a finite number of local minimizers.

The method is constructed by being inspired from the study \([24, 33]\). Now, we are giving information about modus operandi of our global optimization technique.

Assume that \(x_k^*\) is local minimizer of the objective function \(f(x)\) and the corresponding value of \(f\) at \(x_k^*\) is \(f_k^*\). We construct the following function

\[\phi(x, x_k^*) = \min\{f(x), f_k^*\}\]

which has no minimizer upper than \(x_k^*\). Equivalently, we can re-state \(\phi(x, x_k^*)\) as

\[\phi(x, x_k^*) = f_k^* + \min\{f(x) - f_k^*, 0\},\]

It can be observed that the function \(\phi(x, x_k^*)\) has no (strict) minimizer upper than \(x_k^*\) and the minimizers of \(\phi(x, x_k^*)\) lower than \(x_k^*\) are the same as those of the objective function \(f(x)\). However, there are two important difficulties in terms of finding lower minimizer of \(\phi(x, x_k^*)\). The first one is the non-smoothness of the function \(\phi(x, x_k^*)\).

In order to eliminate this problem, we propose to use the smoothing functions for \(\phi(x, x_k^*)\). Before using smoothing approach, we first re-state the function \(\phi(x, x_k^*)\) as

\[\phi(x, x_k^*) = f_k^* + (f(x) - f_k^*) \chi_{A_k}(t),\]

where \(A_k = \{t \in \mathbb{R} : t = f(x) - f_k^* < 0\}\) and \(\chi_{A_k} : \mathbb{R} \to \{0, 1\}\) is characteristic function of the set \(A_k\) which is defined

\[\chi_{A_k}(t) = \begin{cases} 
1, & t < 0, \\
0, & t \geq 0.
\end{cases}\]

Based on the Equation (4) and the results in Section 3, we design the following smoothing function:

\[\tilde{\phi}(x, x_k^*, \beta) = f_k^* + (f(x) - f_k^*) \tilde{\chi}_{A_k}(t, \beta),\]

where \(\tilde{\chi}_{A_k}(x, \beta)\) is characteristic function of \(A_k\) which is defined by

\[\tilde{\chi}_{A_k}(t, \beta) = \begin{cases} 
0, & t > \beta, \\
q(t, \beta), & -\beta \leq t \leq \beta, \\
1, & t < -\beta,
\end{cases}\]

where

\[q(t, \beta) = \frac{1}{4\beta^3} t^3 - \frac{3}{4\beta} t + \frac{1}{2} .\]

**Theorem 4.1.** Let \(\beta > 0\) and \(x_k^*\) be a minimizer of the function \(f\) then, we have the following:

i. The function \(\tilde{\phi}(x, x_k^*, \beta)\) continuously differentiable,

ii. \(\tilde{\phi}(x, x_k^*, \beta)\) approaches \(\phi(x, x_k^*)\), when \(\beta \to 0\).

The second problem is the stability of the function \(\tilde{\phi}(x, x_k^*, \beta)\) for \(x \in \Omega\) such that \(f(x) > f_k^*\). In this condition, finding of the lower minimizer may be very hard and very costly. In order to reach lower local minimizers without expensive costs, we add the function \(H(\|x - x_k^*\|^2)\) to \(\tilde{\phi}(x, x_k^*, \beta)\). The function \(H\) is defined on \(\mathbb{R}_+\) and it satisfies the following properties:
For any $H(u) > 0$

ii. $H(u) < 0$

iii. $\lim_{u \to \infty} H(u) = 0$.

Important examples of $H$ include $\frac{u}{u^2}$, $\arctan(u)$, and $\exp(-u) + \frac{1}{1+u^2}$. Finally, our auxiliary function (Fast Delving function) is defined as follows:

$$\tilde{\phi}(x, x^*_k, \beta, \alpha) = f_k^* + (f(x) - f_k^*) \chi_k(t, \beta) + \alpha H(||x - x_k^*||^2),$$

(14)

where $\alpha$ and $\beta$ are real parameters.

**Theorem 4.2.** Let $x_k^*$ be a local minimizer of the function $f$, then the function $\tilde{\phi}(x, x_k^*, \beta, \alpha)$ which is defined in (14) has no stationary point for all $x \in U_1 = \{x \in \Omega : f(x) > f_k^*\}$.

**Proof.** For any $x \in U_1$, $x \neq x_k^*$ we have $\tilde{\phi}(x, x_k^*, \beta, \alpha) = f_k^* + \alpha H(||x - x_k^*||^2)$ and

$$\nabla\tilde{\phi}(x, x_k^*, \beta, \alpha) = \alpha \nabla H(||x - x_k^*||^2) \neq 0.$$ 

**Theorem 4.3.** Let $x_k^*$ be a local minimizer of the function $f$. If the function $f(x)$ has a local minimizer $x_{k+1}^*$ lower than $x_k^*$, then there exists a local minimizer $\tilde{x}$ of $\tilde{\phi}(x, x_k^*, \beta, \alpha)$ defined in (14) such that $\tilde{x} \in S(x_{k+1}^*)$.

**Proof.** Suppose that the objective function $f(x)$ has a local minimizer $x_{k+1}^*$ lower than $x_k^*$. It can be deduced that the set $U_2 = \{x : f(x) \leq f_k^*, x \in \Omega\}$ is not empty. Moreover, $U_2$ is closed since $f(x)$ continuous and bounded since it is contained by $\Omega$. Thus, the function $\tilde{\phi}(x, x_k^*, \beta, \alpha)$ has a minimizer in $U_2$.

Let $\tilde{x}$ be a local minimizer of the function $\tilde{\phi}(x, x_k^*, \beta, \alpha)$, then we have

$$\nabla\tilde{\phi}(\tilde{x}, x_k^*, \beta, \alpha) = \nabla f(\tilde{x}) + \alpha \nabla H(||\tilde{x} - x_k^*||^2) = 0.$$ 

Therefore, we have $\nabla f(\tilde{x}) = -\alpha \nabla H(||\tilde{x} - x_k^*||^2)$. From the definition of the function $H(||x - x_k^*||^2)$, we have

$$(\tilde{x} - x_k^*) \nabla f(\tilde{x}) = (\tilde{x} - x_k^*) \left(-\alpha \nabla H(||\tilde{x} - x_k^*||^2)\right) > 0.$$ 

Since $\tilde{x} \rightarrow x_{k+1}^*$ as $\alpha \rightarrow 0$, $\tilde{x}$ is close enough to $x_{k+1}^*$, $(\tilde{x} - x_k^*)$ is close enough to $(x_{k+1}^* - x_k^*)$ and $||\tilde{x} - x_k^*|| \geq ||x_{k+1}^* - x_k^*||$. Then, the vectors $(\tilde{x} - x_k^*)$ and $(\tilde{x} - x_{k+1}^*)$ are almost in the same directions. Therefore, we obtain

$$(\tilde{x} - x_k^*) \nabla f(\tilde{x}) = (\tilde{x} - x_{k+1}^*) \left(-\alpha \nabla H(||\tilde{x} - x_k^*||^2)\right) > 0.$$ 

It is clear that $\tilde{x} \in S(x_{k+1}^*)$ and it completes the proof.

By the help of the above auxiliary function, our method works as follows:

1. Find the local minimizer $x_k$ of the given objective function $f(x)$ by applying local solver.
2. Consider the function $\tilde{\phi}(x, x_k^*, \beta, \alpha)$ and find the local minimizer $x_{k+1}^*$ such that $f_k^* < f_{k+1}^*$.
3. Set $x_k^* = x_{k+1}^*$ until the stopping criteria is supplied.

By using the auxiliary function $\phi(x, x_k^*, \beta, \alpha)$, we design the Algorithm 1 (Fast Delving Algorithm) for global optimization process. Before the real-world application, we first apply our Algorithm 1 to test problems and compare the results with the other methods. The proposed algorithm is programmed in Matlab and has been implemented on Intel Core i5-3337U 1.8GHz with 7.8 Gb Ram. We consider the important test problems which are shown in Table 1.
Algorithm 1

Step 0. Set \( k = 1, \beta = 0.1, \alpha = 0.5, \epsilon = 10^{-2}, \varepsilon = 10^{-20}, R = 10 \), boundary \( M \) for the parameter \( \alpha \), the maximum number of directions \( N \), the directions \( d_i \) for \( i = 1, 2, \ldots, N \) and determine boundary of \( \Omega \).

Step 1. Find the first local minimizer \( x_k^* \) of the objective function \( f(x) \) starting from the point \( x_0 \).

Step 2. Construct the function \( \tilde{\phi}(x, x_k^*, \beta, \alpha) \) and set \( i = 1 \).

Step 3. Use \( x_0 = x_k^* + \epsilon d_i \) as a starting point and find the minimizer of \( \tilde{\phi}(x, x_k^*, \beta, \alpha) \) and denote it as \( x_s \).

Step 4. If \( x_s \in \Omega \), then go to Step 5; otherwise, go to Step 6.

Step 5. Take \( x_0 = x_s \) and go to Step 1.

Step 6. If \( i < N \), set \( i = i + 1 \) \((d_i \rightarrow d_{i+1})\) and go to Step 3; otherwise go to Step 7.

Step 7. If \( \alpha \geq M \) or the gradient of \( \tilde{\phi}(x, x_k^*, \beta, \alpha) \) vanishes outside of the search domain or \(|f_k^* - f_{k-1}^*| \leq \epsilon \), stop the algorithm and take the global minimizer \( x^* = x_k^* \); otherwise take \( \alpha = \alpha R \) go to Step 2.

**Table 1.** The list of test problems

| Problem No. | Function Name | Dimension | Region | Optimum value |
|-------------|---------------|-----------|--------|---------------|
| 1           | Two dimensional function \( c = 0.05 \) | 2         | \([-3,3]^2\] | 0              |
| 2           | Two dimensional function \( c = 0.2 \)  | 2         | \([-3,3]^2\] | 0              |
| 3           | Two dimensional function \( c = 0.5 \)  | 2         | \([-3,3]^2\] | 0              |
| 4           | 3-hump function                              | 2         | \([-3,3]^2\] | 0              |
| 5           | 6-hump function                              | 2         | \([-3,3]^2\] | –1.0316        |
| 6           | Treccani function                            | 2         | \([-3,3]^2\] | 3.0000         |
| 7           | Goldstein-Price function                     | 2         | \([-3,3]^2\] | 0              |
| 8           | Shubert function                             | 2         | \([-10,10]^2\] | -186.73091    |
| 9           | Rastrigin function                           | 2         | \([-3,3]^2\] | -2.0000        |
| 10          | Branin function                              | 2         | \([-5,10] \times [10,15]\] | 0.3979        |
| 11          | (S5) Shekel function                         | 4         | \([0,10]^4\] | -10.1532       |
| 12          | (S7) Shekel function                         | 4         | \([0,10]^4\] | -10.4029       |
| 13          | (S10) Shekel function                        | 4         | \([0,10]^4\] | -10.5364       |
| 14,15,16,17 | Sin-square I function                       | 2 \(, 3, 5, 7\) | \([-10,10]^n\] | 0              |
| 18,19,20,21 | Sin-square I function                       | 10, 20, 30, 50 | \([-10,10]^n\] | 0              |

The results on the mean value of the total iteration number (iter-m), the mean value of function evaluations (f.eval-m), the mean value of function values (f-mean), the best value of function values (f-best) and the success rates (SR) of ten trials for our algorithm on Problems 1-21 are illustrated in Table 2. It can be concluded that our method presents satisfactory results on all of the test problems with the success rate at least 60%.

We compare our method with the methods in [16] and [5] in terms of the mean value of the total iteration numbers (iter-m), the mean value of function evaluations (f.eval-m) in Tables 3. It can be seen from the Table 3, our method presents better results than the other methods in terms of the mean value of total iteration numbers and the mean value of function evaluations.

5. Algorithm and its application. In this section, we will apply local smoothing and smooth blending to obtain a smooth objective function as a model representing energy function of threonine (\(C_4H_9NO_3\)) molecule. Then, we will apply our global optimization method to obtain the optimum (minimum) energy value.
of the molecule. The data is obtained from DFT calculations of the Theronine molecule. In the calculations, the molecular structure was taken with two torsion (SC1 = C2C1C4O7 and SC2 = C1C2O6H13) angles and scanned around these
angles as 360° with 20° increments by using the DFT-6-31 G(d) method. The first angle values are increased by a step of 20° starting with 6.78° and ending with 346.78° and the second angle values are again increased by 20° at each steps starting with 19° and ending with 359°. For more details, we refer to [23].

We follow the Algorithm 2 for constructing the smooth model of the data points and for finding the global minimizer.

**Algorithm 2**

Step 0. – Define the partitions of $D$ and determine the subregions $D_{i,j}$, $i, j = 1, 2, \ldots, m$.

– Chose 9 control points for each subregion $D_{i,j}$ assign it as a column matrix $Q_{i,j}$, for $i,j = 1, 2, \ldots, m$, respectively.

Step 1. Construct the Bezier surface patches according to control points $Q_{i,j}$.

Step 2. Transform the parametric form of surface patches to cartesian form.

Step 3. Assign all surface patches to the functions $f_{i,j}$, $i,j = 1, 2, \ldots, m$.

Step 4. Define the smoothing functions of characteristic functions $\chi_{D_{i,j}}$.

Step 5. Construct the function $\hat{f}(x, y, \varepsilon) = \sum_{i,j=1}^{m} f_{i,j}(x, y, \varepsilon)\chi_{D_{i,j}}(x, y, \varepsilon)$

Step 6. Apply global minimum finding method to find global minimizer of the function $f(x, y, \varepsilon)$.

Now, we give the details of the application. First, we determine the region $D = [0, 360] \times [0, 360]$ which includes global minimizer. Then, we first choose 9 rectangular shaped subregions which are $D_{1,1} = [0, 80] \times [0, 80]$, $D_{1,2} = [0, 80] \times [140, 220]$, $D_{1,3} = [0, 80] \times [280, 360]$, $D_{2,1} = [150, 230] \times [0, 80]$, $D_{2,2} = [150, 230] \times [140, 220]$, $D_{2,3} = [150, 230] \times [280, 360]$, $D_{3,1} = [280, 360] \times [0, 80]$, $D_{3,2} = [280, 360] \times [140, 220]$ and $D_{3,3} = [0, 80] \times [280, 360]$ and determine 9 control points for each sub-region. The regions and control points are illustrated in Fig. 1. We construct the Bezier surfaces on these sub-regions, separately. These surfaces are shown in Fig. 2. The formulas of constructed Bezier surfaces are obtained in parametric form. Since, our blending approach is convenient for cartesian form, we convert these parametric equations to their explicit expressions in cartesian form as in Step 2 of Algorithm 2. The functions $f_{i,j}$ represent the explicit expression of the Bezier surface on the region $D_{i,j}$, for $i,j = 1, 2, 3$ and the functions are obtained as:

$$f_{1,1}(x, y) = \frac{13x^2 y^2}{10240000000} - \frac{(21x^2 y)}{2560000000} - \frac{(27xy^2)}{256000000} + \frac{(47xy)}{6400000} - \frac{(3x)}{20000} + \frac{(7y^2)}{320000} + \frac{y}{80000} - \frac{249}{1000},$$

![Figure 1. The subregions of $\Omega = [0, 360] \times [0, 360]$.](image-url)
\[ f_{1,2}(x, y) = \frac{(x^2 y^2)}{2560000000} - \frac{(x^2 y)}{6400000} + \frac{(103x^2)}{6400000} + (xy)/16000000 \]
\[ -\frac{(19x)}{80000} + \frac{(13y^2)}{3200000} - \frac{(59y)}{40000} - 197/1600, \]
\[ f_{1,3}(x, y) = \frac{(29x^2 y^2)}{1024000000} - \frac{(489x^2 y)}{256000000} + \frac{(103x^2)}{3200000} \]
\[ -\frac{(111x^2)}{5120000} + \frac{(919xy)}{6400000} - \frac{(241x)}{10000} - y^2/16000000 \]
\[ +y/10000 - 41/200, \]
\[ f_{2,1}(x, y) = \frac{(3x^2 y^2)}{1024000000} + \frac{(3x^2 y)}{256000000} + \frac{(9x^2)}{3200000} \]
\[ -\frac{(67xy^2)}{512000000} - \frac{(43xy)}{12800000} - \frac{(173x)}{1600000} \]
\[ +\frac{(363y^2)}{204800000} + \frac{(583y)}{2560000} - 5027/3200, \]
\[ f_{2,2}(x, y) = \frac{-(7x^2 y^2)}{512000000} + \frac{(33x^2 y)}{64000000} - \frac{(113x^2)}{2560000} \]
\[ +\frac{(133xy^2)}{256000000} - \frac{(627xy)}{3200000} + \frac{(10783x)}{6400000} \]
\[ -\frac{(2063y^2)}{5120000} + \frac{(973y)}{6400000} - 190213/128000, \]
\[ f_{2,3}(x, y) = \frac{(29x^2 y^2)}{1024000000} - \frac{(489x^2 y)}{256000000} + \frac{(103x^2)}{3200000} \]
\[ -\frac{(109xy^2)}{102400000} + \frac{(9173xy)}{12800000} - \frac{(9653x)}{800000} \]
\[ +\frac{(9761y^2)}{102400000} - \frac{(164909y)}{2560000} + 34087/3200, \]
\[ f_{3,1}(x, y) = \frac{(x^2 y^2)}{1024000000} - \frac{(3x^2 y)}{256000000} + \frac{(9x^2)}{3200000} \]
\[ -\frac{(21xy^2)}{256000000} + \frac{(11xy)}{1280000} - \frac{(14x)}{800000} \]
\[ +\frac{(63y^2)}{3200000} - \frac{(121y)}{800000} + 1/100, \]
\[ f_{3,2}(x, y) = \frac{-(x^2 y^2)}{2048000000} + \frac{(13xy^2)}{64000000} - \frac{(19x^2)}{102400000} \]
\[ +\frac{(83xy^2)}{256000000} - \frac{(27xy)}{200000} + \frac{(7909x)}{6400000} \]
\[ -\frac{(151y^2)}{320000} + \frac{(399y)}{20000} - 3341/1600, \]
\[ f_{3,3}(x, y) = \frac{(x^2 y^2)}{5120000000} - \frac{(23x^2 y)}{128000000} + \frac{(31x^2)}{800000} \]
\[ -\frac{(9xy^2)}{64000000} + \frac{(203xy)}{1600000} - \frac{(27x)}{1000} + (3y^2)/128000 \]
\[ -(1703y)/800000 + 4293/1000. \]

On the other hand, the above surfaces are not adjacent (not continuously joined each other), for this reason it is not possible to apply the smoothing approach, directly. By making some modifications on the surface smoothing approach it can be designed a smoothing approach for these nonadjacent surface patches. Let us define the partition for the variable \( x \) as \( a_0 = 0, a_1 = 80, a_2 = 150, a_3 = 230, a_4 = 280, a_5 = 360 \). The segment \([a_0, a_1]\) is used but \([a_1, a_2]\) is not used. But the function defined on \( D \) have to cover all domains. Therefore, the function \( \tilde{\chi}_{[a_0, a_1]} \) is defined.
where
\[
\tilde{x}_{[a_0, a_1]}(x, \varepsilon_1) = \begin{cases} 
1, & x \in [a_0, a_1], \\
P(x - b_1), & x \in [a_1, a_2], \\
0, & x \in [a_2, a_5],
\end{cases}
\]

the function \( \tilde{x}_{[a_2, a_3]} \) is defined by
\[
\tilde{x}_{[a_2, a_3]}(x; \varepsilon_1, \varepsilon_2) = \begin{cases} 
0, & x \in [a_0, a_1], \\
P_1(-(x - b_1)), & x \in [a_1, a_2], \\
1, & x \in [a_2, a_3], \\
P_1((x - b_2)), & x \in [a_3, a_4], \\
0, & x \in [a_4, a_5],
\end{cases}
\]

and the function \( \tilde{x}_{[a_4, a_5]} \) is
\[
\tilde{x}_{[a_4, a_5]}(x; \varepsilon_1, \varepsilon_2) = \begin{cases} 
0, & x \in [a_0, a_3], \\
P_1(-(x - b_2)), & x \in [a_3, a_4], \\
1, & x \in [a_4, a_5],
\end{cases}
\]

where,
\[
P_1(x - b_i) = \frac{1}{4\varepsilon_i^3}(x - b_i)^3 - \frac{3}{4\varepsilon_i}(x - b_i) + \frac{1}{2},
\]

and \( b_i = \frac{a_2 - a_{2i-1}}{2} \) and \( \varepsilon_i = \frac{a_{2i+2} - a_{2i-1}}{2} \), for \( i = 1, 2 \). The partition for the variable \( y \) is defined as \( c_0 = 0, c_1 = 80, c_2 = 140, c_3 = 220, c_4 = 280, c_5 = 360 \). Therefore, the smoothing functions of characteristic functions are defined by
\[
\tilde{x}_{[c_0, c_1]}(y, \delta_1) = \begin{cases} 
1, & y \in [c_0, c_1], \\
P_1(y - d_1), & y \in [c_1, c_2], \\
0, & y \in [c_2, c_3],
\end{cases}
\]
\[
\tilde{x}_{[c_2, c_3]}(y, \delta_1, \delta_2) = \begin{cases} 
0, & y \in [c_0, c_1], \\
P_1(-(y - d_1)), & y \in [c_1, c_2], \\
1, & y \in [c_2, c_3], \\
P_1((y - d_2)), & y \in [c_3, c_4], \\
0, & y \in [c_4, c_5],
\end{cases}
\]
\[
\tilde{x}_{[c_4, c_5]}(y, \delta_2) = \begin{cases} 
0, & y \in [c_0, c_3], \\
P_1(-(y - d_2)), & y \in [c_3, c_4], \\
1, & y \in [c_4, c_5],
\end{cases}
\]

where
\[
P_1(y - d_j) = \frac{1}{4\delta_j^3}(y - d_j)^3 - \frac{3}{4\delta_j}(y - d_j) + \frac{1}{2},
\]

and \( d_j = \frac{c_{2j-2} - c_{2j-1}}{2} \) and \( \delta_j = \frac{c_{2j+2} - c_{2j-1}}{2} \), for \( j = 1, 2 \). We apply our approach to these functions and we obtain single smooth model function \( \tilde{f}(x) \) defined as
\[
\tilde{f}(x; \varepsilon, \delta) = \sum_{i=1}^{3} \sum_{j=1}^{3} f_{i,j}(x, y) \tilde{x}_{[a_{2i-2}, a_{2i-1}]}(x; \varepsilon) \tilde{x}_{[c_{2j-2}, c_{2j-1}]}(y; \delta).
\]

The graph of the function \( \tilde{f}(x; \varepsilon, \delta) \) is shown in Figure 3. Then, we construct the following optimization problem:
\[
(P) \quad \min_{(x, y) \in [0, 360] \times [0, 360]} \tilde{f}(x; \varepsilon, \delta),
\]
Figure 3. The graph of the function \( \tilde{f}(x, y, \varepsilon, \delta) \) which is constructed by blending Bezier surfaces.

where

\[
\tilde{f}(x, y, \varepsilon, \delta) = \sum_{i=1}^{3} \sum_{j=1}^{3} f_{i,j}(x, y) \tilde{\chi}_{D_{i,j}}(x, y, \varepsilon, \delta).
\]

Since the obtained function is smooth, we can apply the Fast Delving Method in order to solve the problem \((\tilde{P})\). We found the global minimizer as \((x^*, y^*) = (181.6167, 187.5836)\) and the global minimum value as \(\tilde{f}(x^*, y^*, \varepsilon, \delta) = -438.2678\). The detailed results are shown in Table 1.

Table 4. Numerical Results

| \(k\) | \(\alpha\) | \(\beta\) | \(x_0\) | \(x_k^*\) | \(f_k^*\) |
|------|------|------|------|------|------|
| 1    | 0.5  | 0.1  | (160.0000, 280.0000) | (190.2613, 277.4205) | -438.2412 |
| 2    | 0.5  | 0.1  | (190.2613, 277.4205) | (329.0062, 186.9678) | -438.2625 |
| 3    | 0.5  | 0.1  | (329.0062, 186.9678) | (181.6167, 187.5836) | -438.2678 |

6. Conclusion. In this study, we propose a new methodology for smooth data modeling. Within this methodology, we propose new function smoothing and surface blending methods. This article is novel for using Bezier surfaces to obtain local models, blending surface patches to obtain the smooth model function and finding the global minimum of that function. The whole process offers a new methodology for many real-life problems in general, especially for energy conformation problems in this article.

We compare our global optimization method with the methods in [16] and [5]. As a real-life application, we apply our new methodology to construct the model function of the energy structure problem in [23] and minimize this model function.
The global minimum point of the problem is found approximately as \((186^\circ, 179^\circ)\) with the corresponding energy value \(-438.268\) Hartree/particle in [23]. In this study, the global minimum value is obtained exactly as \((181.6167^\circ, 187.5836^\circ)\) with the corresponding energy value \(-438.2678\) Hartree/particle.

The new function smoothing and global optimization approaches are generalized to \(n\)-dimensional case. In case of \(n\)-dimensional local modeling approach is used, the methodology proposed in this study can be used for modeling and minimizing \(n\)-dimensional problems.

**Acknowledgments.** This study has been supported by the Teaching Staff Training Project Units (OYP–05545–DR–13) and Scientific Research Projects Unit (4733–YL–16) of Suleyman Demirel University.

**REFERENCES**

[1] B. Belkhatir and A. Zidna, Construction of flexible blending parametric surfaces via curves, *Math. Comput. Simulat.*, **79** (2009), 3599–3608.
[2] X. Chen, Smoothing methods for nonsmooth, nonconvex minimization, *Math. Program.*, **134** (2012), 71–99.
[3] J. Cheng and X. S. Gao, Constructing blending surfaces for two arbitrary surfaces, *MM Research Preprints*, **22** (2003), 14–28.
[4] R. M. C. Dawson, D. C. Elliot and K. M. Jones, *Data for Biochemical Research*, Clarendon Press, Oxford, 1985.
[5] T. M. El-Gindy, M. S. Salim and A. I. Ahmet, A new filled function method applied to unconstrained global optimization, *Appl. Math. Comput.*, **273** (2016), 1246–1256.
[6] G. E. Farin, *Curves and Surfaces for CAGD: A Practical Guide*, Morgan Kaufmann, San Franciso, 2002.
[7] G. E. Farin, J. Hoschek and M. S. Kim, *Handbook of Computer Aided Geometric Design*, Elsevier, Amsterdam, 2002.
[8] R. P. Ge, A filled function method for finding a global minimizer of a function of several variables, *Math. Program.*, **46** (1990), 191–204.
[9] R. P. Ge, The theory of filled function method for finding global minimizers of nonlinearly constrained minimization problems, *J. Comput. Math.*, **5** (1987), 1–9.
[10] A. Griewank and A. Walther, First-and second-order optimality conditions for piecewise smooth objective functions, *Optim. Method Softw.*, **31** (2016), 904–930.
[11] T. Gu, S. Ji, S. Lin and T. Luo, Curve and surface reconstruction method for measurement data, *Measurement*, **78** (2016), 278–282.
[12] K. A. Guzzetti, A. B. Brizuela, E. Romano and S. A. Brandán, Structural and vibrational study on zwitterions of L-threonine in aqueous phase using the FT-Raman and SCRF calculations, *Mol. Struct.*, **1045** (2013), 171–179.
[13] E. Hartmann, Blending an implicit with a parametric surface, *Comput. Aided Geom. D.*, **12** (1995), 825–835.
[14] W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev.*, **140** (1965), A1133–A1138.
[15] A. V. Levy and A. Montalvo, The tunneling algorithm for the global minimization of functions, *SIAM Sci. Stat. Comput.*, **6** (1985), 15–29.
[16] S. Ma, Y. Yang and H. Liu, A parameter free filled function for unconstrained global optimization, *Appl. Math. Comput.*, **215** (2010), 3610–3619.
[17] A. Mazroui, D. Sbibih and A. Tijini, A simple method for smoothing functions and compressing Hermite data, *Adv. Comput. Math.*, **23** (2005), 279–297.
[18] A. Mazroui, H. Mraoui, D. Sbibih and A. Tijini, A simple method for smoothing functions and compressing Hermite data, *BIT Numerical Mathematics*, **47** (2007), 613–635.
[19] C. K. Ng, D. Li and L. S. Zhang, Global descent method for global optimization, *SIAM J. Optim.*, **20** (2010), 3161–3184.
[20] I. Nowak, J. Smolka and A. J. Nowak, Application of Bezier surfaces to the 3-D inverse geometry problem in continuous casting, *Inverse Probl. Sci. Eng.*, **19** (2011), 75–86.
[21] A. Ozmen, G. W. Weber, I. Batmaz and E. Kropat, RCMARS: Robustification of CMARS with different scenarios under polyhedral uncertainty set, Commun. Nonlinear Sci. Numer. Simulat., 16 (2011), 4780–4787.
[22] R. G. Parr and W. G. Yang, Density Functional Theory of Atoms and Molecules, Oxford University Press, New York, 1989.
[23] A. Sahiner, F. Ucun, G. Kapusuz and N. Yilmaz, Completed optimised structure of threonine molecule by fuzzy logic modelling, Z. Naturforsh. A, 71 (2016), 381–386.
[24] A. Sahiner, N. Yilmaz and G. Kapusuz, A descent global optimization method based on smoothing techniques via Bezier curves, Carpathian J. Math., 33 (2017), 373–380.
[25] Y. D. Sergeyev and D. E. Kvasov, A deterministic global optimization using smooth diagonal auxiliary functions, Commun. Nonlinear Sci. Numer. Simulat., 21 (2015), 99–111.
[26] P. Venkataraman, Solution of inverse ODE using Bezier functions, Inverse Probl. Sci. Eng., 19 (2011), 529–549.
[27] G. W. Weber, I. Batmaz, G. Koksal, P. Taylan and F. Yerlikaya-Ozkurt, CMARS: A new contribution to nonparametric regression with multivariate adaptive regression splines supported by continuous optimization, Inverse. Probl. Eng., 20 (2012), 371–400.
[28] Z. Y. Wu, D. Li and L. S. Zhang, Global descent methods for unconstrained global optimization, J. Glob. Optim., 50 (2011), 379–396.
[29] H. Wu, P. Zhang and G. H. Lin, Smoothing approximations for some piecewise smooth functions, J. Oper. Res. Soc. China, 3 (2015), 317–329.
[30] Y. T. Xu, Y. Zhang and S. G. Wang, A modified tunneling function method for non-smooth global optimization and its application in artificial neural network, Appl. Math. Model., 39 (2015), 6438–6450.
[31] X. Ye, Y. Liang and H. Nowacki, Geometric continuity between adjacent Bézier patches and their constructions, Comput. Aided Geom. D., 13 (1996), 521–548.
[32] N. Yilmaz and A. Sahiner, A new smoothing approximation to piecewise smooth functions and applications, International Conference on Analysis and Application, 1 (2016), p226.
[33] N. Yilmaz and A. Sahiner, New global optimization method for non-smooth unconstrained continuous optimization, AIP Conference Proceedings, 1863 (2017), 250002.
[34] J. Zilinskas, Branch and bound with simplicial partitions for global optimization, Math. Model. Anal., 13 (2008), 145–159.

Received April 2017; 1st revision September 2017; final revision January 2018.

E-mail address: ahmetnur32@gmail.com, ahmetsahiner@sdu.edu.tr
E-mail address: nurullahyilmaz@sdu.edu.tr
E-mail address: guldenkapusuz92@gmail.com