Journal of Computational Mathematics, Vol.18, No.4, 2000, 403–412.

ON TRIANGULAR C^1 SCHEMES: A NOVEL CONSTRUCTION*[1)]

Yin-wei Zhan
(*Key Lab of Digital Image Processing Techniques of Guangdong Province Science Center, Shantou University, Shantou, 515063, China*)

Abstract

In this paper we present a C^1 interpolation scheme on a triangle. The interpolant assumes given values and one order derivatives at the vertices of the triangle. It is made up of partial interpolants blended with corresponding weight functions. Any partial interpolant is a piecewise cubics defined on a split of the triangle, while the weight function is just the respective barycentric coordinate. Hence the interpolant can be regarded as a piecewise quartic. We device a simple algorithm for the evaluation of the interpolant. It's easy to represent the interpolant with B-net method. We also depict the Franke's function and its interpolant, the illustration of which shows good visual effect of the scheme.

Key words: Spline, interpolation scheme, partial interpolants, barycentric coordinates, splits, B-net

1. Introduction

The smooth interpolation on a triangulation of a planar region is of great importance in most applied areas, such as computation of finite element method, computer aided (geometric) design and scattered data processing.

Let Δ be a triangulation of a polygonal domain $\Omega \subset R^2$ and Δ_0, Δ_1 and Δ_2 the sets of vertices, edges and triangles in Δ respectively. Usually the triangulation in practice is formed by a mass of scattered nodes that, covered by the region Ω, are carrying similar types of data, i.e., positions and derivatives.

We have interest in this paper only the C^1 case and consider the following interpolation problem

$$\begin{cases} \text{For } f \in C^1(\Omega), \text{ find a function } g \in C^1 \text{ such that} \\ g(v) = f(v), \\ \frac{\partial}{\partial x} g(v) = \frac{\partial}{\partial x} f(v), \quad \frac{\partial}{\partial y} g(v) = \frac{\partial}{\partial y} f(v), \\ \text{for } v \in \Delta_0. \end{cases} \tag{1}$$

* Received May 19, 1997.
[1)] Supported by the Natural Science Foundation of Guangdong, China.

To solve problem (1), one efficient way is the *local* approach(cf. [3]). It is based on a single triangle. Once a solution of the problem restricted on the triangle is found, and it satisfies the property that the interpolants on any two adjacent triangles connect with C^1 smoothness, the triangle and its interpolant are representative and they together form an interpolation model or a C^1 interpolation scheme. Hence on Ω, a C^1 interpolant is constructed by piecing together the interpolants on all the triangles of triangulation Δ. The interpolation scheme such constructed has an another advantage: any change on a vertex affects only the interpolants on the star region of the vertex. Therefore it shows local significance.

To construct the interpolants, functions that are easy to formulate and evaluate are preferable, such as polynomials, rationals, and their piecewise versions, i.e. splines(see [4] and [6]).

Therefore we consider only the following problem for triangle $T = A_1A_2A_3$. Let e_i denote the opposite edge of A_i and n_i the outer normal vector of e_i, $i = 1, 2, 3$.

$$\begin{cases} \text{For } f \in C^1(T), \text{ find a function } g \in C^1(T) \text{ such that} \\ g(A_i) = f(A_i), \\ \frac{\partial}{\partial x} g(A_i) = \frac{\partial}{\partial x} f(A_i), \quad \frac{\partial}{\partial y} g(A_i) = \frac{\partial}{\partial y} f(A_i), \\ \text{and } g \text{ and } \frac{\partial}{\partial n_i} g \text{ on } e_i \text{ are univariate polynomials} \\ \text{of degrees 3 and 2 respectively, for } i = 1, 2, 3. \end{cases} \tag{2}$$

If the interpolant is a polynomial, derivatives at the vertices of the triangle of order 2 are needed and hence beyond the conditions given in (2)(cf. [7] and [8]). It also introduces a drawback that the degree of the polynomial reaches as high as 5(see [7] and [8]).

By splitting the triangle in HCT type(see [5]), one can find a solution of (2). The interpolant is a piecewise cubics. For a C^1 scheme, we have shown that the triangle must be subdivided and each angle of the triangle is split into 2 parts[8]. This is a crucial rule.

The splitting method lowers the order of interpolation data needed and the degree of interpolant polynomial. But it increases the computational complication. If we want the triangle is less split, the rationals instead of polynomials are suitable candidates. But rationals often introduce singular points which will cause unstable in evaluation.

So, some *compromise* between splits and rationals or polynomials will better the cases.

In this paper we find a solution of (2) by a hybrid of polynomials and splits. The interpolant of (2) is made up of partial interpolants blended with corresponding weight functions. Any partial interpolant is a piecewise cubics defined on a split of the triangle, while the weight function is just the respective barycentric coordinate. Hence the interpolant can be regarded as a piecewise quartics. We device a simple algorithm for the evaluation of the interpolant. It's easy to represent the interpolant with B-net method. We also depict the Franke's function and its interpolant of this scheme, the illustration of which shows good visual effect of the interpolation scheme.

The paper is organized as follows. The B-net method is effective for polynomial interpolation and we summarize it briefly in §2 as preliminaries. The essential ideas for the construction of our scheme are described in §3. In §4 we prove Theorem 2, the main result for the partial interpolants. In §5 we device an algorithm for evaluation of the interpolant. We end the paper by some remarks in §6.

2. Preliminaries

2.1 B-net method

Given is a triangle $T = A_1A_2A_3$. Any point $A \in R^2$ can be uniquely formulated as

$$A = \sum_{i=1}^{3} u_i A_i, \quad \sum_{i=1}^{3} u_i = 1,$$

where $u = (u_1, u_2, u_3)$ are the barycentric coordinates of A with respect to A_1, A_2 and A_3. For an integer $n > 0$ and any multiple index $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ with $|\lambda| = \sum_{i=1}^{3} \lambda_i = n$, define the λ-th Bernstein polynomial of order n as

$$B_{\lambda,n}(u) = \frac{n!}{\lambda_1!\lambda_2!\lambda_3!} u_1^{\lambda_1} u_2^{\lambda_2} u_3^{\lambda_3}.$$

Hence any bivariate polynomial p has its Bernstein-Bézier form (short for B-form)

$$p(u) = \sum_{|\lambda|=n} b_\lambda B_{\lambda,n}(u)$$

where b_λ is the B-ordinate of p with domain point $\sum_{i=1}^{3} \lambda_i A_i/n$.

Suppose $A' = \sum_{i=1}^{3} v_i A_i$ is a point opposite to A_1 across edge A_2A_3, and another polynomial q defined on $T' = A'_1A_2A_3$ has $\{c_\lambda\}$ as its B-ordinates relative to A'_1, A_2 and A_3. Then we have[1]

Theorem 1. *For any integer $\mu > 0$, the polynomials p and q are C^μ smoothly joint on edge A_2A_3 iff the following relations hold*

$$c_\lambda = \sum_{|\rho|=\lambda_1} b_{\eta+\rho} B_{\rho,\lambda_1}(v), \lambda_1 = 0, 1, \cdots, \mu, \eta = (0, \lambda_2, \lambda_3),$$

where $v = (v_1, v_2, v_3)$.

2.2 A representation of cubics by B-net method

For $i = 1, 2, 3$, denote by B_i the middle point of e_i and take the normal vector to e_i as

$$n_i = e_{i+2} - h_i e_i \quad \text{with} \quad h_i = \frac{e_i \cdot e_{i+2}}{e_i \cdot e_i} \tag{3}$$

where i is counted modulo 3 and $\cdot$ denotes the inner product in R^2. Suppose q is a bivariate cubic of the form

$$q(u) = \sum_{\lambda \neq \lambda_0} b_\lambda B_\lambda(u),$$

where $B_\lambda(u)$ is short for $B_{\lambda,3}(u)$ and $\lambda_0 = (1, 1, 1)$.

For given $f \in C^1(T)$, suppose

$$q(A_i) = f(A_i) =: f_i, \quad \frac{\partial}{\partial x} q(A_i) = \frac{\partial}{\partial x} f(A_i) =: f_{xi}, \quad \frac{\partial}{\partial y} q(A_i) = \frac{\partial}{\partial y} f(A_i) =: f_{yi}.$$

Then by Theorem 1, we have

$$\begin{aligned} &b_{300} = f_1, \quad b_{030} = f_2, \quad b_{003} = f_3, \\ &b_{210} = f_1 + D_{12}/3, \quad b_{201} = f_1 + D_{13}/3, \\ &b_{021} = f_2 + D_{23}/3, \quad b_{120} = f_2 + D_{21}/3, \\ &b_{102} = f_3 + D_{31}/3, \quad b_{012} = f_3 + D_{32}/3, \end{aligned} \tag{4}$$

where D_{ij} is the directional derivative of f at A_i along vector A_iA_j and can be expressed by f_{xi} and f_{yi}, i.e. $D_{ij} = f_{xi}(x_j - x_i) + f_{yi}(y_j - y_i)$.

Now suppose

$$q_i(u) = q(u) + b^i B_{111}(u), \quad i = 1, 2, 3.$$

Take $f_{ni} = \frac{\partial}{\partial n_i} f(B_i)$. Then by $\frac{\partial}{\partial n_i} q_i(B_i) = f_{ni}$, b^i is determined. For $i = 1$, we have

$$b^1 = -\frac{2}{3} f_{n1} + [(1 + h_1)b_{030} + (2 + h_1)b_{021} + (1 - h_1)b_{012} - h_1 b_{003} - b_{120} - b_{102}]/2. \tag{5}$$

Usually the normal derivatives are not easy to obtain in practice. Alternatively the normal derivative $\frac{\partial}{\partial n_1} q$ on e_1 is supposed to be linear. In the latter case

$$b^1 = \frac{1}{2}[b_{120} + b_{102} - h_1(2b_{012} - b_{021} - b_{003}) + (1 + h_1)(2b_{021} - b_{030} - b_{012})]. \tag{5'}$$

The formulas for b^2 and b^3 are similar.

In the consequent sections, for integers k and d with $k \geq d \geq 0$, we need a notation $S_k^d(\Delta; \Omega)$ or $S_k^d(\Delta)$ that expresses the bivariate spline space on triangulation Δ of piecewise C^d polynomials of degrees up to k (cf. [6]).

3. Basic Ideas

Now split T by connecting A_i and B_i and denote by T_i the resulted triangulation of T, $i = 1, 2, 3$. There are two subtriangles in T_i, and we would rather denote $T_{ij} = A_iA_jB_i$, for $\{i, j, k\} = \{1, 2, 3\}$. See Fig. 1 for the case $i = 1$.

We also denote by T_0 the triangulation of T by connecting all the lines A_iB_i, $i = 1, 2, 3$. T_0 can be regarded as a composition of the three triangulations T_i, $i = 1, 2, 3$.

We will find a solution of (2) in $S_4^1(T_0) \subset C^1(T)$. For this purpose, we consider at first the *partial* interpolation problem.

$$\begin{cases} \text{For } f \in C^1(T) \text{ and } i \in \{1,2,3\}, \text{ find a function } g_i \in S_3^1(T_i) \text{ such that} \\ g_i(A_j) = f_j, \quad \frac{\partial}{\partial x} g_i(A_j) = f_{xj}, \quad \frac{\partial}{\partial y} g_i(A_j) = f_{yj}, \\ \frac{\partial}{\partial n_k} g_i(B_k) = f_{nk}, \text{ for } j,k = 1,2,3, k \neq i. \end{cases} \tag{6}$$

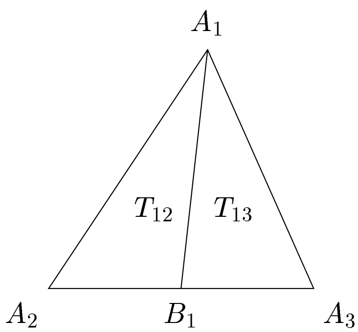


Fig. 1 Split T_1.

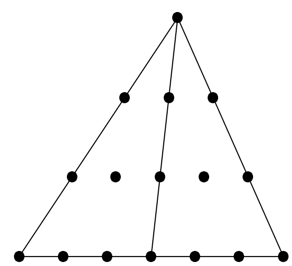


Fig. 2 Stencil of B-net of g_1.

It is easy to see that $\dim S_3^1(T_i) = 13$ (cf. [6]), while the number of data given in (6) is 11, 3 at each vertex and 1 on each edge other than e_i. To determine the other 2 parameters of g_i, we need a restriction in addition to (6) that

$$g_i \text{ on } e_i \text{ is an overall polynomial.} \tag{7}$$

Theorem 2. *Under restriction* (7), *there is a unique solution of* (6), $g_i \in S_3^1(T_i)$.

We will prove this theorem in the next section. Then with simple calculation, we can conclude that

Theorem 3. *Let g_i be the partial interpolant of* (6) *under restriction* (7), $i = 1, 2, 3$. *Define*

$$g(A) = \sum_{i=1}^{3} u_i g_i(A).$$

Then g is a solution of (2). *In detail, for a function $f \in C^1(T)$, g is uniquely determined by the following conditions,*

$$\begin{aligned} &g(A_i) = f_i, \\ &\frac{\partial}{\partial x} g(A_i) = f_{xi}, \quad \frac{\partial}{\partial y} g(A_i) = f_{yi}, \\ &\frac{\partial}{\partial n_i} g(B_i) = f_{ni}, \quad \text{for } \; i = 1,2,3. \end{aligned}$$

Note again that $g_i \in S_3^1(T_i)$. Then by the definition of g, it is easy to see that

Corollary 4. $g \in S_4^1(T_0)$.

4. Proof of Theorem 2

To prove Theorem 2, it needs only to investigate the case $i = 1$. The others are similar.

Suppose g_1 satisfies (6) and (7). The stencil depicted in Fig. 2 illustrates the B-net of g_1.

For a point A on the plane, its barycentric coordinates are supposed to be $u = (u_1, u_2, u_3)$ with T, $u^l = (u_1^l, u_2^l, u_3^l)$ with T_{1l}, $l = 2, 3$.

Let

$$g_1(u) = \begin{cases} g_{12}(u^2) = \sum_{|\lambda|=3} c_\lambda^2 B_\lambda(u^2), & \text{if } A \in T_{12}, \\ g_{13}(u^3) = \sum_{|\lambda|=3} c_\lambda^3 B_\lambda(u^3), & \text{if } A \in T_{13}. \end{cases}$$

Note that A_2, B_1 and A_3 are collinear, the B-net of g_1 is composed of layers, each of which is parallel to $e_1 = A_2A_3$. Therefore the continuity between g_{12} and g_{13} can be identified layer by layer. Each layer can be regarded as the B-net of a univariate spline of two pieces.

Recall the discussion in Section 2.2. The C^0 continuities between the layers of g_{12} and g_{13} simply imply that $c_{m,0,3-m}^2 = c_{m,0,3-m}^3$, $m = 0, 1, 2, 3$. By interpolation conditions on the vertices, c_{300}^l, c_{201}^l, c_{210}^l, c_{030}^l, c_{012}^l, and c_{120}^l are then determined, and hence c_{111}^l by normal derivatives, $l = 2, 3$. Then by C^1 continuity, $c_{102}^2 = c_{102}^3 = (c_{111}^2 + c_{111}^3)/2$.

At last, under restriction (7), i.e. g_{12} and g_{13} are identical on e_1, g_1 on e_1 is an overall cubic with only 4 unknowns, while there are *just* 4 Hermite interpolation conditions given at the end points of e_1 so that make the univariate cubic uniquely solvable. This ends the proof of Theorem 2.

5. An Algorithm for Evaluation

Now suppose g is the interpolant defined in Theorem 3. For a point $A \in T$, let (u_1, u_2, u_3) be its barycentric coordinates with T. The symbols in this section keep their meaning in §2.2 and §3.

The above discussion has introduced a procedure to evaluate g but it is too complicated. We will derive a simple algorithm instead. The key point is that, any piece of the partial interpolant can be transformed to a polynomial defined on the barycentric coordinates system $u = (u_1, u_2, u_3)$ with T.

There are two subtriangles in T_i and they can be characterized as

$$T_{ij} = \{A \in T : u_j \geq u_k\}, \quad \{i, j, k\} = \{1, 2, 3\}.$$

Then we have

Lemma 5. *For a multiple index (i,j,k) with $\{i,j,k\}=\{1,2,3\}$, let $T_{ijk}=\{A\in T: u_i\le u_j\le u_k\}$. Then we have*

$$T_{ijk}=T_{kj}\cap T_{ik}\cap T_{jk} \tag{8}$$

and the subtriangles in T_0 are just $\{T_{ijk}:\{i,j,k\}=\{1,2,3\}\}$.

Let $g_{ij}:=g_i|T_{ij}$. If $A\in T_{ijk}$, then by Lemma 5,

$$g(u)=u_kg_{kj}(u)+u_ig_{ik}(u)+u_jg_{jk}(u). \tag{9}$$

Note that g_{ij} differs from q only in two items that include $u_1u_2u_3$ and $u_iu_k^2$. Fig. 3 shows the stencil of $g_1(u)$ in the barycentric coordinates system (u_1,u_2,u_3). We can rewrite g_{ij} as

$$g_{ij}(u)=q(u)+6b_{ij}u_1u_2u_3+3c_{ij}u_iu_k^2, \tag{10}$$

where the B-ordinates b_{ij} and c_{ij} are easily formulated as

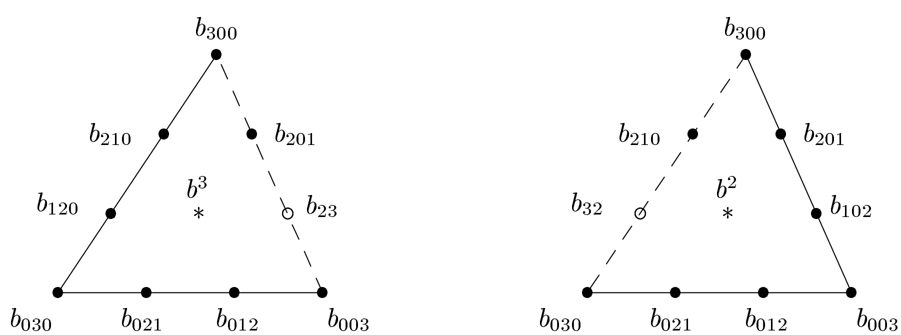
$$b_{ij}=b^k,\quad c_{ij}=b^j-b^k. \tag{11}$$

Therefore,

$$\begin{aligned} g(u) &= u_k[q(u)+6b_{kj}u_1u_2u_3+3c_{kj}u_ku_i^2]\\ &\quad +u_i[q(u)+6b_{ik}u_1u_2u_3+3c_{ik}u_iu_j^2]\\ &\quad +u_j[q(u)+6b_{jk}u_1u_2u_3+3c_{jk}u_ju_i^2]\\ &= q(u)+6b_{111}u_1u_2u_3+3(c_{kj}u_k)u_ku_i^2+3(c_{ik}u_i)u_iu_j^2+3(c_{jk}u_j)u_ju_i^2 \end{aligned} \tag{12}$$

where

$$\begin{aligned} b_{111} &= u_kb_{kj}+u_ib_{ik}+u_jb_{jk}=u_kb^i+u_ib^j+u_jb^i\\ &= u_ib^j+(1-u_i)b^i=b^i+u_i(b^j-b^i). \end{aligned} \tag{13}$$



Left: $g_{12}(u)$. $\quad b_{23}=b_{102}+(b^2-b^3)$ Right: $g_{13}(u)$. $\quad b_{32}=b_{120}+(b^3-b^2)$

Fig. 3 The stencil of g_{12} and g_{13} in the same coordinates system (u_1,u_2,u_3).

Then we can establish a simple algorithm for the evaluation of g at a point u.

Step 1 Order the coordinates

$$\begin{aligned} i_{\max} &= \text{the index of } \max\{u_1, u_2, u_3\}, \\ i_{\min} &= \text{the index of } \min\{u_1, u_2, u_3\}, \\ i_{\text{med}} &= \text{the other index}. \end{aligned}$$

Then take

$$b_{111} = b^{i_{\min}} + u_{i_{\min}}(b^{i_{\text{med}}} - b^{i_{\min}})$$

Step 2

$$\begin{array}{lllll} \texttt{If} & u_2 \le u_3 & \texttt{then} & b_{120} += u_1(b^3 - b^2) \\ & \texttt{else} & b_{102} += u_1(b^2 - b^3) \\ \texttt{If} & u_3 \le u_1 & \texttt{then} & b_{012} += u_2(b^1 - b^3) \\ & \texttt{else} & b_{210} += u_2(b^3 - b^1) \\ \texttt{If} & u_1 \le u_2 & \texttt{then} & b_{201} += u_3(b^2 - b^1) \\ & \texttt{else} & b_{021} += u_3(b^1 - b^3) \end{array}$$

Step 3 Use de Casteliau Algorithm to evaluate $g(u)$ with B-ordinates $\{b_\lambda : |\lambda| = 3\}$.

6. Remarks

In [2], T.N.T. Goodman and H.B. Said derived a C^1 scheme(short for GS scheme), the interpolant of which can be rewritten as

$$P(u) = q(u) + 6u_1u_2u_3b(u), \quad b(u) = \frac{u_1^2u_2^2b^3 + u_2^2u_3^2b^1 + u_3^2u_1^2b^2}{u_1^2u_2^2 + u_2^2u_3^2 + u_3^2u_1^2}.$$

We can compare GS with our scheme in many ways.

(1) In GS, before using de Casteliau Algorithm to evaluate $P(u)$, it needs at least 6 multiplications, while our scheme needs only 4 multiplications;

(2) The interpolant $P(u)$ of GS is a rational, while $g(u)$, the interpolant of our scheme, is a piecewise quartics. The vertices of the triangle are singular points of $P(u)$. The singular points are removable, but it may cause unstable in evaluation of P and the derivatives of P at these points. It is also difficult to do integration with rationals. Our scheme can overcome these drawbacks.

(3) Now take the Franke's function $f(x, y)$ as a test.

$$\begin{aligned} f(x,y) \;\; = \;\; & 0.75e^{-((9x-2)^2+(9y-2)^2)/4} + 0.75e^{-((9x+1)^2/49+(9y+1)/10)} \\ & +0.50e^{-((9x-7)^2+(9y-3)^2)/4} - 0.20e^{-((9x-4)^2+(9y-7)^2)} \end{aligned}$$

Table 1. Errors of Franke's function with the interpolants

interpolants	Max error	Mean error
GS scheme: P	0.045273	0.004627
our scheme: g	0.042543	0.004593

Fig. 4 shows the triangulation of unit square $[0,1]\times[0,1]$ used in many papers(cf. [2] and the references therein). The surfaces of Franke's function f is illustrated in Fig. 5, and the difference (scaled by factor 100) between the interpolant g and f on the triangulation is illustrated in Fig. 6. The comparison of errors is shown in Table 1. It is seen that both the max error and the mean error by using our scheme are less than those by using GS.

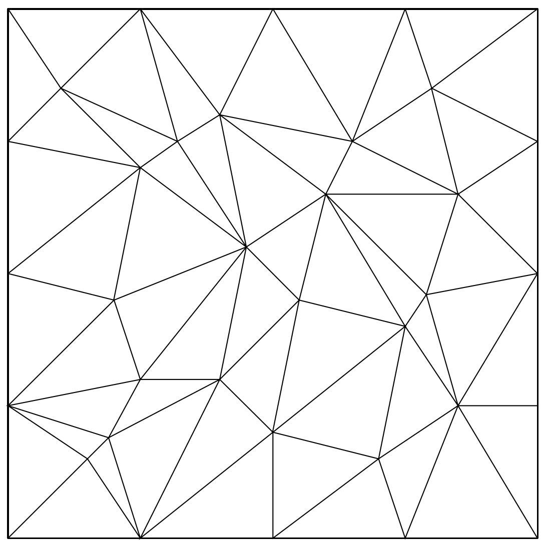


Fig. 4 A triaugulation of $[0,1]\times[0,1]$

Acknowledgement. I would like to thank Mr. Zongbin Mu for helping me with MATLAB to derive the experiment results Table 1 and Figures 5 and 6.

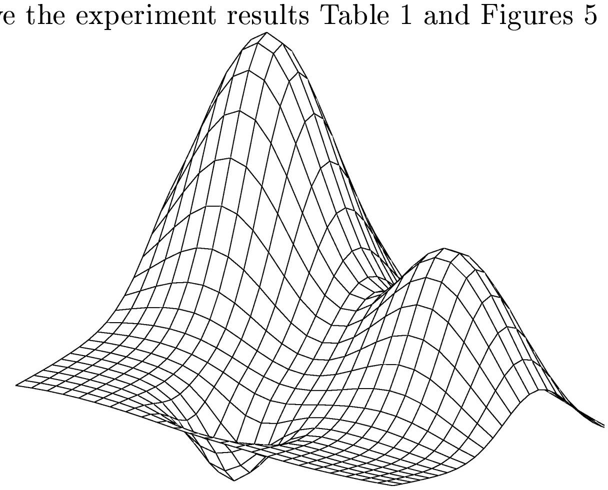


Fig. 5. Franke function with 25×25 samples

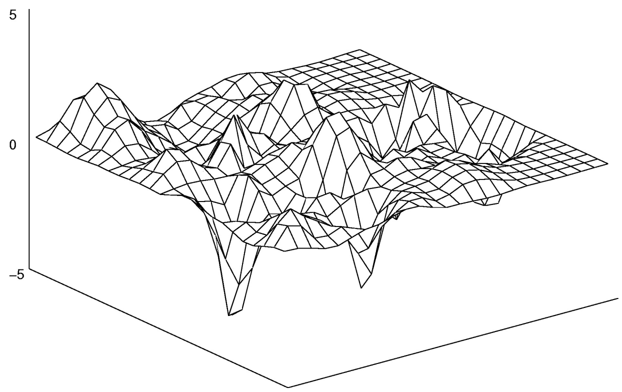


Fig. 6. Error between Franke function and its interpolant (scaled by factor 100)

References

[1] Farin, G., Triangular Berstein-Bézier patches, Computer Aided Geometric Design **3**(1986), 83–127.

[2] Goodman, T.N.T. and H. B. Said, A C^1 triangular interpolant suitable for scattered data interpolation, Communication in Applied Numerical Methods, **7**(1991), 479–485.

[3] Peters, J., Local smooth surface interpolation: a classification, Computer Aided Geometric Design **7**(1990), 191–195.

[4] Schumaker, L., *Spline Functions: Basic Theory*, Wiley, 1981, New York.

[5] Strang, G. and G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, 1973, New York.

[6] Wang, R., The dimension and basis of spaces of multivariate splines, J. Comp. Appl. Math. **12:13**(1975), 163–177.

[7] Ženišek, A., Interpolation polynomials on the triangle, Numer, Math. **15**(1970), 383-296.

[8] Zhan, Y., A geometric feature for finite element schemes, Approx. Th. Appl. **7**(1994), no. 2, 83–91.