A COMPARISON OF NORMALS CALCULATION FOR THE CONSTRUCTION OF INTERPOLANTS ABOVE TRIANGULAR MESHES

Conference Paper · February 2017

Robert Bohdal
Comenius University in Bratislava

Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
16th Conference on Applied Mathematics **APLIMAT 2017**
Proceedings

A COMPARISON OF NORMALS CALCULATION FOR THE CONSTRUCTION OF INTERPOLANTS ABOVE TRIANGULAR MESHES

BOHDAL Róbert (SK)

Abstract. In this paper, we compare selected methods for calculation of normal vectors, which are necessary for the construction of interpolants above the given triangulation. The normals at individual vertices of the underlying triangulation greatly affect the shape and the smoothness of the resulting interpolation surface. We compare results of selected methods for normals calculation with analytically calculated normals of test functions. We also calculate the difference between the created interpolants and the corresponding test functions. The best results were achieved by the method of calculating normals using the local thin plate interpolation spline. The weighted average method, created by combining Little's and Max's method came as the second in order.

Keywords: normals calculation, thin plate spline, Clough-Tocher, Powell-Sabin

Mathematics subject classification: Primary 65D05; Secondary 65D17

1 Introduction

In many applications, we are often confronted with the problem of scattered points interpolation. In case of a small number of input points, it is usually the best to choose so-called global interpolation methods, which interpolate the input points using only one function. The global interpolation methods yield good overall smoothness and accuracy of the interpolation. Interpolation methods using radial basis functions [4] such as Hardy multiquadrics or thin plate splines, etc. are often used. For a bigger number of input points (of the order ten thousand and more), it is not possible to use global interpolation methods because they need to solve large systems of equations. In such case, local interpolation methods are used with an advantage, which use piecewise functions, each interpolates only a few points. The classical examples are the Clough-Tocher method [1, 8], the Powell-Sabin [1, 10], the natural neighbour interpolation [7], and others. Many of them require normal vectors at the given points. Since these vectors are not known, they must be estimated.

The quality of the created interpolation surfaces (in the sense of visual smoothness, continuity, accuracy, etc.) greatly depends on the accuracy of the calculated normals. The gradient of the surface in a given point has often a greater influence on the shape of the surface than the chosen degree of polynomials or the degree of smoothness.

In this paper we compare some known methods for normals calculation. We also suggest a small modification which includes removing "long and thin" triangles on the boundary of the given triangulation and removing "unsuitable" normals from the calculations. Next, we suggest a combination of the Max's and Little's method into one formula. We then compare the calculated normals with the "true" analytical normals, and we also quantify the accuracy with which the interpolation surfaces using these normals match the test functions.

Jin et al. [5] also compare some methods for normals calculation on the selected test models with known analytical normals, but their comparison technique is based on the cumulative histograms of angular discrepancies. Moreover, they exclude vertices of non-closed test surfaces from the comparison.

2 The methods for normals calculation

There are many methods how to estimate the normals. The most commonly used methods are weighted average methods and methods using local interpolants and approximants. Other global methods usually based on minimization of a given integral on the triangular network are used rarely. Another approach for normal calculation based on the linear regression and on the finite difference method we can find in the paper [6].

Almost all methods for normals calculation use an in-advance constructed triangulation of the input points. The normals are then calculated at the triangulation vertices using some weighted average of the normals of adjacent triangles.

We search such method that is robust enough with respect to input points. It often happens that the considered triangulation includes "long and thin" triangles on the boundary that negatively affect the accuracy of calculating normals using the weighted average methods. Thin triangles have at least one angle much smaller than the other ones. In addition, long triangles have at least two sides much longer than all other triangles in the triangulation. The constructed interpolation surface which uses such normals then can create unwanted shapes on the surfaces boundary, see figure 3(a).

2.1 Normals calculated using weighted average

Weighted average methods use the existing triangulation of the input points. It is good to use Delaunay triangulation that minimizes the number of "long and thin" triangles, because it maximizes the minimal angles of all the triangles of the triangulation.

Let we have the vertices $B_i[x_i, y_i, z_i]$ of the triangulation $\mathcal{T}$ created from the set of given points $(x_i, y_i, z_i) \in \mathbb{E}^3$. According to [3], we can calculate the normals at the vertices by the formula

$$\hat{\boldsymbol{n}}_i = \sum_{\mathcal{N}_i} \omega_{ijk} \boldsymbol{n}_{\triangle ijk}, \tag{1}$$

where the weight ω_{ijk} is calculated from the relation

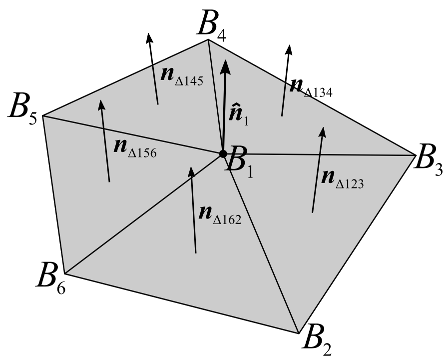


Fig. 1. Example of adjacent triangles for calculation $\hat{\boldsymbol{n}}_1$ with indices belongs to the set $\mathcal{N}_1$

$$\omega_{ijk} = \frac{\sigma_{ijk}}{\sum_{\mathcal{N}_i} \sigma_{ijk}} \tag{2}$$

and where $\boldsymbol{n}_{\triangle ijk}$ denotes the normal of the triangle $B_i B_j B_k$. The sum in equation (1) is enumerated for all triple indices (i, j, k) of the triangulation vertices from the set

$$\mathcal{N}_i = \big\{(i, j, k) \in \mathbb{N}^3; j \neq k, \\ \text{where } B_j, B_k \text{ meet "selection criterion"}\big\}\,.$$

The selection criterion includes all the vertices which either create an edge with the vertex B_i or lie in some neighbourhood of the vertex B_i (see figure 1).

According to [3], the values σ_{ijk} in formula (2) are given by one of the following possibilities (see figure 2):

- The arithmetic average (Gouraud)

  $$\sigma_{ijk} = 1.$$

  Every triangle contributes with the same weight to the calculated normal.

- The inverse value of lengths (Little)

  $$\sigma_{ijk} = \frac{1}{|B_i B_j|^r |B_i B_k|^r},$$

  where $r = 1$, 2 or $1/2$ as a rule. The weight depends inversely on the distance of triangulation vertices adjacent to the vertex B_i. The longer is the triangle, the less it contributes to the resulting normal.

- The angle at the vertex (Thurmer)

  $$\sigma_{ijk} = \alpha_i.$$

  The symbol α_i denotes the angle $\measuredangle B_i B_j B_k$ at the vertex B_i. The triangle that makes the biggest angle between the edges $B_j B_i$ and $B_k B_i$ contributes the most to the resulting normal.

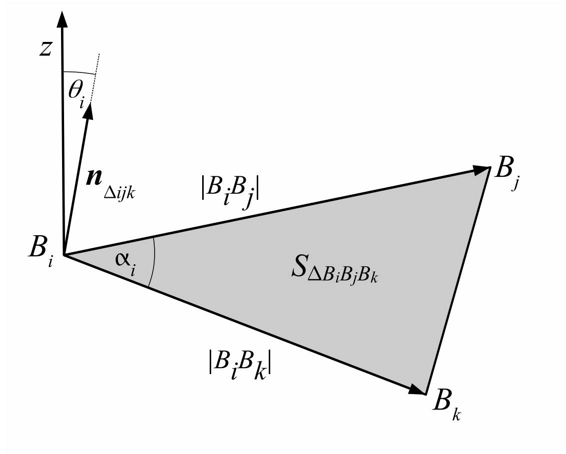


Fig. 2. Elements of a triangle for calculation of normals

- The area of the triangle (Akima)

$$\sigma_{ijk} = S_{\triangle B_i B_j B_k}.$$

  The triangle with the greatest surface contributes the most to the resulting normal. The symbol $S_{\triangle B_i B_j B_k}$ denotes the area of the triangle $B_i B_j B_k$.

- The gradient of the surface (Akima2)

$$\sigma_{ijk} = \cos\theta_i S_{\triangle B_i B_j B_k}.$$

  The symbol θ_i denotes the angle between the z-axis and $\boldsymbol{n}_{\triangle ijk}$. A triangle whose normal makes a smaller angle with the z-axis, contributes more to the resulting normal than one with the same area. This ensures the equality of the triangles which have the same area in projection to the xy plane.

Max in paper [9] proposes to calculate σ_{ijk} using the formula

$$\sigma_{ijk} = \frac{\sin\alpha_i}{|B_i B_j|^r |B_i B_k|^r}.$$

This method prefers triangles whose angle at the vertex B_i approaches the right angle and whose edges are short.

We can also combine Little's and Max's method into one to achieve a better "average accuracy":

$$\sigma_{ijk} = \frac{1 + \sin\alpha_i}{2|B_i B_j|^r |B_i B_k|^r}.$$

Although weighted average methods have low computational complexity, they give low-grade results. They are not suitable for triangulations with smaller number of vertices, in which the gradient of the triangles often changes its direction.

2.2 Normals calculated using local interpolation or approximation

The next possibility how to estimate a normal at a point B_i is to use a local function $f_i(x, y)$, which interpolates or approximates the set of "close neighbours" of the point B_i, and calculate the normal using partial derivatives:

$$\hat{\boldsymbol{n}}_i = \left(\frac{\partial f_i(x_i, y_i)}{\partial x}, \frac{\partial f_i(x_i, y_i)}{\partial y}, -1 \right).$$

Stead compared several kinds of functions in [13], among them:

- The Shepard's interpolant

  $$f(x, y) = \sum_{i=1}^{n} \omega_i(x, y) z_i,$$

  where the weight ω_i is given by $\omega_i(x, y) = \frac{d_i^2(x,y)}{\sum_{j=1}^{n} d_j^2(x,y)}$ and $d_i^2(x, y) = (x - x_i)^2 + (y - y_i)^2$.

- The Hardy's multiquadrics interpolant

  $$f(x, y) = \sum_{i=1}^{n} c_i \sqrt{d_i^2(x, y) + R^2},$$

  where the shaping parameter R is given, and the unknowns c_i are calculated from a set of equations following the interpolation conditions $f(x_i, y_i) = z_i$.

- Linear polynomial function given by the least squares method.

- Quadratic polynomial function given by the least squares method.

According to the performed tests, the best results are achieved by Hardy's multiquadrics interpolant.

Instead of Hardy's multiquadrics $\phi(d_i^2(x, y)) = \sqrt{d_i^2(x, y) + R^2}$ we can use thin plate splines $\phi(d_i^2(x, y)) = d_i^2(x, y) \ln(d_i^2(x, y))$ because they do not require estimation of the parameter R.

Renka and Cline [11] described method for finding the normal at the vertex B_i using partial derivatives of the quadratic polynomial function $f(x, y) = z_i + a(x - x_i)^2 + b(x - x_i)(y - y_i) + c(y - y_i)^2 + d(x - x_i) + e(y - y_i)$, which interpolates the vertex B_i and approximates the set of "close neighbours" in the sense of the least squares method. For each vertex B_j from the set of neighbour vertices, a weight ω_j is calculated so that the furthest vertices do not contribute to the resulting normal

$$\omega_j = \frac{(r_i - |B_i B_j|)_+}{r_i |B_i B_j|},$$

where r_i is the radius of neighbourhood influence of the vertex B_i and $(\cdot)_+ = \max\{0, \cdot\}$.

2.3 Normals calculated using global methods

We can get the best results for normals estimation if we use global methods which are based on finding the minimum of appropriate integral functions. An example is the method of Nielson's minimum norm network described in [12].

Global methods for estimating normals give better results than local methods, but we have to solve large systems of equations for their calculation. Moreover, this estimation is only a little more accurate than results obtained by methods based on local interpolants [3].

3 Testing the methods

For testing purposes, we have used data which were created by 9 test functions, so we have been able to calculate all normals using partial derivatives. We have added two own functions (see figure 4) to the seven test functions presented in [2].

The samples contained from 100, 900, 2500, and 4900 randomly selected points lying in the interval $[0, 1] \times [0, 1]$, and they were different for each test function. As a result, we have gathered 36 different input triangulations. We have compared 7 methods of calculating normals using various weighted averages and one method using the local thin plate spline interpolation function.

Some of the weighted average methods (e.g. the Akima method) make unwanted shapes at the border of the constructed interpolant (see figure 3(a)) if the underlying triangulations include very "long and thin" triangles, which often occur at the boundaries of the triangulations. After deleting them[1], the results significantly improved for the less robust methods (see figure 3(b)). Moreover, the problem of calculating the values of the interpolation function outside of the triangulation occurred. The next attempt for normals calculation on the boundary of the triangulation was not to use normals which differ a lot from the "average normal" (see algorithm 1). The combination of both modifications gave noticeable improvement (see figure 3(c)).

Little's and Max's methods have the best results in the category of methods using the weighted average. Little's method was better for some data, and Max's method was better for the other ones (see the plots in figure 5). Consequently, we have decided to combine both methods into one relation (Little-Max). Using this new method we have achieved a better "average accuracy" (see plots in figure 5).

From the methods using local interpolants, we have selected thin plate splines instead of Hardy's multiquadrics because they do not require estimation of the parameter R.

In our test we have calculated:

1. The deviations of the estimated normals $\hat{\boldsymbol{n}}_i$ from the normals of the test function $\nabla f_t(x_i, y_i)$ calculated using partial derivatives at the given point (x_i, y_i)

$$\mathrm{RMSE}_\alpha = \sqrt{\frac{\sum_{i=1}^{n} \left(\cos^{-1}\left(\frac{\hat{\boldsymbol{n}}_i \cdot \nabla f_t(x_i, y_i)}{|\hat{\boldsymbol{n}}_i||\nabla f_t(x_i, y_i)|}\right)\right)^2}{n}}.$$

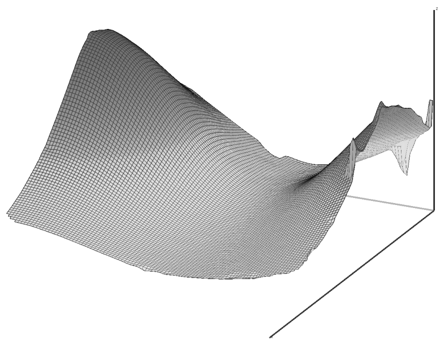
   Results are visualized in the plots in figure 5.

[1]We have deleted triangles whose inner angle exceed 178 degrees.

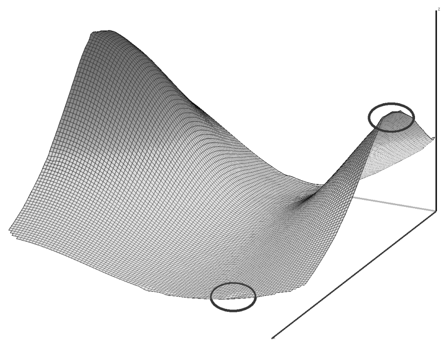
Algorithm 1 "Unsuitable" normals removing
input: $num\_normals$, $trng\_normals[]$, $const = 1.0$
output: $trng\_normals$

---

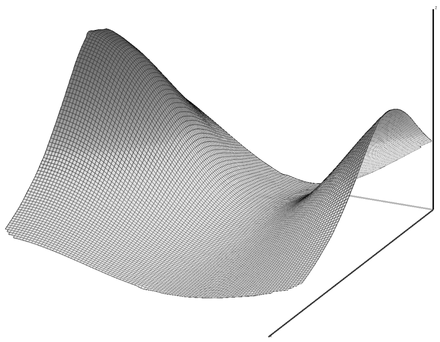
if $num\_normals < 5$ **then**
    return
$sum\_norm = 0$
for all $normal$ in $trng\_normals$ **do**
    $sum\_norm \mathrel{+}= normal$
$avg\_normal = sum\_norm/num\_normals$
$sum\_sqd = 0.0$
for $i = 0$ to $num\_normals$ **do**
    $square\_dif[i] = \|trng\_normals[i] - avg\_normal\|^2$
    $sum\_sqd \mathrel{+}= square\_dif[i]$
$average\_sqd = sum\_sqd/num\_normals$
for $i = 0$ to $num\_normals$ **do**
    $delta\_sqd = |square\_dif[i] - average\_sqd|$
    if $delta\_sqd > const * average\_sqd$ **then**
        remove $trng\_normals[i]$ from list



(a) Constructed interpolant with original normals calculation method



(b) After thin triangles deleted



(c) After "unsuitable" normals deleted

Fig. 3. Effect of a modification for calculating the normals on the shape of constructed interpolant. The normals have been calculated by Akima method

2. The accuracy of "matching" the resulting interpolation surfaces calculated by the Clough-Tocher and the Powell-Sabin method with the corresponding test function. To test the accuracy of the interpolation surface, we have calculated the deviation from the square of the difference between the function value of interpolation function $\hat{f}(x, y)$ and the function value of the test function $f_t(x, y)$ for $m = 2500$ randomly selected points from interval $[0, 1] \times [0, 1]$:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{m}(\hat{f}(x_i, y_i) - f_t(x_i, y_i))^2}{m}}.$$

   The plots in figure 6 present the result. Since the order of accuracy of the tested methods was almost identical for both Clough-Tocher and Powell-Sabin interpolation method, we present only the results for the first of them.

The approximate time in milliseconds of the individual normals calculation methods is presented in the table 3. All methods have been tested on the desktop PC with Intel(R) Core(TM) i5-4670K CPU @3.40GHz processor with 8GB RAM.

4 Conclusion

As expected, the method using the local interpolant (see plots in figures 5 and 6) has come out as the best one from all tested methods. The combination of Little's and Max's method was the second best method in general. Both approaches of calculations were sufficiently robust with respect to the used input data.

For applications not needing great speed of normals calculating, we recommend the method which uses local thin plate spline interpolation. In other cases, we suggest to use the above described combination of Little's and Max's method or one of them.

From table 3 it is clear that the method using local thin plate interpolation splines (TPS) is approximately 5 times more computationally expensive than all the weighted average methods.

Our suggested combination of Little's and Max's methods is sufficiently robust considering the triangulation of the input points and it could be used for the the construction of interpolants with the Clough-Tocher or Powell-Sabin method even without "unsuitable" normals removing.

Methods/number of points	**4900 points**	**8100 points**
All methods except TPS	$\approx$ 90ms	$\approx$ 250ms
TPS method	$\approx$ 480ms	$\approx$ 1310ms

Tab. 1. A comparison of the times of computations

References

[1] AMIDROR, I. *Scattered data interpolation methods for electronic imaging systems: a survey.* Journal of electronic imaging 11, 2 (2002), 157–176.

[2] FRANKE, R. *Scattered data interpolation: Tests of some methods.* Mathematics of computation 38, 157 (1982), 181–200.

[3] HOSCHEK, J., LASSER, D. *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Ltd., Natick, MA, USA, 1993.

[4] ISKE, A. *Radial basis functions: basics, advanced topics and meshfree methods for transport problems.* Rendiconti del Seminario Matematico, Polytechnic University of Turin 61, 3 (2003), 247–285.

[5] JIN, S., LEWIS, R. R., West, D. *A comparison of algorithms for vertex normal computation.* The Visual Computer 21, 1-2 (2005), 71–82.

[6] JIRKA, T., SKALA, V. *Gradient vector estimation and vertex normal computation.* Tech. rep., University of West Bohemia, Pilsen, 2002.

[7] LEDOUX, H., GOLD, C. *An efficient natural neighbour interpolation algorithm for geoscientific modelling.* Developments in Spatial Data Handling. Springer, 2005, pp. 97–108.

[8] MANN, S. *Cubic precision clough-tocher interpolation.* Computer aided geometric design 16, 2 (1999), 85–88.

[9] MAX, N. *Weights for computing vertex normals from facet normals.* Journal of Graphics Tools 4, 2 (1999), 1–6.

[10] POWELL, M., SABIN, M. *Piecewise quadratic approximations on triangles.* ACM Transactions on Mathematical Software (TOMS) 3, 4 (1977), 316–325.

[11] RENKA, R., CLINE, A. *A triangle-based c1 interpolation method.* Rocky Mountain Journal (1984).

[12] SCHEUERMANN, G., TRICOCHE, X., HAGEN, H. *C1-interpolation for vector field topology visualization.* Visualization'99. Proceedings (1999), IEEE, pp. 271–533.

[13] STEAD, S. *Estimation of gradients from scattered data.* Rocky Mountain Journal of Mathematics 14, 1 (1984).

Current address

Róbert Bohdal, RNDR., PhD.
Department of Algebra, Geometry and Didactics of Mathematics
Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava
Mlynska Dolina, 842 48 Bratislava, Slovakia
Tel. number: +421-2-602 95 ext. 185, e-mail: robert.bohdal@fmph.uniba.sk

List of the test functions

$$f_1(x,y) = \frac{3e^{-\frac{(9y+1)^2}{10}-\frac{(9x+1)^2}{49}}}{4} + \frac{3e^{-\frac{(9y-2)^2}{4}-\frac{(9x-2)^2}{4}}}{4} + \frac{e^{-\frac{(9y-3)^2}{4}-\frac{(9x-7)^2}{4}}}{2} - \frac{e^{-(9y-7)^2-(9x-4)^2}}{5}$$

$$f_2(x,y) = \frac{1-\tanh(9y-9x)}{9}$$

$$f_3(x,y) = \frac{\cos\left(\frac{27}{5}y\right)+\frac{5}{4}}{6(3x-1)^2+6}$$

$$f_4(x,y) = \frac{e^{-\frac{81\left(\left(y-\frac{1}{2}\right)^2+\left(x-\frac{1}{2}\right)^2\right)}{16}}}{3}$$

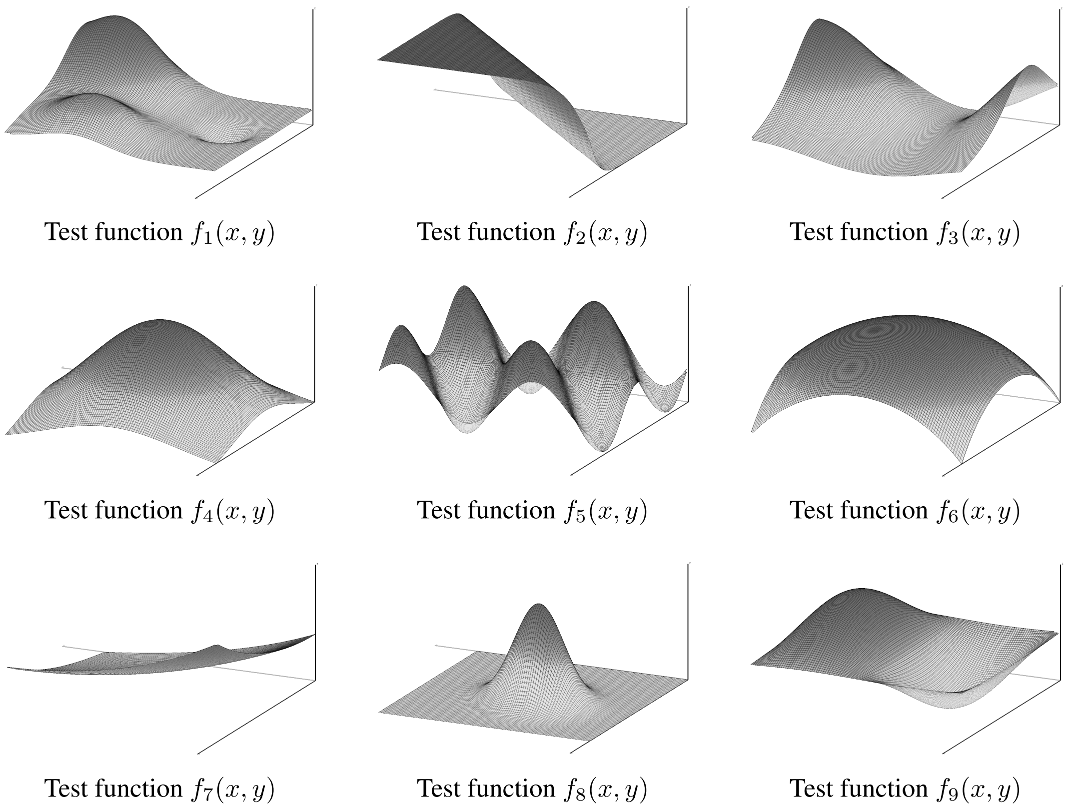
$$f_5(x,y) = \cos(10y) + \sin(10(x-y))$$

$$f_6(x,y) = \frac{\sqrt{64-81\left(\left(y-\frac{1}{2}\right)^2+\left(x-\frac{1}{2}\right)^2\right)}}{9} - \frac{1}{2}$$

$$f_7(x,y) = \frac{1}{\sqrt{2e^{-3\left(\sqrt{y^2+x^2}-\frac{67}{10}\right)}+1}}$$

$$f_8(x,y) = 50e^{-200\left(\left(y-\frac{3}{10}\right)^2+\left(x-\frac{3}{10}\right)^2\right)} + e^{-50\left(\left(y-\frac{1}{2}\right)^2+\left(x-\frac{1}{2}\right)^2\right)}$$

$$f_9(x,y) = \sin(\pi x)\sin(2\pi y)$$



Test function $f_1(x,y)$ Test function $f_2(x,y)$ Test function $f_3(x,y)$

Test function $f_4(x,y)$ Test function $f_5(x,y)$ Test function $f_6(x,y)$

Test function $f_7(x,y)$ Test function $f_8(x,y)$ Test function $f_9(x,y)$

Fig. 4. Pictures of the test functions

Deviations between the calculated and the analytical normals

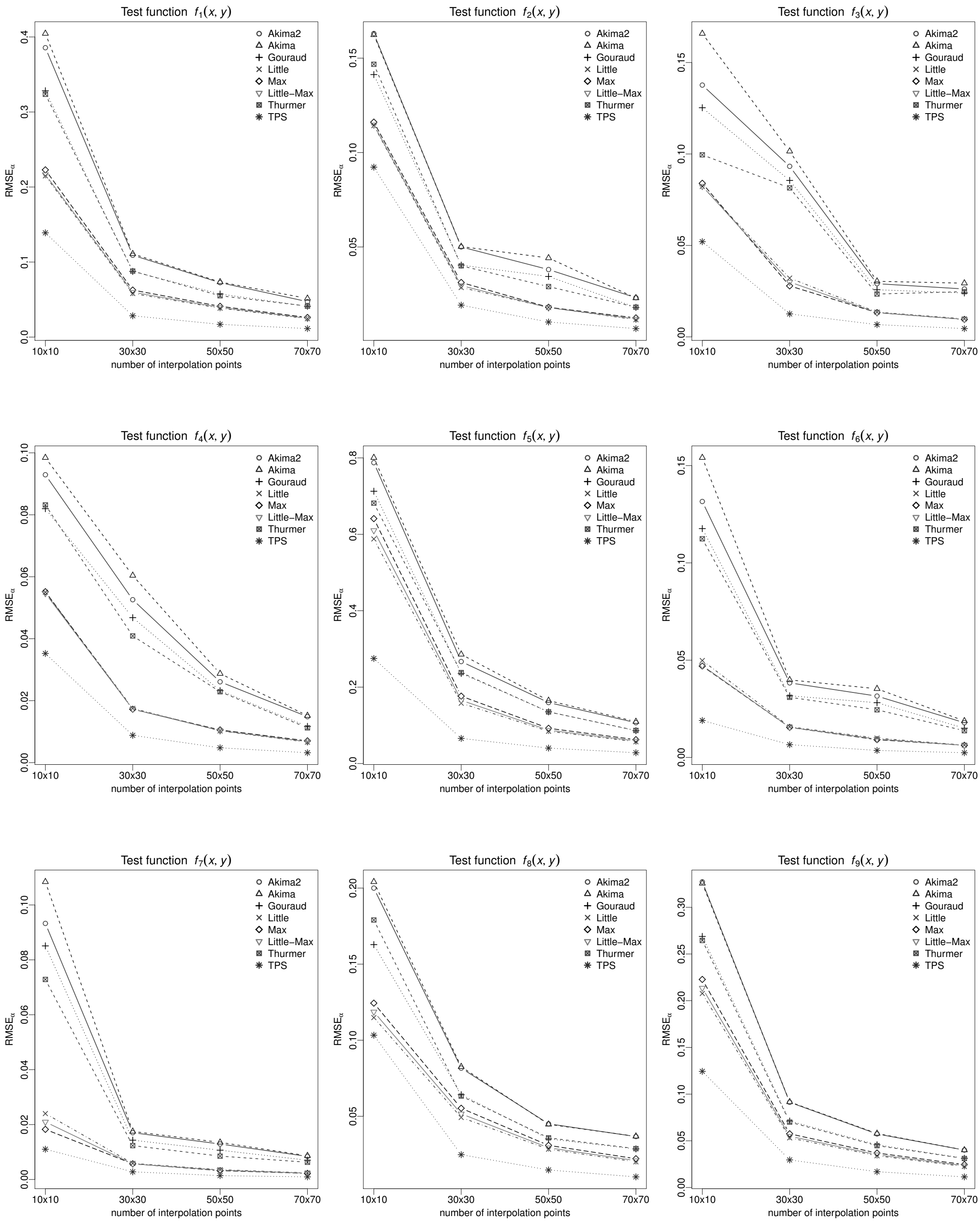


Fig. 5. Plots of deviations between the calculated normal and the analytical normal of the test function

Deviations between the Clough-Tocher interpolant and the test function

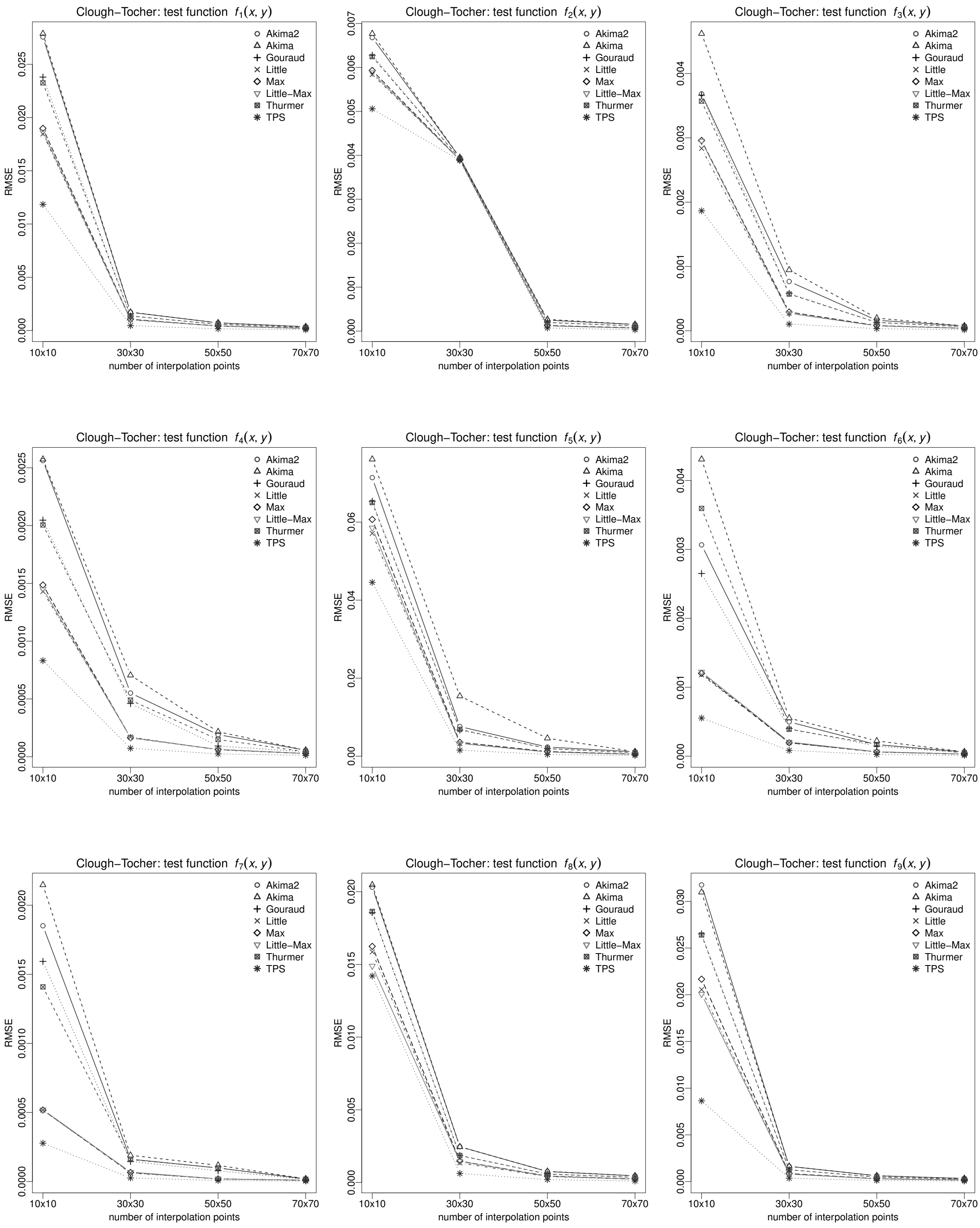


Fig. 6. Plots of deviations between the function value of the interpolation and the test function