PARTIAL DERIVATIVE ESTIMATION USING CONVEX COMBINATION METHODS

L. H. T. CHANG
Academic Services Unit, Engineering Campus, Universiti Sains Malaysia,
14300 NIBONG TEBAL, MALAYSIA.
E-mail: lawrence@eng.usm.my

Y. P. WONG
Faculty of Information Technology, Multimedia University,
63100 CYBERJAYA, MALAYSIA.
E-mail: ypwong@mmu.edu.my

Abstract: This paper will discuss methods to estimate partial derivatives at scattered data points in 3-dimension. In particular, we shall discuss methods based on data points that have been triangulated. One very quaint method is the convex combination method (Goodman *et al.*, 1995) that we shall discuss and improve upon using Martin's weights.

1. Introduction

Let us say we are given a set of scattered data points in 3-dimension and we wish to define a surface which passes through these points. One very popular approach is to start by defining non-overlapping triangular patches with the points being the vertices of these triangles. Some of the methods that we can use to triangulate data points can be found in Barnhill (1977), Brassel and Reif (1979) and Fang and Piegl (1992a, 1992b). After the triangulation process, we interpolate each of the generated triangular patches. This can be done using a choice of several existing methods, e.g. splines, Bézier's methods. Again, the more popular methods define the triangular surfaces based on control points that we can determine depending on how we want the triangular surfaces to join together. Examples of references with basic but very extensive tutorials on how to generate triangular surfaces are Böhm *et al.* (1984) and Farin (1986).

How the triangular surfaces join at the boundaries will determine whether the overall surface looks smooth or not. Therefore, it is important we place the control points for adjoining triangles in such a way that the adjoining surfaces join with the type of continuity we want. A smooth overall surface results when the triangular surfaces join with tangential continuity, or C^1 continuity, at the very least. For this to happen, the control points of every pair of adjoining triangular patches will need to be placed in such a way that the first set of inner control points at the common boundary of both patches are coplanar.

In order for us to calculate the position of these control points that yield a C^1 continuous surface, we need to know the first-order partial derivatives at the vertices of the triangles, or in other words, at all the given data points. In actual practice, we would have collected these data points at field sites or some laboratory experiments and there is no possibility of knowing the partial derivatives at these points. Therefore, if we still want to generate a smooth surface from these data points, we have no choice but to estimate the partial derivatives at these points.

One very common method used to estimate partial derivatives is the least squares method (Renka & Cline, 1984). Although the least squares method is easy to understand, the calculation effort is tremendous. Goodman *et al*. (1995) introduced the convex combination method that has been shown to be more accurate and requires less computation compared to the least squares method. It is well known that for a triangular interpolation to be good, the generated triangles, though most likely scalene or isosceles, should be shaped as close as possible like that of an equilateral triangle, i.e. we do not want triangles that are too thin or too long. However, sometimes the scattered data points that are given are such that when triangulated, thin triangles are inevitably generated. In their paper introducing the convex combination method to estimate partial derivatives, Goodman *et al*. used the reciprocals of the base triangular areas as weights. There is a weakness that Goodman *et al.*'s weights have with thin triangles and in this paper we shall replace their weights with Martin's suggestion of the reciprocals of the base triangular heights to overcome this weakness. We complete this paper with a pseudo computer program or algorithm that we hope readers can easily adapt to real codes.

2. Goodman *et al*.'s convex combination method

The idea for the convex combination method is based on the polynomial interpolating three univariate data points. It is well known that for a polynomial interpolating three data points to be unique, it has to be a quadratic.

Let us consider three consecutive univariate data points, $I_{i-1} = (x_{i-1}, y_{i-1})$, $I_i = (x_i, y_i)$ and $I_{i+1} = (x_{i+1}, y_{i+1})$. We shall also denote the base, or *x*-coordinate, distances between I_{i-1} and I_i as

$$\ell_1 = x_i - x_{i-1} \tag{1}$$

and between I_i and I_{i+1} as

$$\ell_2 = x_{i+1} - x_i \tag{2}$$

This is shown in Figure 1.

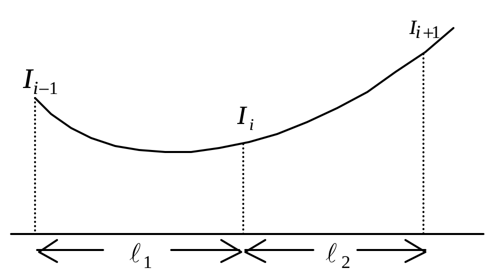

Figure 1. Figure showing three consecutive data points, I_{i-1}, I_i and I_{i+1}.

We shall also denote g_1 as the gradient joining I_{i-1} and I_i,

$$g_1 = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = \frac{y_i - y_{i-1}}{\ell_1} \tag{3}$$

and g_2 as the gradient joining I_i and I_{i+1},

$$g_2 = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{\ell_2} \tag{4}$$

We shall now use Lagrange's method to generate the unique quadratic interpolating I_{i-1}, I_i and I_{i+1}.

$$y = \frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} y_{i-1} + \frac{(x-x_{i-1})(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})} y_i + \frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} y_{i+1} \tag{5}$$

Differentiating (5) with respect to x, we obtain

$$\frac{dy}{dx} = \frac{2x - x_i - x_{i+1}}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} y_{i-1} + \frac{2x - x_{i-1} - x_{i+1}}{(x_i-x_{i-1})(x_i-x_{i+1})} y_i + \frac{2x - x_{i-1} - x_i}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} y_{i+1} \tag{6}$$

Thus, the derivative at point $I_i = (x_i, y_i)$ is

$$\frac{dy}{dx}(x_i, y_i) = \frac{x_i - x_{i+1}}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} y_{i-1} + \frac{2x_i - x_{i-1} - x_{i+1}}{(x_i-x_{i-1})(x_i-x_{i+1})} y_i + \frac{x_i - x_{i-1}}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} y_{i+1} \tag{7}$$

Substituting (1), (2), (3) and (4) into (7) and simplifying, we obtain

$$\frac{dy}{dx}(x_i, y_i) = \frac{\dfrac{g_1}{\ell_1} + \dfrac{g_2}{\ell_2}}{\dfrac{1}{\ell_1} + \dfrac{1}{\ell_2}} \tag{8}$$

We cannot use equation (8) to estimate the derivatives at the end-points because the quadratic interpolant is defined at only one side of each end-point when clearly, equation (8) requires that the quadratic be defined at both sides of each data point. In order to overcome this problem, let us as an example, assume that point $I_{i-1} = (x_{i-1}, y_{i-1})$ is an end-point. The derivative of the interpolating quadratic at this point is

$$\frac{dy}{dx}(x_{i-1}, y_{i-1}) = \frac{2x_{i-1} - x_i - x_{i+1}}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} y_{i-1} + \frac{x_{i-1} - x_{i+1}}{(x_i-x_{i-1})(x_i-x_{i+1})} y_i + \frac{x_{i-1} - x_i}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} y_{i+1} \tag{9}$$

Substituting (1), (2), (3) and (4) into (9) and simplifying, we obtain

$$\frac{dy}{dx}(x_{i-1}, y_{i-1}) = \frac{(2\ell_1 + \ell_2)g_1 - \ell_1 g_2}{\ell_1 + \ell_2} \tag{10}$$

All that is left to do now is to translate the estimation of the partial derivatives in (8) for inner points and (10) for end-points to that for a 3-dimensional case. In the 2-dimensional case, the base distance is the length of the curve projected onto the x-axis (*see* Figure 1). Therefore, in the 3-dimensional case, the base triangle shall be the triangle projected onto the xy-plane.

Figure 2 shows an inner point k surrounded by triangular patches $T_i, i = 1, \cdots, n$.

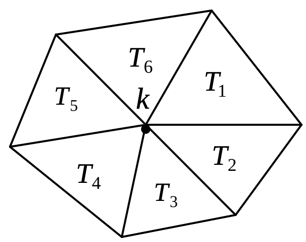

Figure 2. An inner point k surrounded by six triangular patches.

Goodman *et al.* translated the base, or x-coordinate, distances ℓ in equation (8) as the base areas of the triangular patches $T_i, i = 1, \cdots, n$. Therefore, the estimated partial derivatives at point k becomes

$$D_k = \frac{\frac{g_1}{\Delta_1} + \frac{g_2}{\Delta_2} + \cdots + \frac{g_n}{\Delta_n}}{\frac{1}{\Delta_1} + \frac{1}{\Delta_2} + \cdots + \frac{1}{\Delta_n}} = \frac{\sum_{i=1}^{n} \frac{g_i}{\Delta_i}}{\sum_{i=1}^{n} \frac{1}{\Delta_i}} \tag{11}$$

where $\Delta_i, i = 1, \cdots, n$ are the base triangular areas and $g_i, i = 1, \cdots, n$ are the partial derivatives of the triangular planes respectively.

Now, let us look at a boundary point that we shall denote as j, where j is one of the vertices of triangles $T_i, i = 1, \cdots, n$. We denote also triangles $T_i', i = 1, \cdots, n$ as the adjoining triangles to triangles $T_i, i = 1, \cdots, n$ respectively. This is shown in Figure 3.

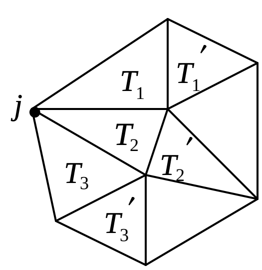

Figure 3. Triangles T_1, T_2 and T_3 having boundary point j as a vertex and adjoining triangles T_1', T_2' and T_3' respectively.

In order to obtain a convex combination estimate for the partial derivatives at the boundary points, Goodman *et al.* translated the partial derivative for an end-point (10) for the univariate case to the bivariate case as follows:

$$D_j = \frac{\sum_{i=1}^{n} \frac{1}{\Delta_i} \left(\frac{\left(2\Delta_i + \Delta_i'\right) g_i - \Delta_i g_i'}{\Delta_i + \Delta_i'} \right)}{\sum_{i=1}^{n} \frac{1}{\Delta_i}} \tag{12}$$

and as before, Δ are the base triangular areas, g are the partial derivatives of the triangular planes respectively and $\frac{1}{\Delta_i}$, $i = 1, \cdots, n$ are the weights.

3. Martin's weights

Some years ago, the first author had the good fortune of meeting up with Dr. Ralph R. Martin who was visiting the campus that the first author was attached to. At that time, the first author was involved in a 3-dimensional scattered data interpolation project and as raw data was involved, the estimation of partial derivatives became necessary. Martin was visiting the laboratory where the project was conducted and he suggested that instead of using the reciprocals of the areas of the base triangles as weights, we should use the reciprocals of the perpendicular heights of the base triangles, taken from the points in question to the opposite sides, as weights. Martin's reasoning was that since base distances played an important role in determining the weights in the 2-dimensional univariate case, base distances should also play the same role in the 3-dimensional bivariate case. If a particular triangle is long or thin, the area of the triangle is small but the height is great. Similarly, a short but fat triangle has a bigger area but a lesser height. Therefore, the height of the base triangle should be a more proper analogy to the base distance compared to the area of the base triangle. This can be observed in Figures 4(a) and 4(b) that show the height of each triangle from the point to the opposite side.

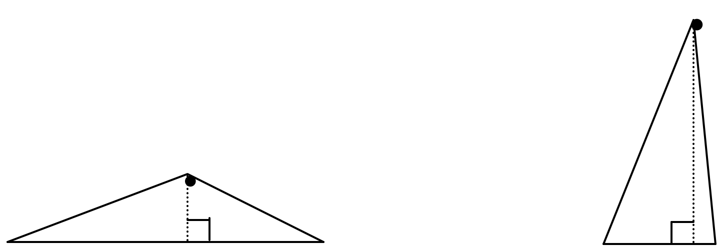

Figure 4(a) Figure 4(b)

Figures 4(a) & 4(b). The height of each triangle from the point in question to the opposite side.

A triangle has three perpendicular heights thereby giving three weights. For the estimation of partial derivatives of the inner points, the considered weight from each triangle is the reciprocal of the height from the point in question to the opposite side (*see* Figure 5). For boundary points, where we have to also consider each adjoining triangle (*see* Figure 6), we propose that for the adjoining triangle, the perpendicular distance from the common boundary to the opposite vertex shall be the base distance taken into consideration.

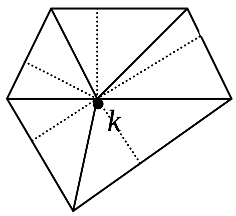

Figure 5. The heights of triangles with point k as a vertex.

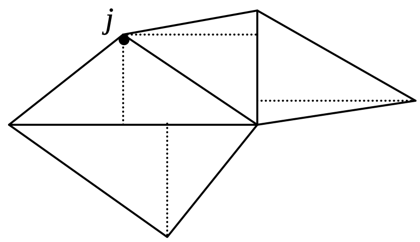

Figure 6. The heights of triangles with point j as a vertex together with heights of the adjoining triangles.

Rewriting equations (11) and (12) with Martin's weights, we have for the estimation of partial derivatives at the inner points

$$D_k = \frac{\sum_{i=1}^{n} \frac{g_i}{h_i}}{\sum_{i=1}^{n} \frac{1}{h_i}} \tag{13}$$

and for the estimation of partial derivatives at the boundary points

$$D_j = \frac{\sum_{i=1}^{n} \frac{1}{h_i} \left(\frac{\left(2h_i + h_i'\right) g_i - h_i g_i'}{h_i + h_i'} \right)}{\sum_{i=1}^{n} \frac{1}{h_i}} \tag{14}$$

where h symbolise the base triangular heights and g symbolise the partial derivatives of the triangular planes.

4. Results and Conclusion

We tested the partial derivative estimation methods mentioned above using two well-known test functions, i.e.

1. Franke's exponential function.

$$F1(x,y) = 0.75e^{-\frac{(9x-2)^2+(9y-2)^2}{4}} + 0.75e^{-\left(\frac{(9x+1)^2}{49}+\frac{9y+1}{10}\right)} + 0.50e^{-\frac{(9x-7)^2+(9y-3)^2}{4}} - 0.20e^{-\left((9x-4)^2+(9y-7)^2\right)}$$

2. Saddle function.

$$F2(x,y) = \frac{1.25+\cos(5.4y)}{6+6(3x-1)^2}$$

The test domain that we employ here is the one used by Whelan (1984) which is a unit square with 36 points taken from within the square (*see* Table 1). These 36 points were triangulated using Delaunay's triangulation method with the resulting triangles shown in Figure 7.

Table 1. Planar data points used in our tests.

Point No.	**Coordinates**		**Point No.**	**Coordinates**	
	x	*y*		*x*	*y*
1	0.00	0.00	19	0.80	0.85
2	0.50	0.00	20	0.85	0.65
3	1.00	0.00	21	1.00	0.50
4	0.15	0.15	22	1.00	1.00
5	0.70	0.15	23	0.50	1.00
6	0.50	0.20	24	0.10	0.85
7	0.25	0.30	25	0.00	1.00
8	0.40	0.30	26	0.25	0.00
9	0.75	0.40	27	0.75	0.00
10	0.85	0.25	28	0.25	1.00
11	0.55	0.45	29	0.00	0.25
12	0.00	0.50	30	0.75	1.00
13	0.20	0.45	31	0.00	0.75
14	0.45	0.55	32	1.00	0.25
15	0.60	0.65	33	1.00	0.75
16	0.25	0.70	34	0.19	0.19
17	0.40	0.80	35	0.32	0.75
18	0.65	0.75	36	0.79	0.46

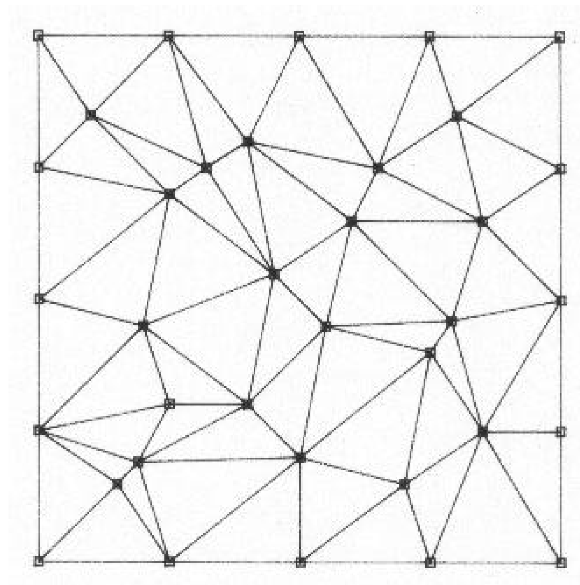

Figure 7. A Delaunay triangulation of the 36 data points in Table 1.

The partial derivatives of both test functions at the 36 data points were estimated using both weights suggested by Goodman *et al.* and Martin. The results are shown in the Appendices that can be found in the electronic version of this paper. For a quick comparison between the two weights, we table the maximum absolute errors and the mean absolute errors in Table 2.

Table 2. A comparison of the test results.

Function and weight used.	**Max. abs. error**		**Mean abs. error**	
	$\frac{dF}{dx}$	$\frac{dF}{dy}$	$\frac{dF}{dx}$	$\frac{dF}{dy}$
Franke's exponential function, weight $=\frac{1}{\Delta}$	0.972078	1.138082	0.340011	0.391382
Franke's exponential function, weight $=\frac{1}{h}$	0.835827	1.124918	0.311411	0.375929
Saddle function, weight $=\frac{1}{\Delta}$	0.298376	0.380398	0.084480	0.107790
Saddle function, weight $=\frac{1}{h}$	0.290634	0.408416	0.081809	0.103958

Although Table 2 shows that the maximum absolute error of the estimated derivative $\frac{dF}{dy}$ of the saddle function increases when using Martin's weights, all the other errors importantly the mean absolute errors decrease showing that Martin's weights do result in an improvement in the estimation of partial derivatives compared to Goodman *et al.*'s weights. Using Martin's weights requires extra calculations because there are three perpendicular heights in each base triangle but the extra programming required is not much of an extra burden and the difference in computation time too is hardly noticeable.

5. Algorithm to estimate partial derivatives for scattered data points

Below, we present a pseudo program to estimate partial derivatives for scattered data points that we hope readers can easily implement in real code.

```
Begin
    Input data points;
    Triangulate data points;
    For each triangle
      begin
          Calculate the derivatives of the triangular plane;
          Calculate the three perpendicular heights of the triangle projected onto the xy-plane;
      end;
    For each data point i
      begin
          Numerator for derivative with respect to x  = 0;
          Numerator for derivative with respect to y = 0;
          Denominator = 0;
          For each triangle j
                  If point i is a vertex of triangle j
                    begin
                          If point i is a boundary point
                                  For each triangle k
                                          If triangle k is adjoining triangle j
                                                  Add values from equation (14) to both
                                                  numerators with respect to x and y;
                          If point i is an inner point
                                  Add values from equation (13) to both numerators with
                                  respect to x and y;
                          Add values from either equation (13) or equation (14) to denominator;
                    end;
          Estimate both partial derivatives by respective numerator/denominator;
      end;
End.
```

Acknowledgements

Both authors wish to thank Ms. Wei-Li Pang for having done the initial programming for this work.

The first author also wishes to thank Multimedia University for making possible his trip to the Faculty of Information Technology at the said university.

References

Barnhill, R.E. (1977). Representation and approximation of surfaces. In *Mathematical Software III* (Rice, J.R.,ed.), p.68-119. Academic Press.

Böhm, W., Farin, G. & Kahmann, J. (1984). A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design* **1**, 1-60.

Brassel, K.E. & Reif, D. (1979). A procedure to generate Thiessen polygons. *Geographical Analysis* **11**(3), 289-303.

Fang, T.P. & Piegl, L.A. (1992a). Algorithm for Delaunay triangulation and convex-hull computation using a sparse matrix. *Computer-Aided Design* **24**(8), 425-436.

Fang, T.P. & Peigl, L.A. (1992b). Delaunay triangulation in the plane using a uniform grid. *Proceedings of the 5th International Conference on Engineering, Computer Graphics and Descriptive Geometry*, 250-255. Melbourne, Australia.

Farin, G. (1986). Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design* **3**, 83-127.

Goodman, T.N.T., Said, H.B. & Chang, L.H.T. (1995). Local derivative estimation for scattered data interpolation. *Applied Mathematics and Computation* **68**, 41-50.

Renka, R.J. & Cline, A.K. (1984). A triangle-based C^1 interpolation method. *Rocky Mountain Journal of Mathematics* **14**(1), 223-237.

Whelan, T. (1986). A representation of a C^2 interpolant over triangles. *Computer Aided Geometric Design* **3**, 53-66.