Triangulations and Applications

Figures algorithms and some equations as an aid to oral exam.

November 27, 2007
Contents

1 Delaunay Triangulations and Voronoi Diagrams ............ 1
2 Algorithms for Delaunay Triangulation ...................... 9
3 Data Dependent Triangulations ............................... 19
4 Constrained Delaunay Triangulation ......................... 29
5 Delaunay Refinement Mesh Generation ....................... 35
6 Least Squares Approximation of Scattered Data ............. 47
   6.1 Approximation over Triangulations of Subsets of Data .... 49
   6.2 Existence and Uniqueness .................................. 51
   6.3 Sparsity and Symmetry .................................... 52
   6.4 Penalized Least Squares ................................. 53
   6.5 Smoothing Terms for Penalized Least Squares ........... 54
   6.6 Approximation over General Triangulations ............. 58
   6.7 Weighted Least Squares ................................. 60
6.8 Constrained Least Squares ................................. 61
   6.9 Approximation over Binary Triangulations ............... 62
   6.10 Numerical Examples for Binary Triangulations .......... 63
References .............................................................. 69
List of Algorithms

1.1 Circumcircle test ........................................ 1
1.2 LOP, Local Optimization Procedure ................. 1
2.1 Simple Delaunay triangulation ......................... 9
2.2 Radial Sweep .......................................... 9
2.3 recSwapDelaunay(Edge $e_i$) ......................... 10
3.1 LOP for data dependent triangulations.............. 19
3.2 Simulated annealing .................................... 20
4.1 Eliminate $u_m$, with $\alpha_m < \pi$, from $Q^{e_c,L}$ .... 29
4.2 Include $e_c$ as an edge in a triangulation .......... 29
4.3 recSwapDelaunayConstr(Edge $e_i$) ................. 29
5.1 Delaunay refinement .................................... 35
6.1 General multilevel scheme ............................. 62
Algorithm 1.1 Circumcircle test

1. if \((\cos\alpha < 0 \text{ and } \cos\beta < 0)\)
2. \hspace{1em} return FALSE \hspace{0.5em} // swap the edge
3. if \((\cos\alpha > 0 \text{ and } \cos\beta > 0)\)
4. \hspace{1em} return TRUE
5. if \((\cos\alpha \sin\beta + \sin\alpha \cos\beta < 0)\)
6. \hspace{1em} return FALSE \hspace{0.5em} // swap the edge
7. else
8. \hspace{1em} return TRUE

Algorithm 1.2 LOP, Local Optimization Procedure

1. Make an arbitrary legal triangulation \(\Delta\) of a point set \(P\).
2. if \(\Delta\) is locally optimal,
\hspace{1em} stop.
3. Let \(e_i\) be an interior edge of \(\Delta\) which is not locally optimal.
4. Swap \(e_i\) to \(e'_i\), thus transforming \(\Delta\) to \(\Delta'\).
5. Let \(\Delta := \Delta'\).
6. goto 2.
Fig. 1.1. Two triangulations of the same point set that satisfy the MaxMin angle criterion. The triangulation in (a) is a Delaunay triangulation.

Fig. 1.2. A neutral case for the MaxMin angle criterion. Only $\Delta^c$ satisfy the MinMax criterion. The triangulation $\Delta^d$ is optimal with respect to the MaxMin angle criterion.
Fig. 1.3. The Voronoi region of a point $p_i$ in the plane.

Fig. 1.4. The Voronoi diagram of a set of points in the plane.
Fig. 1.5. The straight line dual of the Voronoi diagram which yields a Delaunay triangulation.

Fig. 1.6. Almost cocircular points \( p_1, p_2, p_3 \) and \( p_4 \) left, and cocircular points right. In the cocircular case there are two possible solutions when constructing the Delaunay triangulation.
Fig. 1.7. A Voronoi diagram and its straight-line dual, the Delaunay triangulation, with some of the circumcircles.

Fig. 1.8. The circimecircle $C(t_{1,2,3})$ of a triangle in a Delaunay triangulation and the Voronoi regions of $p_1$, $p_2$ and $p_3$. A point $p_4$ inside $C(t_{1,2,3})$ is not in agreement with the definition of a Voronoi diagram.
Fig. 1.9. Illustration for the proof of Lemma ??.

Fig. 1.10. Inscribed angles $\alpha$ on the same arc are equal, and $\gamma < \alpha < \beta$.

Fig. 1.11. Illustration for the proof of Lemma ??.
Fig. 1.12. Example showing that choosing the diagonal with the minimum length violates the circle criterion.

Fig. 1.13. Illustration for circumcircle test based on interior angles.

Fig. 1.14. Illustration for the proof of Theorem ??.
Fig. 1.15. Voronoi diagram of a point set $P$ on the left, and Voronoi diagram of a subset $\bar{P} \subset P$ on the right.

Fig. 1.16. Illustration for the proof of Theorem ??.
Algorithms for Delaunay Triangulation

Algorithm 2.1 Simple Delaunay triangulation
1. Compute \( \text{conv}(P) \).
2. Apply Algorithm ?? to the vertices of \( \text{conv}(P) \) to find an initial triangulation \( \Delta' \).
3. Apply Algorithm ?? to insert points of \( P \), that are interior to \( \text{conv}(P) \), into \( \Delta' \). This gives a new triangulation \( \Delta'' \) after all points have been inserted.
4. Apply the LOP repeatedly on the edges of \( \Delta'' \) until no edge-swap occurs, and thus obtain a triangulation \( \Delta^* \) which has all of its edges locally optimal.

Algorithm 2.2 Radial Sweep
1. Choose a point \( p \) near the centroid of \( P \) and connect \( p \) by radiating edges to all other points of \( P \), Figure 2.1(a).
2. Sort and order the points \( \{P \setminus p\} \) by orientation and distance from \( p \) and connect the ordered sequence by edges as in Figure 2.1(b). The result from this step is a triangulation with a star-shaped domain as seen from \( p \). Triangles may be degenerate since points may have identical orientation relative to \( p \).
3. Form a triangle for each triple of points \( (p_{i-1}, p_i, p_{i+1}) \) on the boundary of the triangulation. If the edge between \( p_{i-1} \) and \( p_{i+1} \) is outside the existing boundary, include the triangle in the triangulation and update the boundary. Repeat this step until no more triangles can be added. The resulting triangulation has a convex boundary and all points are included in the triangulation, Figure 2.1(c).
4. Apply the LOP repeatedly to the edges until no edge-swap occurs, to obtain a Delaunay triangulation, Figure 2.1(d).
# Algorithm 2.3 \texttt{recSwapDelaunay(Edge } e_i \texttt{)}

1. if (circumcircleTest\(e_i\) == OK) \quad // Algorithm 1.1 in Section ??
2. return
3. swapEdge\(e_i\) \quad // the swapped edge \(e'_i\) is incident with \(p\)
4. \texttt{recSwapDelaunay}(e_{i,1}) \quad // call this procedure recursively
5. \texttt{recSwapDelaunay}(e_{i,2}) \quad // call this procedure recursively

\[ P_0^0 = P \]

\[ \begin{array}{cccccc}
    & P_0^1 & \quad & P_1^1 \\
    P_0^2 & & & P_3^2 \\
    P_0^3 & P_1^3 & P_2^3 & P_3^3 \\
\end{array} \]

\[ \begin{array}{cccccc}
    \Delta_0^3 & \Delta_1^3 & \Delta_2^3 & \Delta_3^3 \\
    \Delta_0^2 & \Delta_1^2 & \Delta_2^2 & \Delta_3^2 \\
    \Delta_0^1 & \Delta_1^1 \\
\end{array} \]

\[ \Delta_0^0 = \Delta(P) \]
Fig. 2.1. The Radial Sweep algorithm.
Fig. 2.2. Circle growing from the base line $e_b$ to find a point $p$ to form a new triangle with $e_b$. 
Fig. 2.3. Influence polygons $Q^p$ in (a) and (c) when $p$ is inserted interior and exterior to an existing triangulation. Delaunay triangulations after $p$ has been inserted are shown in (b) and (d).
Fig. 2.4. Swapping procedure when inserting a point $p$ into a Delaunay triangulation. From (b) to the final triangulation in (e), each picture shows the triangulation after one new edge has been swapped.
Fig. 2.5. Starting the recursive swapping procedure.

Fig. 2.6. Illustration for Lemma ??.
Fig. 2.7. (a): The initial edges are Delaunay. (b): Edges that are swapped to \( p \) are Delaunay.

Fig. 2.8. The curves \( y = x \), \( y = x \log x \) and \( y = x^2 \) from bottom to top for illustrating the difference between run-time order of \( O(N) \), \( O(N \log N) \) and \( O(N^2) \).
Fig. 2.9. Illustration of a worst case example for incremental Delaunay triangulation algorithms.

Fig. 2.10. Merging two Delaunay triangulations in the divide-and-conquer algorithm.
Algorithm 3.1 LOP for data dependent triangulations
1. Make an arbitrary legal triangulation \( \Delta \) of a point set \( P \).
2. If \( \Delta \) is locally optimal, that is, if (??) holds for all interior edges in \( \Delta \),
   stop.
3. Let \( e_i \) be an interior edge of \( \Delta \) which is not locally optimal.
4. Swap \( e_i \) to \( e'_i \), transforming \( \Delta \) to \( \Delta' \).
5. Let \( \Delta := \Delta' \).
6. goto 2.

Fig. 3.1. Test function \( F_1(x, y) = (\tanh(9y - 9x) + 1)/9 \).
Algorithm 3.2 Simulated annealing

1. do \( k = 1, \ldots, n_{\text{temps}} \)
2. \( t_k = r^k t_0, \quad 0 < r < 1, \ \text{e.g.,} \ r = 0.95 \)
3. do \( l = 1, \ldots, n_{\text{limit}} \)
4. while the number of “good swaps” \( \leq g_{\text{limit}} \)
5. let \( \Delta \) be the current triangulation; choose a random edge \( e_i \) in \( \Delta \)
6. if \( e_i \) is swappable
7. let \( \Delta' \) be the result of swapping \( e_i \) and let \( d = C_p(\Delta') - C_p(\Delta) \)
be the corresponding change of the global cost function (??).
8. if \( d < 0 \), i.e., if the global cost decreases
9. swap \( e_i \) (“good swap”)
10. else
11. choose a random number \( \theta \), \( 0 \leq \theta \leq 1 \)
12. if \( \theta \leq e^{-d/t_k} \)
13. swap \( e_i \) (“bad swap”)
14. endif
15. endif

(a) (b)

Fig. 3.2. Delaunay triangulation of grid data and level curves.
Fig. 3.3. Triangulation optimized with the LOP algorithm using ABN criterion and $l_1$ norm.

Fig. 3.4. Edges that must be checked when considering swapping the edge $e_i$. 
Fig. 3.5. Geometric embedding information used when considering edge-swap of an edge $e_i$ in a convex quadrilateral defined by two triangles $t_1$ and $t_2$. $Q_1(x,y)$ and $Q_2(x,y)$ are the equations of the planes defined by $t_1$ and $t_2$. The dashed lines display the projection of the quadrilateral in the $(x, y)$-plane. $n = (n_x, n_y)$ is a unit vector orthogonal to the projection of $e_i$ in the $(x, y)$-plane.

Fig. 3.6. The SCO data dependent swapping criterion.
Fig. 3.7. A “bad swap” of $e_2$ followed by a swap of $e_1$ which reduces the global cost.

Fig. 3.8. Probability of making bad swaps in Step 12 of Algorithm 3.2.
Fig. 3.9. Triangulation optimized with the simulated annealing algorithm using ABN criterion and $l_1$ norm.

Fig. 3.10. 1108 points measured from an interior detail of a car.
Fig. 3.11. Delaunay triangulation of the points in Figure 3.10, and level curves.
Fig. 3.12. Data dependent triangulation from LOP algorithm, SCO criterion and $l_1$ norm.
Fig. 3.13. Data dependent triangulation from the simulated annealing algorithm, SCO criterion and $l_1$ norm.
Constrained Delaunay Triangulation

Algorithm 4.1 Eliminate $u_m$ with $\alpha_m < \pi$, from $Q^{e_c,L}$
1. while ($r \geq 1$)
2. Let $(u_m, w_s)$ be a diagonal in a convex quadrilateral $(u_m, w_{s-1}, w_s, w_{s+1})$.
3. Swap $(u_m, w_s)$ to $(w_{s-1}, w_{s+1})$
4. $r \leftarrow r - 1$

Algorithm 4.2 Include $e_c$ as an edge in a triangulation
1. while ($n > 1$)
2. Find a point $u_m$, $1 \leq m \leq n$ where $\alpha_m < \pi$
3. Apply Algorithm 4.1 to $u_m$
4. $n \leftarrow n - 1$
5. $Q^{e_c,L} \leftarrow (p_a, u_1, \ldots, u_{m-1}, u_{m+1}, \ldots, u_n, p_b)$

Algorithm 4.3 recSwapDelaunayConstr(Edge $e_i$)
1. if $(e_i \in E_c)$
2. return
3. if (circumcircleTest($e_i$) == true) // Algorithm 1.1 in Section ??
4. return
5. swapEdge($e_i$) // the swapped edge $e'_i$ is incident with $p$
6. recSwapDelaunay($e_{i,1}$) // call this procedure recursively
7. recSwapDelaunay($e_{i,2}$) // call this procedure recursively
Fig. 4.1. (a): A planar straight-line graph $G(P, E_c)$. (b): Conventional Delaunay triangulation of the point set $P$. (c): Constrained Delaunay triangulation of $G(P, E_c)$. (d): Illustration of the modified circle criterion for constrained Delaunay triangulation.

Fig. 4.2. The triangulation in (a) with $e$ as an edge is a conventional Delaunay triangulation. The triangulation in (b) is a constrained Delaunay triangulation due to the constrained edge $e'$. 
Fig. 4.3. (a) The influence region of a constrained edge $e_c$ that is inserted into an existing triangulation. (b) the influence polygons $Q_c^L$ and $Q_c^R$ of $e_c$.

Fig. 4.4. The growing circle reaches a point $p'$ first, but $p'$ is separated from $e_b$ by a constrained edge and cannot form a triangle with $e_b$. 
Fig. 4.5. Retriangulation of the influence region of a constrained edge $e_c$. (a) to (c) show how new triangles are constructed when retriangulating the left influence polygon $Q^{e_c,L}$. The growing circles (dotted) are shown when they have reached a point where a new triangle can be formed with the base line. In (d) the constrained Delaunay triangulation of the whole influence region is shown.
Fig. 4.6. Illustration for edge insertion and swapping. (b) shows the situation when the point $u_1$ has been isolated from the influence polygon $Q_{e_c,L}$, and (c) shows the situation when the last edge is swapped and takes on the role as the constrained edge $e_c$. 
Fig. 4.7. Illustration for Lemma ??.

Fig. 4.8. (a): The influence polygon $Q^p$ of a point $p$ in a CDT $\Delta(P, e_\epsilon)$ is shown with bold edges. (b): The updated CDT $\Delta(P \cup p, e_\epsilon)$. 
Algorithm 5.1 Delaunay refinement

1. Make the initial CDT of the PSLG.
   Remove triangles outside the triangulation domain.
2. while skinny triangles remain  // (controls termination)
3.     while any segment $s$ is encroached upon
4.         SplitSegment($s$)
5.     Let $t$ be a skinny triangle and $v$ the circumcenter of $t$.
6.     if $v$ encroaches upon any segments $s_1, s_2, \ldots, s_k$   // “look-ahead”
7.         for $i = 1, \ldots, k$
8.            SplitSegment($s_i$)
9.     goto 3
10. else
11.     KillTriangle($t$)
12.     goto 5
Case 1. *(KillTriangle)* $v$ is inserted at the circumcenter of a skinny triangle $t$.

The parent node $p$ is chosen to be one of the two endpoints of the shortest edge of $t$. (Figure 5.9).

Case 2. *(SplitSegment)* $v$ is a node inserted at the midpoint of a segment $s$ that is encroached upon by a node $p$.

Thus $p$ lies inside the diametral circle of $s$. If more than one node encroaches upon $s$, assume without loss of generality that $p$ is the closest node to $v$ that encroaches upon $s$. The shortest edge connected to $v$, which defines the insertion radius $r_v$, has $p$ as the other endpoint unless $p$ is not yet inserted and thus rejected. This follows from the Delaunay property. Four possible roles of $p$ must be considered under SplitSegment.

Case 2a. $p$ is an input node, or $p$ is a node inserted in a segment not incident to $s$. (Figure 5.10(a))

Case 2b. $p$ is at the circumcenter of a skinny triangle and thus rejected since it encroaches upon $s$. (Figure 5.10(b), and Step 6–8 of Algorithm 5.1.)

Case 2c. $p$ is a node on a segment $s'$ incident to $s$ that makes an angle $45^\circ \leq \alpha < 90^\circ$ with $s$. (Figure 5.10(c))

Case 2d. As Case 2c with $\alpha \leq 45^\circ$. (Figure 5.10(d))
A sample input planar straight line graph (PSLG).

Constrained Delaunay triangulation of the PSLG. Encroached segments are bold.

One encroached segment has been bisected.

A second encroached segment has been bisected.

A third encroached subsegment has been bisected.

The last encroached subsegment has been bisected and a skinny triangle is found.

The skinny triangle’s circumcenter is inserted. Find another skinny triangle.

This circumcenter encroaches upon a segment, and is therefore rejected.

Although the vertex was rejected, the segment it encroached upon is still marked for bisection.

The encroached segment is split, and the skinny triangle that led to its bisection is eliminated.

A circumcenter is successfully inserted, creating another skinny triangle.

The triangle’s circumcenter is rejected since it encroaches upon a segment.

The encroached segment will be bisected.

The skinny triangle was not eliminated. Try to insert its circumcenter again.

This time, its circumcenter is inserted successfully. Only one skinny triangle is left.

The final mesh with no interior angle smaller than \( \arcsin \frac{1}{2} \approx 20.7^\circ \).

Fig. 5.1. The Delaunay refinement algorithm step-by-step with upper bound \( B = \sqrt{2} \) on the circumradius-to-shortest-edge ratio. Illustration and most figure texts from Shewchuk [1].
Fig. 5.2. Relationship between circumradius-to-shortest-edge ratio $r/l$ and the minimum angle $\alpha_{\text{min}}$ of a triangle: $r/l = 1/(2 \sin \alpha_{\text{min}})$. 
Fig. 5.3. (a): A skinny triangle $t$ in a Delaunay triangulation. (b): $t$'s circumcircle.
(c): The Voronoi diagram and the circumcenter of $t$ positioned at a Voronoi point.
(d): Updated Delaunay triangulation after insertion of a node at $t$'s circumcenter.
Fig. 5.4. Spatial graded mesh uppermost, and a uniform mesh below. Illustration from Shewchuk [1].
Fig. 5.5. Recursive bisection of a segment that is encroached upon. (a): Two nodes encroach upon the segment initially. (b): After splitting the segment at its midpoint, it is still encroached upon by a node. (c): After the second bisection there is no encroachment.

Fig. 5.6. Recursive bisection of incident segments that never terminates when $\alpha \leq 45^\circ$. 
Fig. 5.7. Illustration for Lemma ??.

Fig. 5.8. Illustration of local feature size $\ell f(s)$ at some points relative to a planar straight line graph. An arbitrary point in the plane marked with $\times$ has local feature size equal to the radius of the circle drawn with center at $\times$. 
Fig. 5.9. Case 1 where $v$ is inserted at the center of a skinny triangle $t$. The parent node $p$ is chosen as one of the endpoints of the shortest edge of $t$. 
Fig. 5.10. Different roles of a parent node $p$ which encroaches upon a segment $s$.
(a): $p$ is on the input PSLG.
(b): $p$ is at the circumcenter of a skinny triangle (and rejected).
(c) and (d): $p$ is on a segment $s'$ incident with $s$ with $\alpha \geq 45^\circ$ and $\alpha < 45^\circ$ respectively. The filled bullet $\bullet$ at $p$ indicates the position of $p$ which defines the lower bounds on the insertion radius $r_v$ in each case of Lemma ??.
(e): illustrates the same case as (b), but with $p$ in a position which defines the lower bound on $r_v$. 
Fig. 5.11. Corner-pping when angles of the input PSLG are less than 60°. Illustration from Shewchuk [2].
Least Squares Approximation of Scattered Data

\[ S^0_1(\Delta) = \{ f \in C^0(\Omega) : f|_{t_i} \in H^1, \ i = 1, \ldots, |T| \}, \quad (6.1) \]

\[ (N_1(x, y), N_2(x, y), \ldots, N_n(x, y)), \]

\[ N_i(v_j) = \delta_{ij}, \quad j = 1, \ldots, n, \quad (6.2) \]

---

**Fig. 6.1.** A basis function \( N_i(x, y) \) for the function space \( S^0_1(\Delta) \) and its (compact) support \( \Omega_i \).

\[ f(x, y) = \sum_{i=1}^{n} c_i N_i(x, y), \quad (6.3) \]

\[ \nabla g = \begin{pmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{pmatrix}. \]

\[ n = \begin{pmatrix} -\frac{\partial g}{\partial x} \\ -\frac{\partial g}{\partial y} \\ 1 \end{pmatrix}. \quad (6.4) \]
Least Squares Approximation of Scattered Data

Fig. 6.2. Triangle patch and normal vector.

\[
n' = v_1 \times v_2 = (-\eta_a c_a - \eta_b c_b - \eta_c c_c, -\mu_a c_a - \mu_b c_b - \mu_c c_c, 2A) \quad (6.5)
\]

\[
\eta_a = (y_b - y_c), \quad \eta_b = (y_c - y_a), \quad \eta_c = (y_a - y_b),
\]
\[
\mu_a = (x_c - x_b), \quad \mu_b = (x_a - x_c), \quad \mu_c = (x_b - x_a),
\]

\[
A = \frac{1}{2} ((x_b - x_a)(y_c - y_a) - (y_b - y_a)(x_c - x_a)). \quad (6.6)
\]

\[
\nabla g = (\eta_a c_a + \eta_b c_b + \eta_c c_c, \mu_a c_a + \mu_b c_b + \mu_c c_c) / 2A. \quad (6.7)
\]

\[
\nabla g = \frac{1}{2A} \left( \sum_{i=1}^{n} \eta_i c_i, \sum_{i=1}^{n} \mu_i c_i \right). \quad (6.8)
\]
6.1 Approximation over Triangulations of Subsets of Data

Fig. 6.3. Triangulation generated from a subset of a given data set.

\[ V = \{ v_k = (x_k, y_k) \}_{k=1}^n. \]  \hfill (6.9)

\[ f(x_k, y_k) \approx z_k, \quad k = 1, \ldots, m. \]

\[ I(c) = \sum_{k=1}^m (f(x_k, y_k) - z_k)^2. \]  \hfill (6.10)

\[ I(c) = \sum_{k=1}^m \left( \sum_{j=1}^n c_j N_j(x_k, y_k) - z_k \right)^2 = \|Bc - z\|_2^2. \]  \hfill (6.11)

\[ B = \begin{pmatrix}
N_1(x_1, y_1) & N_2(x_1, y_1) & \cdots & N_n(x_1, y_1) \\
N_1(x_2, y_2) & N_2(x_2, y_2) & \cdots & N_n(x_2, y_2) \\
\vdots & \vdots & \ddots & \vdots \\
N_1(x_m, y_m) & N_2(x_m, y_m) & \cdots & N_n(x_m, y_m)
\end{pmatrix}. \]  \hfill (6.12)

\[ ||u||_2 = (u_1^2 + \cdots + u_m^2)^{1/2}. \]
\[
\frac{\partial I}{\partial c_i} = 2 \sum_{k=1}^{m} \left( \sum_{j=1}^{n} c_j N_j(x_k, y_k) - z_k \right) N_i(x_k, y_k) = 0, \quad i = 1, \ldots, n
\]

\[
\sum_{j=1}^{n} \sum_{k=1}^{m} N_i(x_k, y_k) N_j(x_k, y_k) c_j = \sum_{k=1}^{m} N_i(x_k, y_k) z_k, \quad i = 1, \ldots, n.
\]

\[
(B^T B) c = B^T z, \quad (6.13)
\]

\[
(B^T B)_{ij} = \sum_{k=1}^{m} N_i(x_k, y_k) N_j(x_k, y_k)
\]

\[
(B^T z)_i = \sum_{k=1}^{m} N_i(x_k, y_k) z_k.
\]

**Fig. 6.4.** Least squares approximation to 4500 scattered data points sampled from Franke’s test function. The triangulation has approximately 500 nodes. Triangles are Gouraud-shaded.

\[
f(x, y) = \frac{3}{4} e^{-\frac{(9x-2)^2 + (9y-2)^2}{4}} + \frac{3}{4} e^{-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}} + \frac{1}{2} e^{-\frac{(9x-7)^2 + (9y-3)^2}{4}} + \frac{1}{5} e^{-\frac{(9x-4)^2}{4} - (9y-7)^2}. \quad (6.14)
\]
6.2 Existence and Uniqueness

\[
\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix} = \begin{pmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
N_1(x_{n+1}, y_{n+1}) & N_2(x_{n+1}, y_{n+1}) & \cdots & N_n(x_{n+1}, y_{n+1}) \\
N_1(x_{n+2}, y_{n+2}) & N_2(x_{n+2}, y_{n+2}) & \cdots & N_n(x_{n+2}, y_{n+2}) \\
\vdots & \vdots & \ddots & \vdots \\
N_1(x_m, y_m) & N_2(x_m, y_m) & \cdots & N_n(x_m, y_m)
\end{pmatrix}
\] (6.15)
6.3 Sparsity and Symmetry

\[(B^T B)_{ij} = \sum_{k=1}^{m} N_i(x_k, y_k)N_j(x_k, y_k), \quad (6.16)\]

Fig. 6.5. A possible non-zero off-diagonal element in the system matrix \((B^T B)_{ij}\) corresponds to an edge between the vertices \(v_i\) and \(v_j\) in the triangulation. It is non-zero if one or more data points fall strictly inside \(\Omega_i \cap \Omega_j\).

Fig. 6.6. Sparsity pattern of a 500 × 500 system matrix \(B^T B\) for the least squares problem.
6.4 Penalized Least Squares

Fig. 6.7. Least squares approximation to 4500 scattered data points sampled from Franke’s test function. Random noise was added to the $z$-values of the data points when generating the surface on the right.

\begin{align*}
J(c) &= c^T Ec, \quad (6.17) \\
I(c) &= \sum_{k=1}^{m} (f(x_k, y_k) - z_k)^2 + \lambda J(c) = \sum_{k=1}^{m} (f(x_k, y_k) - z_k)^2 + \lambda c^T Ec \\
&= \|Bc - z\|_2^2 + \lambda c^T Ec \\
(B^T B + \lambda E) c &= B^T z, \quad (6.18) \\
\lambda_d &= \frac{\|B^T B\|_F}{\|E\|_F}. \quad (6.19)
\end{align*}
6.5 Smoothing Terms for Penalized Least Squares

\[ \int |\nabla g|^2 = \int \left[ \left( \frac{\partial g}{\partial x} \right)^2 + \left( \frac{\partial g}{\partial y} \right)^2 \right] \] (6.20)

and the thin-plate energy

\[ \int \left[ \left( \frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 g}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 g}{\partial y^2} \right)^2 \right] \] (6.21)

Membrane energy functional.

\[ \nabla g_k = \left( \frac{\partial g_k}{\partial x}, \frac{\partial g_k}{\partial y} \right) = \frac{1}{2A_k} \left( \sum_{i=1}^{n} \eta_k^i c_i, \sum_{i=1}^{n} \mu_k^i c_i \right). \]

\[ J_1(c) = \sum_{k=1}^{T} A_k |\nabla g_k|^2 = \sum_{k=1}^{T} A_k \left[ \left( \frac{\partial g_k}{\partial x} \right)^2 + \left( \frac{\partial g_k}{\partial y} \right)^2 \right] \]

\[ = \sum_{k=1}^{T} \frac{1}{4A_k} \left[ \left( \sum_{i=1}^{n} \eta_k^i c_i \right)^2 + \left( \sum_{i=1}^{n} \mu_k^i c_i \right)^2 \right] \]

\[ = \sum_{k=1}^{T} \frac{1}{4A_k} \left[ \left( \sum_{i=1}^{n} \eta_k^i c_i \right) \left( \sum_{j=1}^{n} \eta_k^j c_j \right) + \left( \sum_{i=1}^{n} \mu_k^i c_i \right) \left( \sum_{j=1}^{n} \mu_k^j c_j \right) \right] \]

\[ = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{T} \left( \eta_k^i \eta_k^j + \mu_k^i \mu_k^j \right) / 4A_k \] \text{c}_i \text{c}_j = \mathbf{c}^T \mathbf{E} \mathbf{c},

where

\[ E_{ij} = \sum_{k=1}^{T} \left( \eta_k^i \eta_k^j + \mu_k^i \mu_k^j \right) / 4A_k. \]

The Umbrella-operator.

\[ \mathcal{M}_k f = \frac{1}{n_k} \sum_{l \in \partial \Omega} \eta_{kl} - \eta_{kl} \]

\[ \mathcal{M}_k f = \sum_{l=1}^{n} \mu_l^k c_l, \quad \mu_l^k = \begin{cases} -1, & l = k \\ \frac{1}{n_k}, & \text{if } (v_k, v_l) \text{ is an edge in } \Delta \\ 0, & \text{otherwise.} \end{cases} \] (6.22)
6.5 Smoothing Terms for Penalized Least Squares

Fig. 6.8. The umbrella-operator.

Fig. 6.9. Illustration of which triangle vertices that generate non-zero off-diagonal elements in the system matrix together with the vertex \( v_i \). The '1'-vertices generate non-zeros with \( v_i \) by the membrane energy and in the basic least squares problem (matrix \( B^TB \)); the '1' and '2'-vertices generate non-zeros by the thin-plate energy term; and '1', '2' and '3'-vertices generate non-zeros by the umbrella-operator.

\[
\mathcal{J}_2(\mathbf{c}) = \sum_{k=1}^{n} (\mathbf{M}_k f)^2 = \sum_{k=1}^{n} \left[ \sum_{i=1}^{n} \rho_i^k c_i \right]^2 \\
= \sum_{k=1}^{n} \left[ \sum_{i=1}^{n} \rho_i^k c_i \right] \left[ \sum_{j=1}^{n} \rho_j^k c_j \right] \\
= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \rho_i^k \rho_j^k c_i c_j = \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij} c_i c_j = \mathbf{c}^T \mathbf{E} \mathbf{c}, \quad (6.23)
\]

where

\[
E_{ij} = \sum_{k=1}^{n} \rho_i^k \rho_j^k.
\]
\[ \tilde{M}_k = \frac{1}{W_k} \sum_{l \in \partial \Omega_k} \omega_k c_l - c_k, \]

\[ \tilde{\rho}_l^k = \begin{cases} -1, & l = k \\ \frac{\omega_k}{W_k}, & \text{if } (v_k, v_l) \text{ is an edge in } \Delta \\ 0, & \text{otherwise.} \end{cases} \]

**Fig. 6.10.** Stencil for the second order divided difference \( D_k^2 f \) used to make the discrete thin-plate energy measure.

**Discrete thin-plate energy functional.**

\[
\frac{\partial g_i}{\partial n_{ek}} = (\frac{\partial g_i}{\partial x}, \frac{\partial g_i}{\partial y}) \cdot n_{ek} = \nabla g_i \cdot n_{ek}, \quad i = 1, 2.
\]

\[ T_k f = \frac{\partial g_2}{\partial n_{ek}} - \frac{\partial g_1}{\partial n_{ek}} \\
= \left( \frac{\partial g_2}{\partial x} - \frac{\partial g_1}{\partial x}, \frac{\partial g_2}{\partial y} - \frac{\partial g_1}{\partial y} \right) \cdot n_{ek} \\
= (\nabla g_2 - \nabla g_1) \cdot n_{ek} \quad (6.24) \]

\[
\nabla g_1 = (\eta c_t + \eta^1_s c_s + \eta^1_t c_t, \mu c_t + \mu^1_s c_s + \mu^1_t c_t) / 2A_{st},
\]

\[
\nabla g_2 = (\eta c_r + \eta^2_s c_s + \eta^2_t c_t, \mu c_r + \mu^2_s c_s + \mu^2_t c_t) / 2A_{rs}, \quad (6.25)
\]

\[
\mathbf{n}_{ek} = (y_t - y_s, x_s - x_t) / L_{ek}, \quad (6.26)
\]
\[ T_k f = \sum_{i \in \omega(c_k)} \beta_k^i c_i, \]  

where

\[ \beta_k^i = -\frac{L_{e_k}}{2A_{lst}} \]
\[ \beta_k^r = -\frac{L_{e_k}}{2A_{rts}} \]
\[ \beta_k^s = \frac{L_{e_k} A_{ltr}}{2A_{lst}A_{rts}} \]
\[ \beta_k^t = \frac{L_{e_k} A_{srl}}{2A_{lst}A_{rts}}. \]

\[ J_3(c) = \sum_{k=1}^{[E_1]} L_{e_k} (T_k f)^2 = \sum_{k=1}^{[E_1]} L_{e_k} \left[ \sum_{i=1}^{n} \beta_k^i c_i \right]^2 \]
\[ = \sum_{k=1}^{[E_1]} L_{e_k} \left[ \sum_{i=1}^{n} \beta_k^i c_i \right] \left[ \sum_{j=1}^{n} \beta_k^j c_j \right] \]
\[ = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[ \sum_{k=1}^{[E_1]} L_{e_k} \beta_k^i \beta_k^j \right] c_i c_j = \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij} c_i c_j = c^T Ec, \]

where

\[ E_{ij} = \sum_{k=1}^{[E_1]} L_{e_k} \beta_k^i \beta_k^j. \]

\[ T_\lambda(\lambda) = \frac{1}{m} \sum_{k=1}^{m} \left( f_\lambda(x_k, y_k) - z_k \right)^2 \]  

(6.28)
6.6 Approximation over General Triangulations

![Image of point set and triangulation]

**Fig. 6.12.** Point set for penalized least squares. The shaded region is a domain of a basis function $\Omega_j$ that do not cover any data points.

\[
x^T (B^T B)x = (Bx)^T (Bx) = \|Bx\|_2^2 \geq 0.
\]
\[ c^T (B^T B + \lambda E) c = c^T (B^T B) c + \lambda c^T Ec = 0, \quad (6.29) \]

Uniqueness with the membrane energy functional.

\[ J_1(c) = \sum_{k=1}^{[T]} A_k \left[ \left( \frac{\partial g_k}{\partial x} \right)^2 + \left( \frac{\partial g_k}{\partial y} \right)^2 \right] = c^T Ec. \]

\[ c^T (B^T B)c = (Bc)^T (Bc) = \|Bc\|_2^2 = 0, \quad (6.30) \]

Uniqueness with the umbrella-operator.

\[ J_2(c) = \sum_{k=1}^{n} (M_k f)^2 = \sum_{k=1}^{n} \left[ \frac{1}{n_k} \sum_{l \in \partial \Omega_k} c_l - c_k \right]^2. \]

\[ c_k = \frac{1}{n_k} \sum_{l \in \partial \Omega_k} c_l, \quad k = 1, \ldots, n. \quad (6.31) \]

\[ c_k = \frac{1}{W_k} \sum_{l \in \partial \Omega_k} \omega_k c_l, \quad k = 1, \ldots, n \]

Uniqueness with the thin-plate energy functional.

\[ J_3(c) = \sum_{k=1}^{\{|E_r|\}} (T_k f)^2 = \sum_{k=1}^{\{|E_r|\}} \left[ (\nabla g_2 - \nabla g_1) \cdot n_{ek} \right]^2 = c^T Ec. \]
6.7 Weighted Least Squares

\[ I(c) = \sum_{k=1}^{m} w_k \left[ f(x_k, y_k) - z_k \right]^2 = \sum_{k=1}^{m} w_k \left[ \sum_{j=1}^{n} c_j N_j(x_k, y_k) - z_k \right]^2. \]

\[ \frac{\partial I}{\partial c_i} = 2 \sum_{k=1}^{m} w_k \left[ \sum_{j=1}^{n} c_j N_j(x_k, y_k) - z_k \right] N_i(x_k, y_k) = 0, \quad i = 1, \ldots, n, \]

\[ \sum_{j=1}^{n} \sum_{k=1}^{m} w_k N_i(x_k, y_k) N_j(x_k, y_k) c_j = \sum_{k=1}^{m} w_k N_i(x_k, y_k) z_k, \quad i = 1, \ldots, n. \]

\[ (B^T B) c = B^T z, \]

\[ (B^T B)_{ij} = \sum_{k=1}^{m} w_k N_i(x_k, y_k) N_j(x_k, y_k), \quad i, j = 1, \ldots, n, \quad (6.32a) \]

\[ (B^T z)_i = \sum_{k=1}^{m} w_k N_i(x_k, y_k) z_k, \quad i = 1, \ldots, n. \quad (6.32b) \]
6.8 Constrained Least Squares

\[ \Gamma = \{(x_r, y_r, z_r)\}_{r=n+1}^{n+\gamma}. \]

\[ f(x, y) = \sum_{j=1}^{n+\gamma} c_j N_j(x, y). \]

\[ f(x, y) = \sum_{j=1}^{n} c_j N_j(x, y) + \sum_{r=n+1}^{n+\gamma} z_r N_r(x, y). \]

\[ I(c) = \sum_{k=1}^{m} \left[ \sum_{j=1}^{n} c_j N_j(x_k, y_k) + \sum_{r=n+1}^{n+\gamma} z_r N_r(x_k, y_k) - z_k \right]^2. \]

\[ \frac{\partial I}{\partial c_i} = 2 \sum_{k=1}^{m} \left[ \sum_{j=1}^{n} c_j N_j(x_k, y_k) + \sum_{r=n+1}^{n+\gamma} z_r N_r(x_k, y_k) - z_k \right] N_i(x_k, y_k) = 0, \]

\[ \sum_{j=1}^{n} \sum_{k=1}^{m} N_i(x_k, y_k) N_j(x_k, y_k) c_j = \sum_{k=1}^{m} N_i(x_k, y_k) \left[ z_k - \sum_{r=n+1}^{n+\gamma} z_r N_r(x_k, y_k) \right], \]

where

\[ (B^T B)_{i,j} = \sum_{k=1}^{m} N_i(x_k, y_k) N_j(x_k, y_k), \quad i, j = 1, \ldots, n, \quad \text{and} \]

\[ (B^T z)_i = \sum_{k=1}^{m} N_i(x_k, y_k) \left[ z_k - \sum_{r=n+1}^{n+\gamma} z_r N_r(x_k, y_k) \right], \quad i = 1, \ldots, n. \]
6.9 Approximation over Binary Triangulations

The general multilevel scheme.

Algorithm 6.1 General multilevel scheme
1. for \( k = 1, 2, 3, \ldots \)
2. Find a surface approximation \( f_{\Delta_k} \).
3. if \(|f_{\Delta_k}(x_i, y_i) - z_i| < \epsilon, i = 1, \ldots, m, \)
   stop.
4. else
5. Refine \( \Delta_k \) locally where \( \epsilon \) is exceeded to produce \( \Delta_{k+1} \),
   and use \( f_{\Delta_k} \) to make an initial guess for \( f_{\Delta_{k+1}} \).

Approximation Scheme for Binary Triangulations.

![Diagram](image)

Fig. 6.13. Starting with the triangulation \( \Delta_1 \) and the regular grid \( \Psi_1 \) on the left, vertex \( v_{3,1}^2 \) in \( \Psi_2 \) is first activated together with its two parents \( v_{1,0}^1 \) and \( v_{1,1}^2 \) in \( \Psi_1 \). The resulting triangulation is \( \Delta_2 \) in the middle. Next, \( v_{3,3}^3 \) in \( \Psi_3 \) is activated together with ancestors belonging to both \( \Psi_1 \) and \( \Psi_2 \) to obtain \( \Delta_3 \) on the right.

\[
\Psi_k = \{ v_{i,j}^k \}_{i,j=0,0}^{2^k,2^k}.
\]

\[
c_{i,j}^{k+1} = f_k(x_i, y_j).
\]

\[
S_1^0(\Delta_1) \subset S_1^0(\Delta_2) \subset \cdots \subset S_1^0(\Delta_h),
\]
6.10 Numerical Examples for Binary Triangulations

Approximation of data sampled from Franke’s function

Approximation of noisy data from Franke’s function

Approximation of parametrized 3D scattered data.

Terrain modeling of mountain area.
Fig. 6.14. Approximation to Franke’s test function; the resulting binary triangulation imposed on the surface.
Fig. 6.15. The same binary triangulation as in Figure 6.14, and input data for numerical examples. The lower left corner corresponds to the nearest corner in Figure 6.14.

Fig. 6.16. Approximation with huge smoothing parameter.
Fig. 6.17. Approximation to Franke’s function from a data set with noise.

Fig. 6.18. Approximation of terrain data and the given hypsographic data.
Fig. 6.19. Approximation of terrain data, and triangulation imposed on the surface.

Fig. 6.20. Terrain modeling of an area in Jotunheimen, Norway.
Fig. 6.21. Approximation of parametrized 3D scattered data.
References
