On Algorithms for Best L_2 Fits to Continuous Functions with Variable Nodes

M.J. Baines

Numerical Analysis Report 1/93

Department of Mathematics
University of Reading
P.O. Box 220
Reading, UK

The work reported here forms part of the research programme of the Oxford/Reading Institute for Computational Fluid Dynamics.

Abstract

This report gives details of a direct variational approach (with non-standard variations) used to generate algorithms to determine optimal discontinuous piecewise linear and piecewise constant L_2 fitsto a continuous function of one or two variables with adjustable nodes.Algorithms are presented which are fast and robust, and the mesh cannot tangle. An extension to higher dimensions is also given.

In recent years there has been much interest in the use of irregular grids for the representation of quantities in computational modelling. This applies both to economic representation of individual features and tracking of such features as they move. Two approaches to generate such grids are through best fits with variable nodes and through equidistribution. Work on linear splines with free knots has been carried out by de Boor [4], [5], Chiu

$$2$$

$$2$$

$$2$$

piecewise constant $_2$ fits with adjustable nodes on variable triangulations of a region. One of these algorithms, for piecewise constant functions, is particularly robust and successful.

The algorithms are demonstrated on various test functions. In one dimension both the full and approximate methods are fast and robust and give excellent results without any possibility of mesh tangling. In two dimensions on triangles, owing to the complexity of the problem, only the simplest algorithm is demonstrated, on functions with a single severe feature.

The plan of the report is as follows. In section 2 we obtain expeditious natural conditions in one dimension for the $_2$ error between a given continuous function and a piecewise linear discontinuous function with variable nodes to have an extremum. These conditions are then used in section 3 as the basis of a new iterative algorithm designed to obtain the required best fit. The conditions also have a useful geometrical interpretation. Section 3 also contains results on two test functions. The ideas of sections 2 and 3 are repeated in section 4 for the case of piecewise constant functions with variable nodes. Approximate versions

k

k 1

k k

k

k 1

$n+1$

o

1

in (2.7) to obtain

$$(\ ^{*}) \qquad ^{*}(\ ^{*})^{\ 2}{}_{j} = 0 \tag{2 10}$$

The simultaneous solution of (2.8) and (2.10) gives the required solution $^{*}(\ ^{*})$.

Using for left and right values at the (variable) node (see Fig. 1a), (2.10) may be written

$$(\ ^{*}_{j}) \qquad ^{*}_{L}(\ ^{*}_{j})^{\ 2} = \quad (\ ^{*}_{j}) \qquad ^{*}_{R}(\ ^{*}_{j})^{\ 2} \tag{2 11}$$

It follows that, if $(\ ^{*}_{j}) \qquad ^{*}_{L}(\ ^{*}_{j})$ and $(\ ^{*}_{j}) \qquad ^{*}_{R}(\ ^{*}_{j})$ have the same sign, i.e. if $_{L} \quad _{R}$ lie on the same side of $(\ ^{*}_{j})$ (see Fig. 2a),

$$(\ ^{*}_{j}) \qquad ^{*}_{L}(\ ^{*}_{j}) = \quad (\ ^{*}_{j}) \qquad ^{*}_{R}(\ ^{*}_{j})$$

$$^{*}_{L}(\ ^{*}_{j}) = \quad ^{*}_{R}(\ ^{*}_{j}) \tag{2 12}$$

irrespective of $(\ )$ (as long as it is continuous), and therefore that * is continuous at the new position of the node. On the other hand, if $(\ ^{*}_{j}) \qquad ^{*}_{R}(\ ^{*}_{j})$ and $(\ ^{*}_{j}) \qquad ^{*}_{L}(\ ^{*}_{j})$ have opposite signs, i.e. if $_{L} \quad _{R}$ lie on the same side of $(\ ^{*}_{j})$ (see Fig. 2b),

$$(\ ^{*}_{j}) \qquad ^{*}_{L}(\ ^{*}_{j}) = \qquad (\ ^{*}_{j}) \qquad ^{*}_{R}(\ ^{*}_{j})$$

$$\frac{1}{2} \quad ^{*}_{L}(\ ^{*}_{j}) + \quad ^{*}_{R}(\ ^{*}_{j}) \quad = \quad (\ ^{*}_{j}) \tag{2 13}$$

in which case * is discontinuous at $^{*}_{j}$, its jump being bisected by bisected by $(\ ^{*}_{j})$.

Now it is known (Chui [7], Loach Wathen [10]) that for for continuous functions $(\ )$ the best $_{2}$ fit amongst discontinuous piecewise linear functions with variable nodes is continuous, which clearly corresponds to (2.11). The case (2.12), with a definite discontinuity in * at $^{*}_{j}$, therefore cannot correspond to the best least squares fit when $(\ )$ is continuous, and canfiwi.e.kFq[(bwcwi)])(4q5Pq]44(5we.kFI

An algorithm to find optimal piecewise linear $_2$ fits with variable nodes can be constructed in two stages (carried out alternately until convergence is obtained), corresponding to the particular choices of variations referred to in section 2 above.

Stage (i) $_j = 0$, ($= 1$)

$$= \ _{k1} \text{ or } \ _{k2} \ (\ = 1 \qquad + 1) \tag{3 1}$$

This stage of the algorithm is governed by (2.8) and corresponds to the best $_2$

D

j j

j x j

j

2

j j

j

j

k

$$w_k(x) = w_{k1}\phi_1(x) + w_{k2}\phi_2(x) \tag{3.3}$$

(see Fig 1b) and substituting it for $w_k^*(x)$ into (2.8), yielding the equations

$$h_k \begin{bmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} w_{k1} \\ w_{k2} \end{bmatrix} = \int_{k1}^{k2} f(x) \begin{bmatrix} \phi_{k1}(x) \\ \phi_{k2}(x) \end{bmatrix} dx \tag{3.4}$$

where $h_k = x_{k2} - x_{k1}$, while stage (ii) involves taking the $w_k(x)$ which come from (3.4) and solving

$$(f(x_j) - w_{jL}(x_j))^2 - (f(x_j) - w_{jR}(x_j))^2 = 0 \tag{3.5}$$

for a new x_j (see Fig. 2).

From (2.8) we observe that $f(x) - w_k(x)$ must pass through zero at least once between x_{k-1} and x_k. Let x_{OL} and x_{OR} be the zeros closest to node j from the element either side. Then the function

$$F(x) = (f(x) - w_{jL}(x))^2 - (f(x) - w_{jR}(x))^2 \tag{3.6}$$

(*c.f.* (3.5)) has the properties

$$F(x_{0L}) < 0 \qquad F(x_{0R}) > 0, \tag{3.7}$$

(excluding the special case $w_{jL}(x_{0R}) = w_{jR}(x_{0L})$). It follows that there is at least one root of $F(x)$ between x_{0L} and x_{0R}. Choose this root (or the one nearest the old x_j if there are two) to be the new x_j. Note that if this root is chosen, all such roots lie between pairs of intersection points and mesh tangling cannot occur.

In fact, from (3.5), the new x_j must satisfy

$$(w_{jL}(x) - w_{jR}(x))(f(x_j) - \frac{1}{2}(w_{jL}(x) + w_{jR}(x))) = 0 \tag{3.8}$$

from which we either have

$$w_{jL}(x_j) = w_{jR}(x_j) \tag{3.9}$$

or

$$\frac{1}{2}(w_{jL}(x_j) + w_{jR}(x_j)) = f(x_j) \tag{3.10}$$

We shall call (3.9) the intersection construction (independent of the function ( ), note) and (3.10) the averaging construction. They are represented graphically in Fig. 2.

Further information about the direction in which the nodes move may be obtained from the sign of ($_j$). Solving (3.4) for $_{k1}$ and $_{k2}$ and using Simpson's rule for the integration gives

$$\begin{matrix} _{k1} \\ _{k2} \end{matrix} = \begin{matrix} 4 & 2 \\ 2 & 4 \end{matrix} \begin{matrix} \frac{1}{6}\ (\ _{k1}) + \frac{1}{6}\ (\ _{km}) + \ (\ ^4_k) \\ \frac{1}{3}\ (\ _{km}) + \frac{1}{6}\ (\ _{k2}) + \ (\ ^4_k) \end{matrix}$$

$$= \begin{matrix} \frac{2}{3}\ (\ _{k1}) + \frac{2}{3}\ (\ _{km}) & \frac{1}{3}\ (\ _{k2}) + \ (\ ^4_k) \\ \frac{1}{3}\ (\ _{k1}) + \frac{2}{3}\ (\ _{k2}) + \frac{2}{3}\ (\ _{k2}) + \ (\ ^4_k) \end{matrix} \qquad (3\ 11)$$

where $_{km}$ is the mid-point of the element . Hence

$$\begin{matrix} (\ _j) & _{jL}(\ _j) = \frac{1}{3}\ (\ _{j-1}) & \frac{2}{3}\ (\ _{j-\frac{1}{2}}) + \frac{1}{3}\ (\ _j) + \ (\ ^4_{j-\frac{1}{2}}) \\ (\ _j) & _{jR}(\ _j) = \frac{1}{3}\ (\ _j) & \frac{2}{3}\ (\ _{j+\frac{1}{2}}) + \frac{1}{3}\ (\ _{j+1}) + \ (\ ^4_{j+\frac{1}{2}}) \end{matrix} \qquad (3\ 12)$$

and, from (3.6),

$$(\ _j) = \frac{1}{3}\ ^2\ _{j-\frac{1}{2}} + 0(\ ^4_{j-\frac{1}{2}})\ ^2 \qquad \frac{1}{3}\ ^2\ _{j+\frac{1}{2}} + \ (\ ^4_{j+\frac{1}{2}})\ ^2$$

$$= \frac{1}{9}(\ ^2\ _{j-\frac{1}{2}} + \ ^2\ _{j+\frac{1}{2}})(\ ^2\ _{j-\frac{1}{2}} \qquad ^2\ _{j+\frac{1}{2}}) + \ (\max(\ ^4_{j-\frac{1}{2}} \quad ^4_{j+\frac{1}{2}})) \qquad (3\ 13)$$

For sufficiently small $_{j-\frac{1}{2}}$ $_{j+\frac{1}{2}}$, therefore, the movement of node is governed by the sign of

$$(\ ^2\ _{j-\frac{1}{2}} + \ ^2\ _{j+\frac{1}{2}})(\ ^2\ _{j-\frac{1}{2}} \qquad ^2\ _{j+\frac{1}{2}}) \qquad (3\ 14)$$

moving left if this quantity is positive and right if it is negative.

Since in stage (ii) ( ) is restricted in elements by $= _x$, then in elements and , respectively,

$$(\ )\qquad _L = (\qquad _j)\ _{k-1}$$
$$(\ )\qquad _R = (\qquad _j)\ _k$$

k x k $k-1$ m o v i n g 5

k $k-1$

$^{new}_{j}$ j $\dfrac{_{jR} \qquad _{jL}}{_k \qquad _{k-1}}$

where x_j is the previous approximation and x_j^{new} the one currently sought. Similarly, if $m_k + m_{k-1} \neq 0$, the averaging construction (3.10) gives

$$x_j^{new} - x_j = \frac{2f(x_j^{new}) - (w_{jL} + w_{jR})}{m_k + m_{k-1}}. \tag{3.17}$$

Note that the calculation of x_j^{new} is implicit since $f(x_j^{new})$ occurs on the r.h.s.

Near to inflection points the averaging construction (3.10) may well occur (see Fig. 2b) and the fit obtained by this method will be a (discontinuous) local minimum.

For regions in which $f(x)$ is convex the new approximation to x_j is provided by the displacement (3.16), i.e. the intersections of lines in adjacent elements (see Fig. 2a), since in this case the expression $f(x) - w$ is of the same sign when approached from left or right. The fit is therefore continuous. Where $f(x)$ has an inflection point the intersection construction is replaced by the averaging construction (3.17): this occurs when the $f(x) - w$ are of opposite sign when approached from left or right, as in Fig. 2b. For these exceptional points the fit obtained by this method will be discontinuous. (One possible remedy is to change the number of points locally by one, thus breaking the symmetry, a device which seems to work well.)

In order to simplify the solution of (3.17) it is possible to make use of the outer iteration to move towards the converged x_j by using the x_j^{old} values at the previous step in the calculation of $f(x)$. In the very special case $m_{k-1} = m_k = 0$, equation (3.15) shows that x^{new} is indeterminate and there is no advantage in moving the node at all,

If $f(x)$ is convex we see from (2.12) that the result of the converged iteration (stage (i) — stage (ii) — repeated alternately) is the best continuous L_2 fit using piecewise linear approximation. If $f(x)$ is not convex there may possibly be isolated discontinuities in the fitted function at inflection points, where only a local minimum occurs. It is possible to replace such a discontinuous function locally by a continuous approximation, by say simply averaging the two nodal values (in which case the result is the function value itself). This is of course at the expense of abandoning the optimal fit at these isolated points. The resulting approximation may however be used as an initialisation for other algorithms completely dedicated to continuous best fits, see Loach Wathen [10].

In summary the algorithm is:

1. Set up the initial grid

2. Project elementwise into the space of piecewise linear discontinuous functions on the current grid as in (3.4) (stage (i))

3. Determine the next grid by the intersection construction (3.16) or (exceptionally) the averaging construction (3.17) (stage (ii))

4. If the new grid is too different from the previous grid, go to 2.

The algorithm, which is fast and robust, finds in appropriate cases optimal linear spline approximations with variable knots: indeed, by concentrating on piecewise linear discontinuous fits, the procedure effectively linearises the problem and avoids many of the difficulties generated by restricting the search to continuous fits at the outset.

ach step (i)+(ii) of the algorithm bears a striking resemblance to the Moving Finite lement procedure in the two step form described by Baines [2] and Baines Wathen [1]. The similarity is pursued by Baines [2].

We show results for two examples, in Fig. 3(a),3(b).

(a) tanh 20(0 5)	11 interior nodes
(b) $10^{-10x} + 20$ $1 + 400($ 0 7)^2	9 interior nodes

In each case the fixed interval is [0,1] and the initial grid is equally spaced. xample (a) is a severe front with a single inflection. xample (b) is a test example suggested by Pryce [11].

In each example the trajectories of the nodes (i) are shown as they move towards their final positions together with the function (ii) and the fit obtained (iii). The process is taken to be converged when the $_\infty$ norm of the nodal position updates is less than 10^{-4}. The number of iterations appears on the ordinate axis of the trajectories. In general an extra order of magnitude reduction is obtained in the $_2$ error over the equispaced case.

Although the theory has been derived only for ^{1} functions numerical experiments show that the algorithm also gives optimal fits to functions which are

only piecewise [1]. A simple example shows that the intersection construction drives one node towards an isolated slope discontinuity (. Fig. 2(a)), where it remains while the fits either side converge.

The algorithm also gives piecewise linear best fits to functions which have isolated discontinuities. In this case there are extra jump discontinuity terms in (2.7) arising from the variation of the integral which vanish only when a node is located at a discontinuity itself. In numerical experiments nodes move towards such a point from either side where they remain while once again the fits either side converge. This can be understood in terms of an isolated discontinuity, being

—

2

2

2

1

1 2

2 2

is negative where intersects in element 1 and positive where intersects in element (see Fig. 4). There is therefore at least one root between these points which may be chosen as the new position of $_j$. Moreover, if this root is chosen, all roots (for different) are separated by these intersections and mesh tangling cannot occur.

Further information about the direction in which the nodes move can be obtained from the sign of ( ). Using the trapezium rule for the integration in (4.7) gives

$$k = \frac{1}{2} \quad k1 \quad kL \quad ^2_k$$

$$j \quad jL \quad \tfrac{1}{2} \quad j \quad j{-}1 \quad ^2_{j-\frac{1}{2}}$$
$$j \quad jR \quad \tfrac{1}{2} \quad j \quad j{+}1 \quad ^2_{j+\frac{1}{2}}$$

$$j \quad - \quad j \quad j{-}1 \quad ^2_{j-\frac{1}{2}} \quad ^2$$

$$- \quad j \quad j{+}1 \quad ^2 \quad ^2_{j+\frac{1}{2}} \quad ^2$$

$$- \quad j{-}1 \quad j \quad j{+}1 \quad j{-}1 \quad j{+}1 \quad ^2_{j-\frac{1}{2}} \quad ^2_{j+\frac{1}{2}}$$

$$j{-}\tfrac{1}{2} \quad j{+}\tfrac{1}{2}$$

$$j{-}1 \quad j \quad j{+}1 \quad j{-}1 \quad j{+}1$$

$$- \quad jL \quad jR \quad j$$

$$jL \quad jR$$

$$j$$

to be almost as good, for which convergence proofs can be given, and which are very useful for generalisations to higher dimensions.

These algorithms are based upon using interpolants of the function $f(x)$ at each stage of the iteration, rather than the function itself. The resulting fit is therefore not to $f(x)$ but the interpolant of $f(x)$ at the limit. The degradation is rather small, however, and the algorithms have very positive advantages.

We begin this time with the piecewise constant fits of section 4. Instead of fitting $f(x)$ we shall fit the current linear interpolant $f_I(x)$ (linear in each element) at each stage of the iteration. This means that (4.7) becomes

$$w_k = \frac{1}{2}(f(x_{k1}) + f(x_{k2})) \tag{5.1}$$

and that (4.14) becomes

$$\frac{1}{2}(w_{jL} + w_{jk}) = f_I(x_j). \tag{5.2}$$

Since $f_I(x)$ is linear in each element to the left and right of node j, it is possible to write down the solution of (5.2) for x_j (called here x_j^{new}) which (using (5.1)) is given (see Fig. 4) by

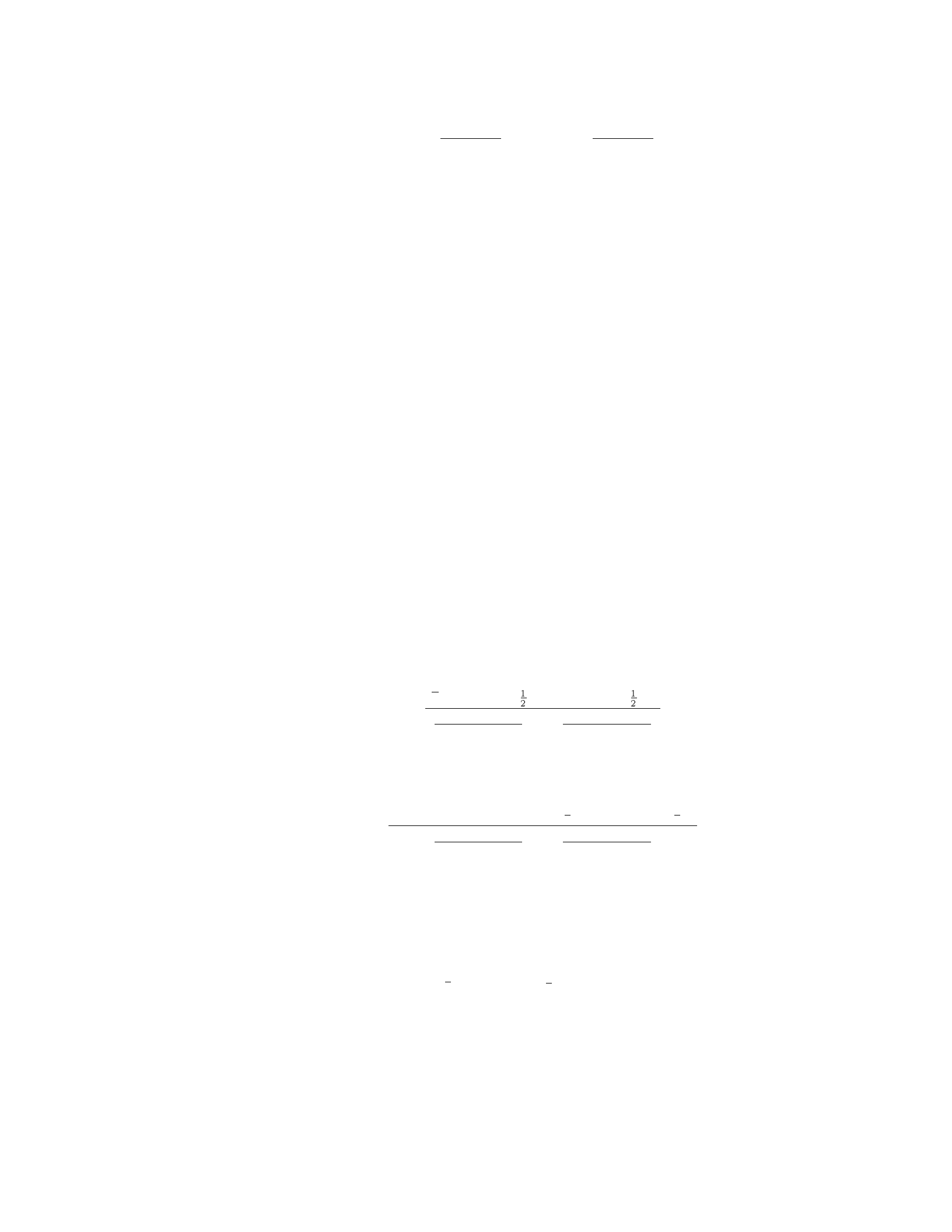
$$x_j^{new} - x_j = \frac{\frac{1}{4}(f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))}{\max\left[\frac{|f(x_{j+1})-f(x_j)|}{\Delta x_{j+\frac{1}{2}}}, \frac{|f(x_j)-f(x_{j-1})|}{\Delta x_{j-\frac{1}{2}}}\right]} \tag{5.3}$$

This simple iteration replaces the two stage iteration of section 4. If it converges, the limit values satisfy

$$f(x_{j+1}^*) - f(x_j^*) = f(x_j^*) - f(x_{j-1}^*) \tag{5.4}$$

and the grid is the one that produces equi-spaced $f(x_i^*)$.

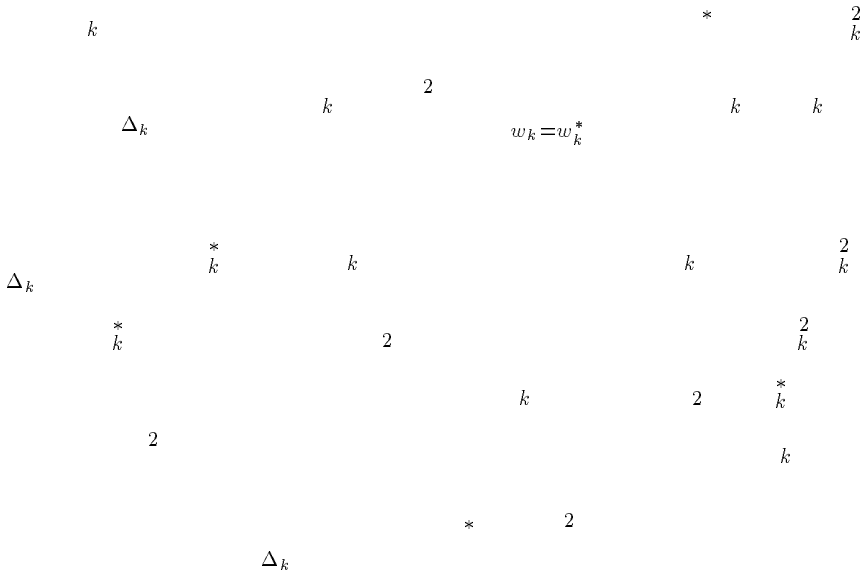
Convergence of the algorithm may be discussed via (4.13). Note that the $O(h_k^2)$ term of (4.10) is now missing so that the node j moves to the left or the right according as whether (4.13) is positive or negative. Thus, except in the vicinity of nodes where (4.13) changes sign, nodal movement is uni-directional. We may exclude the possibility of (4.13) changing sign by assigning fixed nodes to points of maxima, minima and inflection points of $f(x)$. Between these fixed points the



satisfied if (3.14) is not zero. By fixing nodes at the inflection points and at points when $^{(iv)}(\ )$ vanishes, this condition is satisfied and convergence may be proved as before. Finally, we observe that for the quadratic interpolant the solution in -space is equispaced points on the particular quadratic $\hat{}_Q(\ ) = \ _Q(\ (\ ))$ which passes through the points $\ _A \ \frac{1}{2}(\ _A + \ _B) \ \ _B$, between any two points $\ _A$ and $\ _B$ for which the intersection construction is unique, i.e. away from inflection points. Finding the points $\ _j$ still requires the inversion of the functions $\ _Q$ or , however.

The generalisation of these techniques to two dimensions raises a number of difficulties. In principle, the same approach yields algorithms for obtaining best discontinuous fits to given continuous functions on a tessellation of the plane. The solution of the nodal position stage of the algorithm is more difficult, however, and requires additional numerical techniques. Furthermore, there is not the same simple connection in two dimensions between discontinuous linear fits and continuous ones. With these important provisos, however, we describe methods and algorithms which obtain good representation of sharp functions in two dimensions, and generalise to higher dimensions.

Let ( ) be a given ^{1} function of the two variables and in a domain Ω and let $_k(\ \ )$ be a member of the family ^2_k of linear functions on a triangular



k

Choosing $\delta x, \delta y = 0$ and δw to be in the space of piecewise linear discontinuous functions gives for the best discontinuous fit, denoted by w^*, x^* and y^*, the conditions

$$\int_{\Delta_k^*} \{f(x,y) - w^*(x,y)\}\phi_{k1}\, dxdy \;\; = 0 \quad (i = 1,2,3) \tag{6.8}$$

where ϕ_{k1}, ϕ_{k2}, ϕ_{k3} are local linear basis functions in the element k (see Fig. 6b). As other choices, remembering that $\delta x_j, \delta y_j$ must lie in the space of piecewise linear continuous functions, and letting α_j (see Fig. 6a) be the two-dimensional linear finite element basis function at node j, we may set (separately)

$$\left.\begin{array}{ll} & \delta x_j = \alpha_j, \delta y_j = 0, \delta w_j = w_x^* \delta x_j \\ \text{and} & \\ & \delta x_j = 0, \delta y_j = \alpha_j, \delta w_j = w_y^* \delta y_j \end{array}\right\} \tag{6.9}$$

(c.f.(2.9)) in turn in (6.7) to obtain

$$\int_{j\text{-star}} \{f(x,y) - w^*(x,y)\}^2 \alpha_j \underline{\hat{\mathbf{n}}}\, ds = 0 \tag{6.10}$$

for x_j^* and y_j^*, where $\underline{\hat{\mathbf{n}}} = (n_1, n_2)$ and "j-star" indicates the spokes, i.e. the union of the sides of the triangles, passing through the node j at x_j^*, y_j^*, (see Fig. 7).

The simultaneous solution of (6.8) (6.10) gives the required solution $w^*(x^*, y^*)$. Note that (6.10) can be written

$$\sum_{\ell=\ell_1}^{\ell_s} \int_{\ell} \{f(x,y) - w_{\ell A}^*(x,y)\}^2 - \{f(x,y) - w_{\ell C}^*(x,y)\}^2 \;\alpha_j \underline{\mathbf{n}}\, ds = 0 \tag{6.11}$$

where ℓ runs over the spokes ℓ_1 to ℓ_s of j-star, and $w_{\ell A}, w_{\ell C}$ refer to the values of w on the spoke looking anticlockwise and clockwise, respectively. Another useful form is

$$\sum_{\ell=\ell_1}^{\ell_s} (w_{\ell A}(x,y) - w_{\ell C}(x,y))\{(f(x,y) - \frac{1}{2}(w_{\ell A}(x,y) - w_{\ell C}(x,y))\} \;\alpha_j \underline{\mathbf{n}}\, ds = 0 \tag{6.12}$$

An algorithm to find the best discontinuous linear $_2$ fit with variable nodes is constructed in two stages (carried out repeatedly until convergence), corresponding to the choice of variations referred to in section 6 above.

Stage (i)

$$_j = 0 \quad _j = 0 \quad = \quad _{k1} \quad _{k2} \text{ or } \quad _{k3} \tag{7 1}$$

This stage of the algorithm corresponds to the best $_2$ fit amongst discontinuous piecewise linear functions on a prescribed grid, as in (6.1),(6.2), and (6.8) above.

Stage (ii), variations

$$_j = \quad _j \quad _j = 0 \quad _j \quad _x \quad _j = 0 \quad (\quad = 1 \ 2 \quad) \tag{7 2}$$

Stage (ii), variations

$$_j = 0 \quad _j = \quad _j \quad _j \quad _y \quad _j = 0 \quad (\quad = 1 \ 2 \quad) \tag{7 3}$$

Stage (ii), which combines and (or) variations to give variations in "following the motion" in the (or) directions, corresponds to finding $_j$ (or $_i$) such that (6.10) (or (6.11)) holds. Geometrically, we see from (7.2) or (7.3)

ydra(i)i)hstage the constructed

$$k \qquad \overset{2}{k}$$

$$\overset{3}{k} \qquad _{ki} \ _{ki}$$

$$\overset{i=1}{}$$

M $\qquad k$

$$k\text{—}k \quad \text{—}k$$

where $\_\_k = \quad ki \quad \_k = \quad ki \quad ki = \_{\Delta_k} \ (\quad) \ ki \quad$, and

$$= \frac{k}{12} \begin{matrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{matrix} \qquad (7\ 6)$$

where $\ k$ is the area of the triangular element .

The other problems, those of finding $\ j$ satisfying (6.10) with $\ j = \ x \ j$ and $\ j$ satisfying (6.11) with $\ j = \ y \ j$, are more difficult non-linear problems. To make progress we shall hold the $\ j$ in $(\quad)$ constant in solving for the new $\ j$, and embed the associated iteration in the overall iteration, as in the "averaging" construction algorithm of section 3. Similarly for the $\ j$. This device was used in section 3 (equation (3.17)) to obtain converged solutions for $\ ^*_j$, in effect a "lagged" form of the equation being solved as the overall iteration converges.

Let $\ = \ 1 \ , \ e$ denote the elements surrounding the node and let $\ 1 \ 2$ denote the edges of the element emanating from node (see Fig. 7). Then (6.12) may be written

$$\sum_{\ell=\ell_1}^{\ell_s} \ \ell A(\quad) \quad \ell C(\quad)$$

$$(\quad) \quad \frac{1}{2}(\ \ell A(\quad) + \ \ell C(\quad))^{\ 2} \ _{j\_} \quad \ell = 0 \qquad (7\ 7)$$

Since $(\quad)$ is restricted in element by $\ = \ x \ , \quad = 0$, then if $\ jk$ is the value of the fit obtained from stage (i) at node in element , we have in element

$$(\quad) \quad \ jk = \ k(\quad \ j) + \ k(\quad \ j) \qquad (7\ 8)$$

where $\ k = (\ x)k, \ \ k = (\ x)k$, to be substituted into (7.7).

$j \quad j \quad \ell \quad \ell$

$^{(i+1)}_{j} \quad ^{(i)}_{j}$

$^{(i)}_{j} \quad j$

$^{(i+1)}_{j}$

where

$$= \quad {}^{(i+1)}_{j} \qquad j \tag{7 10}$$

$$= \sum_{\ell=\ell_1}^{\ell_s} \left(\quad {}^{2}_{\ell A} \qquad {}^{2}_{\ell C} \right) \quad j \quad \ell \tag{7 11}$$

$$= \sum_{\ell=\ell_1}^{\ell_s} \left(\quad \ell A \qquad \ell C \right) + \left(\quad \ell A \qquad \ell C \right)\left(\quad \ell \qquad j \right) +$$

$$\left(\quad \ell A \qquad \ell C \right)\left(\quad \ell \qquad j \right) \quad k \quad j \quad \ell \tag{7 12}$$

$$= \sum_{\ell=\ell_1}^{\ell_s} \left(\quad \ell \quad \ell \right) \qquad JA \qquad \ell A\left(\quad \ell \qquad j \right) \qquad \ell A\left(\quad \ell \qquad j \right)$$

$$\left(\quad \ell \quad \ell \right) \qquad jC \qquad \ell C\left(\quad \ell \qquad j \right) \qquad \ell C\left(\quad \ell \qquad j \right)^{2} \quad j \quad \ell \tag{7 13}$$

and (provided that ^{2} 4) solved for . The integrals in (7.13) may be evaluated by a quadrature rule. Both Gaussian quadrature and the trapezium rule have been tried. In the latter case (7.13) simplifies considerably with little degradation to the results.

Two real solutions of (7.9) may be regarded in simple situations as analogous to the "intersection" solution and "averaged" solution encountered in the 1-D case discussed in section 3, corresponding to convex or concave parts and inflection points of the function , respectively. In the present two-dimensional case the dimensionality and the many contributions to blur the simple

Since the non-tangling property in one-dimension is no longer guaranteed, there may still be the possibility of nodes being carried across element boundaries, leading to triangles with negative area. In these situations a relaxation parameter is introduced which restricts each node to stay within the surrounding triangles.

ven then there are rare occasions when a triangle area may go negative, in which case a local smoothing can be applied as an emergency measure, and the algorithm continued. These features greatly reduce the effectiveness of the algorithm and prompt the simplified algorithm described in section 9.

The calculation of $y^{(i+1)}$ proceeds in a similar way.

This algorithm gives an approximate optimal discontinuous linear fit on triangles. To obtain a useful continuous piecewise linear approximation we may take an average of the $_{jk}$ values at a given node from each adjacent element to give an approximate nodal value $\overline{\ }_j$, or use the present approximation as a first guess in an algorithm dedicated to finding a continuous best fit.

In summary the algorithm is:

1. Set up the initial grid
2. Project ( ) elementwise into the space of piecewise linear discontinuous functions on the current grid using (7.5) (stage (i))
3. Determine the next grid by solving (7.10) (and its -direction counterpart) with a relaxation factor to prevent tangling (stage (ii))
4. If the new grid is too different from the previous grid or if the $_2$ error is decreasing, go to 2.

Results are shown in Fig. 8(a-c) for three examples, each being a sharp front with a different orientation:

(a) tanh 20($\frac{1}{2}$)

(b) tanh 20(+ 1)

(c) tanh 20(^{2} + ^{2} $\frac{1}{2}$)

all on the unit square with 49 interior grid points. In each case the initial grid is uniform (Fig. 8)

Figure 8(a) shows the grid and profile for example (a) after convergence of the algorithm, while Figures 8(b) and 8(c) show the corresponding results in the case of examples (b) and (c), respectively. Note that the profiles show piecewise continuous linear plots (obtained by averaging at the nodes) whereas the true plots should be piecewise linear discontinuous.

The $_2$ errors are shown in Table 1. rrors from the corresponding piecewise linear continuous function (obtained by averaging nodal values) are shown in brackets.

	Initial error	Final error	No. of steps
(a)	3 77 10^{-3}	2 37 10^{-5}	40
	(2 49 10^{-2})	(5 28 10^{-5})	
(b)	4 06 10^{-3}	5 89 10^{-6}	80
	-2	-5	
	-3	-4	
	-2	-4	

2

x y

k Δ_k

$$_k \quad _{\partial\Delta_k} \quad (\quad) \quad (\quad)^{\,2} (\quad) \hat{} \quad = 0 \qquad (8\,1)$$

With as the characteristic function $_k(\quad)$ on element (Fig. 6c), and taken successively, as in section 4, to be the local "hat" function associated with node we have that the conditions for the best piecewise constant $_2$ fit to ( ), denoted by $^*_k$ $^*_j$ and $^*_j$, are (c.f. (6.8)-(6.10))

$$_{\Delta^*_k} \quad (\quad) \quad ^*_k \quad = 0 \qquad (8\,2)$$

$$_{j-\text{star}} \quad (\quad) \quad \sum_{k=k_1}^{k_e} \; ^*_k \; ^*_k(\quad) \quad _{j\_}^{\,2} \quad = 0 \qquad (8\,3)$$

where -star is as in Fig. 8, $_j$ is as in Fig. 7a, runs over the elements surrounding node and

$$^*(\quad) = \sum_{k=k_1}^{k_e} \; ^*_k \; ^*_k(\quad) \qquad (8\,4)$$

By solving (8.2) and (8.3) simultaneously, we obtain the required fit $^*(\quad)$

This leads to the following algorithm.

Stage (i)

$$_j = \quad _j = 0 \qquad = \; _k \qquad (8\,5)$$

This stage of the algorithm corresponds to the best $_2$ fit amongst piecewise constant functions on a prescribed grid ((8 6))

$$_j \qquad _j \qquad\qquad _j \qquad _j$$

$$_k \qquad \overline{\qquad}_k \quad _k$$

$$\sum_{\ell=\ell_1}^{\ell_s} \qquad _{\ell A} \qquad _{\ell C} \qquad _\ell \quad _\ell \quad - \quad _{\ell A} \qquad _{\ell C} \qquad _j \quad _\ell$$

$\overset{new}{j}$ j

$\overset{new}{j}$ j

ℓ_s

—

ℓ=ℓ_1

1

2

2

2

	Initial error	Final error	No. of steps
(a)	1.8	$1.55 \quad 10^{-3}$	40
(b)	1.8	$8.54 \quad 10^{-4}$	40
(c)	1.84	$2.34 \quad 10^{-3}$	20

Table 2: L_2 errors for piecewise constant best fits.

boundary. Again this cleans up a lot of the noise generated by the special behaviour of the boundary nodes and the resulting pollution as it spreads into the interior.

Now, following section 5, we develop simplified forms of the two-dimensional algorithms in sections 6-8, using the current interpolant during the iterations instead of the function itself.

We begin with the piecewise constant case of section 8. Replacing $f(x, y)$ by its linear interpolant $f_I(x, y)$, (8.9) becomes

$$w_k = \frac{1}{3}(f_{k1} + f_{k2} + f_{k3}) \tag{9.1}$$

where

$$f_{ki} = f(x_{ki}, y_{ki}), \quad i = 1, 2, 3 \tag{9.2}$$

and the ki are the three vertices of the triangle k.

When $f(x, y)$ is replaced by $f_I(x, y)$ in (8.9) or (8.10), the integrand is quadratic in x or y, leading to

$$\sum_{\ell=\ell_1}^{\ell_s} (\quad_{\ell A} \qquad _{\ell C}) \; \frac{2}{3} \; _I(\; ^{new}_{j} \quad _j) + \frac{1}{3} \; (\; _{j\ell} \quad _{j\ell}) \quad \frac{1}{2}(\; _{\ell A} + \quad _{\ell C}) \; (\; _{j\ell} \qquad _j) = 0 \quad (9\,3)$$

$$\sum_{\ell=\ell_1}^{\ell_s} (\quad_{\ell A} \qquad _{\ell C}) \; \frac{2}{3} \; _I(\; _j \quad ^{new}_{j}) + \frac{1}{3} \; (\; _{j\ell} \quad _{j\ell}) \quad \frac{1}{2}(\; _{\ell A} + \quad _{\ell C}) \; (\; _{j\ell} \qquad _j) = 0 \quad (9\,4)$$

$j\ell \quad j\ell$

ℓ_s

$\ell A \quad \ell C \quad I \quad j^{new} \quad j \quad - \quad j\ell A \quad j\ell C \quad j\ell \quad j$

$\ell{=}\ell_1$

ℓ_s

$\ell A \quad \ell C \quad I \quad j \quad j^{new} \quad - \quad j\ell A \quad j\ell C \quad j\ell \quad j$

$\ell{=}\ell_1$

$\ell A, \quad \ell C$

$mA \quad - \quad j \quad \ell A \quad mC \quad - \quad j \quad \ell C$

$j-$ $-$

ℓ_s

$\ell A \quad \ell C \quad I \quad j^{new} \quad j^{new} \quad - \quad mA \quad mC \quad \ell$

$\ell{=}\ell_1$

ℓ

$-$ $j^{new} \quad j^{new}$ $j \quad j$

$-$

$\ell \quad \ell A \quad \ell C \quad \ell$

$^{s} \quad ^{s}$

$-$

$= {}_1 \quad = {}_1$

$-$

$— \quad - \quad —$

$\frac{1}{2}$

by the spokes of j-star. Any positively averaged value will therefore intersect one of the spokes (the one with steepest f) at a point closer to (x_j, y_j) than the mid-point of the spoke, thus ensuring a displacement which cannot cause tangling.

These arguments suggest a modification to (9.10), taking $\underline{\hat{\mathbf{r}}}$ to be in the direction of the maximum slope of f along the spokes of j-star, giving $\theta = \hat{\theta}$, say, and replacing the weights W_ℓ of (9.9) in (9.10) by the positive weights

$$W_\ell^+ = |f_{\ell A} - f_{\ell C}|\, \Delta\ell\, |\sin\hat{\theta}_\ell|. \tag{9.11}$$

This modification gives the correct behaviour in the direction of steepest f but also ensures no tangling in any direction. The point (x_j^{new}, y_j^{new}) is restricted to lie on the spoke of j-star with the greatest slope.

The resulting algorithm is very simple to code and much faster and more robust than the full algorithm of section 8. Also it requires no relaxation parameter or test to see if the grid has tangled. A particularly easy version which simply takes the $W_\ell = 1$ is also viable. Graphs for the three problems of section 8 are shown in Figs. 10 (a)-(c), with the initial grid of Fig. 8, and the corresponding errors are shown in Table 3:

	Initial error	Final error	No. of steps
(a)	1.8	2.37×10^{-3}	15
(b)	1.8	2.36×10^{-3}	15
(c)	1.84	6.00×10^{-3}	15

Table 3: L_2 errors for the algorithm of section 9.

The corresponding approach for piecewise linear fits will use as interpolant a function which must be higher order than linear in any triangle but the precise choice will depend on a balance between simple quadrature and accuracy. For example, a bilinear interpolant or a full quadratic interpolant could be used, the latter being harder to integrate, the former being more subject to singularity. We shall not pursue the analysis here except to note that, by analogy with the 1-D piecewise linear case, it is the intersection construction which will dominate

In particular, in the one-dimensional linear case the fits obtained are optimal piecewise linear continuous fits a.e.

In two dimensions the algorithms are less robust and harder to implement, needing relaxation parameters to prevent mesh tangling. Simplified versions have therefore been developed which avoid mesh tangling and hence the need for these parameters.

We demonstrate in the Appendix the strong connection between piecewise discontinuous fits in one dimension and equidistribution. The extension to three dimensions is straightforward. The main difference in the theory is that in (6.7) the two types of integral are over tetrahedra and their faces. The spokes of -star then become the faces of the triangles which have node as a vertex. A very simple algorithm in 3-D which avoids mesh tangling is then (9.10), the being the edges emanating from node and with the $_\ell$ taken equal to 1.

Apart from the grid generation aspects, this approach is also seen as an in-

[11] Pryce, J.D. (1989). *On the Convergence of Iterated Remeshing*, IMA J. Num. An., **9**, 315-335.

[12] White, A.B. (1979). *On the selection of equidistributing meshes for two-point boundary value problems*, SIAM J. Numer. An., **16**, 473-502.

In this section, following Carey Dinh [6], we derive asymptotic equidistribution results for the linear and constant cases in one dimension, showing the link between equidistribution and approximation by piecewise discontinuous linear and

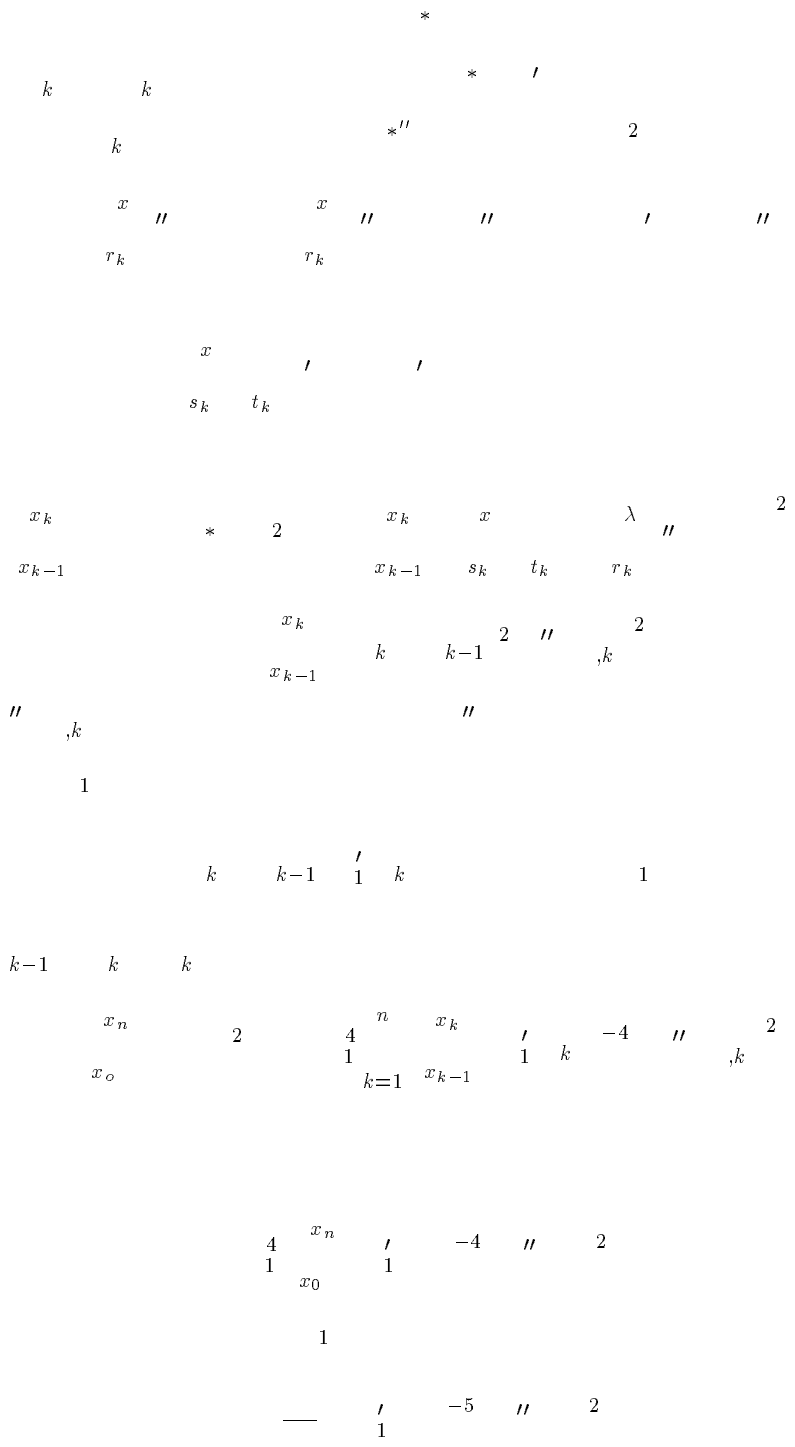


Figure Captions

1. Basis Functions in One Dimension
2. Linear Fits to (a) Convex and (b) Non-convex Functions
3. Results for Piecewise Linear Fits in One Dimensions
   (i) Trajectories (ii) Function (iii) Fit
4. Constant Fits to (a) Monotonic and (b) Non-monotonic Functions
5. Results for Piecewise Constant Fits in One Dimension
   (i) Trajectory (ii) Function (iii) Fit
6. Basis Functions in Two Dimensions
7. Node Connections in Two Dimensions
8. Results for Piecewise Linear Fits in Two Dimensions
9. Results for Piecewise Constant Fits in Two Dimensions
10. Results for the simplified algorithm of section 9 for Piecewise Constant Fits in Two Dimensions.