MATHEMATICS OF COMPUTATION
VOLUME 62, NUMBER 206
APRIL 1994, PAGES 645–669

ALGORITHMS FOR OPTIMAL DISCONTINUOUS PIECEWISE LINEAR AND CONSTANT L_2 FITS TO CONTINUOUS FUNCTIONS WITH ADJUSTABLE NODES IN ONE AND TWO DIMENSIONS

M. J. BAINES

ABSTRACT. In this paper a direct variational approach (with nonstandard variations) is used to generate algorithms to determine optimal *discontinuous* piecewise linear and piecewise constant L_2 fits to a continuous function of one or two variables with adjustable nodes. In the one-variable case the algorithm is fast and robust, the mesh cannot tangle, and the resulting fits are continuous a.e. In the two-variable case, on an adjustable triangular grid, the algorithm is less robust but gives good results for particular functions possessing a single steep feature. The extension to higher dimensions is straightforward.

1. INTRODUCTION

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], Chui et al. [7], Barrow et al. [3], and, more recently, Loach and Wathen [10]. The equidistribution approach is described in White [12], and references therein, Kautsky and Nichols [9], Carey and Dinh [6], and Pryce [11]. A comprehensive up-to-date bibliography is given in Grosse [8].

In this paper we approach the problem of finding optimal L_2 fits to continuous functions with adjustable nodes via piecewise linear *discontinuous* functions. Using a direct variational approach but with nonstandard variations, interpreted numerically, new algorithms are devised, based on a two-stage iteration process whose limit is the required best approximation. In this way we reduce the nonlinearity of the problem (eliminating it altogether for linear fits in one dimension) and obtain algorithms which are fast and robust in comparison with existing methods. Using the same approach, we derive similar algorithms providing best piecewise *constant* L_2 fits with adjustable nodes.

It is known that for continuous functions in one dimension the best piecewise linear fit amongst discontinuous functions with adjustable nodes is continuous

Received by the editor August 17, 1992 and, in revised form, May 24, 1993.
1991 *Mathematics Subject Classification.* Primary 41A30, 65D15.
Key words and phrases. Best fits, adjustable nodes.
The work reported here forms part of the research program of the Oxford/Reading Institute for Computational Fluid Dynamics.

0025-5718/94 $1.00 + $.25 per page

[7]. This result also comes out of the present analysis, except for certain cases where isolated discontinuities can occur. Thus, the piecewise linear algorithm in one dimension generates piecewise linear continuous L_2 fits with adjustable nodes a.e.

In two dimensions the algorithms are based on variable triangulations of the plane, although with invariant connectivity. In this more complex case the algorithms are less robust, and a relaxation procedure is used. Also, owing to the numerical intricacies in the implementation, we have developed simplified forms of the algorithms, which give approximately optimal discontinuous linear and piecewise constant fits with adjustable nodes to a given continuous function on a variable triangulation of the plane.

The algorithms are demonstrated on various test functions. In one dimension, the methods are fast and robust, and give excellent results without any mesh tangling. In two dimensions, owing to the complexity of the problem, only simple functions (with a single severe feature) have been fitted.

The plan of the paper is as follows. In §2 we obtain expeditious natural conditions in one dimension for the L_2 error between a given continuous function and a piecewise linear discontinuous function with adjustable nodes to have an extremum. These conditions are then used in §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 §§2 and 3 are repeated in §4 for the case of piecewise constant functions with adjustable nodes. In §5 similar conditions are obtained in two dimensions for the L_2 error between a given continuous function and a piecewise linear discontinuous function on a variable triangular grid with adjustable nodes to have an extremum. These are used as the basis for a two-dimensional algorithm in §6, which also includes a simplified implementation of the algorithm. Once again, the pattern is repeated for piecewise constant functions in §7. Finally, in §8, the connection between such best fits and equidistribution is studied (in one dimension).

2. Piecewise linear fits in one dimension

Let $f(x)$ be a given C^1 function of a scalar variable x in the interval (x_0, x_{n+1}), and let $u_k(x)$ be any member of the family S_k of linear functions in the interval (x_{k-1}, x_k), where $x_0 \le x_{k-1} < x_k \le x_{n+1}$. Then there exists a unique member $u_k^*(x)$ of S_k such that

$$\delta \int_{x_{k-1}}^{x_k} (f(x) - u_k(x))^2 \, dx \Big|_{u_k = u_k^*} = 0 \tag{2.1}$$

or, equivalently,

$$\int_{x_{k-1}}^{x_k} (f(x) - u_k^*(x)) \delta u_k \, dx = 0 \quad \forall \delta u_k \in S_k . \tag{2.2}$$

The function $u_k^*(x)$ is the best L_2 fit to $f(x)$ from the family S_k.

For the interval (x_0, x_{n+1}), the union of the intervals (x_{k-1}, x_k) $(k = 1, \ldots, n+1)$, the best L_2 fit $u^*(x)$ to $f(x)$ from the family S of piecewise

linear discontinuous functions $u(x)$ with (arbitrary) jumps at $x = x_k$ $(k = 1, \ldots, n)$ satisfies

$$(2.3)\quad \delta \int_{x_0}^{x_{n+1}} \{f(x) - u(x)\}^2 \, dx\big|_{u=u^*} = \delta \sum_{k=1}^{n+1} \int_{x_{k-1}}^{x_k} \{f(x) - u_k(x)\}^2 \, dx\big|_{u_k=u_k^*} = 0$$

and is also given by (2.1) or (2.2) $(k = 1, \ldots, n)$, since $S = \bigoplus S_k$ $(k = 1, \ldots, n)$ and the problem decouples. The solution is $u^*(x) = \bigcup u_k^*(x)$.

Consider now the problem of determining the best L_2 fit to $f(x)$ from the family S_D of piecewise linear discontinuous functions having arbitrary jumps at $x = x_k$ $(k = 1, \ldots, n)$ on a *variable* partition $(x_1, \ldots, x_k, \ldots, x_n)$ of the fixed interval (x_0, x_{n+1}). The solution again satisfies (2.3) but, since the x_k $(k = 1, \ldots, n)$ are to be varied as well as the u_k, the problem does not decouple in an obvious way. However, as we shall see, it is possible to regard it as the limit of a sequence of problems, which include the decoupled problem (2.2).

It is convenient to introduce here a new independent variable ξ, which remains fixed, while x joins u as a dependent variable, both now depending on ξ and denoted by $\hat{x}$ and $\hat{u}$, respectively. Then, with $\hat{u}(\xi) = u(\hat{x}_k(\xi))$, (2.3) becomes (reserving suffixes for interval end points only)

$$(2.4)\quad \delta \sum_{k=1}^{n+1} \int_{x_{k-1}}^{x_k} \{f(\hat{x}(\xi)) - \hat{u}(\xi)\}^2 \frac{d\hat{x}}{d\xi} \, d\xi = 0.$$

Taking the variations of the integral in (2.4) gives

$$(2.5)\quad \int_{x_{k-1}}^{x_k} \Big\{ 2\{f(\hat{x}(\xi)) - \hat{u}(\xi)\}\{f'(\hat{x}(\xi))\delta\hat{x} - \delta\hat{u}(\xi)\} \frac{d\hat{x}}{d\xi} + \{f(\hat{x}(\xi)) - \hat{u}(\xi)\}^2 \frac{d}{d\xi}(\delta\hat{x}) \Big\} \, d\xi.$$

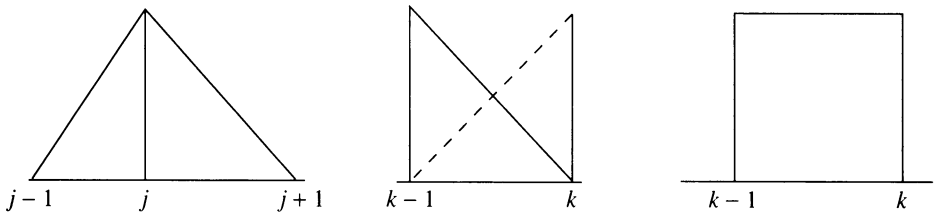
Integrating the last term by parts leads to

$$(2.6)\quad -\int_{x_{k-1}}^{x_k} 2\{f(\hat{x}(\xi)) - \hat{u}(\xi)\}\{f'(\hat{x}(\xi)) \frac{d\hat{x}}{d\xi} - \frac{d\hat{u}}{d\xi}\}\delta\hat{x} \, d\xi - (f(\hat{x}(\xi)) - \hat{u}(\xi))_{k-1}^2 \delta\hat{x}_{k-1} + (f(\hat{x}(\xi)) - \hat{u}(\xi))_k^2 \delta\hat{x}_k.$$

Substituting (2.6) for the last term in (2.5), collecting terms, and returning to the x, u notation, we obtain from (2.4)

$$(2.7)\quad \sum_{k=1}^{n+1} \int_{x_{k-1}}^{x_k} 2\{f(x) - u(x)\}(\delta u - u_x \delta x) \, dx + \sum_{j=1}^{n} [\{f(x) - u(x)\}^2]_j \delta x_j = 0,$$

where the second summation is now over nodes j and the square bracket notation $[\,]_j$ denotes the jump in the relevant quantity at the node j. With the constraint $\delta x = 0$, this leads back to (2.3) and (2.2) and to equations for the best piecewise linear discontinuous L_2 fit to $f(x)$ with fixed nodes.



(a) (b) (c)

FIGURE 1. Basis functions in one dimension

Choosing $\delta x = 0$ and δu to be in the space of piecewise linear discontinuous functions, (2.7) yields the conditions

$$\int_{x_{k-1}^*}^{x_k^*} \{f(x) - u^*(x)\}\phi_{ki}\, dx = 0 \qquad (i = 1, 2) \tag{2.8}$$

for the best fit, denoted by u^* and x^*, where ϕ_{k1}, ϕ_{k2} are the local linear basis functions in element k (see Figure 1b).

Alternatively, remembering that for continuity δx must lie in the space of continuous functions, we may set $\delta x = \alpha_j$ (where α_j is the standard basis function for continuous piecewise linear functions: see Figure 1a), together with the particular constraint

$$\delta u = u_x^* \delta x, \tag{2.9}$$

in (2.7) to obtain

$$[(f(x^*) - u^*(x^*))^2]_j = 0. \tag{2.10}$$

The simultaneous solution of (2.8) and (2.10) gives the required fit $u^*(x^*)$.

With L, R denoting the left and right values at the (variable) node j (see Figure 1a), it follows from (2.10) that if u_L, u_R lie on the same side of $f(x_j^*)$ (cf. Figure 2(a)),

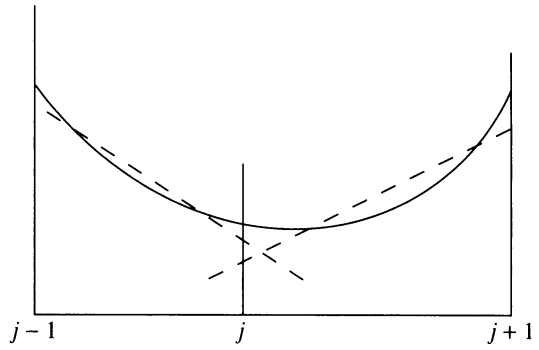
$$f(x_j^*) - u_L^* = f(x_j^*) - u_R^* \Rightarrow u_L^* = u_R^* \tag{2.11}$$

(irrespective of $f(x)$ as long as it is continuous) and therefore that u^* is continuous at the new position of the node. On the other hand, if u_L, u_R lie on opposite sides of $f(x_j^*)$ (cf. Figure 2(b)), then

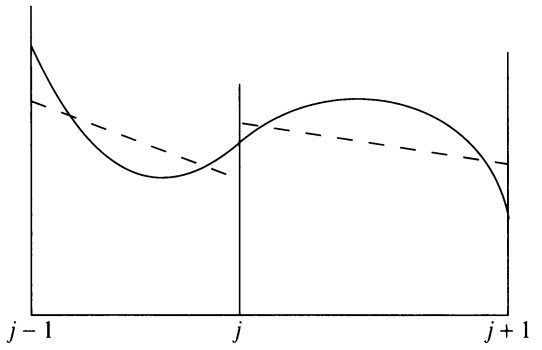
$$-(f(x_j^*) - u_L^*) = f(x_j^*) - u_R^* \Rightarrow u_L^* + u_R^* = 2f(x_j^*), \tag{2.12}$$

in which case u^* is discontinuous and its jump is bisected by $f(x_j^*)$.

Now it is known [7, 10] that for continuous functions with variable nodes, $f(x)$, the best L_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 u^* at x_j^*, therefore, cannot correspond to the best least squares fit when $f(x)$ is continuous, and must correspond to only a local minimum.



(a)



(b)

FIGURE 2. Linear fits to (a) convex and (b) nonconvex functions

3. AN ALGORITHM FOR VARIABLE NODE DISCONTINUOUS PIECEWISE LINEAR FITS

An algorithm to find optimal piecewise linear L_2 fits with variable nodes is now constructed in two stages (carried out alternately until convergence is obtained), corresponding to the particular choices of variations referred to in §2 above.

Stage (i).

$$(3.1) \qquad \delta x_j = 0 \quad (j = 1, \ldots, n), \qquad \delta u = \phi_{k1} \text{ or } \phi_{k2} \quad (k = 1, \ldots, n+1).$$

This stage of the algorithm is governed by (2.8) and corresponds to the best L_2 fit $u(x)$ amongst the family S_D of linear functions discontinuous at *prescribed* nodes, as in (2.1), (2.2).

Stage (ii).

$$(3.2) \qquad \delta x_j = \alpha_j, \qquad \delta u_j - u_x \delta x_j = 0 \qquad (j = 1, \ldots, n).$$

This stage, which combines both u- and x-variations to give variations in u "following the motion", corresponds to finding x_j such that (2.10) holds. Geometrically, we see from (2.9) that variations of x, u are restricted to points lying on the lines of the current piecewise linear discontinuous approximation (possibly linearly extrapolated).

The algorithm is analogous to minimizing a quadratic function $q(x, y)$ using two search directions $d1$ and $d2$ spanning the plane. Starting from some initial guess, we may alternately minimize q in the directions $d1$ and $d2$. Similarly, in the present case, to find the best L_2 fit, we may begin with an initial guess $\{x_j\}$, $\{u_j\}$; stage (i) is to find the minimum in the linear manifold specified by the variations given in (3.1) and so solve (2.8) for new, generally discontinuous, values w_1, w_2 of u at the point j with the x_j fixed; stage (ii) is to find the minimum in the linear manifold specified by the variations given in (3.2) and so solve (2.10) approximately for new $\{x_j\}$ by the implementation of (2.11), (2.12), as discussed below. Repetition of these stages gives a sequence which, if convergent, provides a solution of (2.4) or (2.7). As with similar problems of this type, the limit may correspond only to a local minimum.

Since in stage (ii), $u(x)$ is restricted in element k by $\delta u = u_x \delta x$, then, if it passes through the point (x_j, w_L), say, we have

$$u(x) - w_L = (x - x_j)(u_x)_k . \tag{3.3}$$

Hence, (2.10) becomes

$$\{f(x^*) - w_L - (x^* - x_j)(u_x)_k\}^2 - \{f(x^*) - w_R - (x^* - x_j)(u_x)_{k-1}\}^2 = 0, \tag{3.4}$$

where w_L and w_R are the values of the current stage-(i) approximations to the left and right of node j. In the case corresponding to (2.11) we deduce that, if $(u_x)_k \neq (u_x)_{k-1}$,

$$x_j^{\text{new}} - x_j^{\text{old}} = \frac{w_R - w_L}{(u_x)_k - (u_x)_{k-1}} . \tag{3.5}$$

Here, x_j^{old} is the previous approximation and x_j^{new} the one currently sought. Call this the intersection construction (see Figure 2(a)). Similarly, in the case corresponding to (2.12) we have

$$w_L + w_R + (x_j^{\text{new}} - x_j^{\text{old}})\{(u_x)_{k-1} + (u_x)_k\} = 2f(x_j^{\text{new}}), \tag{3.6}$$

giving, if $(u_x)_k + (u_x)_{k-1} \neq 0$,

$$x_j^{\text{new}} - x_j^{\text{old}} = \frac{2f(x_j^{\text{new}}) - (w_L + w_R)}{(u_x)_k + (u_x)_{k-1}} . \tag{3.7}$$

Call this the averaging construction (see Figure 2(b)).

Observe that the expression in (3.4) is negative at the nearest point (to x_j) that u intersects f in element $k-1$, and is positive at the nearest point (to x_j) that u intersects f in element k (see Figure 2). There is therefore a root of (3.4) between these points. Moreover, if this root is chosen, all roots (for different k) lie between such pairs of intersection points, and therefore mesh tangling cannot occur.

Note that near to inflection points the averaging construction (3.7) may occur and the "untangled" limit will be only a local minimum. (One possible way round this difficulty is to change the number of points locally by one, thus breaking the symmetry. Numerical experience indicates that this does indeed avoid the problem.)

For regions in which $f(x)$ is convex the new approximation to x_j is provided by the displacement (3.5), i.e., the intersections of lines in adjacent elements (see Figure 2a), since in this case the curly brackets in (3.4) are of the same

sign when approached from left or right. Where $f(x)$ has an inflection point the intersection construction is replaced by the averaging construction (3.7): this occurs when values of the curly brackets in (3.4) are of opposite sign when approached from left or right, as in Figure 2b.

Note that the calculation of x_j^{new} from (3.7) is implicit, since $f(x_j^{\text{new}})$ occurs on the right-hand side. In order to simplify the solution of (3.7), we may make use of the outer iteration in this case to move towards the converged x_j by using the x_j^{old}-values at the previous step. In the special case $(u_x)_{k-1} = (u_x)_k = 0$, equations (3.4), (3.6) show that x_j^{new} is indeterminate and there is no advantage in moving the node at all.

If $f(x)$ is convex, we see from (2.11) 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 initialization for other algorithms dedicated to *continuous* best fits [10].

In summary the algorithm is:

1. set up the initial grid;
2. project f elementwise into the space of piecewise linear discontinuous functions on the current grid (stage (i));
3. determine the next grid by the intersection construction (3.5) or (exceptionally) the averaging construction (3.7) (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 linearizes the problem and avoids many of the difficulties generated by restricting the search to continuous fits at the outset. Further details are given in Baines [1].

One step of the algorithm bears a striking resemblance to the Moving Finite Element procedure in the two-step form described by Baines and Wathen [2]. The connection is described more fully in a future publication.

Results are shown for two examples, in Figures 3(a), 3(b) (see next page):

(a) $\tanh\{20(x-0.5)\}$, 11 interior nodes;

(b) $10e^{-10x} + 20/\{1 + 400(x-0.7)^2\}$, 9 interior nodes.

In each case the fixed interval is $[0, 1]$ and the initial grid is equally spaced. Example (a) is a severe front with a single inflection. Example (b) is a test example posed 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 have converged when the l_∞ 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 L_2 error over the equispaced case.

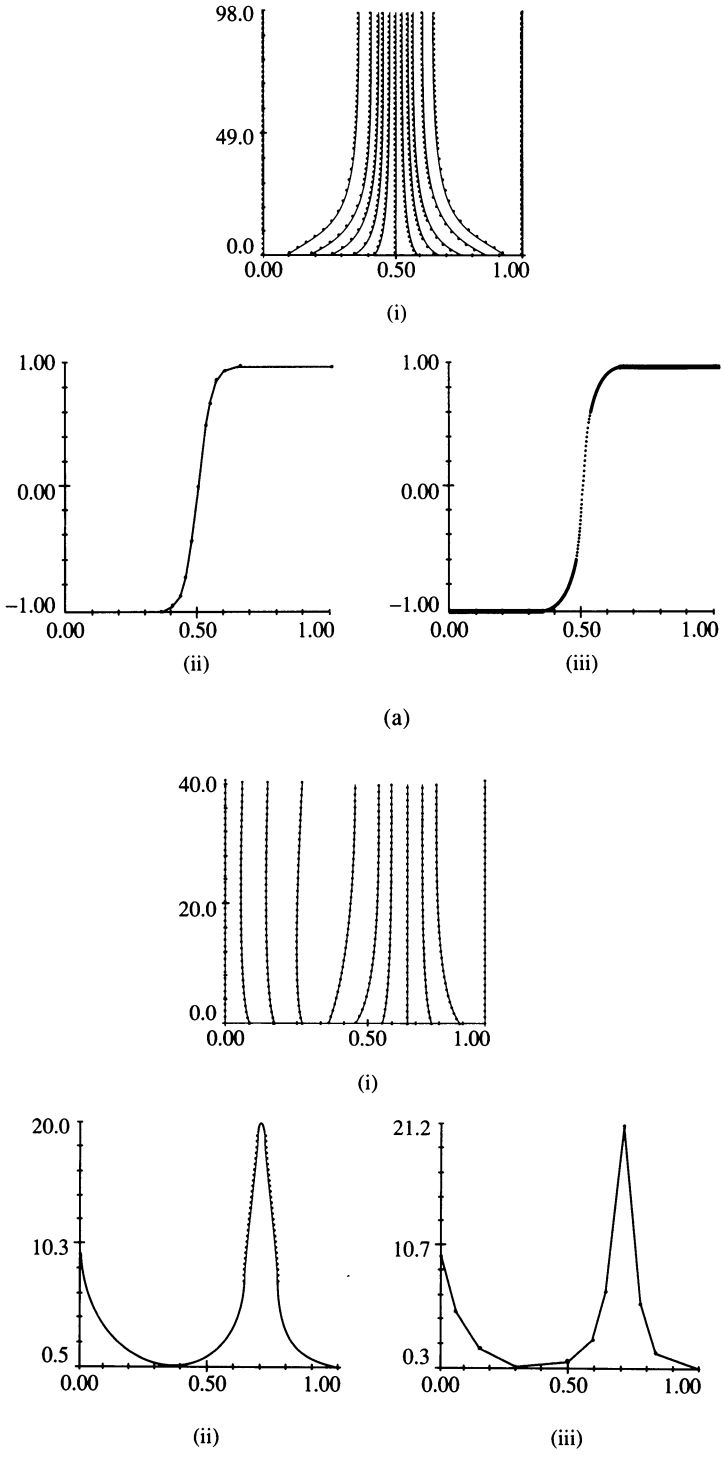


FIGURE 3. Results for piecewise linear fits in one dimension (i) trajectories, (ii) function, (iii) fit

Although the theory has been derived only for C^1 functions, numerical experiments show that the algorithm also gives optimal fits to functions which are only piecewise C^1. A simple example shows that the intersection construction drives the nearest node towards the slope discontinuity (cf. Figure 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 the nearest node moves towards such a point, where it remains, while once again the fits either side converge.

A simplified algorithm may be obtained by fitting instead of $f(x)$ its current quadratic interpolant in each element, using the value at the midpoint of the element as the third matched value. The resulting algorithm avoids quadrature and gives a simple formula for the iteration grid generator. For further details see Baines [1].

4. Piecewise constant fits and adjustable nodes in one dimension

The approach is readily adapted for best piecewise constant fits with variable nodes. In this case the conditions for the best fit, denote by u^*, and the grid, denoted by x^*, are

$$\int_{x_{k-1}}^{x_k} \{f(x) - u^*(x)\}\pi_k(x)\,dx = 0 \tag{4.1}$$

(cf. (2.8)), where $\pi_k(x)$ is the characteristic function in the element k (see Figure 1c), and

$$[(f(x^*) - u^*)^2]_j = 0 \tag{4.2}$$

(cf. (2.10)).

With L, R denoting values to the left and right of the (variable) node j, it follows from (4.2) that, as in §2, if u_L, u_R lie on the same side of $f(x_j^*)$,

$$f(x_j^*) - u_L^* = f(x_j^*) - u_R^* \Rightarrow u_L^* = u_R^* \tag{4.3}$$

or, if u_L, u_R lie on opposite sides of $f(x_j^*)$,

$$-(f(x_j^*) - u_L^*) = f(x_j^*) - u_R^* \Rightarrow u_L^* + u_R^* = 2f(x_j^*). \tag{4.4}$$

It is easy to see that the latter corresponds to monotonic behavior of f while, although the former may exceptionally occur at maxima or minima, it gives no information about the position of x_j^* (see Figures 4a, 4b).

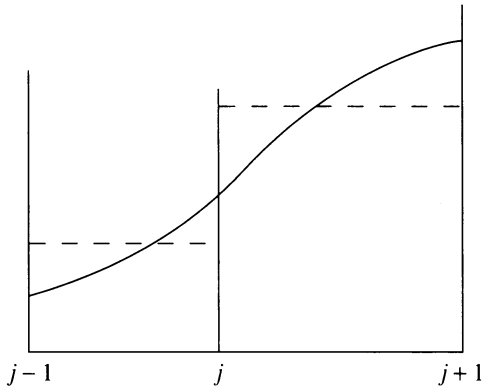
The solution is therefore the set of best constant fits in separate elements which have the averaging property (4.4).

The corresponding algorithm to find the best piecewise constant L_2 fit with variable nodes is again constructed in two stages (carried out alternately until convergence), corresponding to the above.

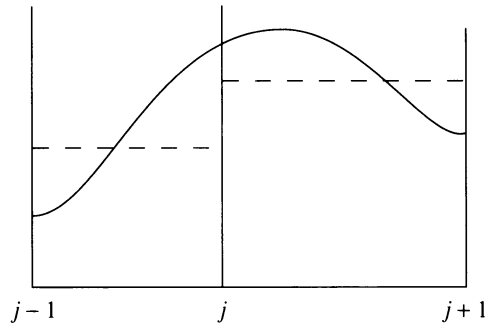
Stage (i).

$$\delta x_j = 0 \quad (j = 1, \ldots, n), \qquad \delta u = \pi_k \quad (k = 1, 2, \ldots, n+1). \tag{4.5}$$

This stage of the algorithm is governed by (4.1) and corresponds to the best L_2 fit amongst the family Π_D of piecewise constant functions on a fixed grid.



(a)



(b)

FIGURE 4. Constant fits to (a) monotonic and (b) nonmonotonic functions

Stage (ii).

$$\delta x_j = \alpha_j, \quad \delta u = 0 \quad (j = 1, 2, \ldots, n). \tag{4.6}$$

This stage corresponds to finding x_j such that (4.2) holds, with variations of u restricted to points lying on the current piecewise constant discontinuous approximation in element k (possibly extrapolated).

As in §2, the expression in (4.2) is negative where u intersects f in element $k-1$, and positive where u intersects f in element k (see Figure 4). There is therefore a root between these points. Moreover, if this root is chosen, all roots (for different k) are separated by these intersections and mesh tangling cannot occur.

Since, in stage (ii), $u(x)$ is restricted in element k by $\delta u = 0$, then $u(x)$ is equal to the value of the current stage-(i) approximation within the whole element. Hence, (4.4) becomes

$$w_L + w_R = 2f(x_j) \tag{4.7}$$

cf. (3.6), where w_L and w_R are the values of the current stage-(i) approximation to the left and right of node j. Any standard algorithm may be used to extract x_j: here we have used a bisection method.

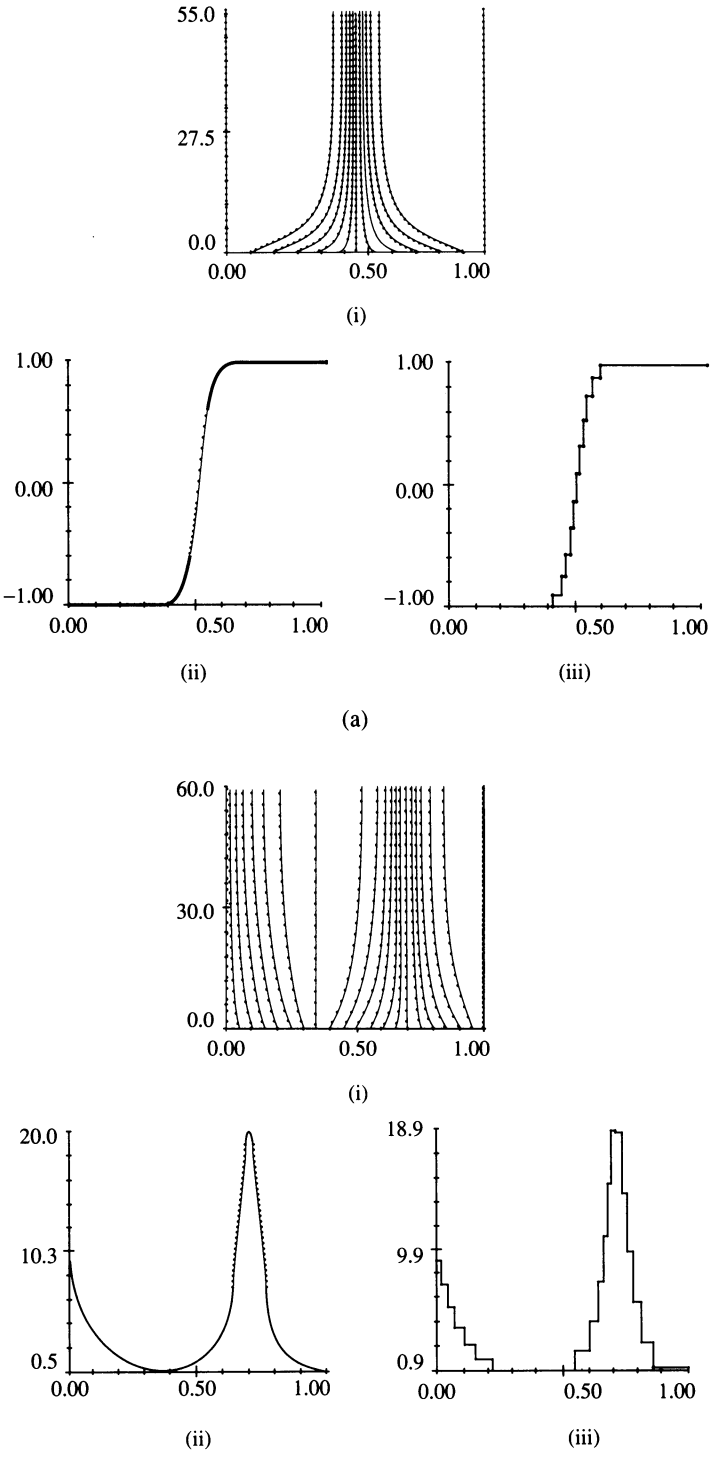


FIGURE 5. Results for piecewise constant fits in one dimension (i) trajectory, (ii) function, (iii) fit

In the case of (4.3) there is no solution for x_j unless $u_L = u_R$. In this exceptional case any x_j in the element is a solution, and there is therefore no reason to adjust the node position at the current iteration.

In summary the algorithm is:

1. set up an initial grid;
2. project f elementwise into the space of piecewise constant functions on the current grid (stage (i));
3. determine the new grid by the averaging construction (4.9) (stage (ii));
4. if the new grid is too different from the previous grid, go to 2.

Results are given for the same test examples as in §3, shown in Figures 5(a), 5(b), except that for better resolution example (b) is done with 19 interior nodes:

(a) $\tanh\{20(x-0.5)\}$, 11 interior nodes;

(b) $10e^{-10x} + 20/\{1 + 400(x-0.7)^2\}$, 19 interior nodes.

In both cases the interval is $[0, 1]$ and the initial grid is again equally spaced. 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 have converged when the l_∞ norm of the nodal position updates is less than 10^{-4}. The number of iterations appears on the ordinate axis of the trajectories. An order-of-magnitude reduction in the L_2 error over the fixed node case is obtained.

Again, numerical experiments indicate that the algorithm also gives best piecewise constant fits to C^0 functions which are only piecewise continuous, with a node moving towards a discontinuity, and remaining there, while the rest of the fit converges.

A simplified algorithm, avoiding quadrature, may be obtained by fitting instead of $f(x)$ its current linear interpolant in each element, giving a very simple formula for the iterative grid generator. For details see Baines [1].

5. Piecewise linear fits in two dimensions

The generalization 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 tesselation 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 a method and an algorithm which is at least partially successful in that it obtains good representation of sharp functions in two dimensions, and generalizes to higher dimensions.

Let $f(x, y)$ be a given C^1 function of the two variables x and y in a domain Ω, and let $u_k(x, y)$ be a member of the family S_k^2 of linear functions on a triangular subdomain Δ_k of Ω. Then there exists a unique member $u^*(x, y)$ of S_k^2 such that

$$\delta \int_{\Delta_k} \{f(x, y) - u_k(x, y)\}^2 \, dx \, dy \big|_{u_k = u_k^*} = 0, \qquad u_k \in S_k, \tag{5.1}$$

or, equivalently,

$$\int_{\Delta_k} \{f(x, y) - u_k^*(x, y)\}\delta u_k(x, y)\, dx\, dy = 0 \quad \forall \delta u_k(x, y) \in S_k^2. \tag{5.2}$$

The function $u_k^*(x, y)$ is the best L_2 fit to $f(x, y)$ from the family S_k^2.

For the region Ω, the union of triangles Δ_k, the best L_2 fit $u_k^*(x, y)$ to $f(x, y)$ from the family S^2 of piecewise linear *discontinuous* functions $u_k(x, y)$ satisfies

$$\begin{aligned} &\delta \int_{\Delta} \{f(x, y) - u^*(x, y)\}^2\, dx\, dy \\ &\quad = \delta \sum_k \int_{\Delta_k} \{f(x, y) - u_k^*(x, y)\}^2\, dx\, dy = 0 \end{aligned} \tag{5.3}$$

and is also given by (5.1) or (5.2), since $S^2 = \bigoplus S_k^2$ and the problem decouples. The solution is $u^*(x, y) = \bigcup u_k(x, y)$.

Now consider the problem of determining the best L_2 fit to $f(x, y)$ from the family S_D^2 of piecewise linear discontinuous functions on a variable triangulation $\bigcup_k \Delta_k$ of the fixed domain Ω, where the internal vertices of the Δ_k are varied.

It is again convenient to introduce new independent variables ξ, η, which remain fixed, while x and y join u as dependent variables, all three now depending on ξ and η and being denoted by $\hat{x}$, $\hat{y}$, and $\hat{u}$, respectively. Then, with $\hat{u}(\xi, \eta) = u(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta))$, (5.3) becomes

$$\delta \sum_k \int_{\Delta_k} \{f(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta)) - \hat{u}(\xi, \eta)\}^2 J\, d\xi\, d\eta = 0, \tag{5.4}$$

where $J = \frac{\partial(x, y)}{\partial(\xi, \eta)}$ is the Jacobian of the transformation.

Taking the variations of the integral in (5.4) gives

$$\begin{aligned} &\int_{\Delta_k} \{2\{f(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta)) - \hat{u}(\xi, \eta)\} \\ &\quad \cdot \{f_x(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta))\delta\hat{x}(\xi, \eta) \\ &\qquad + f_y(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta))\delta\hat{y}(\xi, \eta) - \delta\hat{u}(\xi, \eta)\}J \\ &\qquad + \{f(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta)) - \hat{u}(\xi, \eta)\}^2 \delta J\}\, d\xi\, d\eta. \end{aligned} \tag{5.5}$$

Integrating the last term by parts leads to

$$\begin{aligned} &-\int_{\Delta_k} 2\{f(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta)) - \hat{u}(\xi, \eta)\} \\ &\quad \cdot \{\nabla_{(x,y)} f(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta))J - \nabla_{(\xi,\eta)}\hat{u}\}\, \delta\hat{x}\, d\xi\, d\eta \\ &\qquad + \int_{\partial\Delta_k} \{f(\hat{x}(\xi, \eta), \hat{y}(\xi, \eta)) - \hat{u}(\xi, \eta)\}^2 (\delta x, \delta y) \cdot \hat{\mathbf{n}}\, ds, \end{aligned} \tag{5.6}$$

where $\hat{\mathbf{n}}$ is the outward drawn normal to an element ds of the boundary $\partial\Delta_k$ of Δ_k.

Substituting (5.6) into (5.5), collecting terms, and returning to the x, y, u notation, we obtain from (5.4)

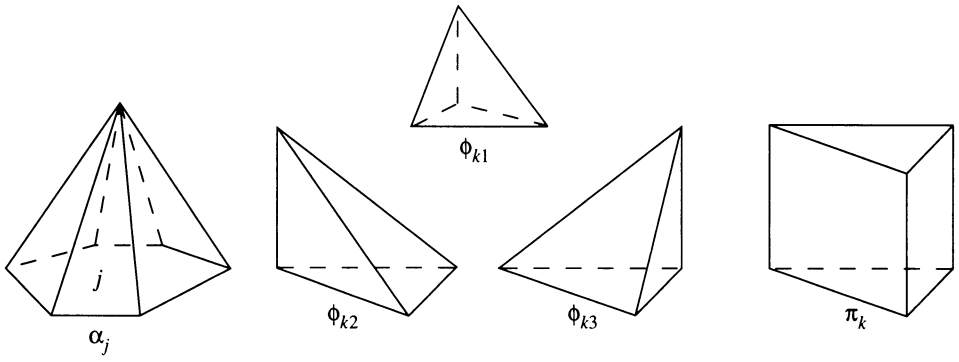


FIGURE 6. Basis functions in two dimensions

$$\sum_k \int_{\Delta_k} 2\{f(x,y)-u(x,y)\}\{\delta u - u_x\delta x - u_y\delta y\}\,dx\,dy + \sum_k \int_{\partial\Delta_k} \{f(x,y)-u(x,y)\}^2(\delta x,\delta y)\cdot\hat{\mathbf{n}}\,ds = 0. \tag{5.7}$$

With the constraints $\delta x,\ \delta y = 0$, this leads back to (5.3) and (5.2) and to equations for the best piecewise linear discontinuous L_2 fit to $f(x,y)$ with fixed nodes.

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

$$\int_{\Delta_k^*} \{f(x,y)-u^*(x,y)\}\phi_{ki}\,dx\,dy = 0 \qquad (i=1,2,3), \tag{5.8}$$

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

$$\delta x_j = \alpha_j, \quad \delta y_j = 0, \quad \delta u_j = u_x\delta x_j \quad \text{and} \quad \delta x_j = 0, \quad \delta y_j = \alpha_j, \quad \delta u_j = u_y = \delta y_j \tag{5.9}$$

(cf. (2.9)) in turn in (5.7) to obtain

$$\int_{j\text{-star}} (f(x,y)-u^*(x^*,y^*))^2\alpha_j n_1\,ds = 0 \tag{5.10}$$

for x_j^*, and

$$\int_{j\text{-star}} (f(x,y)-u^*(x^*,y^*))^2\alpha_j n_2\,ds = 0 \tag{5.11}$$

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

The simultaneous solution of (5.8) and (5.10)–(5.11) gives the required best fit $u^*(x^*,y^*)$.

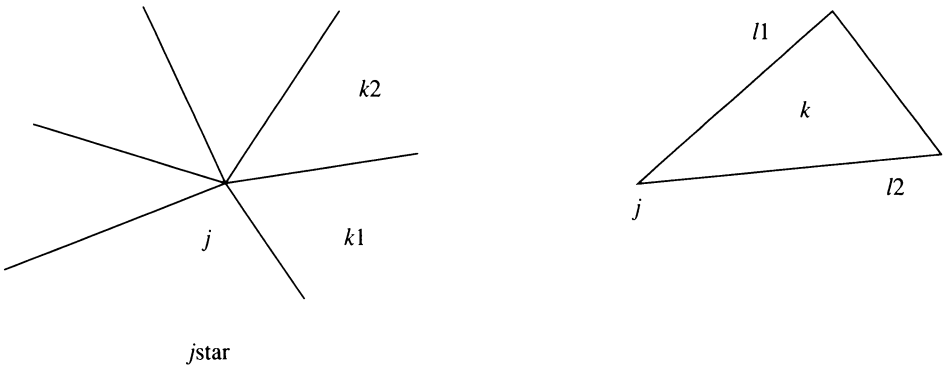


FIGURE 7. Node connections in two dimensions

6. AN ALGORITHM FOR VARIABLE NODE DISCONTINUOUS PIECEWISE LINEAR FITS ON A VARIABLE TRIANGULATION IN TWO DIMENSIONS

The algorithm used to find best discontinuous linear L_2 fit with variable nodes is constructed in two stages (carried out repeatedly until convergence), corresponding to the choice of variations referred to in §5 above.

Stage (i).

$$\delta x_j = 0, \quad \delta y_j = 0, \quad \delta u = \phi_{k1}, \ \phi_{k2}, \text{ or } \phi_{k3}. \tag{6.1}$$

This stage of the algorithm corresponds to the best L_2 fit amongst discontinuous piecewise linear functions on a prescribed grid, as in (5.1), (5.2), and (5.8) above.

Stage (ii). x variations

$$\delta x_j = \alpha_j, \quad \delta y_j = 0, \quad \delta u_j - u_x \delta x_j = 0 \qquad (j = 1, 2, \ldots, n). \tag{6.2}$$

Stage (ii). y variations

$$\delta x_j = 0, \quad \delta y_j = \alpha_j, \quad \delta u_j - u_y \delta y_j = 0 \qquad (j = 1, 2, \ldots, n). \tag{6.3}$$

Stage (ii), which combines u- and x- (or y-) variations to give variations in u "following the motion" in the x- (or y-) directions, corresponds to finding x_j (or y_j) such that (5.10) (or (5.11)) holds. Geometrically, we see from (6.2) or (6.3) that variations of x, u (or (y, u)) are restricted to points lying on certain planes constructed from the stage-(i) solution (possibly extrapolated) in each of the elements k surrounding j.

The problem of finding $u_k(x, y)$, belonging to S_k^2, which satisfies (5.8) is standard. Setting

$$u_k(x, y) = \sum_{i=1}^{3} w_{ki} \phi_{ki}(x, y) \tag{6.4}$$

in element k, where i ranges over the corners of Δ_k, we substitute into (5.8) and find that

$$C_k \mathbf{w}_k = \mathbf{b}_k, \tag{6.5}$$

where $\mathbf{w}_k = \{u_{ki}\}$, $\mathbf{b}_k = \{b_{ki}\}$, $b_{ki} = \int_{\Delta_k} f(x, y) \phi_{ki} \, dx \, dy$, and

$$C = \frac{A_k}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \tag{6.6}$$

with A_k being the area of the triangular element k.

The other problems, those of finding x_j satisfying (5.10) with $\delta u_j = u_x \delta x_j$ and y_j satisfying (5.11) with $\delta u_j = u_y \delta y_j$, are more difficult nonlinear problems. To make progress, we solve them approximately, with the following simplifications:

(a) replace the line integrals in (5.10) and (5.11) by a simple quadrature rule;
(b) hold the x_j in $f(x, y)$ constant in solving for the new x_j, and embed the necessary iteration in the overall iteration, as in the "averaging" construction algorithm of §3; similarly for the y_j.

The device (b) is as used in §3 (equation (3.7)) to obtain converged solutions for x_j^*, in effect a "lagged" form of the equation being solved as the overall iteration converges.

Let $k = k_1, \ldots, k_e$ denote the elements surrounding the node j, and let l_1, l_2 denote the edges of the element k emanating from node j (see Figure 7). Then (5.10) may be written

$$\sum_{k=k_1}^{k_e} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} \{f(x, y) - u(x, y)\}^2 (-\sin\theta_l) \alpha_j \, ds_l = 0, \tag{6.7}$$

where θ_l is the angle between the edge l and the x-axis, so that $n_1 = -\sin\theta_l$. Since $u(x, y)$ is restricted in element k by $\delta u = u_x \delta x$, $\delta y = 0$, then, if it passes through the point (x_j, y_j, w_{jk}), say, where w_{jk} is the value of the fit obtained from stage (i) at node j in element k, we have

$$u(x, y) - w_{jk} = (x - x_j)(u_x)_k + (y - y_j)(u_y)_k, \tag{6.8}$$

so that, writing $\sin\theta_l \, ds_l = \delta y_l$, the integral (6.7) becomes

$$\sum_{k=k_1}^{k_e} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} \{f(x_l, y_l) - w_{jk} - (x_l - x_j)(u_x)_k - (y_l - y_j)(u_y)_k\}^2 \phi_l \, dy_l = 0, \tag{6.9}$$

where ϕ_l is a linear basis function along the side l (the restriction of α_j to the edge l, with the value 1 at j and 0 at the other end of the line), to be solved for x_j. This is a highly nonlinear equation, bearing in mind the dependence of the range of integration on the unknown x_j, but we may reduce it to a quadratic as follows.

As in §3, we introduce an iteration (to be run in tandem with the main iteration) in which we solve (6.9) for $x_j^{(i+1)}$ in terms of $x_j^{(i)}$, where f and u_x are evaluated at $x_j^{(i)}$ while x_j and the range of integration are evaluated at $x_j^{(i+1)}$.

Equation (6.9) can then be written

$$AX^2 - BX + C = 0, \tag{6.10}$$

where

$$X = x_j^{(i+1)} - x_j, \tag{6.11}$$

(6.12)

$$A = \sum_{k=k_1}^{k_2} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} (u_x)_k^2 \phi_l \, dy_l ,$$

$$B = \sum_{k=k_1}^{k_2} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} \{f(x_l, y_l) - w_{jk} - (x_l - x_j)(u_x)_k - (y_l - y_j)(u_y)_k\} \phi_l \, dy_l ,$$

$$C = \sum_{k=k_1}^{k_2} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} \{f(x_l, y_l) - w_{jk} - (x_l - x_j)(u_x)_k - (y_l - y_j)(u_y)_k\}^2 \phi_l \, dy_l ,$$

and (provided that $B^2 > 4AC$) solved for X. The integrals in (6.12) may be evaluated by a quadrature rule. Both Gaussian quadrature and the trapezium rule have been tried. In the latter case (6.12) simplifies considerably.

Two real solutions of (6.10) may be regarded in simple situations as analogous to the "intersection" solution and "averaged" solution encountered in the 1-D case discussed in §3, corresponding to convex/concave parts and inflection points of the function f, respectively. In the present two-dimensional case, the dimensionality and the several contributions to A, B, C blur the simple 1-D interpretation, but for consistency we choose the root corresponding to least movement. If $B^2 = 4AC$ in (6.10), the roots coalesce, while if $B^2 < 4AC$, imaginary roots occur. In the latter case we go for the "nearest" real solution, which is the equal-roots case.

Numerical difficulties arise when A, B, and/or C become very small, which may be due to nearly plane patches in f or simply closeness to the best fit. A threshold parameter is therefore introduced which protects the roots from the resultant singularities. If $|A|$, $|B|$, or $|C|$ fall short of the threshold parameter, special solutions are taken. In particular, note that if $|C|$ is small, we are already close to convergence.

Since the nontangling property in one dimension is no longer guaranteed, there may still be the possiblity 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. Even 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.

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 w_{jk}-values at a given node j from each adjacent element k to give an approximate nodal value $\overline{w}_j$, 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 $f(x, y)$ elementwise into the space of piecewise linear discontinuous functions on the current grid (stage (i));

3. determine the next grid by solving (6.10) (and its y-direction counterpart) with a relaxation factor to prevent tangling (stage (ii));

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

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

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

(b) $\tanh 20(x + y - 1)$,

(c) $\tanh 20(x^2 + y^2 - \frac{1}{2})$,

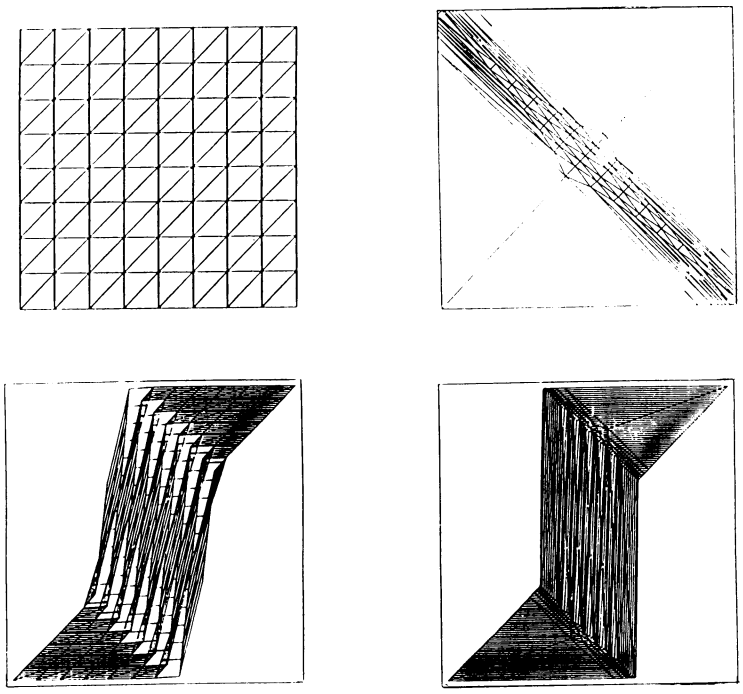
all on the unit square with 49 interior grid points. In each case the initial grid is uniform (Figure 8).

Figure 8(a) shows the grid and profile for example (a) before and 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.

Table 1 (see p. 664) gives a listing of L_2 errors. Errors from a corresponding piecewise linear *continuous* function (obtained by simple averaging of the nodal values) are shown in parentheses.

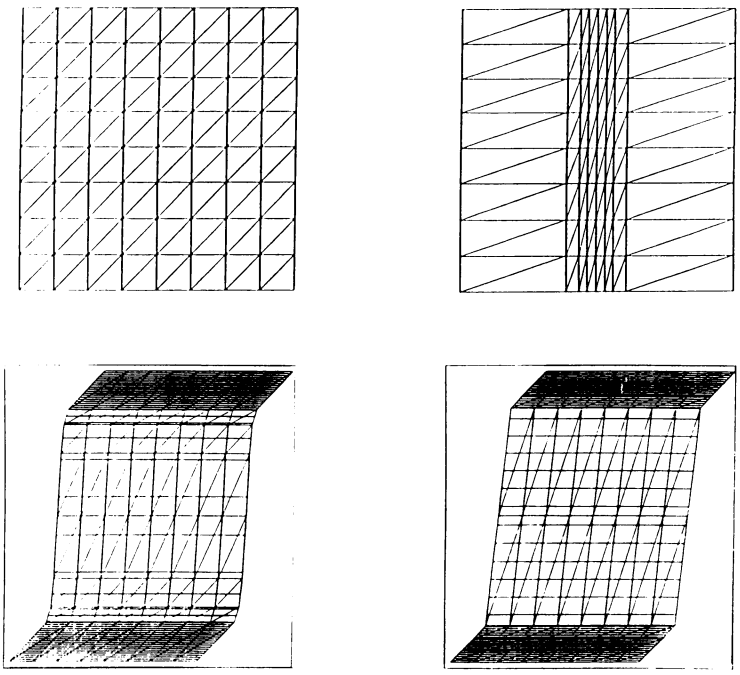
In examples (a) and (c), boundary node displacements along the boundary are set equal to the corresponding displacements on the next grid line in from the boundary. This cleans up a lot of the noise generated by the special behavior of the boundary nodes and the resulting pollution as it spreads into the interior, giving an extra order-of-magnitude accuracy in this way.

As in one dimension, a simplified algorithm exists which fits instead of $f(x, y)$ a quadratic interpolant version. This results in simple formulae for stage (i), although stages (ii) and (iii) are still tricky.

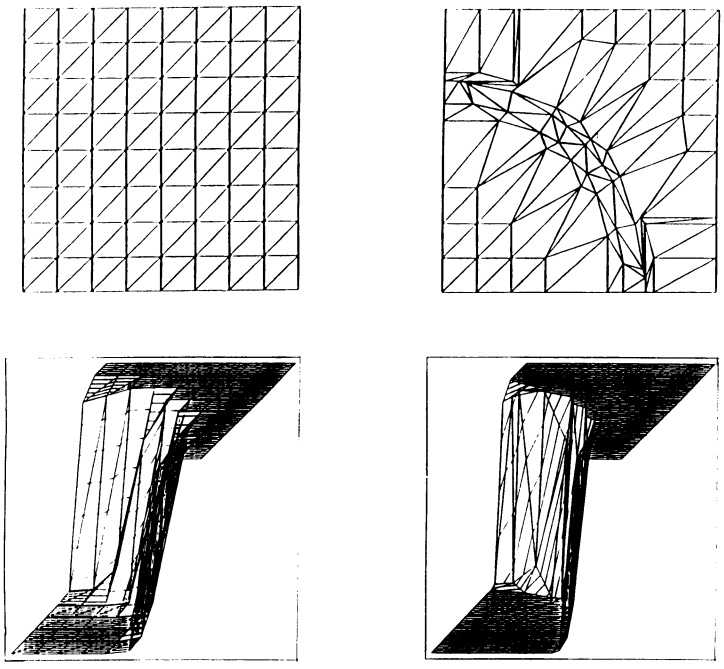


(a)

FIGURE 8. Results for piecewise linear fits in two dimensions



(b)



(c)

FIGURE 8. (Continued)

TABLE 1. L_2 errors for piecewise linear discontinuous best fits

	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
	(3.90×10^{-2})	(1.37×10^{-5})	
(c)	6.62×10^{-3}	2.43×10^{-4}	40
	(2.86×10^{-2})	(4.53×10^{-4})	

7. PIECEWISE CONSTANT FITS IN TWO DIMENSIONS

In the case of best piecewise constant fits with adjustable nodes in two dimensions, $u_x = u_y = 0$, and (5.7) reduces to

$$\sum_k \int_{\Delta_k} 2\{f(x, y) - u(x, y)\}\delta u \, dx \, dy + \sum_k \int_{\partial\Delta_k} \{f(x, y) - u(x, y)\}^2 (\delta x, \delta y) \cdot \hat{n} \, ds = 0. \tag{7.1}$$

With δu as the characteristic function $\pi_k(x, y)$ on element k (Figure 6c), and δx, δy taken successively, as in §4, to be the local "hat" function associated with node j, we have that the conditions for the best piecewise constant L_2 fit to $f(x, y)$, denoted by u^*, x_j^*, and y_j^*, are (cf. (5.8)–(5.11))

$$\int_{\Delta_k^*} \{f(x, y) - w_k^*\} \, dx \, dy = 0, \tag{7.2}$$

$$\int_{j\text{-star}} \left\{ f(x, y) - \sum_k w_k^* \pi_k^*(x, y) \right\}^2 \alpha_j n_1 \, ds = 0, \tag{7.3}$$

$$\int_{j\text{-star}} \left\{ f(x, y) - \sum_{k^*} w_k^* \pi_k^*(x, y) \right\}^2 \alpha_j n_2 \, ds = 0, \tag{7.4}$$

where j-star is as in Figure 7, α_j is as in Figure 6a, k runs over the elements surrounding node j, and

$$u^*(x, y) = \sum_k w_k^* \pi_k^*(x, y). \tag{7.5}$$

From (7.2),

$$w_k^* = \frac{1}{\Delta_k^*} \int_{\Delta_k^*} f(x, y) \, dx \, dy, \tag{7.6}$$

while from (7.3) and (7.4) we may obtain new values of x_j, y_j.

This leads to the following algorithm:

Stage (i).

$$\delta x_j = \delta y_j = 0, \qquad \delta u = \pi_k. \tag{7.7}$$

This stage of the algorithm corresponds to the best L_2 fit amongst piecewise constant functions on a prescribed grid (cf. (7.6)).

Stage (ii). x variations

$$\delta u_j = \delta y_j = 0, \qquad \delta x_j = \alpha_j . \tag{7.8}$$

Stage (ii). y variations

$$\delta u_j = \delta x_j = 0, \qquad \delta y_j = \alpha_j . \tag{7.9}$$

Stage (ii) corresponds to finding x_j (or y_j) such that (7.3) (or (7.4)) holds. Equation (7.3) may be written as

$$\sum_{k=k_1}^{k_e} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} \{f(x_l, y_l) - w_k\}^2 \phi_l \, dy_l = 0 \tag{7.10}$$

(cf. (6.9)), to be solved for x_j with y_j fixed, and (7.4) as

$$\sum_{k=k_1}^{k_e} \sum_{l=l_1}^{l_2} \int_{\text{edge } l} \{f(x_l, y_l) - w_k\}^2 \phi_l \, dx_l = 0 \tag{7.11}$$

to be solved for y_j with x_j fixed.

To solve (7.10), (7.11) for the new node positions x_j, y_j, respectively, we simplify by using trapezium rule quadrature and bisection routines. Again, since the nontangling property is not guaranteed in two dimensions, a relaxation parameter must be introduced to prevent nodes crossing element boundaries.

In summary the algorithm is:

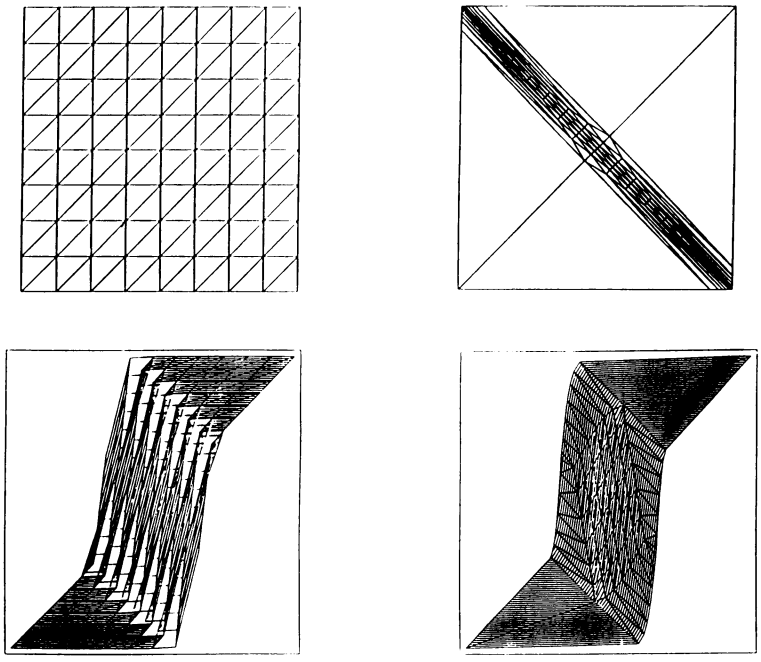
1. set up the initial grid;
2. project $f(x, y)$ elementwise into the space of piecewise constant functions w_k in each element k (stage (i));
3. determine the new grid by solving (7.10) and (7.11) for x_j, y_j, respectively, using bisection, with a relaxation factor to prevent tangling (stage (ii));
4. if the grid is too different from the previous grid go to 2.

Results are shown in Figures 9(a–c) (next page) for the same three examples as in §6 on the same unit square with the same number of interior grid points. The initial grid is again uniform (Figure 8). Figure 9(a) shows grids and profiles for example (a) before and after convergence of the algorithm, while Figures 9(b) and 9(c) show the corresponding results in the case of examples (b) and (c), respectively. Note that, owing to the graphics, the figures show piecewise continuous linear plots, whereas the true plots should be piecewise constant.

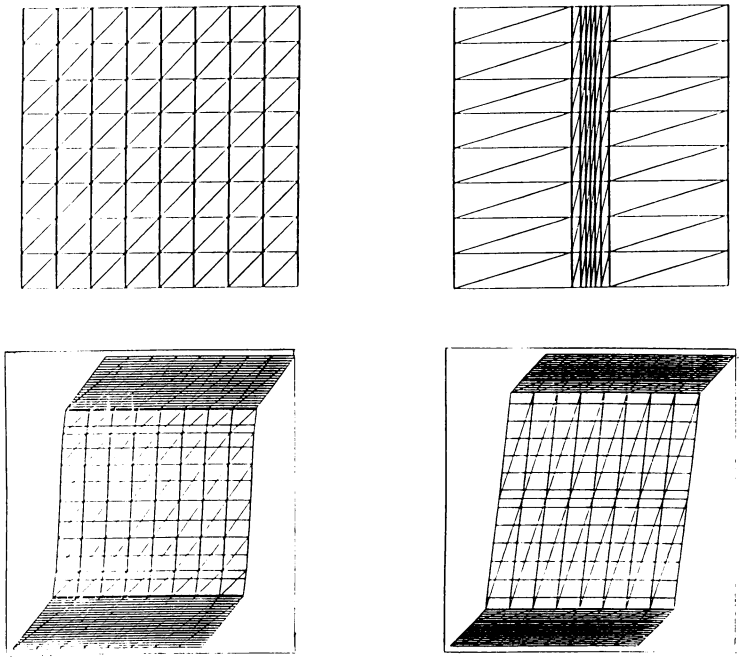
The corresponding list of L_2 errors is shown in Table 2.

TABLE 2. L_2 errors for piecewise constant best fits

	initial error	final error	no. of steps
(a)	1.8	1.55×10^{-3}	40
(b)	1.8	8.54×10^{-4}	40
(c)	1.84	2.34×10^{-3}	20

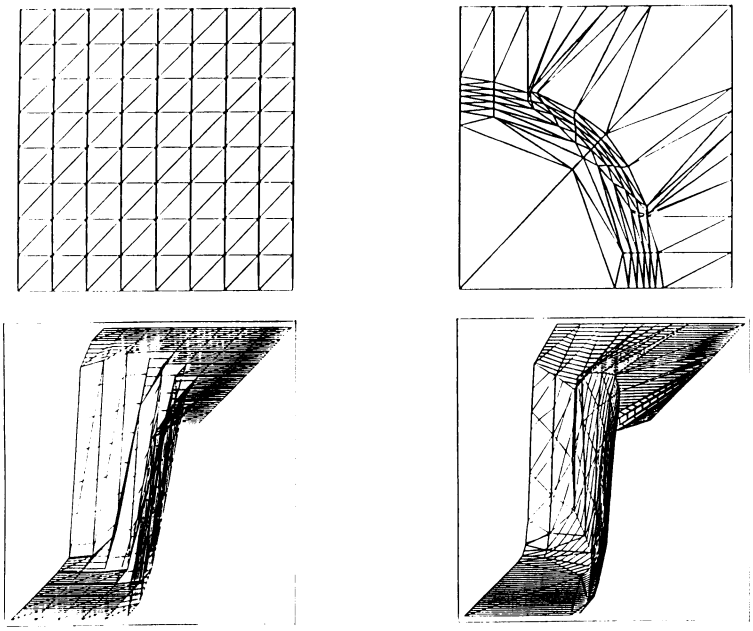


(a)



(b)

FIGURE 9. Results for piecewise constant fits in two dimensions



(c)

FIGURE 9. (Continued)

In example (a), node displacements along the boundary are again set equal to the corresponding displacements on the next grid line in from the boundary. Again, this cleans up a lot of the noise generated by the special behavior of the boundary nodes and the resulting pollution as its spreads into the interior.

Instead of fitting $f(x, y)$, it is possible to fit the linear interpolant of $f(x, y)$ and still achieve a good result. Indeed, by modifying this approach and making it more closely resemble the 1-D case, a nontangling version of the algorithm in 2-D may be obtained (see Baines [1]).

8. Approximate equidistribution results in one dimension

In this section, following [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 constant functions with adjustable nodes.

Using the standard interpolation bound for linear interpolation in element k, we have

$$|f(x) - u^*(x)| \leq \tfrac{1}{8}(x_k - x_{k-1})^2 \max_k |f''| \tag{8.1}$$

and, if $E_1(x)$ is an equidistributing function,

$$(x_k - x_{k-1})E_1'(\theta_k) = \text{ a constant}, \quad C_1, \text{ say}, \tag{8.2}$$

where $x_{k-1} < \theta_k < x_k$. Hence, we get

$$\int_{x_0}^{x_n} (f-u)^2\,dx \leq \frac{1}{64} C_1^4 \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} \{E_1'(\theta_k)\}^{-4} \max_k |f''|^2\,dx. \tag{8.3}$$

Finally, as in [6], we approximate the right-hand side of (8.3) by the integral

$$\frac{1}{64}C_1^4\int_{x_0}^{x_n}\{E_1'(x)\}^{-4}\{f''(x)\}^2\,dx \tag{8.4}$$

and minimize over functions $E_1(x)$, yielding

$$\frac{d}{dx}[\{E_1'(x)\}^{-5}\{f''(x)\}^2] = 0, \tag{8.5}$$

$$E_1(x) \propto \int^x \{f''(\sigma)\}^{2/5}\,d\sigma, \tag{8.6}$$

which may be regarded as the asymptotically equidistributed function.

Similarly, in the piecewise constant case the standard interpolation bound gives

$$|f(x) - u_k^*| \le \frac{1}{\sqrt{6}}(x_k - x_{k-1})\max_k |f'| \tag{8.7}$$

and, if $E_0(x)$ is the equidistributing function,

$$(x_k - x_{k-1})E_0'(\theta_k) = \text{ a constant}, \quad C_0, \text{ say}, \tag{8.8}$$

where $x_{k-2} < \theta_k < x_k$. Hence, we have

$$\int_{x_0}^{x_n}(f - u_k)^2\,dx \le \frac{1}{6}C_0^2\sum_{k=1}^{n}\int_{x_{k-1}}^{x_k}\{E_0'(\theta_k)\}^{-2}\max_k |f'|^2. \tag{8.9}$$

Finally, as before, we approximate the right-hand side of (8.9) by the integral

$$\frac{1}{6}C_0^2\int_{x_0}^{x_n}\{E_0'(x)\}^{-2}\{f'(x)\}^2\,dx \tag{8.10}$$

and minimize over functions $E_0(x)$, yielding

$$\frac{d}{dx}[\{E_0'(x)\}^{-3}\{f'(x)\}^2] = 0, \tag{8.11}$$

or

$$E_0(x) \propto \int^x \{f'(\sigma)\}^{2/3}\,d\sigma, \tag{8.12}$$

which may be regarded as the asymptotically equidistributed function.

These results are approximately borne out by the results in §§3 and 4, which therefore correspond to approximate equidistribution of the functions (8.6) and (8.12), respectively.

9. Conclusions

We have shown that a particular variational approach to finding optimal L_2 fits to a continuous function among piecewise discontinuous linear or constant functions can be used to generate fast and robust algorithms for obtaining such fits. In one dimension the algorithms are conceptually simple, avoid mesh tangling, and are easy to implement. In particular, in the one-dimensional linear case the fits obtained are optimal L_2 piecewise linear *continuous* fits a.e.

We have also demonstrated the strong connection between piecewise discontinuous fits with adjustable nodes and equidistribution.

In two dimensions the algorithms are less robust and harder to implement, needing relaxation parameters to prevent mesh tangling. However, versions using simple quadrature have been shown to be effective for functions with steep fronts.

The extension to three dimensions is straightforward. The main difference in the theory is that in (5.7) the two types of integral are over tetrahedra and their faces. The spokes of the j-star contour become the faces of the triangles which have node j as one of their vertices. Versions using simple quadrature formulae and the closely related versions fitting interpolants of f are easily constructed.

Acknowledgments

I am indebted to Neil Carlson (Purdue) for important early discussions and to Peter Sweby, Carol Reeves, and Nick Birkett for programs and programming assistance.

Bibliography

1. M. J. Baines, *On algorithms for best L_2 fits to continuous functions with variable nodes*, Numerical Analysis Report 1/93, Dept. of Mathematics, University of Reading, U.K., 1993.
2. M. J. Baines and A. J. Wathen, *Moving finite element methods for evolutionary problems: Theory*, J. Comput. Phys. **79** (1988), 245–269.
3. D. L. Barrow, C. K. Chui, P. W. Smith, and J. D. Ward, *Unicity of best mean approximation by second order splines with variable knots*, Math. Comp. **32** (1978), 1131–1143.
4. C. de Boor, *Good approximation by splines with variable knots*, Spline Functions and Approximation Theory (A. Meir and A. Sarma, eds.), Internat. Ser. Numer. Methods, vol. 21, Birkhäuser, Basel, 1973, pp. 57–72.
5. ———, *Good approximation by splines with variable knots*, II, Numerical Methods for ODEs (Dundee 1973), Lecture Notes Math., vol. 3, Springer, New York and Berlin, 1974, pp. 12–20.
6. G. F. Carey and H. T. Dinh, *Grading functions and mesh redistribution*, SIAM J. Numer. Anal. **22** (1985), 1028–1040.
7. C. K. Chui, P. W. Smith, and J. D. Ward, *On the smoothness of best L_2 approximants from nonlinear spline manifolds*, Math. Comp. **31** (1977), 17–23.
8. E. Grosse, *A catalog of algorithms for approximation*, Proc. Conf. on Algorithms for Approximation (Shrivenham, England, 1988), Chapman and Hall, 1988.
9. J. Kautsky and N. K. Nichols, *Equidistributing meshes with constraints*, SIAM J. Sci. Statist. Comput. **1** (1980), 449–511.
10. P. D. Loach and A. J. Wathen, *On the best least squares approximation of continuous functions using linear splines with free knots*, IMA J. Numer. Anal. **11** (1991), 393–409.
11. J. D. Pryce, *On the convergence of iterated remeshing*, IMA J. Numer. Anal. **9** (1989), 315–335.
12. A. B. White, *On the selection of equidistributing meshes for two-point boundary value problems*, SIAM J. Numer. Anal. **16** (1979), 473–502.

Department of Mathematics, University of Reading, P.O. Box 220, Reading, United Kingdom

E-mail address: m.baines@reading.ac.uk