DEPARTMENT OF MATHEMATICS

ON PIECEWISE LINEAR AND CONSTANT L_2 FITS WITH ADJUSTABLE NODES IN 2-D

by

M.J. Baines

Numerical Analysis Report 15/90

UNIVERSITY OF READING

ON PIECEWISE LINEAR AND CONSTANT L_2 FITS WITH ADJUSTABLE NODES IN 2-D

M.J. Baines

Numerical Analysis Report 15/90

Department of Mathematics

University of Reading

P O Box 220

Reading

Keywords: Best Fits, Adjustable Nodes

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

Abstract

In this report simple procedures are used to determine piecewise linear and piecewise constant L_2 fits to a function of two variables on a triangulation of the plane with adjustable nodes.

0. Introduction

In [1],[2] simple procedures for determining optimal or near-optimal piecewise linear and piecewise constant L_2 fits to functions of a single variable were described. In this report we extend these ideas to two dimensions and report further developments.

The situation is more complicated in 2D in several ways. Here, however, we have simply generalised the 1-D ideas given in [1],[2]. Sometimes this works but not always and we have therefore also developed a variation on the procedure taking our cue from related work on moving finite elements [3]. The results obtained exhibit two effective procedures for determining L_2 fits with adjustable nodes, one for piecewise linear approximation and one for piecewise constants.

In sections 1 and 2 generalisations of the theory of [1] and [2], respectively, are given. In section 3 implementation of algorithms arising from the theory is described. In section 4 an alternative algorithm for the piecewise linear case is presented, drawing on similarities with the moving finite element method. Finally in section 5 some results are presented.

§1. Linear Fits: Theory

Let $f(x,y)$ be a given function of x and y and denote by $u_k(x,y)$ the best linear L_2 fit to $f(x,y)$ in a triangle Δ_k. Then

$$\delta \int_{\Delta_k} \left\{f(x,y)-u_k(x,y)\right\}^2 dx\, dy = 0 \qquad u_k \epsilon\, S_k \tag{1.1}$$

or

$$\int_{\Delta_k} \left\{f(x,y)-u_k(x,y)\right\} \delta u_k(x,y)\, dx\, dy = 0 \qquad \delta u_k(x,y) \epsilon\, S_k \tag{1.2}$$

where S_k is the family of planes with support on the triangle Δ_k. For a region Δ, the union of triangles Δ_k, the best L_2 fit to $f(x,y)$ amongst piecewise linear <u>discontinuous</u> functions is also given by (1.1) or (1.2), since the problems decouple.

Now consider the problem of determining the best L_2 fit $u(x,y)$ to $f(x,y)$ amongst discontinous piecewise linear functions in a fixed region Δ on a variable triangulation $\Delta_k (k=1,\ldots,n)$. Then

$$\delta \int_{\Delta} \left\{f(x,y)-u(x,y)\right\}^2 dx\, dy = \delta \sum_k \int_{\Delta_k} \left\{f(x,y)-u(x,y)\right\}^2 dx\, dy = 0 \tag{1.3}$$

where the internal vertices of the Δ_k are also varied.

It is convenient to introduce new independent variables ξ, η which remain fixed, while x and y join u as dependent variables, all now depending on ξ and η and denoted by $\hat{x}, \hat{y}$ and $\hat{u}$ respectively.

Then, with $\hat{u}(\xi,\eta) = u(x(\xi,\eta),\ y(\xi,\eta))$, (1.3) becomes

$$\delta \sum_k \int_{\Delta_k} \left\{ f(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta)) - \hat{u}(\xi,\eta) \right\}^2 J\ d\xi\ d\eta = 0 \tag{1.4}$$

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

Taking the variations of the integral in (1.4) gives

$$\int_{\Delta_k} \Bigg\{ 2\left\{ f(\hat{x}(\xi,)\hat{y},(\xi,\eta)) - \hat{u}(\xi,\eta) \right\}$$

$$\left\{ f_1(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta))\delta\hat{x} + f_2(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta))\delta\hat{y} - \delta\hat{u}(\xi,\eta) \right\} J$$

$$+ \left\{ f(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta)) - \hat{u}(\xi,\eta) \right\}^2 \delta J \Bigg\} d\xi d\eta. \tag{1.5}$$

Integrating the last term by parts leads to

$$- \int_{\Delta_k} 2 \left\{ f(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta)) - \hat{u}(\xi,\eta) \right\} \left\{ \nabla f(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta))J - \underline{\nabla}\hat{u} \right\} \delta\hat{x}\ d\xi d\eta$$

$$+ \int_{\partial\Delta_k} \{ f(\hat{x}(\xi,\eta),\hat{y}(\xi,\eta)) - \hat{u}(\xi,\eta) \}^2 \ (\delta x,\delta y).\hat{\underline{n}}\ ds \Bigg\} . \tag{1.6}$$

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

Collecting terms and returning to the x,y,u notation, (1.4) yields

$$\sum_k \int_{\Delta_k} 2\left\{f(x,y) - u(x,y)\right\} \left\{\delta u - u_x \delta x - u_y \delta y\right\} dxdy +$$

$$\sum_k \int_{\partial\Delta_k} \{f(x,y) - u(x,y)\}^2 (\delta x, \delta y).\hat{\underline{n}}\, ds = 0 \qquad (1.7)$$

With $\delta x, \delta y = 0$ this leads back to (1.2) and equations for the best piecewise linear discontinuous L_2 fit to $f(x,y)$. In terms of independent variations $\delta u_j, \delta x_j, \delta y_j$ at node j the full conditions are

$$\int_{\Delta_k} \left\{f(x,y)-u(x,y)\right\} \delta u_j \, dxdy = 0 \qquad \forall\ \delta u_j \qquad (1.8)$$

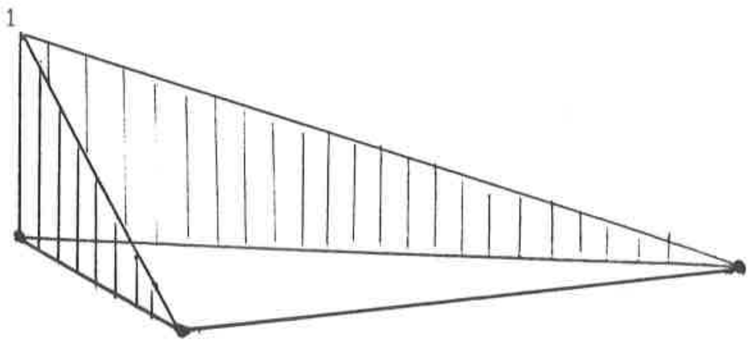
$$\sum_k \int_{\Delta_k} 2\left\{f(x,y)-u(x,y)\right\}(-u_x)\delta x_j \, dxdy + \int_{\partial\Delta_k} \{f(x,y) - u(x,y)\}^2 n_1 \delta x_j \, ds = 0$$
$$\forall\ \delta x_j \ . \qquad (1.9)$$

$$\sum_k \int_{\Delta_k} 2\left\{f(x,y)-u(x,y)\right\}(-u_x)\delta y_j \, dxdy + \delta\Delta_k \int_{\partial\Delta_k} \{f(x,y) - u(x,y)\}^2 n_2 \delta y_j \, ds = 0$$
$$\forall\ \delta y_j \ . \qquad (1.10)$$

where $\hat{\underline{n}} = (n_1, n_2)$ and k runs over elements surrounding node j .

With δu_j in the space of piecewise linear discontinuous functions the orthogonality condition (1.8) is equivalent [1] to the conditions

$$\left.\begin{aligned} &\iint_{\Delta_k} \left\{f(x,y)-u(x,y)\right\} \phi_{k1}\,dxdy = 0 \\ &\iint_{\Delta_k} \left\{f(x,y)-u(x,y)\right\} \phi_{k2}\,dxdy = 0 \\ &\iint_{\Delta_k} \left\{f(x,y)-u(x,y)\right\} \phi_{k3}\,dxdy = 0 \end{aligned}\right\} \qquad (1.11)$$



Basis Function ϕ_{k1}

fig. 1

where ϕ_{k1}, ϕ_{k2}, ϕ_{k3} are linear basis functions in element k (see fig. 1). On the other hand, since $\delta x_j, \delta y_j$ lie in the space of piecewise linear continuous functions, letting α_j be the linear finite element basis function at node j we may set

$$\delta x_j = \alpha_j, \ \delta y_j = 0, \ \delta u_j = u_x \delta x_j$$

and

$$\delta x_j = 0, \ \delta y_j = \alpha_j, \ \delta u_j = u\ \delta y_j$$

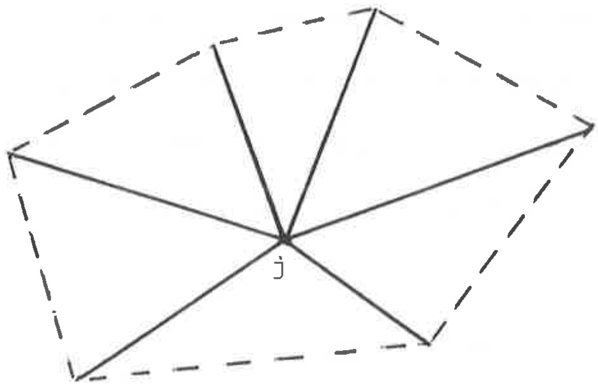
in turn in (1.7) to obtain

$$\int_{j\ \mathrm{star}} \left[f(x,y)-u(x,y)\right]^2 \alpha_j n_1 \, ds = 0 \ . \qquad (1.12)$$

for x_j, and

$$\int_{j\ \mathrm{star}} \left[f(x,y)-u(x,y)\right]^2 \alpha_j n_2 \, ds = 0 \qquad (1.13)$$

for y_j, where $\hat{\underline{n}} = (n_1, n_2)$ and "j star" indicates the sides of the triangles passing through the node j (see fig. 2).



The spokes of j^*

fig. 2

The problem of finding $u(x,y)$, belonging to the union of the S_k, which satisfies (1.11) is standard. Setting

$$u_k(x) = \sum_i w_{ki} \, \phi_{ki} \qquad (1.14)$$

in element k, we substitute into (1.11) and find that

$$C_k \underline{w}_k = \underline{b}_k \tag{1.15}$$

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

$$C = \frac{A_k}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{1.16}$$

where A_k is the area of element k .

The other problems, those of finding x_j satisfying (1.12) with $\delta u_j = u_x \delta x_j$ and y_j satisfying (13) with $\delta u_j = u_y \delta y_j$, are much more difficult non-linear problems. To make progress we solve the problem approximately, with one or both of the following simplifications.

(a) replace the line integrals in (1.12) and (1.13) by a suitable 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. Similarly for the y_j .

The device (b) was used in [1] to obtain converged solutions for x_j, in effect a "lagged" form of the equation being solved as the overall iteration converged.

§2 Constant Fits: Theory

In the case of best piecewise constant fits with adjustable nodes, $u_x = u_y = 0$ and (1.7) reduces to

$$\sum_k \int_{\Delta_k} 2\{f(x,y)-u(x,y)\}\delta u \, dxdy + \sum_k \int_{\partial\Delta_k} \{f(x,y) - u(x,y)\}^2(\delta x, \delta y).\hat{n} \, ds = 0 \tag{2.1}$$

With δu as the characteristic function $\pi_k(x,y)$ on element k, and $\delta x, \delta y$ taken successively, as in §1, to be the local "hat" function associated with node j we have the conditions

$$\int_{\Delta_k} \{f(x,y) - w_k\} \, dxdy = 0 \tag{2.2}$$

$$\int_{j \text{ star}} \{f(x,y) - \sum_k w_k \pi_k(x,y)\}^2 \, \alpha_j \, n_1 ds = 0 \tag{2.3}$$

$$\int_{j \text{ star}} \{f(x,y) - \sum_{k^*} w_k \pi_k(x,y)\}^2 \, \alpha_j \, n_2 ds = 0 \tag{2.4}$$

where j star is as in (1.12),(1.13), k runs over the elements surrounding node j and

$$u(x,y) = \sum_k w_k \pi_k(x,y) \, . \tag{2.5}$$

From (2.2)

$$w_k = \frac{1}{\Delta_k} \int_{\Delta_k} f(x,y) \, dxdy \tag{2.6}$$

while from (2.3),(2.4) we obtain new values of x_j, y_j .

§3. The Algorithms

The algorithms used to find near-best discontinuous L_2 fits with variable nodes are in three stages (carried out repeatedly until convergence), corresponding to the choices of variations referred to in §1 and §2 above.

(a) Piecewise linears

Stage (i)

$$\delta x_j = 0,\ \delta y_j = 0\ ,\quad \delta u = \phi_{k1},\ \phi_{k2}\ \text{ or }\ \phi_{k3} \tag{3.1}$$

This stage of the algorithm corresponds to the best L_2 fit amongst discontinuous linear functions on a prescribed grid, as in (1.1),(1.2), and (1.15) above.

Stage (ii)

$$\delta x_j = \alpha_j\ ,\ \delta y_j = 0\ ,\ \delta u_j - u_x \delta x_j = 0\ (j=1,2,\ldots,\ n) \tag{3.2}$$

Stage (iii)

$$\delta x_j = 0,\ \delta y_j = \alpha_j,\ \delta u_j - u_y \delta y_j = 0\ (j=1,2,\ldots,\ n) \tag{3.3}$$

Stage (ii) corresponds to finding x_j such that (1.12) holds, with variations of x,u restricted to points lying on the planes of the stage (i) solution (possibly extrapolated) in each of the elements k^* surrounding j . Stage (iii) works similarly for y_j .

The algorithm is analogous to minimising a quadratic function Q(X,Y,Z) using three search directions v1, v2 and v3 spanning a space. Starting from some initial guess we may minimise Q(X,Y,Z) in the directions v1, v2 and v3 in turn. Similarly, in the present case to find the near-best L_2 fit we may begin with an initial guess $\{x_j\},\{y_j\},\{u_j\}$. Stage (i) is to find the minimum in the linear manifold specified by the variations given in (3.2) and so solve (1.10)-(1.11) for new discontinuous values $\underline{w}$ of u (see (1.15)) with the x_j, y_j fixed. Stage (ii) is to seek the minimum in the manifold specified by the variations given in (3.2) and so solve (1.12) approximately for new $\{x_j\}$ by the implementation of §1(a),(b), more fully described below. Stage (iii) corresponds to stage (ii) but with x,y interchanged.

Let $k = k_1, \ldots, k_2$ 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 fig. 2). Then (1.12) may be written

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} \left\{f(x,y)-u(x,y)\right\}^2 (-\sin\theta_\ell)\, \alpha_j ds_\ell = 0 \qquad (3.4)$$

where θ_ℓ is the angle between the edge ℓ and the x-axis so that $n_1 = -\sin\theta_\ell$. Since $u(x,y)$ is restricted in element k by $\delta u = u_x\,\delta x$, $\delta y = 0$, and passes through the point (x_j, y_j, w_{jk}), say, we have

$$u(x,y) - w_{jk} = (x-x_j)(u_x)_k + (y-y_j)(u_y)_k \qquad (3.5)$$

so that, writing $\sin\theta_\ell \, ds_\ell = \delta y_\ell$, the integral (3.4) becomes

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} \left\{ f(x_\ell, y_\ell) - w_{jk} - (x_\ell - x_j)(u_x)_k - (y_\ell - y_j)(u_y)_k \right\}^2 \phi_\ell \, dy_\ell = 0 \tag{3.6}$$

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.

As in [1] we introduce an iteration (to be run in tandem with the main iteration,) in which we solve for $x_j^{(i+1)}$ in terms of $x_j^{(i)}$ where

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} \left\{ f(x_\ell, y_\ell) - w_{jk} - (x_\ell - x_j)(u_x)_k - (y_\ell - y_j)(u_y)_k \right\}^2 \phi_\ell \, dy_\ell = 0 \tag{3.7}$$

where f and u_x are based on $x_j^{(i)}$ and x_j and the range of integration is based on $x_j^{(i+1)}$.

Equation (3.6) can then be written

$$AX^2 - BX + C = 0 \tag{3.8}$$

where $X = x_j^{(i+1)} - x_j$

$$\left.\begin{aligned}
A &= \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\substack{\text{edge}\\ \ell}} (u_x)_k^2 \phi_\ell dy_\ell \\
B &= \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\substack{\text{edge}\\ \ell}} \left\{f(x_\ell,y_\ell) - w_{jk} - (x_\ell - x_j)(u_x)_k - (y_\ell - y_j)(u_y)_k\right\} \phi_\ell dy_\ell \\
C &= \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\substack{\text{edge}\\ \ell}} \left\{f(x_\ell,y_\ell) - w_{jk} - (x_\ell - x_j)(u_x)_k - (y_\ell - y_j)(u_y)_k\right\}^2 \phi_\ell dy_\ell
\end{aligned}\right\} \quad (3.9)$$

and (provided that $B^2 > 4AC$) solved for X. The integrals in (3.9) may be evaluated by a quadrature rule. Both Gaussian quadrature and the trapezium rule may been used. In the latter case (3.9) simplifies considerably and becomes

$$\left.\begin{aligned}
A &= \tfrac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \tfrac{1}{2}(u_x)_k^2 (y_{\ell_2} - y_{\ell_1}) \\
B &= \tfrac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \tfrac{1}{2}\left\{f(x_j,y_j) - w_{jk}\right\}(u_x)_k (y_{\ell_2} - y_{\ell_1}) \\
C &= \tfrac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \tfrac{1}{2}\left\{f(x_j,y_j) - w_{jk}\right\}^2 (y_{\ell_2} - y_{\ell_1})
\end{aligned}\right\} \quad (3.10)$$

Call this method MBFLT.

Two real solutions of (3.8) may be regarded in simple situations as analogous to the "intersection" solution and "averaged" solution encountered in the 1-D case discussed in [1], corresponding to convex/concave parts and inflections of the function f, respectively. In the present two-dimensional case the many contributions to A,B,C blur the simple 1-D interpretation but for consistency we choose the root corresponding to the intersection solution whenever the function f is convex/concave: at other points it is not clear which root to take (but see below).

If $B^2 = 4AC$ in (3.8) 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 small, which may be due to non-convex or nearly planar patches in f or simply closeness to the best fit. A regularisation parameter is therefore introduced which protects the roots from the effects of the potential resultant singularities. By adding a term

$$\epsilon(X - X_{ave})^2 \qquad (3.10)$$

to (3.8), with the appropriate sign, where X_{ave} is the current mean value of the coordinate displacements for the nodes surrounding the node j, singularity is avoided and when (3.10) dominates, node j is simply moved along with its neighbours.

An alternative approach to the iteration described above for (3.7) is to base one of two factors in the square in (3.7) wholly on $x_j^{(i)}$

with the other factor treated as before. This gives

$$-BX + C = 0 \qquad (3.11)$$

instead of (3.8) and an unambiguous (but weaker) iteration procedure. Another approach, corresponding more closely to an ℓ_1 approximation, would be to replace B,C by

$$\left.\begin{aligned} B &= \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} (u_x)_k S_k \phi_\ell dy_\ell \\ C &= \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} \left\{ f(x_\ell, y_\ell) - w_{jk} - (x_\ell - x_j)(u_x)_k - (y_\ell - y_j)(u_y)_k \right\} S_k \phi_\ell dy_\ell \end{aligned}\right\} \qquad (3.12)$$

where

$$S_k = \text{sgn} \left\{ f(x_j, y_j) - w_{jk} \right\} . \qquad (3.13)$$

The corresponding forms using the trapezium rule are

$$\left.\begin{aligned} B &= \frac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} (u_x)_k (y_{\ell_2} - y_{\ell_1}) S_k \\ C &= \frac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} |f(x_j, y_j) - w_{jk}| (y_{\ell_2} - y_{\ell_1}) \end{aligned}\right\} \qquad (3.14)$$

Call this method MBFLT1.

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 prevents this happening.

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

To obtain a continuous piecewise linear approximation we may take an average of the w_{jk} values in each element adjacent to a given node to give an approximate nodal value.

(b) Piecewise Constants

Stage (i)

$$\delta x_j = \delta y_j = 0 \ , \quad \delta u = \pi_k \ . \tag{3.10}$$

This stage of the algorithm corresponds to the best L_2 fit amongst piecewise constant functions on a prescribed grid (see (2.6)).

Stage (ii)

$$\delta u_j = \delta y_j = 0 \ , \quad \delta x_j = \alpha_j \tag{3.11}$$

Stage (iii)

$$\delta u_j = \delta x_j = 0 \ , \quad \delta y_j = \alpha_j \tag{3.12}$$

Stage (ii) corresponds to finding x_j such that (2.3) holds, while stage (iii) corresponds to finding y_j such that (2.4) holds.

As in the case of linear fits, the algorithm is analogous to minimising a quadratic function $Q(X,Y,Z)$ using three search directions (see above).

Equation (2.3) may be written as

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} \{f(x_\ell, y_\ell) - w_k\}^2 \, \phi_\ell \, dy_\ell = 0 \tag{3.13}$$

(c.f. (3.6)), and (2.4) as

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \int_{\text{edge } \ell} \{f(x_\ell, y_\ell) - w_{jk}\}^2 \; \phi_\ell dx_\ell = 0 \tag{3.14}$$

(c.f. (3.7)).

To solve (3.13), (3.14) for the new node positions x_j, y_j, respectively, we use quadrature and bisection routines. Again, a relaxation parameter is introduced to prevent nodes crossing element boundaries. Call this method MBFCT.

§4 A Unified Algorithm for the Displacements

While the algorithm for piecewise constant approximation given in §3(b) works well, that for piecewise linears is less robust and it seems worthwhile to develop an alternative method in this case based on a different approximation argument inspired by the ideas of Moving Finite Elements [3].

Starting from (3.6) and noting that

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_2}^{\ell_2} \int dy_\ell = 0 \tag{4.1}$$

we observe that (3.6) expresses a kind of balance of the terms

$$\{f(x_\ell, y_\ell) - w_{jk} - (x_\ell - x_j)(u_x)_k - (y_\ell - y_j)(u_y)_k\}^2 \phi_\ell \ . \tag{4.2}$$

In the 1-D case this is obvious, since (3.6) reduces to

$$\{f(x) - w_{jL}-(x-x_j)(u_x)_L\}^2 - \{f(x) - w_{jR}-(x-x_j)(u_x)_R\}^2 = 0. \quad (4.3)$$

To imitate the situation in 2-D we first approximate the integrals in (1.12),(1.13) by the trapezium rule, giving (with (3.5))

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \{f(x,y) - w_{jk}-(x-x_j)(u_x)_k - (y-y_j)(u_y)_k\}^2(y_{\ell_2}-y_{\ell_1}) = 0 \quad (4.4)$$

(c.f. (3.10) and

$$\sum_{k=k}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \{f(x,y) - w_{jk}-(x-x_j)(u_x)_k - (y-y_j)(u_y)_k\}^2(x_{\ell_2}-x_{\ell_1}) = 0 . \quad (4.5)$$

In §3, equation (4.4) with $y = y_j$ was solved for x and equation (4.5) with $x = x_j$ is solved for y. Here we consider the single equation

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \{f(x,y) - w_{jk}-(x-x_j)(u_x)_k - (y-y_j)(u_y)_k\}^2(r_{\ell_2}-r_{\ell_1}) = 0 \quad (4.6)$$

where r may be y or x, or indeed any other cartesian coordinate, and attempt to solve for x,y simultaneously. As above, since

$$\sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} (r_{\ell_2} - r_{\ell_1}) = 0 \ , \tag{4.7}$$

(4.6) expresses a certain balance between the terms

$$\{f(x,y) - w_{jk} - (x-x_j)(u_x)_k - (y-y_j)(u_y)_k\}^2 \ . \tag{4.8}$$

Now (4.6), (4.7) are satisfied if we can find x,y such that the terms (4.8) are equal $\forall k$. As we shall see, this is not generally possible, but x,y can be found such that it is approximately true.

First, suppose that the expressions in the brackets in (4.8) have the same sign (corresponding to convex or concave f(x,y)) so that we may take the square roots (c.f. (4.3) and [1]). Then equality of the terms in (4.8) is equivalent to equality of the terms

$$\{f(x,y) - w_{jk} - (x-x_j)(u_x)_k - (y-y_j)(u_y)_k\} \qquad \forall k \tag{4.9}$$

c.f.(3.12)-(3.14), where for the moment we take $S_k = +1$.
Since there are only two unknowns x and y , in general, and more than two equations in (4.9), no solution for x,y exists.

The next best thing is a least squares solution which minimises

$$\sum \{f(x,y) - w_{jk} - (x-x_j)(u_x)_k - (y-y_j)(u_y)_k + e\}^2 \tag{4.10}$$

where $-e$ is the common quantity to which all the expressions (4.9) are equal. Minimising (4.10) over x, y and e yields the equations

$$\begin{bmatrix} \Sigma 1 & -\Sigma(u_x)_k & -\Sigma(u_y)_k \\ -\Sigma(u_x)_k & \Sigma(u_x)_k^2 & \Sigma(u_x)_k(u_y)_k \\ -\Sigma(u_y)_k & \Sigma(u_x)_k(u_j)_k & \Sigma(u_x)_k^2 \end{bmatrix} \begin{bmatrix} e \\ x-x_j \\ y-y_j \end{bmatrix} = \begin{bmatrix} \Sigma(w_{jk} - f(x,y)) \\ -\Sigma(u_x)_k(w_{jk} - f(x,y)) \\ -\Sigma(u_y)_k(w_{jk} - f(x,y)) \end{bmatrix} \tag{4.11}$$

which may be readily solved for x and y , if non-singular. The matrix in (4.11) is positive semi-definite: if the determinant is close to zero no attempt to update x and y is made. This is method MBFMFE.

If the expressions in the brackets in (4.8) are not all of the same sign then there is a ± symbol multiplying e in (4.10) (see (3.12)-(3.14)) and the corresponding least squares equations must be modified to include extra factors S_k (see (3.13)). Call this method MBFMFE1.

The algorithms of this section are motivated by the similarity between the present problem and that of Moving Finite Elements (see [3]).

§5 Results

We show results for the algorithms MBFLT and MBFCT on three examples,

(a) tanh 20(x-½)

(b) tanh 20(x+y-1)

(c) tanh $20(x^2 + y^2 - ½)$

each on the unit square with 49 interior grid points. In each case the initial grid is uniform, as shown in fig 0. The initial profiles are shown as piecewise linear functions on the initial grid in figs. 1(a), 1(b), 1(c), respectively. The algorithm MBFLT1 gives results which differ only slightly from those of MBFLT, while those from algorithms MBFMFE and MBFMFE1 are significantly worse in most cases.

Figures 2L(a), 3L(a) show example (a) after convergence of the algorithm MBFLT, while figures 2C(a), 3C(a) show example (a) after convergence of the algorithm MBFCT. Note that the figures show piecewise continuous linear plots whereas the true plots should be piecewise linear discontinuous or piecewise constant. For the case of MBFLT the errors from the corresponding piecewise linear continuous function obtained by averaging nodal values are shown in brackets.

Figures with brackets (b) and (c) show the corresponding results in the case of examples (b) and (c), respectively.

A table of L_2 errors follows:

	initial error	final error	no. of steps	ϵ
(a) MBFLT	3.77×10^{-3} (2.49×10^{-2})	3.00×10^{-4} (8.54×10^{-4})	40	10^{-2}
MBFCT	2.88×10^{-2}	1.37×10^{-2}	70	-
(b) MBFLT	4.06×10^{-3} (3.90×10^{-2})	4.90×10^{-6} (1.18×10^{-5})	170	10^{-8}
MBFCT	1.01×10^{-1}	9.69×10^{-4}	140	-
(c) MBFLT	6.62×10^{-3} (2.86×10^{-2})	1.22×10^{-4} (2.70×10^{-4})	38	10^{-2}
MBFCT	8.20×10^{-2}	2.34×10^{-3}	21	-

Table 1 L_2 errors

The piecewise constant algorithm MBFCT is more robust and gives apparently more satisfactory grids than the piecewise linear algorithm MBFLT. The amount of regularisation in MBFLT (through the coefficient ϵ) also varies considerably with the problem and further work is needed on this aspect of the algorithm.

To illustrate the effect of boundaries, example (a) is repeated in figs. 4L(a), 5L(a), 4C(a), 5C(a), with boundary node displacements along the boundary 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 behaviour of the boundary nodes and the resulting pollution as it spreads into the interior. This is particularly true of MBFLT, where an extra order of magnitude accuracy is obtainable this way. The value of ϵ is 10^{-2}.

In conclusion it appears that the grids obtained by seeking best fits with discontinuous piecewise linear elements and adjustable nodes are as efficient as those obtained by enforcing continuity in the same situation. That is to say, allowing the fit to be discontinuous and then averaging local values to give continuity gives as good a fit as forcing continuity from the outset.

§6. Acknowledgements

I am indebted to Neil Carlson for useful discussions and to Peter Sweby, Carol Reeves and Nick Birkett for programmes and programming assistance. The graphical output is from the GP2 package of Lee Buzbee.

§7. References

[1] Baines, M.J. and Carlson N.N. (1990) On Best Piecewise Linear L_2 fits with Adjustable Nodes, Numerical Analysis Report 6/90. Department of Mathematics, University of Reading.

[2] Baines, M.J. (1990) On Best Piecewise Constant L_2 fits with Adjustable Nodes, Numerical Analysis Report 12/90. Department of Mathematics, University of Reading.

[3] Baines, M.J. (1991) On the Relationship between the Moving Finite Procedure and Best Piecewise Linear L_2 Fits with Adjustable Nodes. Numerical Analysis Report 2/91. Department of Mathematics, University of Reading.

Appendix A

In this appendix we extend the result in the main body of the report to general extremals (where they exist).

For the problem of finding the extremal of the integral

$$\int F(x,y,u)\ dxdy \tag{A1}$$

over piecewise linear discontinuous functions $u(x,y)$ with variable nodes, we follow the same procedure as in §1, obtaining

$$\int_{\Delta_k} F_u(x,y,u)\delta u_j\ dxdy = 0 \qquad \forall \delta u_j \tag{A2}$$

$$\int_{\Delta_k} F_u(x,y,u)(-u_x)\delta x_j\ dxdy + \int_{\partial\Delta_k} F(x,y,u)n_1\ \delta x_j\ ds = 0 \qquad \forall k \tag{A3}$$

$$\int_{\Delta_k} F_u(x,y,u)(-u_y)\delta y_j dxdy + \int_{\partial\Delta_k} F(x,y,u)n_2\ \delta y_j\ ds = 0 \qquad \forall k \tag{A4}$$

in place of (1.8),(1.9) and (1.10). Then (1.11), and (1.12) and (1.13) become

$$\int_{\Delta_k} F_u(x,y,u)\ \phi_{k,i}\ dxdy = 0 \qquad (i = 1,2,3) \tag{A5}$$

$$\int_{j\ \text{star}} F(x,y,u)\ \alpha_j\ n_1 ds = 0\ . \tag{A6}$$

$$\int_{j\ \text{star}} F(x,y,u)\alpha_j\ n_2 ds = 0\ . \tag{A7}$$

The corresponding algorithm is to solve (A5) for $u_k(x)$ of the form (1.14) in each element with fixed x_j, y_j (stage (i)) and then to solve (A6),(A7) successively for the new x_j, y_j with u restricted to the stage (i) solution, possibly extrapolated (stages (ii) and (iii)). Both problems are nonlinear and may or may not have solutions which are unique.

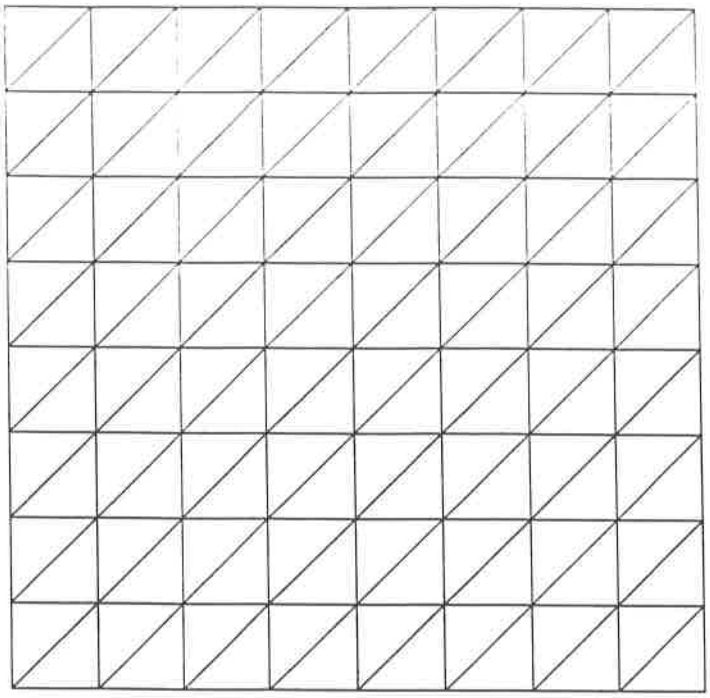


fig. 0

Initial Grid

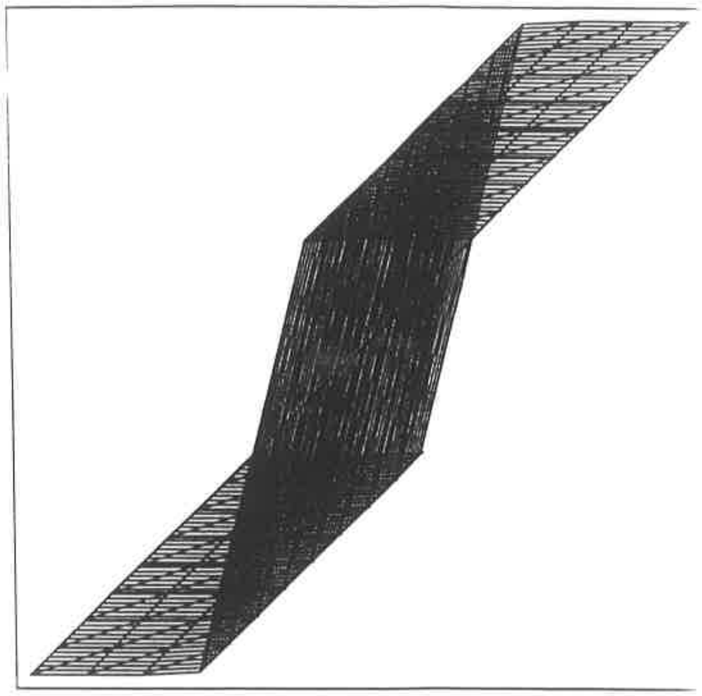


fig.1(a)

Profile for example (a)

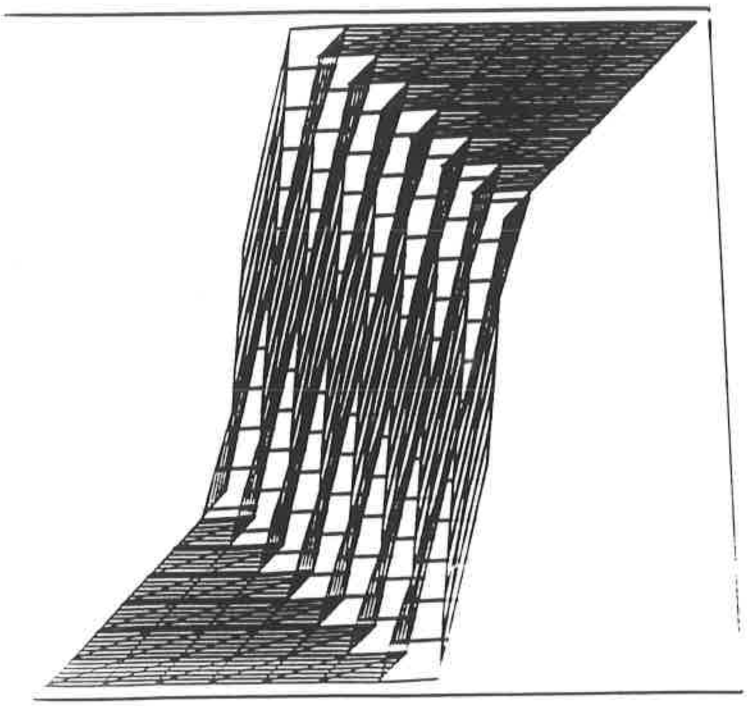


fig. 1(b)

Profile for example(b)

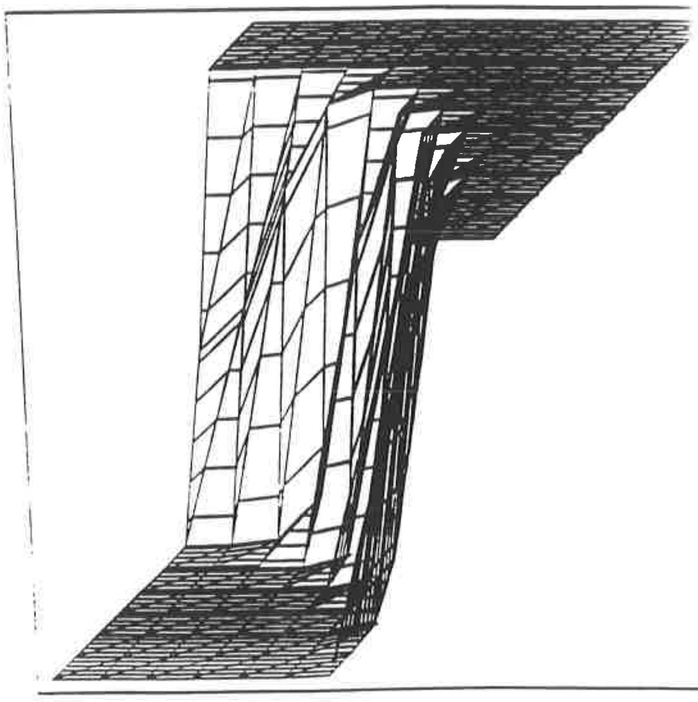


fig. 1(c)

Profile for example (c)

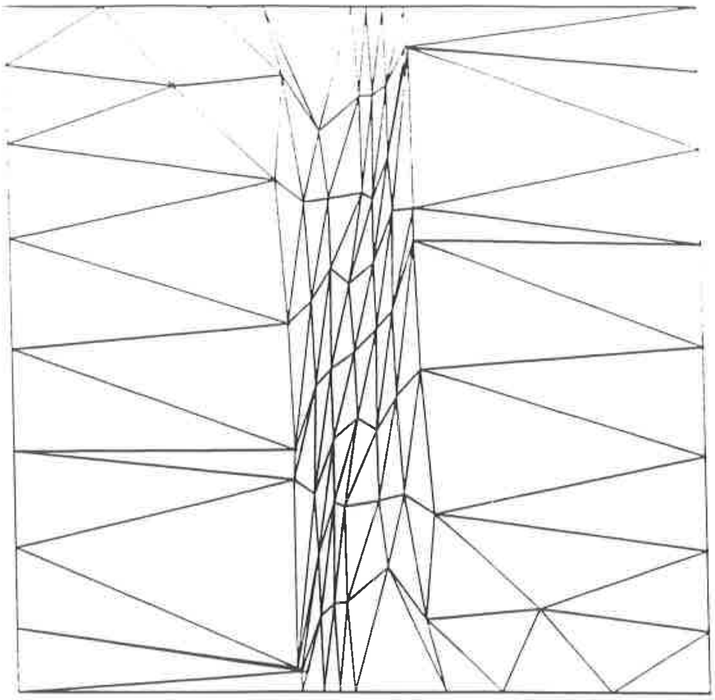


fig. 2L(a)
Final Grid - MBFLT

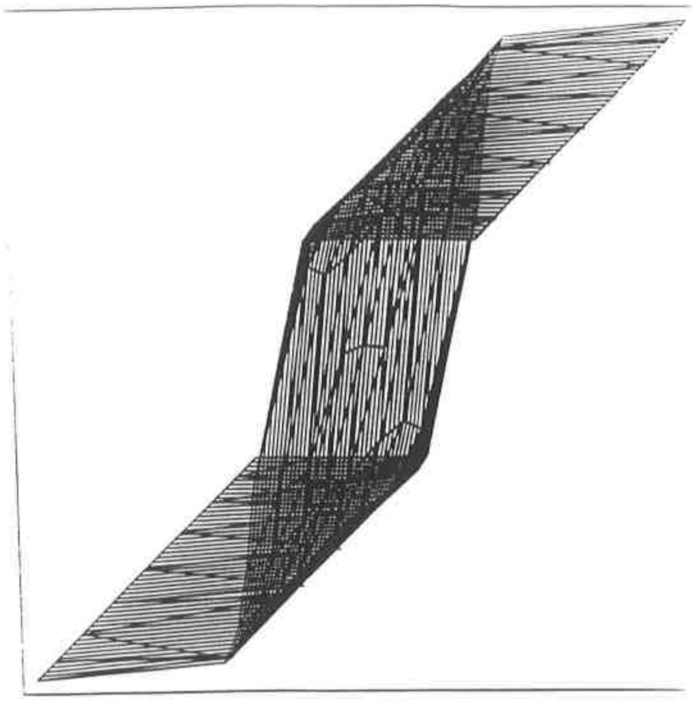


fig. 3L(a)
Final Profile - MBFLT

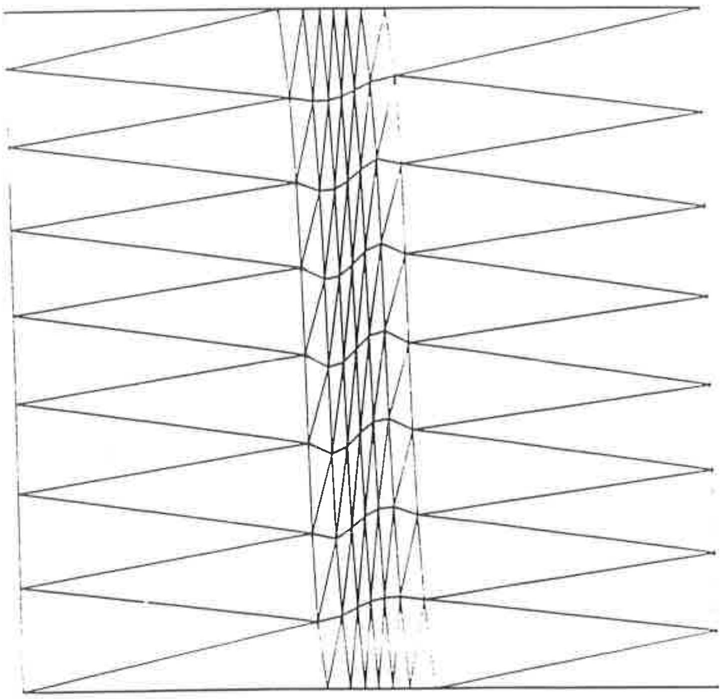


fig. 2C(a)
Final Grid - MBFCT

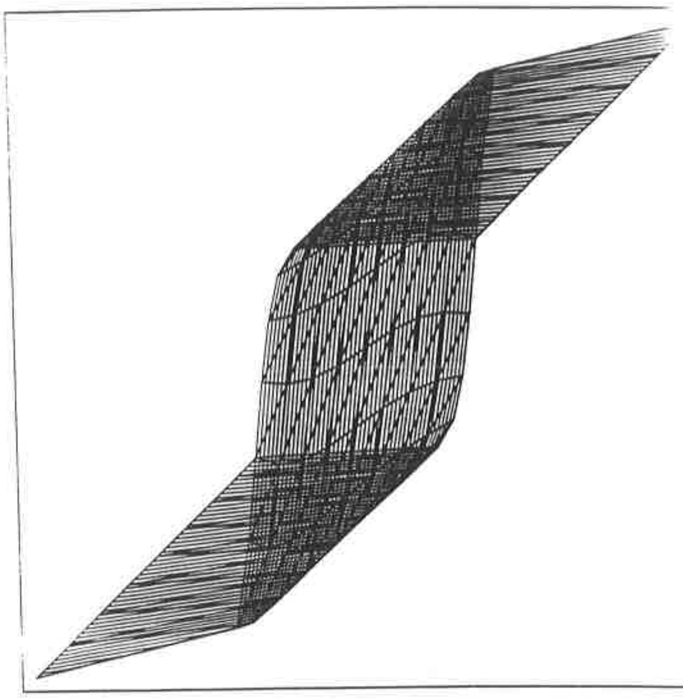


fig. 3C(a)
Final Profile - MBFCT

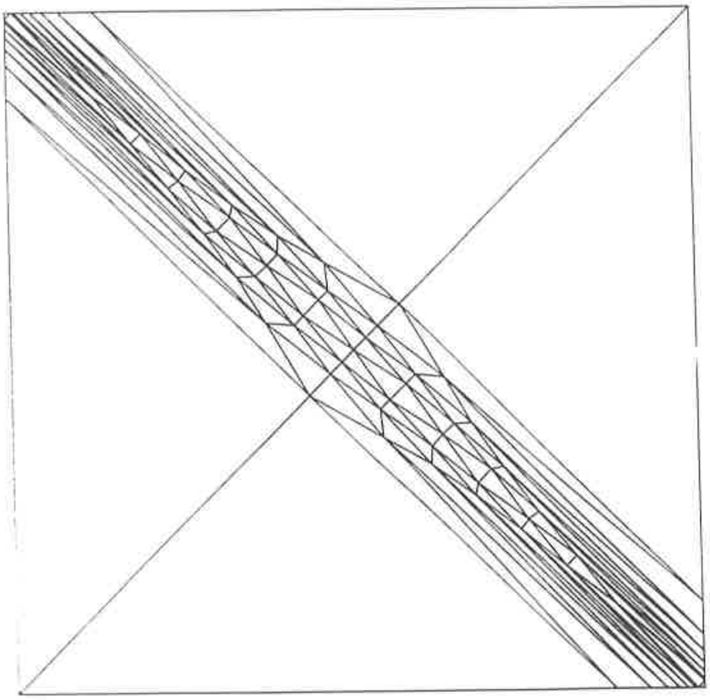


fig. 2L(b)
Final Grid - MBFLT

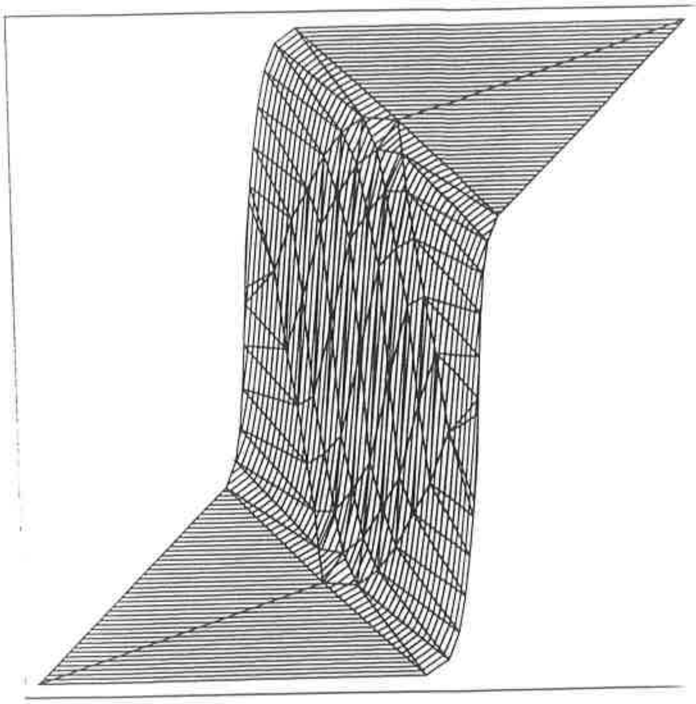


fig. 3L(b)
Final Profile - MBFLT

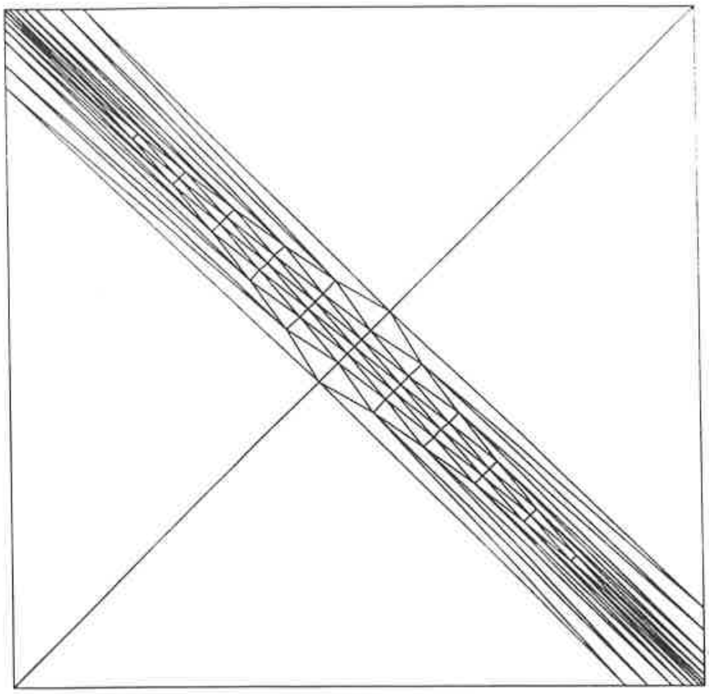


fig. 2L(c)
Final Grid - MBFCT

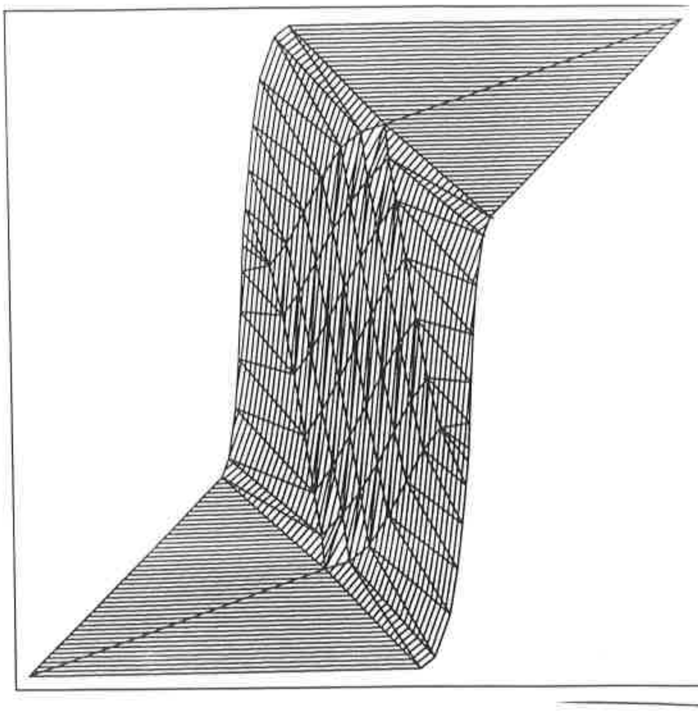


fig. 3L(c)
Final Profile - MBFCT

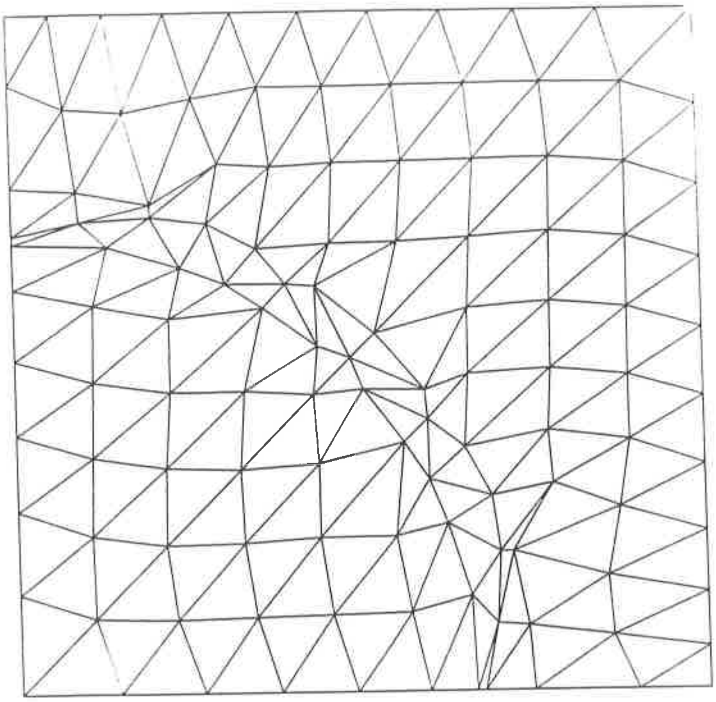


fig. 2L(c)

Final Grid - MBFLT

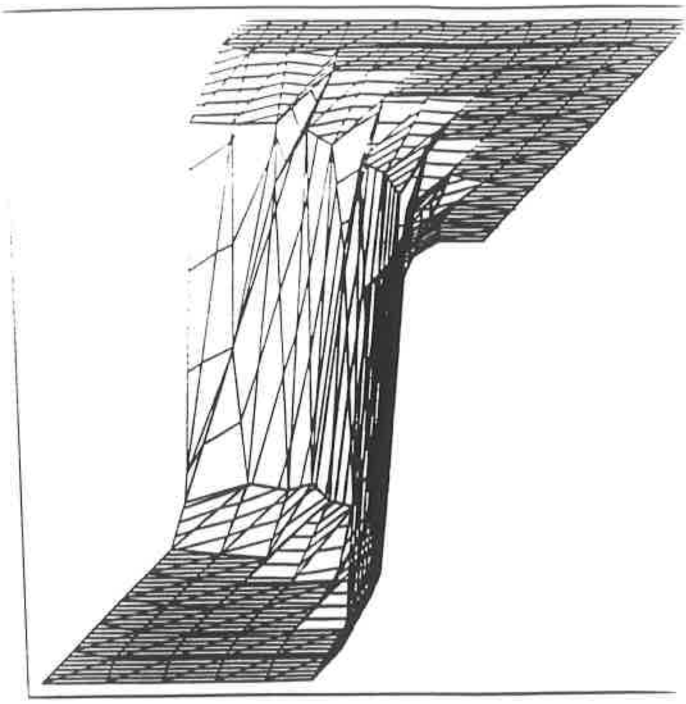


fig. 3L(c)

Final Profile - MBFLT

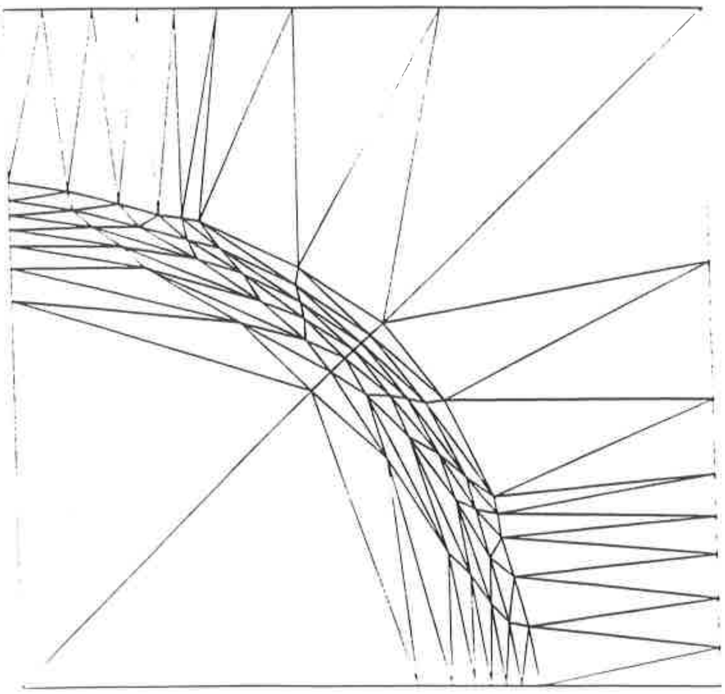


fig. 2C(c)

Final Grid - MBFCT

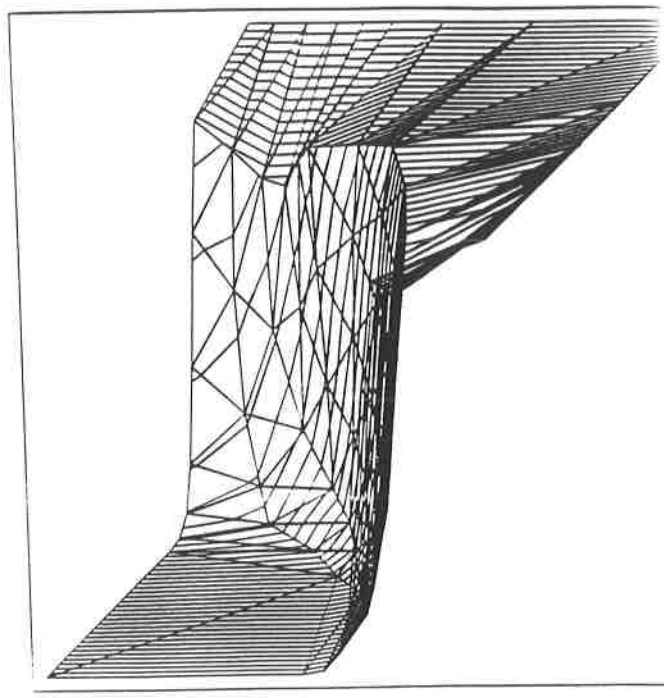


fig. 3(c)

Final Profile - MBFCT

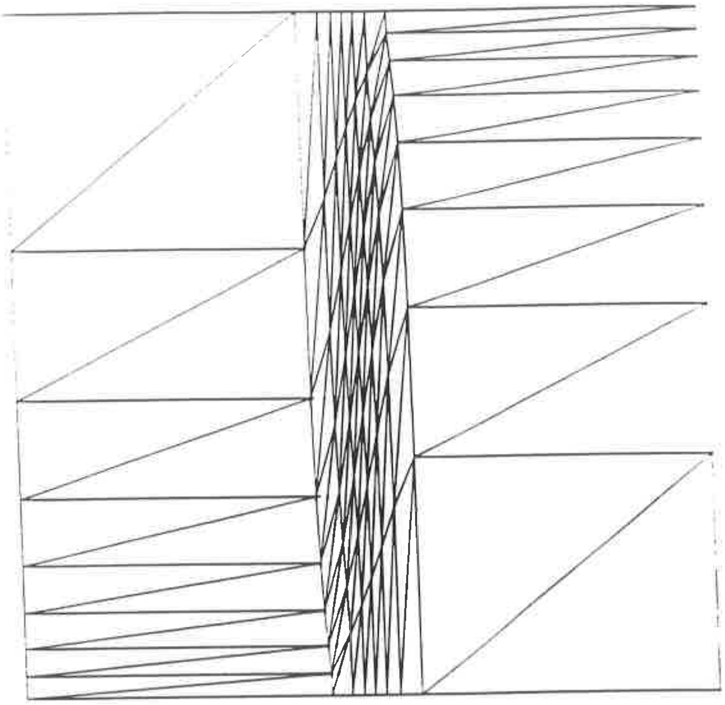


fig. 4L(a)
Final Grid - MBFLT

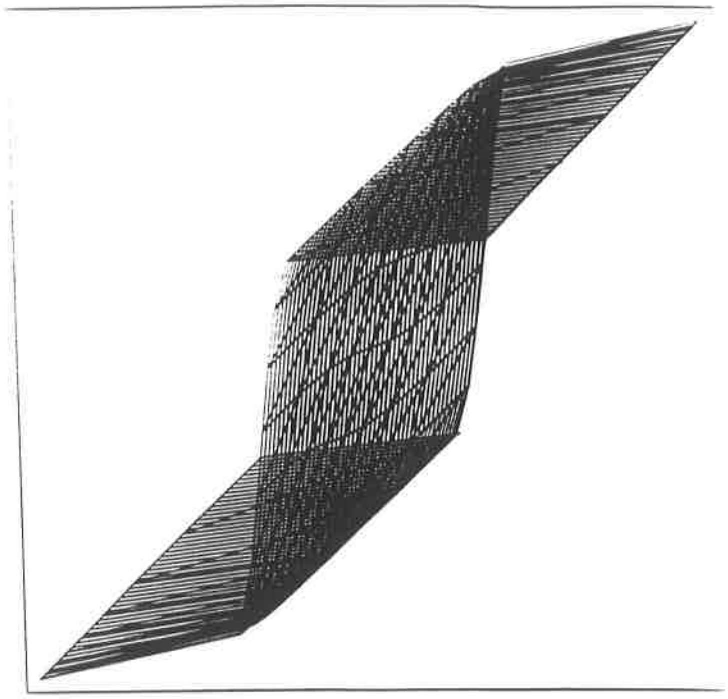


fig. 5L(a)
Final Profile - MBFLT

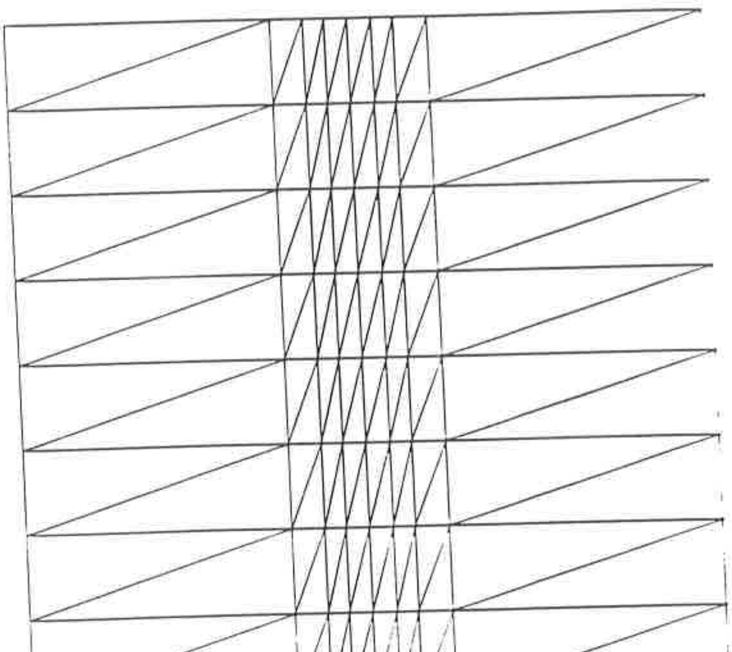


fig. 4C(a)
Final Grid - MBFCT

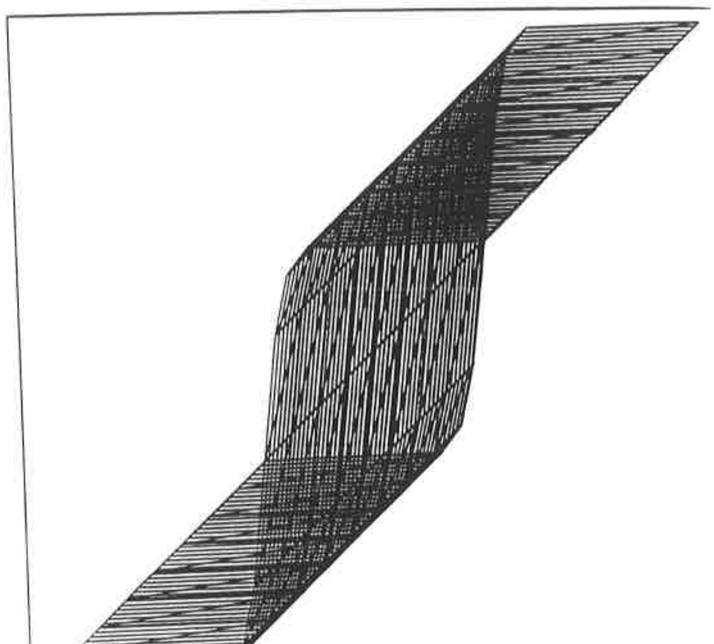


fig. 5C(a)
Final Profile - MBFCT