MATHEMATICS OF COMPUTATION
Volume 66, Number 218, April 1997, Pages 623–650
S 0025-5718(97)00823-5

ANALYSIS OF AN ALGORITHM FOR GENERATING LOCALLY OPTIMAL MESHES FOR L_2 APPROXIMATION BY DISCONTINUOUS PIECEWISE POLYNOMIALS

Y. TOURIGNY AND M. J. BAINES

Abstract. This paper discusses the problem of constructing a locally optimal mesh for the best L_2 approximation of a given function by discontinuous piecewise polynomials. In the one-dimensional case, it is shown that, under certain assumptions on the approximated function, Baines' algorithm [M. J. Baines, *Math. Comp.*, 62 (1994), pp. 645-669] for piecewise linear or piecewise constant polynomials produces a mesh sequence which converges to an optimal mesh. The rate of convergence is investigated. A two-dimensional modification of this algorithm is proposed in which both the nodes and the connection between the nodes are self-adjusting. Numerical results in one and two dimensions are presented.

1. Introduction

The interest in piecewise polynomial approximation on non-uniform meshes has grown enormously in the last two decades. The motivation is obvious: by allowing *arbitrary* rather than just uniform or quasi-uniform meshes, many more functions can be approximated to a high order by piecewise polynomials [14]. This is of special relevance to the numerical analysis of partial differential equations by finite element or finite volume methods: even if the solution is quite rough, good results can often be obtained by carefully adjusting the mesh [1], [13].

In the one-dimensional case, the concept of equidistribution [19], [5], [11], [17] has proved useful in constructing good meshes. However, in spite of recent advances [6], [7], [18], the extension of this idea to higher-dimensional domains with a non-trivial geometry remains problematic.

Another approach is to construct the mesh by seeking to minimize the error functional directly. This has been investigated by several authors in the context of continuous piecewise polynomial finite element approximations (cf. Delfour et al. [12] and the references therein). The main disadvantages of this approach are: firstly, the complexity of the resulting (nonlinear) optimization problem; secondly, the possibility (and, in higher dimension, common occurrence) of mesh tangling. D'Azevedo and Simpson [9], [10], [21] consider the simpler case of a given function defined on the whole plane and present an elegant solution of the optimization problem, based on geometrical considerations, which avoids those difficulties. The

Received by the editor July 27, 1995 and, in revised form, March 13, 1996.

1991 *Mathematics Subject Classification.* Primary 41A30; Secondary 65D15.

Key words and phrases. L_2 approximation, discontinuous piecewise polynomials, adjustable nodes, grid generation, triangulation.

impact of their findings on numerical practice in the case of a bounded domain is still unclear.

In many applications, a *continuous* piecewise polynomial approximation is required. However, it has been noted by some authors that the best L_2 approximation by *discontinuous* piecewise linear polynomials on the optimal mesh is often continuous (for instance in the one-dimensional case of a convex function [3]) or “nearly continuous” [20]. On the basis of this observation, and with a view to making the optimization problem as *local* as possible, Baines [2] recently promoted the use of discontinuous piecewise polynomials as the basic approximation tool. Baines described an iterative algorithm which, for a mesh with a fixed number of nodes and a given connectivity between the nodes, seeks the nodal positions which provide the best L_2 approximation to a given function by piecewise constant or piecewise linear polynomials. Loosely speaking, each iteration of the algorithm consists of two stages. In the first, linear stage, the function is projected onto the finite-dimensional space induced by the current mesh. In the second, nonlinear stage, the nodes are updated by requiring that some “approximate” gradient of the error vanish. The remarkable feature of the algorithm is that the second stage, which involves the solution of nonlinear systems of size equal to the domain's dimension, is generally much cheaper than the first stage. In terms of computational cost, we may therefore think of each iteration as being dominated by the linear projection stage.

This paper is primarily devoted to the analysis and development of Baines' algorithm. We first show that this iterative procedure reduces the error monotonically. For piecewise constant or piecewise linear polynomials in one dimension, under some assumptions concerning the function to approximate, we can use the monotonicity property to prove that the mesh sequence converges to the optimal mesh. For the many-dimensional case, we propose a modification of Baines' algorithm which adjusts the nodes in such a way as to retain the monotonicity property. Combining this algorithm with Lawson's local optimization procedure [15], [16] for connecting the nodes, we obtain an effective tool for the computation of locally optimal triangulations.

The remainder of the paper is structured as follows: in §2, we introduce some notation, describe the one-dimensional version of the algorithm and demonstrate the monotonic behaviour of the error. Convergence results are proved in §3 for piecewise linear polynomials. The basic ingredients of the proof extend easily to the piecewise constant case and the corresponding results are stated. In §4, we examine the behaviour of the algorithm as the number of iterations increases. It turns out that the convergence slows down once the residual (the gradient of the error at the current mesh) becomes dominated by the “low-frequency modes” (particular eigenvectors of the hessian at the optimal mesh). A strategy is suggested to speed up the convergence and numerical results are presented. We also study numerically the order of convergence of the best L_2 approximation on the computed meshes (as the number of nodes increases) in the case where the given function cannot be approximated to an optimal order on uniform meshes. The results are compared with those obtained by use of equidistributing meshes. This confirms the well-known fact [19] that, for a correct choice of the “distribution function”, equidistribution yields an optimal mesh as the number of nodes tends to infinity. The two-dimensional modification of Baines' algorithm is presented in §5. In §6, a description of Lawson's procedure for optimizing the connectivity between the nodes is given. We also suggest a recipe for removing redundant nodes and elements when mesh tangling would otherwise occur. In §7, numerical results are discussed for the approximation of some interesting "test" functions defined on the unit square. Finally, §8 is devoted to a summary of our conclusions.

The calculations reported in this paper were carried out on a Silicon Graphics Irix machine purchased with the EPSRC grant GR/J/75258. We are grateful to Chris Budd and Andy Wathen for generously providing access to this resource.

2. The algorithm in one dimension: basic properties

Let f be a given *continuous* function defined on the interval $[0,1]$. In this section, we shall describe an algorithm for finding a locally optimal L_2 approximation of f by piecewise polynomials with free knots.

We begin by introducing some notation. For $N \in \mathbb{N}$, let Δ be a partition of $[0,1]$ with N interior points, i.e.

$$\Delta : \quad 0 = x_0 < x_1 < \ldots < x_N < x_{N+1} = 1.$$

The set of all such partitions will be denoted by $\mathcal{T}_N$. Let $K_j = (x_{j-1}, x_j)$ and, for a positive integer r, let $S_N^r(\Delta)$ be the subset of $L_2[0,1]$ consisting of those functions which, when restricted to the subintervals K_j, are polynomials of degree less than r. The L_2 projection of f onto $S_N^r(\Delta)$ is the unique element $u = u(\Delta) \in S_N^r(\Delta)$ which, for $j = 1, \ldots, N+1$ and $s = 0, \ldots, r-1$, satisfies

$$\int_{K_j} (u - f)\, x^s \, dx = 0. \tag{2.1}$$

Our aim is to find the partition in $\mathcal{T}_N$ which minimizes the error functional

$$E(\Delta) = \| \, u(\Delta) - f \, \|^2,$$

where $\| \cdot \|$ denotes the $L_2[0,1]$ norm. It is known [3] that, save for the trivial case where f is a polynomial of degree less than r, the optimal partition Δ^{opt} satisfies

$$E'(\Delta^{opt}) = 0, \tag{2.2}$$

where E' is the gradient of the error. More precisely, E' is the mapping from $\mathcal{T}_N$ to $\mathbb{R}^N$ with components

$$\begin{aligned} E'_j(\Delta) = \frac{\partial E}{\partial x_j}(\Delta) &= [u(x_j-) - f(x_j)]^2 - [u(x_j+) - f(x_j)]^2 \\ &= [u(x_j-) - u(x_j+)]\,[u(x_j-) + u(x_j+) - 2f(x_j)]\,. \end{aligned}$$

It is obvious from the last equality that, at each node of the optimal mesh, the best approximation is either continuous or else the average of its right and left limits equals f. For instance, in the special case where $f(x) = x^r$, we obtain

$$u(x_j-) = f(x_j) - \left(\frac{1}{2}h_j\right)^r L_r(1) \quad \text{and} \quad u(x_j+) = f(x_j) - \left(\frac{1}{2}h_{j+1}\right)^r L_r(-1),$$

where, $h_j = x_j - x_{j-1}$ and L_r is the monic Legendre polynomial of degree r. Note that L_r has the same parity as r. The uniform mesh is therefore a stationary point of the error functional: u is continuous for r even, and has the averaging property for r odd.

In the general case, we shall seek to solve eq. (2.2) iteratively by constructing a sequence $\{\Delta^n\}_{n\in\mathbb{N}} \subset \mathcal{T}_N$ with nodes x_j^n, subintervals K_j^n and corresponding projections $u^n = u(\Delta^n) \in S_N^r(\Delta^n)$. For $j = 1, \ldots, N$, it is convenient to introduce the functions $\Phi_j^n : [x_{j-1}^n, x_{j+1}^n] \to \mathbb{R}$ defined by

$$\Phi_j^n(x) = \left[\sum_{s=0}^{r-1} \frac{1}{s!}(x - x_j^n)^s \frac{d^s u^n}{dx^s}(x_j^n-) - f(x)\right]^2 - \left[\sum_{s=0}^{r-1} \frac{1}{s!}(x - x_j^n)^s \frac{d^s u^n}{dx^s}(x_j^n+) - f(x)\right]^2 .$$

Note that $\Phi_j^n(x_j^n) = E_j'(\Delta^n)$. The scalar $\Phi_j^n(x)$ may therefore be viewed as an approximation, obtained by "extrapolation" in the interval (x_{j-1}^n, x_{j+1}^n), of the jth component of the error gradient evaluated at the mesh obtained from Δ^n by substituting x for x_j^n.

By eq. (2.1), the restriction of $u^n - f$ to K_j^n is a continuous function with at least $\min(r, 2)$ zeroes in K_j^n. Let $\underline{x}_j^n$ be the zero of this function in the closure of K_j^n which is nearest to x_j^n. Likewise, we denote by $\overline{x}_j^n$ the zero of $u^n - f$ in the closure of K_{j+1}^n which is nearest to x_j^n. It follows from those definitions that

$$\Phi_j^n(\underline{x}_j^n) \le 0 \quad \text{and} \quad \Phi_j^n(\overline{x}_j^n) \ge 0. \tag{2.3}$$

The algorithm for obtaining the mesh Δ^{n+1} from Δ^n may then be described as follows:

Algorithm 1 (Baines)**.** 1. (Linear stage). Project f onto $S_N^r(\Delta^n)$ and obtain u^n.

2. (Nonlinear stage). For each j from 1 to N, there are three possibilities:

(a) $\Phi_j^n(x_j^n) = 0$. In this case, set $x_j^{n+1} = x_j^n$.

(b) $\Phi_j^n(x_j^n) > 0$, so that Φ_j^n has a zero in $[\underline{x}_j^n, x_j^n)$. Let x_j^{n+1} be the zero, in that interval, which is nearest to x_j^n.

(c) $\Phi_j^n(x_j^n) < 0$, so that Φ_j^n has a zero in $(x_j^n, \overline{x}_j^n]$. Let x_j^{n+1} be the zero, in that interval, which is nearest to x_j^n.

With this choice of x_j^{n+1}, we have

$$x_j^{n+1} \le \overline{x}_j^n \le \underline{x}_{j+1}^n \le x_{j+1}^{n+1}.$$

For $r > 1$, the second inequality is strict and therefore mesh tangling cannot occur. For $r = 1$, the equalities

$$x_j^{n+1} = \overline{x}_j^n = \underline{x}_{j+1}^n = x_{j+1}^{n+1}$$

imply

$$u^n(x_j^n-) = f(\overline{x}_j^n) = u^n(x_j^n+)$$

and therefore $\Phi_j^n(x_j^n) = 0$. This yields $x_j^{n+1} = x_j^n$ and we obtain the contradiction $\overline{x}_j^n = x_j^n < \underline{x}_{j+1}^n$. We conclude that Δ^{n+1} does indeed belong to $\mathcal{T}_N$.

Note that this particular way of defining x_j^{n+1} yields the inequality

$$(x_j^{n+1} - x_j^n)\, \frac{\partial E}{\partial x_j}(\Delta^n) \le 0, \tag{2.4}$$

with equality iff the partial derivative of the error vanishes. We may thus think of the algorithm as a gradient method with "optimal" step. The attractive feature is that, at each iteration, the nodes can be updated without the need for intermediate local projections. The above inequality has the following important consequence:

Theorem 2.1.

$$E(\Delta^{n+1}) \leq E(\Delta^n)$$

with equality iff $E'(\Delta^n) = 0$.

Proof. We introduce the one-parameter family of partitions $\{\Delta(t)\}_{0\leq t\leq 1} \subset \mathcal{T}_N$ where

$$x_j(t) = tx_j^{n+1} + (1-t)x_j^n.$$

For each $t \in [0,1]$, define $\hat{u}(\cdot,t) \in S_N^r(\Delta(t))$ uniquely by means of the relations

$$\hat{u}(x,t) = \sum_{s=0}^{r-1} \frac{1}{s!}(x - x_j^n)^s \frac{d^s u^n}{dx^s}(x_j^n -) \quad \text{for } x \in K_j(t).$$

Note that

$$\hat{u}(\cdot,0) = u^n \in S_N^r(\Delta^n) \quad \text{and} \quad \hat{u}(\cdot,1) \in S_N^r(\Delta^{n+1}).$$

We may write

$$E(\Delta^{n+1}) - E(\Delta^n) = \| u^{n+1} - f \|^2 - \| \hat{u}(\cdot,1) - f \|^2 + \int_0^1 \frac{d}{dt} \| \hat{u}(\cdot,t) - f \|^2 \, dt.$$

By definition of u^{n+1}, we have

$$\| u^{n+1} - f \|^2 \leq \| \hat{u}(\cdot,1) - f \|^2.$$

Hence

$$E(\Delta^{n+1}) - E(\Delta^n) \leq \int_0^1 \sum_{j=1}^{N+1} \frac{d}{dt} \int_{K_j(t)} [\hat{u}(x,t) - f(x)]^2 \, dx \, dt. \tag{2.5}$$

Now, since $\partial\hat{u}/\partial t$ obviously vanishes for $x \in K_j(t)$, we obtain

$$\begin{aligned} \frac{d}{dt} \int_{K_j(t)} [\hat{u}(x,t) - f(x)]^2 \, dx = {} & [\hat{u}(x_j(t)-) - f(x_j(t))]^2 \, \dot{x}_j(t) \\ & - [\hat{u}(x_{j-1}(t)+,t) - f(x_{j-1}(t))]^2 \, \dot{x}_{j-1}(t), \end{aligned}$$

where the dot indicates differentiation with respect to t. Eq. (2.5) then yields

$$\begin{aligned} & E(\Delta^{n+1}) - E(\Delta^n) \\ & \quad \leq \sum_{j=1}^{N} \int_0^1 \left\{ [\hat{u}(x_j(t)-,t) - f(x_j(t))]^2 - [\hat{u}(x_j(t)+,t) - f(x_j(t))]^2 \right\} \dot{x}_j(t) \, dt. \end{aligned}$$

Recalling the definitions of $\hat{u}$ and Φ_j^n, we find that

$$E(\Delta^{n+1}) - E(\Delta^n) \leq \sum_{j=1}^{N} (x_j^{n+1} - x_j^n) \int_0^1 \Phi_j^n(tx_j^{n+1} + (1-t)x_j^n) \, dt. \tag{2.6}$$

By construction, there is no zero of Φ_j^n between x_j^n and x_j^{n+1}. The theorem is then a consequence of the inequality (2.4) □

It follows immediately from this theorem that the sequence $\{E(\Delta^n)\}_{n\in\mathbb{N}}$ converges. Further, since, in the inequality (2.6), each individual term in the sum is non-positive, we easily obtain the

Corollary 2.2. *For each $j = 1, \dots, N$,*

$$\lim_{n\to\infty} (x_j^{n+1} - x_j^n) \int_0^1 \Phi_j^n(tx_j^{n+1} + (1-t)x_j^n)\,dt = 0.$$

3. Approximation by piecewise linear and piecewise constant polynomials in one dimension

Theorem 2.1 and its corollary hold for any $f \in C[0,1]$ and any $r \in \mathbb{N}$. From the theoretical point of view, the outstanding questions are: (1) does the sequence $\{\Delta^n\}_{n\in\mathbb{N}}$ converge to a limit in $\mathcal{T}_N$? (2) If so, is the limit a stationary point of the error? Is it a minimum or, at least, a local minimum? In this section, we shall provide a partial answer to those questions in the special case where $r = 1$ or 2 and the derivative $\frac{d^r f}{dx^r}$ is bounded away from zero. More precisely, we shall prove the following

Theorem 3.1. *For $r = 1$ or 2, let $f \in C^r[0,1]$. Assume that there exist two constants m and M such that*

$$0 < m \le f^{(r)}(x) \le M < \infty \quad \forall x \in [0,1].$$

Then $\{\Delta^n\}_{n\in\mathbb{N}}$ has a limit point $\Delta^* \in \mathcal{T}_N$ such that $E'(\Delta^*) = 0$.

With some additional assumptions, we can elucidate the nature of the stationary point and the type of convergence.

Theorem 3.2. *In addition to the hypothesis of Theorem 3.1, assume either that*

1. $f \in C^{r+1}[0,1]$ and $f^{(r+1)}/f^{(r)}$ is non-increasing

or

2. $f \in C^{r+3}[0,1]$ and N is sufficiently large.

Then the entire sequence $\{\Delta^n\}_{n\in\mathbb{N}}$ converges to a global minimum $\Delta^{opt} \in \mathcal{T}_N$. Further, there exists two constants $0 < \alpha_N < 1$ and $C_N > 0$ independent of n such that

$$\max_{1\le j\le N} |\, x_j^n - x_j^{opt}| \le C_N\,(\alpha_N)^n,$$

where x_j^{opt} is the jth node of Δ^{opt}.

Even under such strong assumptions, the proofs are somewhat intricate. Before addressing the technical details, it is helpful to sketch out the basic ingredients.

For the first theorem, the proof's main inspiration is the analysis of the gradient method (with optimal step) for the minimization of a quadratic functional [8]. We use the relation

$$(3.1) \qquad (x_j^{n+1} - x_j^n) \int_0^1 \Phi_j^n(tx_j^{n+1} + (1-t)x_j^n)\,dt = \int_{x_j^n}^{x_j^{n+1}} (x_j^n - x)\frac{d\Phi_j^n}{dx}(x)\,dx,$$

together with the fact that $d\Phi_j^n/dx$ is strictly positive, to conclude that

$$\lim_{n\to\infty} (x_j^{n+1} - x_j^n) = 0.$$

We can invoke a compactness argument to extract from $\{\Delta^n\}_{n\in\mathbb{N}}$ a subsequence which converges and use the above result to show that the limit is a stationary point of the error. The unusual conditions in the second theorem ensure that there is only one stationary point [3]. Those conditions also yield favourable bounds on the eigenvalues of the error hessian, and the inequality in the theorem follows easily.

We shall carry out this program in detail only for the case $r = 2$, which corresponds to piecewise linear approximation. The case $r = 1$ can be treated in essentially the same way (the calculations involved are in fact simpler). Hence, for the remainder of this section, let $r = 2$. In this case, the algorithm reduces to the simple recurrence relation

$$x_j^{n+1} - x_j^n = \frac{u^n(x_j^n-) - u^n(x_j^n+)}{\frac{du^n}{dx}(x_j^n+) - \frac{du^n}{dx}(x_j^n-)}. \tag{3.2}$$

The proof of the following lemma follows easily by mapping the subinterval to $[0, 1]$, using the convexity of f and the definition of u^n.

Lemma 3.3. *Under the hypothesis of Theorem 3.1, the function $u^n - f$ has exactly two interior zeroes $\overline{x}_{j-1}^n < \underline{x}_j^n$ in K_j^n. We have*

$$u^n - f > 0 \ \ in\ (\overline{x}_{j-1}^n, \underline{x}_j^n) \quad and \quad u^n - f < 0 \ \ in\ (x_{j-1}^n, \overline{x}_{j-1}^n) \cup (\underline{x}_j^n, x_j^n).$$

Further, there exists a constant c, depending only on m and M, such that

$$\underline{x}_j^n - \overline{x}_{j-1}^n \geq c h_j^n,$$

where $h_j^n = x_j^n - x_{j-1}^n$.

Lemma 3.4. *Under the hypothesis of Theorem 3.1, we have*

$$\lim_{n\to\infty} \left(x_j^{n+1} - x_j^n\right) = 0 \quad for \quad j = 1, \dots, N.$$

Proof. Assume the contrary. Then, there exists $1 \leq j \leq N$, $\epsilon > 0$ and a subsequence $\{x_j^{n_k}\}_{k\in\mathbb{N}} \subset \{x_j^n\}_{n\in\mathbb{N}}$ such that

$$\left| x_j^{n_k+1} - x_j^{n_k} \right| \geq \epsilon \quad \forall k \in \mathbb{N}. \tag{3.3}$$

There are two possibilities:

$$\underline{1}\ \ x_j^{n_k+1} - x_j^{n_k} \geq \epsilon \quad \text{or} \quad \underline{2}\ \ x_j^{n_k+1} - x_j^{n_k} \leq -\epsilon.$$

To simplify the notation, let us drop the subscript k. Consider the first possibility. In this case, since $x_j^{n+1} \in (x_j^n, \overline{x}_j^n]$, we have

$$h_{j+1}^n \geq \epsilon. \tag{3.4}$$

Now, for $x \in [x_j^n, x_j^{n+1}]$, we can write

$$\begin{aligned}\frac{d\Phi_j^n}{dx}(x) = 2 &\left[u^n(x_j^n-) + (x - x_j^n)\frac{du^n}{dx}(x_j^n-) - f(x)\right] \left[\frac{du^n}{dx}(x_j^n-) - f'(x)\right] \\ &- 2\left[u^n(x) - f(x)\right] \left[\frac{du^n}{dx}(x_j^n+) - f'(x)\right].\end{aligned} \tag{3.5}$$

For $x \in (\underline{x}_j^n, \overline{x}_j^n)$, we can use the mean-value theorem and the fact that f' is monotonically increasing to obtain

$$f'(x) \geq \frac{f(\underline{x}_j^n) - f(\overline{x}_{j-1}^n)}{\underline{x}_j^n - \overline{x}_{j-1}^n} = \frac{u^n(\underline{x}_j^n) - u^n(\overline{x}_{j-1}^n)}{\underline{x}_j^n - \overline{x}_{j-1}^n} = \frac{du^n}{dx}(x_j^n-).$$

Thus,

$$f(x) - u^n(x_j^n-) - (x - x_j^n)\frac{du^n}{dx}(x_j^n-) = f(\underline{x}_j^n) - u^n(x_j^n-) - (\underline{x}_j^n - x_j^n)\frac{du^n}{dx}(x_j^n-)$$
$$+ \int_{\underline{x}_j^n}^{x} [f'(s) - \frac{du^n}{dx}(x_j^n-)]\, ds = \int_{\underline{x}_j^n}^{x} [f'(s) - \frac{du^n}{dx}(x_j^n-)]\, ds \geq 0,$$

for $x \in (\underline{x}_j^n, \overline{x}_j^n)$. We conclude from (3.5) that

$$\frac{d\Phi_j^n}{dx}(x) \geq -2\,[u^n(x) - f(x)] \left[\frac{du^n}{dx}(x_j^n+) - f'(x)\right].$$

Now,

$$\left[\frac{du^n}{dx}(x_j^n+) - f'(x)\right] = \frac{1}{\underline{x}_{j+1}^n - \overline{x}_j^n} \int_{\overline{x}_j^n}^{\underline{x}_{j+1}^n} \int_x^s f''(\sigma)\, d\sigma\, ds$$
$$\geq \frac{m}{2}(\underline{x}_{j+1}^n - \overline{x}_j^n) \geq \frac{m}{2} c h_{j+1}^n \geq \frac{m}{2} c\epsilon,$$

where we have used Lemma 3.3 and eq. (3.4). Hence, for $x_j^n \leq x \leq x_j^{n+1} \leq \overline{x}_j^n$,

$$\frac{d\Phi_j^n}{dx}(x) \geq mc\epsilon\,[f(x) - u^n(x)]\,.$$

On the other hand, we may write

$$f(x) - u^n(x) = \frac{1}{2}(\overline{x}_j^n - x)(\underline{x}_{j+1}^n - x) f''(\xi_x),$$

where ξ_x lies in the smallest interval containing x, $\overline{x}_j^n$ and $\underline{x}_{j+1}^n$. Thus, by eq. (3.1) and Lemma 3.3,

$$-\,(x_j^{n+1} - x_j^n) \int_0^1 \Phi_j^n(t x_j^{n+1} + (1-t)x_j^n) dt = \int_{x_j^n}^{x_j^{n+1}} (x - x_j^n) \frac{d\Phi_j^n}{dx}(x)\, dx$$
$$\geq \frac{1}{2} m^2 c\epsilon\, (\underline{x}_{j+1}^n - \overline{x}_j^n) \int_{x_j^n}^{x_j^{n+1}} (x - x_j^n)(\overline{x}_j^n - x)\, dx$$
$$\geq \frac{1}{2} m^2 c^2 \epsilon^2 \int_{x_j^n}^{x_j^{n+1}} (x - x_j^n)(x_j^{n+1} - x)\, dx.$$

We conclude that

$$(3.6) \qquad \frac{1}{12} m^2 c^2 \epsilon^2 \left| x_j^{n+1} - x_j^n \right|^3 \leq -(x_j^{n+1} - x_j^n) \int_0^1 \Phi_j^n(t x_j^{n+1} + (1-t)x_j^n)\, dt.$$

If we consider the second possibility instead, namely $x_j^{n+1} - x_j^n \leq -\epsilon$, then $h_j^n \geq \epsilon$. By a similar calculation, we arrive once again at the inequality (3.6). However, by Corollary 2.2, the right-hand side tends to zero as $n \to \infty$. This contradicts the assumption (3.3) and the proof is complete □

Proof of Theorem 3.1. With these lemmata at our disposal, the proof of Theorem 3.1 is now straightforward. Using the definition of u^n and integration by parts, we may deduce [3]

$$
\begin{aligned}
u^n(x_j^n-) - u^n(x_j^n+) = \left(h_{j+1}^n\right)^2 \int_0^1 f''(x_j^n + sh_{j+1}^n)\, s(1-s)^2\, ds \\
- \left(h_j^n\right)^2 \int_0^1 f''(x_{j-1}^n + sh_j^n)\, s^2(1-s)\, ds
\end{aligned}
\tag{3.7}
$$

and

$$
\begin{aligned}
\frac{du^n}{dx}(x_j^n+) - \frac{du^n}{dx}(x_j^n-) = h_{j+1}^n \int_0^1 f''(x_j^n + sh_{j+1}^n)\, (1-s)^2(2s+1)\, ds \\
+ h_j^n \int_0^1 f''(x_{j-1}^n + sh_j^n)\, s^2(3-2s)\, ds.
\end{aligned}
\tag{3.8}
$$

Since the sequence $\{x_j^n\}_{n\in\mathbb{N}}$ lies in the compact interval $[0,1]$, we can extract a subsequence $\{x_j^{n_k}\}_{k\in\mathbb{N}}$ which converges to some $x_j^* \in [0,1]$. N is a fixed number, so we can use the same subscript k for all $j = 0, \dots, N+1$ and conclude that

$$\lim_{k\to\infty} \max_{0\le j\le N+1} \left| x_j^{n_k} - x_j^* \right| = 0.$$

By construction, $h_j^{n_k} > 0$ and thus $h_j^* = x_j^* - x_{j-1}^* \ge 0$. We will now show that $h_j^* > 0$.

Replacing n by n_k in eq. (3.2) and using Lemma 3.4, we may take the limit as $k \to \infty$ to obtain, for $j = 1, \dots, N$,

$$\left(h_{j+1}^*\right)^2 \int_0^1 f''(x_j^* + sh_{j+1}^*)\, s(1-s)^2\, ds = \left(h_j^*\right)^2 \int_0^1 f''(x_{j-1}^* + sh_j^*)\, s^2(1-s)\, ds. \tag{3.9}$$

Note that this equality is valid even if h_{j+1}^* or h_j^* vanishes. Since

$$\sum_{j=1}^{N+1} h_j^* = \lim_{k\to\infty} \sum_{j=1}^{N+1} h_j^{n_k} = 1,$$

$h_j^* > 0$ for at least one j. Because $0 < m \le f''(x) \le M < \infty$, it follows from eq. (3.9) that none of the h_j^*'s can vanish. Hence, if we let Δ^* denote the partition with the nodes x_j^*, we have proved that the sequence $\{\Delta^n\}_{n\in\mathbb{N}}$ has a limit point $\Delta^* \in \mathcal{T}_N$.

To complete the proof, let $u^* = u(\Delta^*) \in S_N^2(\Delta^*)$ denote the projection of f. By (3.7) and (3.9), u^* is continuous and Δ^* is therefore a stationary point of E □

Proof of Theorem 3.2. The proof of Theorem 3.1 uses a compactness argument. If there is only one stationary point of the error functional E, then the entire sequence $\{\Delta^n\}_{n\in\mathbb{N}}$ must converge to it. Barrow et al. [3] showed that the conditions in the hypothesis are sufficient for the uniqueness of the stationary point.

In order to establish the rate of convergence, we shall use the notation $\mathbf{x} = (x_1, \dots, x_N)$ to denote the vector in $\mathbb{R}^N$ with components equal to the interior nodes of the partition $\Delta \in \mathcal{T}_N$. In the neighbourhood of $\mathbf{x}^{opt}$, we can define a mapping Φ with values in $\mathbb{R}^N$ and jth component

$$\Phi_j(\mathbf{x}) = \frac{u(x_j-) - u(x_j+)}{\frac{du}{dx}(x_j+) - \frac{du}{dx}(x_j-)},$$

where $u = u(\Delta)$. The right-hand side of the above equality may be expressed directly in terms of $\mathbf{x}$ by removing the superscript n in (3.7) and (3.8). Note that

$\Phi(\mathbf{x}^{opt}) = \mathbf{0}$. Since Φ is a smooth mapping in a sufficiently small neighbourhood of $\mathbf{x}^{opt}$, the theorem will be proved if we can show that

$$\| I + \Phi'(\mathbf{x}^{opt}) \|_2 := \sup_{|\mathbf{x}| \le 1} |\, \mathbf{x} + \Phi'(\mathbf{x}^{opt})\mathbf{x} \,| < 1, \tag{3.10}$$

where $|\cdot|$ is the usual euclidean norm in $\mathbb{R}^N$ and $\Phi'(\mathbf{x}^{opt})$ is the jacobian matrix of Φ at $\mathbf{x}^{opt}$. Upon calculation, we find that

$$\Phi'(\mathbf{x}^{opt}) = \operatorname{diag}(d_j^{-1})\, A,$$

where

$$\begin{aligned} d_j =& h_{j+1}^{opt} \int_0^1 f''(x_j^{opt} + s h_{j+1}^{opt})\,(1-s)^2(2s+1)\,ds \\ &+ h_j^{opt} \int_0^1 f''(x_{j-1}^{opt} + s h_j^{opt})\, s^2(3-2s)\,ds \end{aligned}$$

and A is the tridiagonal matrix with entries

$$a_{j,j-1} = 2\, h_j^{opt} \int_0^1 f''(x_{j-1}^{opt} + s h_j^{opt})\, s(1-s)^2\,ds,$$

$$\begin{aligned} a_{j,j} = h_{j+1}^{opt} \int_0^1 f''(x_j^{opt} \;+\; s h_{j+1}^{opt})\,(1-s)^2(2s-1)\,ds \\ -\; h_j^{opt} \int_0^1 f''(x_{j-1}^{opt} \;+\; s h_j^{opt})\, s^2(2s-1)\,ds \end{aligned}$$

and

$$a_{j,j+1} = 2\, h_{j+1}^{opt} \int_0^1 f''(x_j^{opt} \;+\; s h_{j+1}^{opt})\, s^2(1-s)\,ds.$$

Note that $a_{j,j-1}$, d_j and $a_{j,j+1}$ are all positive. It follows that A and $\Phi'(\mathbf{x}^{opt})$ are tridiagonal quasi-symmetric matrices [22]. Such matrices are diagonalizable and have real eigenvalues. Thus, in order to establish (3.10), it will suffice to show that the eigenvalues λ of $\Phi'(\mathbf{x}^{opt})$ satisfy

$$-2 < \lambda < 0. \tag{3.11}$$

Let us observe also that the hessian matrix H of the error functional E at $\mathbf{x}^{opt}$ is given by

$$H = 2\operatorname{diag}(u_j^{opt} - f(x_j^{opt}))\, A,$$

where $u_j^{opt} = u^{opt}(x_j^{opt}+) = u^{opt}(x_j^{opt}-)$. We have $u_j^{opt} - f(x_j^{opt}) < 0$, so that $\Phi'(\mathbf{x}^{opt}) = \Sigma\, H$, where Σ is a diagonal matrix with negative diagonal entries. The hypothesis implies that Δ^{opt} is a minimum of the error functional, so that H is positive definite. We conclude that $\Phi'(\mathbf{x}^{opt})$ is negative definite. This proves the second inequality in (3.11).

For the first inequality, we shall use Gerschgorin's theorem. We have

$$\begin{aligned} a_{j,j} + d_j = 4\, h_{j+1}^{opt} \int_0^1 f''(x_j^{opt} + s h_{j+1}^{opt})\,(1-s)^2 s\,ds \\ +\; 4\, h_j^{opt} \int_0^1 f''(x_{j-1}^{opt} + s h_j^{opt})\, s^2(1-s)\,ds \ge \frac{1}{3} m(h_{j+1}^{opt} + h_j^{opt}) \end{aligned}$$

and

$$\frac{1}{2}m(h_{j+1}^{opt}+h_j^{opt}) \le d_j \le \frac{1}{2}M(h_{j+1}^{opt}+h_j^{opt}).$$

Thus

$$\frac{a_{j,j}}{d_j} \ge -1 + \frac{2}{3}\frac{m}{M}$$

and Gerschgorin's theorem yields

$$\begin{aligned}\lambda \ge \frac{1}{d_j}(a_{j,j}-a_{j,j-1}-a_{j,j+1}) &= 2\frac{a_{j,j}}{d_j} - \frac{1}{d_j}(a_{j,j-1}+a_{j,j}+a_{j,j+1})\\ &\ge -2 + \frac{4}{3}\frac{m}{M} - \frac{1}{d_j}(a_{j,j-1}+a_{j,j}+a_{j,j+1}).\end{aligned}$$

Integration by parts leads to

$$\begin{aligned}a_{j,j-1}+a_{j,j}+a_{j,j+1} &= \left(h_{j+1}^{opt}\right)^2\int_0^1 f'''(x_j^{opt}+sh_{j+1}^{opt})\,s(1-s)^2\,ds\\ &\quad - \left(h_j^{opt}\right)^2\int_0^1 f'''(x_{j-1}^{opt}+sh_j^{opt})\,s^2(1-s)\,ds.\end{aligned} \tag{3.12}$$

We now use the fact that

$$\begin{aligned}&\left(h_{j+1}^{opt}\right)^2\int_0^1 f''(x_j^{opt}+sh_{j+1}^{opt})\,s(1-s)^2\,ds\\ &\qquad = \left(h_j^{opt}\right)^2\int_0^1 f''(x_{j-1}^{opt}+sh_j^{opt})\,s^2(1-s)\,ds.\end{aligned} \tag{3.13}$$

If f'''/f'' is non-increasing, it follows easily that $a_{j,j-1}+a_{j,j}+a_{j,j+1} \le 0$. Then (3.12) yields

$$\lambda \ge -2 + \frac{4}{3}\frac{m}{M}$$

and the first inequality in (3.11) holds. If, on the other hand, $f \in C^5[0,1]$ and N is large enough, then (3.13) implies that Δ^{opt} is a quasi-uniform partition [3]. By using Taylor's theorem, we readily obtain from (3.12) that

$$\lambda \ge -2 + \frac{4}{3}\frac{m}{M} - C\,N^{-1}$$

for some constant C independent of N. The first inequality in (3.11) therefore holds for N large enough. The proof is complete □

4. The rate of convergence. Numerical results in one dimension

The previous theorem suggests that the mesh sequence generated by the algorithm converges geometrically. In order to gain further insight, let us consider the simple case in which a quadratic function f is approximated by piecewise linear polynomials. The algorithm then reduces to

$$x_j^{n+1} - x_j^n = \frac{1}{6}\left(x_{j+1}^n - 2x_j^n + x_{j-1}^n\right),$$

where $x_0^n = 0$ and $x_{N+1}^n = 1\ \forall n$. The optimal mesh is uniform. In vector form, the difference $\mathbf{e}^n = \mathbf{x}^n - \mathbf{x}^{opt}$ therefore satisfies

$$\mathbf{e}^n = (I + \Phi'(\mathbf{x}^{opt}))^n\mathbf{e}^0,$$

where I is the identity matrix and $\Phi'(\mathbf{x}^{opt}) = \frac{1}{6}\text{tridiag}(1, -2, 1)$. The eigenvalues of $\Phi'(\mathbf{x}^{opt})$ are

$$\lambda_k = -\frac{2}{3}\sin^2\frac{k\pi}{2(N+1)}$$

for $k = 1, \dots, N$, and the corresponding (normalized) eigenvectors $\mathbf{v}^k$ have components

$$v_j^k = \left(\frac{2}{N+1}\right)^{\frac{1}{2}} \sin\frac{jk\pi}{N+1}.$$

Hence, we may write

$$\mathbf{e}^n = \sum_{k=1}^{N} (1+\lambda_k)^n c_k^0 \mathbf{v}^k, \tag{4.1}$$

where $c_k^0 = (\mathbf{e}^0, \mathbf{v}^k)$ and $(\cdot,\cdot)$ denotes the usual inner product in $\mathbb{R}^N$.

It follows that, for $N/2 < k \leq N$, the components $(\mathbf{e}^n, \mathbf{v}^k)$ of $\mathbf{e}^n$ are damped adequately as the iteration proceeds. The "smoothing factor", or collective damping rate, of those high-frequency components is

$$\mu = \max_{\pi/2 \leq \theta \leq \pi} |1 - \frac{2}{3}\sin^2(\theta/2)| = \frac{2}{3}.$$

Hence, for the first few iterations, the gradient $E'(\Delta^n)$ of the error functional, or "residual", should decrease rapidly. Soon, however, the low-frequency components (those corresponding to $1 \leq k \leq N/2$) dominate, the residual becomes smooth and the overall convergence deteriorates. In fact, we easily find that

$$\alpha_N = \max_{1 \leq k \leq N} |1 + \lambda_k| = 1 - \frac{2}{3}\sin^2\frac{\pi}{2(N+1)} = 1 - O(N^{-2})$$

as $N \to \infty$. This undesirable behaviour seems typical of gradient methods for mesh optimization and can be observed also in the numerical experiments reported by Delfour et al. [12].

In order to deal with the low-frequency components of the residual, it is natural to consider the possible application of multigrid techniques [4]. The basic idea of multigrid is to remove the low-frequency components by correcting on coarser and coarser grids until those components look sufficiently rough to be eliminated effectively by the basic iteration procedure. In view of the particular case (4.1), we may, in our context, think of the nodal values x_j^n as discrete unknowns defined on a uniform mesh with N interior points. Starting at a "fine grid" level, we might first apply the algorithm to smooth the residual and, when convergence slows down, seek to compute a correction at a coarser level involving only $(N-1)/2$ points. Unfortunately, it is not clear how the algorithm can meaningfully be applied to solve for the correction at the coarser level.

In this section, we shall adopt a different strategy, in which we begin at the coarsest level (corresponding to $N = 1$) and move to finer levels. At each level, Baines' algorithm is used to remove the high-frequency components of the residual. When convergence slows down, Newton's method is attempted. If the error increases, the Newton step is rejected and Baines' iteration resumed. Otherwise, we continue with Newton's method.

The starting mesh for the coarsest level is uniform. After optimization at a given level, the starting mesh for the next finer level is obtained by interpolation.

The motivation for this simple approach is that, at each new level, the difference $\mathbf{e}^0$ has small low-frequency components, so that the first few iterations of Baines' algorithm are particularly effective. Thereafter, Newton's method should be successful, because convergence is at hand. At each Newton iteration, a linear system of equations involving the hessian matrix of the error functional must be solved. The calculation of this matrix requires the evaluation of the first derivative of f. Hence, this approach can, in principle, only be applied to functions in $C^1[0,1]$. As the numerical experiments will demonstrate, however, this requirement may be relaxed in practice. Note that, in the one-dimensional case, the hessian matrix is tridiagonal, so that the resulting system may be solved directly in $O(N)$ operations.

In all the numerical examples reported in this paper, the integrals were computed to machine accuracy by standard quadrature rules. For this section, the size of the residual was measured in the norm

$$|E'(\Delta^n)|_\infty = \max_{1 \le j \le N} \left| \frac{\partial E}{\partial x_j}(\Delta^n) \right|.$$

At each level, a minimum of five Baines iterations were performed. If the ratio η_n of consecutive residual norms exceeded a certain parameter η, Newton's method was attempted. If it failed, five more iterates by Baines' algorithm were computed. The stopping criterion which we have used is

$$|E'(\Delta^n)|_\infty < 10^{-10}.$$

Following Brandt [4], a sensible choice for η is

$$\eta = \frac{1+3\mu}{4},$$

where μ is the smoothing factor. Note that $\mu = 1/2$ for $r = 1$ and $\mu = 2/3$ for $r = 2$. In all cases, we have in fact used $\eta = 3/4$.

Example 4.1.

$$f(x) = \tanh(160(x - \frac{7}{20})) - 2\tanh(160(x - \frac{9}{10})).$$

We approximate this function by piecewise constant polynomials. The results displayed in Table 1 indicate that, while Newton's method is able to drive the norm of the residual to zero very quickly, it is Baines' algorithm which accounts for most of the reduction in the L_2 error.

Example 4.2.

$$f(x) = x - x^{\frac{3}{5}}.$$

We consider the approximation of this strictly convex function by piecewise linear polynomials. Note that $f'' \notin L_2[0,1]$, so that this function does not have the required Sobolev regularity to be approximated to an optimal order on *uniform* meshes as $N \to \infty$. The derivatives of f are in fact unbounded at $x = 0$ and, therefore, the analysis of the previous section does not strictly apply. Nevertheless, for each N, the Baines–Newton procedure described above yields a sequence of meshes which converges to a limit $\Delta^* \in \mathcal{T}_N$. Alongside the corresponding result for the uniform mesh Δ^{uni}, Table 2 shows the error $E(\Delta^*)$ as N increases. The local order of convergence may be defined as

$$\gamma_N = \frac{\log(e_N/e_{2N})}{\log 2},$$

Table 1. Baines and Newton iterations for the best piecewise constant approximation of the function $f(x) = \tanh(160(x - \frac{7}{20})) - 2\tanh(160(x - \frac{9}{10}))$

$N+1$	n	$\lvert E'(\Delta^n)\rvert_\infty$	η_n	$\sqrt{E(\Delta^n)}$
2				
	0	0.13200000D+01	–	0.12796484D+01
	1	0.56282120D+00	0.43	0.11414127D+01
	2	0.58071911D−02	0.01	0.11412472D+01
	5	0.65471711D−08	0.01	0.11412472D+01
	6	0.68172079D−10	0.01	0.11412472D+01
4				
	0	0.15051471D+01	–	0.10320340D+01
	1	0.95839029D+01	6.37	0.22475654D+00
	2	0.53481130D+00	0.06	0.14627594D+00
	5	0.10154120D+00	0.58	0.14136375D+00
	10	0.76304214D−02	0.60	0.14106012D+00
	20	0.52471419D−04	0.61	0.14105798D+00
	40	0.28370944D−08	0.61	0.14105798D+00
	47	0.91710001D−10	0.61	0.14105798D+00
8				
	0	0.29079904D+00	–	0.13626575D+00
	1	0.13604911D+01	4.68	0.11138203D+00
	2	0.51343361D+00	0.38	0.75649228D−01
	5	0.10035160D+00	0.80	0.60005877D−01
Newton	+7	0.11657342D−13	–	0.58937702D−01
16				
	0	0.30122464D+00	–	0.43854817D−01
	1	0.59679981D−01	0.20	0.35114814D−01
	2	0.50296637D−01	0.84	0.31681540D−01
	5	0.31060265D−01	0.83	0.28822527D−01
	10	0.13419761D−01	0.86	0.28015346D−01
Newton	+5	0.98532293D−14	–	0.27753325D−01
32				
	0	0.33734554D−01	–	0.17546228D−01
	1	0.71400207D−02	0.21	0.15206486D−01
	2	0.60026248D−02	0.84	0.14424609D−01
	5	0.31082789D−02	0.82	0.13808575D−01
	10	0.18099425D−02	0.92	0.13657572D−01
Newton	+5	0.83093254D−14	–	0.13545728D−01

where e_N is the L_2 error. The numerical evidence clearly suggests that

$$\sqrt{E(\Delta^*)} = \|u(\Delta^*) - f\| = O(N^{-2}) \quad \text{as } N \to \infty.$$

By contrast, we see that the order of approximation based on the uniform mesh $\Delta^{uni} \in \mathcal{T}_N$ is $\gamma = 11/10$ and, hence, suboptimal.

The algorithm that we have described may be used to evaluate the effectiveness of other approaches to the construction of good meshes. In the one-dimensional

TABLE 2. Decay of the L_2 error as N increases for the approximation of the function $f(x) = x^{\frac{3}{5}} - x$ by piecewise linear polynomials

$N+1$	$\sqrt{E(\Delta^*)}$	γ_N	$\sqrt{E(\Delta^{uni})}$	γ_N	$\sqrt{E(\Delta^{equi})}$	γ_N
2	0.12530788D−01	–	0.18308897D−01	–	0.13519264D−01	–
4	0.36478337D−02	1.78	0.85655148D−02	1.10	0.38607478D−02	1.81
8	0.99265952D−03	1.88	0.39992837D−02	1.10	0.10263518D−02	1.91
16	0.25955179D−03	1.94	0.18661729D−02	1.10	0.26427465D−02	1.96
32	0.66375759D−04	1.97	0.87064854D−03	1.10	0.67033400D−04	1.98
64	0.16796645D−04	1.98	0.40616257D−03	1.10	0.16881148D−04	1.99
128	0.42249409D−05	1.99	0.18938208D−03	1.10	0.42356502D−05	1.99
256	0.10594930D−05	2.00	0.88241833D−04	1.10	0.10608338D−05	2.00

case, McGlure [19] has shown that the mesh Δ^{equi} such that

$$\int_{x_{j-1}^{equi}}^{x_j^{equi}} |f^{(r+1)}(x)|^{\frac{2}{2r+1}}\,dx = \frac{1}{N+1}\int_0^1 |f^{(r+1)}(x)|^{\frac{2}{2r+1}}\,dx$$

for $1 \le j \le N+1$, is optimal asymptotically as $N \to \infty$, for the best L_2 approximation of f by piecewise polynomials of degree less than r. How good is this equidistributing mesh for finite values of N? For the special case considered in Example 4.2, a simple calculation shows that, for $r = 2$, we have

$$x_j^{equi} = \left(\frac{j}{N+1}\right)^{\frac{25}{11}}, \quad 1 \le j \le N+1.$$

The L_2 error for this mesh is shown in Table 2. The results confirm its asymptotic optimality. It is also clear that, even for moderate values of N, the equidistributing mesh would provide an excellent starting mesh for our algorithm.

5. Piecewise polynomials on triangles with self-adjusting nodes

We now discuss the extension to higher dimension of the algorithm described in §2. For simplicity, however, we shall confine our attention to the problem of approximating a given continuous function defined on the unit square $\Omega = (0,1) \times (0,1)$.

A triangulation Δ of Ω is a set of non-overlapping triangles $\{K_i\}_{i=1}^N$ such that no vertex of a triangle lies along the edge of another. For practical purposes, one usually represents Δ by numbering the vertices (nodes) $\{x_j\}_{j=1}^M$ and forming a "connectivity matrix" such that the ith row lists (in counter-clockwise order) the number of each of the nodes which form the ith triangle. We shall denote by $\mathcal{T}_N$ the set of all the triangulations of Ω with N triangles. For $\Delta \in \mathcal{T}_N$, let $S_N^r(\Delta)$ be the subset of $L_2(\Omega)$ consisting of those functions which, in each triangle $K_i \in \Delta$, are polynomials of degree less than r. As before, we denote by $u = u(\Delta) \in S_N^r(\Delta)$ the projection of f onto $S_N^r(\Delta)$ and seek to minimize the error functional $E : \mathcal{T}_N \to \mathbb{R}$ defined by

$$E(\Delta) = \|u(\Delta) - f\|^2.$$

In practice, we shall seldom find the optimal triangulation. Rather, we shall have to be content with a mesh that is better than any "nearby" triangulations in

$\mathcal{T}_N$. Algorithms for computing such meshes in higher dimension must necessarily consider both the distribution of the nodes and the choice of connectivity between the nodes. In this section, we shall address only the problem of reducing the error by adjusting the nodes x_j, $j = 1, \ldots, M$, for a given connectivity. A complementary algorithm for improving the connectivity will be described in the next section.

As in the one-dimensional case, Baines' algorithm is an iterative procedure for the minimization of E. In order to describe the iteration step, we shall first explain how a particular node is updated, keeping all the other nodes fixed. Let then $\Delta \in \mathcal{T}_N$ be a given triangulation. Consider the particular interior node x_j associated with Δ. Let Ω_j denote the union of those triangles in Δ which share that vertex.

Definition 5.1. We say that $\hat{x} \in \Omega_j$ is *admissible* if a new triangulation $\Delta(\hat{x}) = \{K_i(\hat{x})\}_{i=1}^N \in \mathcal{T}_N$ may be obtained from Δ by substituting $\hat{x}$ for x_j. In this notation, $K_i(\hat{x}) = K_i$ unless x_j is a vertex of K_i, in which case $K_i(\hat{x})$ is the triangle obtained from K_i by substituting $\hat{x}$ for x_j (cf. Figure 1 (b)).

We seek a new admissible position x_j^{new} for the jth node such that

$$E(\Delta(x_j^{new})) \leq E(\Delta), \tag{5.1}$$

with equality if and only if the derivative $\frac{\partial E}{\partial x_j}$ vanishes at Δ.

For this purpose, we introduce the *continuous piecewise linear* function $\phi_j \in S_N^r(\Delta)$ such that

$$\phi_j(x_k) = \delta_{jk} \quad \text{for } k = 1, \ldots, M,$$

where δ_{jk} is the Kronecker delta. Its counterpart in $S_N^r(\Delta(\hat{x}))$ will be denoted by $\phi_j(\cdot, \hat{x})$. In each triangle $K_i \in \Delta$, the projection $u = u(\Delta)$ is a polynomial of degree less than r. Let $u_i : \mathbb{R}^2 \to \mathbb{R}$ denote the polynomial which agrees with u in K_i. For each admissible $\hat{x}$, define

$$\Phi_j(\hat{x}) = \sum_{K_i(\hat{x}) \subseteq \Omega_j} \int_{\partial K_i(\hat{x})} |u_i - f|^2 \, \phi_j(\cdot, \hat{x}) \, n \, ds,$$

where n is the unit normal pointing outward and s is the arclength parameter. As usual, the paths for the line integrals are in the counter-clockwise direction (cf. Figure 1 (a)). Note that

$$\Phi_j(x_j) = \frac{\partial E}{\partial x_j}(\Delta).$$

We may thus think of $\Phi_j(\hat{x})$ as an approximation of the derivative of E, evaluated at $\Delta(\hat{x})$. In Baines' original formulation [2], a new position x_j^{new} for the jth vertex is sought that satisfies

$$\Phi_j(x_j^{new}) = 0.$$

Unfortunately, in contrast with the one-dimensional case, the existence of an admissible zero of Φ_j is not guaranteed. Nevertheless, it is possible to obtain "approximate" solutions by using a simple quadrature rule for the line integrals and choosing the new vertex position to make $|\Phi(x_j^{new})|$ as small as possible. We refer to [2] for further details.

In this paper, we shall instead seek x_j^{new} along the direction of "steepest gradient", i.e.

$$x_j^{new} = x_j - \tau \, \Phi_j(x_j) \quad \text{for some } \tau \in [0, \alpha_j \, \tau_j^{max}], \tag{5.2}$$

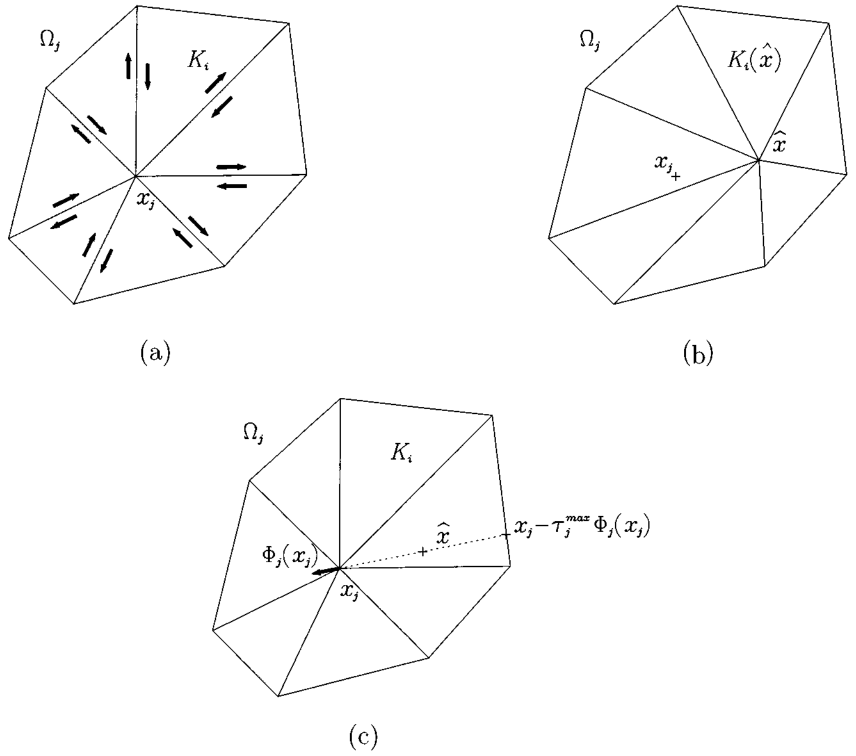


FIGURE 1. The triangles (a) K_i and (b) $K_i(\hat{x})$ in Ω_j. (c) Descent path

where $0 < \alpha_j < 1$. In this expression (cf. Figure 1 (c)),

$$\tau_j^{max} = \sup\{\tau > 0 : x_j - \tau'\,\Phi_j(x_j) \text{ is admissible} \,\forall\, 0 \le \tau' \le \tau\}\,.$$

Assuming that the projection $u = u(\Delta)$ has been computed, the algorithm for updating x_j is as follows:

Algorithm 2 (for updating interior nodes)**.** If $\Phi_j(x_j) = 0$, set $x_j^{new} = x_j$. Otherwise, let τ in (5.2) be the number in $(0, \alpha_j\,\tau_j^{max}]$ such that

$$-\Phi_j(x_j)\cdot\int_0^\tau \Phi_j(x_j - \tau'\,\Phi_j(x_j))\,d\tau'$$

is minimised, where the dot denotes the usual inner product in $\mathbb{R}^2$.

With this definition of x_j^{new}, we have the inequality

$$(x_j^{new} - x_j)\cdot\int_0^1 \Phi_j(tx_j^{new} + (1-t)x_j)\,dt \le 0. \tag{5.3}$$

We shall now prove that (5.3) implies the inequality (5.1). As in section 2, let us introduce the family of triangulations $\Delta(t) = \{K_i(t)\}_{i=1}^N = \Delta(x_j(t))$ where

$$x_j(t) = tx_j^{new} + (1-t)x_j.$$

Let $\hat{u}(\cdot,t) \in S_N^r(\Delta(t))$ be defined by

$$\hat{u}(x,t) = u_i(x) \quad \text{for } x \in K_i(t)$$

and note that, in each $K_i(t) \in \Delta(t)$, we have $\frac{\partial \hat{u}(\cdot,t)}{\partial t} = 0$. Proceeding as in the proof of Theorem 2.1, we may write

$$\begin{aligned} E(\Delta(x_j^{new})) - E(\Delta) &\le \|\hat{u}(\cdot,1) - f\|^2 - \|\hat{u}(\cdot,0) - f\|^2 \\ &= \int_0^1 \frac{d}{dt}\|\hat{u}(\cdot,t) - f\|^2 \, dt = \int_0^1 \sum_{i=1}^N \frac{d}{dt} \int_{K_i(t)} |\hat{u}(x,t) - f(x)|^2 \, dx \, dt. \end{aligned}$$

Using the notation $\phi_j(\cdot,t) = \phi_j(\cdot, x_j(t))$ and recalling that only the jth vertex depends on t, we find that

$$\begin{aligned} \frac{d}{dt}\int_{K_i(t)} |\hat{u}(\cdot,t) - f|^2 \, dx &= (x_j^{new} - x_j) \cdot \int_{\partial K_i(t)} |\hat{u}(\cdot,t) - f|^2 \, \phi_j(\cdot,t) \, n \, ds \\ &\quad + 2 \int_{K_i(t)} [\hat{u}(\cdot,t) - f] \, \frac{\partial \hat{u}}{\partial t}(\cdot,t) \, dx \\ &= (x_j^{new} - x_j) \cdot \int_{\partial K_i(t)} |u_i - f|^2 \, \phi_j(\cdot,t) \, n \, ds. \end{aligned}$$

Therefore

$$\begin{aligned} E(\Delta(x_j^{new})) - E(\Delta) &\le (x_j^{new} - x_j) \cdot \int_0^1 \sum_{K_i(t) \subseteq \Omega_j} \int_{\partial K_i(t)} |u_i - f|^2 \, \phi_j(\cdot,t) \, n \, ds \, dt \\ &= (x_j^{new} - x_j) \cdot \int_0^1 \Phi_j(x_j(t)) \, dt, \end{aligned}$$

so that (5.1) follows from (5.3).

We have thus shown that Algorithm 2 for updating interior nodes does, as in the one-dimensional case, reduce the error monotonically. Obvious modifications are needed to deal with vertices that lie on the boundary of Ω: the nodes at the corners remain fixed ; nodes lying on a side are constrained to remain on that same side and, to construct x_j^{new}, one uses, instead of the vector $\Phi_j(x_j)$, the projection of that vector on the side. Again, this ensures the monotonic decay of the error. A complete iteration of the algorithm consists of projecting f onto $S_N^r(\Delta)$ and then computing x_j^{new} for each $j = 1, \ldots, M$.

Unlike in the one-dimensional case, there is no guarantee that, for the given connectivity, the updated nodes $\{x_j^{new}\}_{j=1}^M$ define a triangulation of Ω. In order to avoid triangles with non-positive area, one may envisage a "Gauss-Seidel" implementation of the algorithm in which the nodes are updated sequentially and the new vertices used immediately they are computed. By adjusting the parameter α_j in (5.2), it is a simple matter to ensure that each triangle area remains greater than some pre-defined threshold $\delta > 0$. However, since one then needs to recalculate the projection each time a node is updated, the cost of a Gauss-Seidel iteration is roughly three times that of the "Jacobi" iteration described earlier (in which the nodes are updated simultaneously using only the old vertices). For reasons of efficiency, we therefore favour the Jacobi implementation. In the next section, we shall explain how to alter the connectivity so as to cater for mesh tangling.

6. Improving the connectivity

We first remark that the occurrence of mesh tangling need not be interpreted as the manifestation of a basic deficiency in the design of the algorithm. Rather, it may reflect the fact that the optimal triangulation cannot be obtained from the given mesh by a "smooth deformation". In order to illustrate this point, consider the approximation of a function $f : \Omega \to \mathbb{R}$ of the form $f(x) = F(x_1)$, where F is a univariate function and $x = (x_1, x_2)$. For such a function, the optimal mesh has no vertices on the vertical boundaries of Ω. If the given initial mesh *does* have vertices on those boundaries, then Algorithm 2 will seek to drive them to the corners and mesh tangling will result.

We therefore adopt a constructive attitude to the occurrence of mesh tangling and view it as a signal that the offending triangles are redundant and need to be removed. Let then $\delta > 0$ be some (small) threshold. If, having computed a Jacobi iteration, it is found that the triangle K_i^{new} has an area less than δ, we reject the Jacobi update and proceed to remove the triangle K_i.

Algorithm 3 (for removing the triangle K_i)**.** There are two cases to consider:

$\underline{1}$ K_i has an edge on the boundary, the edges are comparable in size and the projection of the interior node onto the boundary lies between the boundary nodes. In this case, replace the interior node by its projection onto the part of the boundary along which the edge lies. Remove all the elements with zero area.

$\underline{2}$ Otherwise, identify the shortest edge. Remove one of the nodes that form the edge and replace it by the other node. Then, remove the two triangles which shared the edge.

In the first case, we say that the edges of the triangle are comparable in size if the ratio of the longest edge to the shortest edge does not exceed a certain threshold. The threshold value which we used for the computations reported in the next section is 10. In the second case, a possible criterion to decide which of the nodes should be removed is to compare the corresponding derivatives $\frac{\partial E}{\partial x_j}$ and remove the node with the largest derivative. If the resulting set of nodes does not define a triangulation of Ω, one should remove the other node instead.

There is no guarantee that this simple recipe for removing the offending triangle results in a valid triangulation of Ω. Nevertheless, our computational experience suggests that it always succeeds if δ is sufficiently small.

Since, due to the occurrence of mesh tangling, one must necessarily alter the original connectivity, it is natural to consider further means of improving it. The computational complexity of comparing all the possible triangulations of the same vertices is clearly too great. For this reason, we shall resort to the well-known local optimization procedure (LOP) proposed by Lawson and described in [15], [16].

Consider a triangulation $\Delta \in \mathcal{T}_N$ and let e be an internal edge associated with Δ. In other words, e is a triangle side that does not lie on the boundary of Ω. The union Q of the two triangles which share that edge is either a triangle or a quadrilateral. As shown in Figure 2, if Q is a convex quadrilateral, then there are two possible ways of "triangulating" Q.

Definition 6.1. An internal edge e is said to be *locally optimal* if either

(a) Q is not a convex quadrilateral or

(b) Q is a convex quadrilateral and

$$E(\Delta) \leq E(\Delta'),$$

where Δ' is the triangulation obtained from Δ by substituting for e the other diagonal of Q.

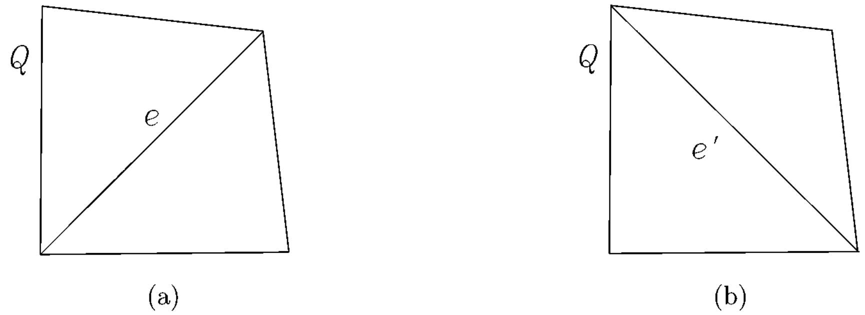


FIGURE 2. Alternative triangulations of the convex quadrilateral Q

Lawson's procedure for constructing from a given triangulation Δ a new triangulation Δ^{new} for which all the edges are locally optimal may be described as follows:

Algorithm 4 (Lawson)**.** Step 1. Test each edge defined by Δ for local optimality. If all the edges are locally optimal, set $\Delta^{new} = \Delta$ and stop. Else go to step 2. Step 2. Let e be an internal edge that is not locally optimal. Set $\Delta = \Delta'$ (cf Definition 6.1) and go to step 1.

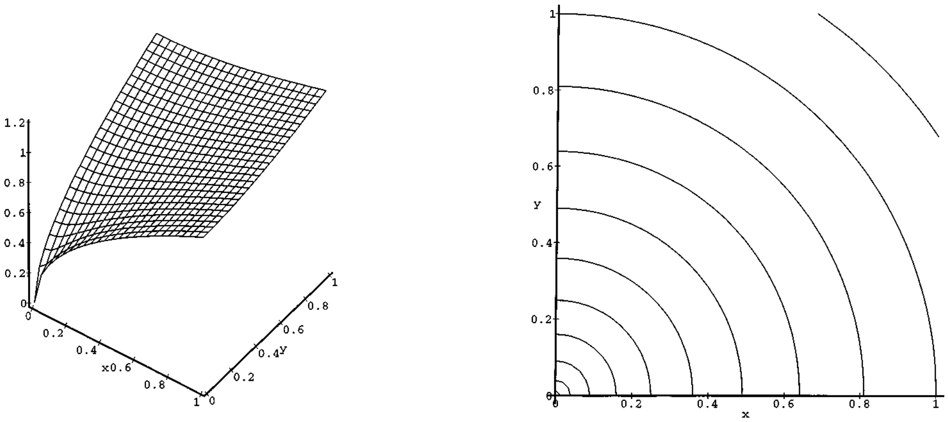
It is clear that the resulting triangulation depends upon the order in which one sweeps through the internal edges. Dyn, Levin and Rippa [16] have carried out numerical experiments and compared a number of possible strategies. Their results suggest that, in our context, it would be advantageous to swap the edge that leads to the largest reduction in the error functional E. However, in order to simplify the book-keeping, we have instead opted for the following strategy: before each sweep, we make a list of all the edges that are not locally optimal and order them according to the size of error reduction. We sweep through the list in descending order, replacing those edges that are not locally optimal. When the sweep is completed, we form a new list of those edges that are not locally optimal since, in general, more than a single sweep may be necessary before they are all eliminated.

7. NUMERICAL RESULTS IN TWO DIMENSIONS

We now combine the algorithms of the previous two sections and compute triangulations of the unit square for the approximation of interesting "test functions" f by piecewise linear polynomials. The triangulations which we shall obtain are locally optimal in the sense that (a) the internal edges are locally optimal and (b) the derivative of E with respect to the nodes vanishes.

As in the one-dimensional case, we start with a given initial triangulation $\Delta \in \mathcal{T}_N$ and construct a sequence of triangulations $\{\Delta^n\}_{n\in\mathbb{N}}$ iteratively. Since triangles may be removed as the iteration proceeds, we shall generally have

$$\Delta^n \in \bigcup_{k=2}^{N} \mathcal{T}_k$$



(a)

(b)

FIGURE 3. (a) Graph and (b) contour lines of the function $f(x) = \sqrt{|x|}$

rather than $\Delta^n \in \mathcal{T}_N$. We obtain Δ^{n+1} from Δ^n by applying Algorithms 4, 2 and 3 consecutively. We use the stopping criterion

$$\|E'(\Delta^n)\|_\infty := \max_{1\le j\le M} \left| \frac{\partial E}{\partial x_j}(\Delta^n) \right|_\infty < 10^{-10},$$

where $|\cdot|_\infty$ denotes the maximum norm in $\mathbb{R}^2$. In the above expression, $\frac{\partial E}{\partial x_j}(\Delta^n)$ should in fact be replaced by zero if x_j^n is at a corner, and by the appropriate horizontal or vertical component if x_j^n lies elsewhere on the boundary. We make the choice $\delta = 10^{-10}$ in Algorithm 3 and $\alpha_j = 9/10$ in Algorithm 2. The integrals are computed to machine accuracy by means of composite high-order Gaussian quadrature rules. As in section 4, the calculations begin at the coarsest level ($N = 8$). Once convergence is obtained, a uniform refinement of the locally optimal triangulation is used as a starting mesh for the next finer level. As before,

$$\eta_n = \frac{\|E'(\Delta^n)\|_\infty}{\|E'(\Delta^{n-1})\|_\infty}$$

is the factor by which the "residual" is reduced at the nth iteration.

Example 7.1.

$$f(x) = \sqrt{|x|},$$

where $|\cdot|$ is the euclidean norm in $\mathbb{R}^2$.

The graph and contour lines of f are shown in Figure 3. Our calculations are summarised in Tables 3 and 4. Table 3 suggests that the asymptotic performance of the algorithm as N increases is qualitatively much the same as in the one-dimensional case.

Note that f has a singularity at the origin and cannot be approximated to an optimal order by piecewise polynomials on quasi-uniform meshes as $N \to \infty$. This is confirmed by the results displayed in Table 4. In this table, γ_N is the local rate

TABLE 3. Iterations in the computation of locally optimal triangulations for the approximation of the function $f(x) = \sqrt{|x|}$ by piecewise linear polynomials

N	n	*edge swaps*	$\|E'(\Delta^n)\|_\infty$	η_n	$\sqrt{E(\Delta^n)}$
8					
	0	–	0.42459422D−03	–	0.14029464D−01
	1	4	0.10367787D−03	0.24	0.90462685D−02
	2	0	0.73561779D−04	0.71	0.84609623D−02
	5	0	0.25517273D−04	0.70	0.79791818D−02
	10	0	0.10965694D−04	0.70	0.74940892D−02
	20	0	0.34195277D−06	0.71	0.74851905D−02
	40	0	0.49730317D−09	0.73	0.74851847D−02
	46	0	0.73156246D−10	0.73	0.74851847D−02
32					
	0	–	0.13604532D−04	–	0.23427740D−02
	1	2	0.94188498D−05	0.69	0.21539671D−02
	2	0	0.67101547D−05	0.71	0.20820072D−02
	5	0	0.26946165D−05	0.75	0.20194351D−02
	10	0	0.26750307D−05	0.80	0.18842340D−02
	20	0	0.61825238D−06	0.88	0.18422231D−02
	40	0	0.63962876D−07	0.94	0.18349505D−02
	120	0	0.45798966D−09	0.94	0.18349243D−02
	145	0	0.98448068D−10	0.94	0.18349243D−02
128					
	0	–	0.93047387D−06	–	0.51189443D−03
	1	6	0.63258753D−06	0.68	0.48663096D−03
	2	2	0.44654850D−06	0.71	0.47661968D−03
	5	2	0.30771953D−06	0.64	0.46538827D−03
	10	0	0.54005721D−07	0.74	0.46021732D−03
	20	0	0.17465723D−07	0.93	0.45721019D−03
	40	0	0.66404327D−08	0.96	0.45575799D−03
	80	0	0.32103565D−08	0.99	0.45509076D−03
	160	0	0.97696194D−08	0.98	0.45486095D−03
	352	0	0.98924223D−10	0.99	0.45482596D−03

of convergence defined by

$$\gamma_N = \frac{\log(e_N/e_{4N})}{\log 2},$$

where e_N is the L_2 error for the mesh with N triangles. For $N = 2$, the triangulation Δ^{uni} consists of two triangles sharing the diagonal that joins the origin to the point $(1, 1)$. For larger values of N, Δ^{uni} is obtained from that mesh by successive uniform refinement. The results suggest that the corresponding rate of convergence is $\gamma = 3/2$. By contrast, the triangulations Δ^* generated by the algorithm yield the optimal rate of convergence $\gamma = 2$. Some of those triangulations are shown in Figure 4 for (a) 128, (b) 510 and (c) 2036 triangles respectively. Figure 4(d) shows an enlargement of this last mesh near the singularity.

TABLE 4. Decay of the L_2 error as N increases for the approximation of $f(x) = \sqrt{|x|}$ by piecewise linears

N	$\sqrt{E(\Delta^*)}$	γ_N	N	$\sqrt{E(\Delta^{uni})}$	γ_N
2	0.27408054D−01	–	2	0.27408054D−01	–
8	0.74851847D−02	1.87	8	0.10243603D−01	1.42
32	0.18349243D−02	2.03	32	0.37297791D−02	1.46
128	0.45482596D−03	2.01	128	0.13380112D−02	1.48
510	0.11312414D−03	2.01	512	0.47646763D−03	1.49
2034	0.28290617D−04	2.00	2048	0.16905744D−03	1.49
8136	0.70656407D−05	2.00	8192	0.59876793D−04	1.50
32544	0.17775092D−05	1.99	32768	0.21188352D−04	1.50

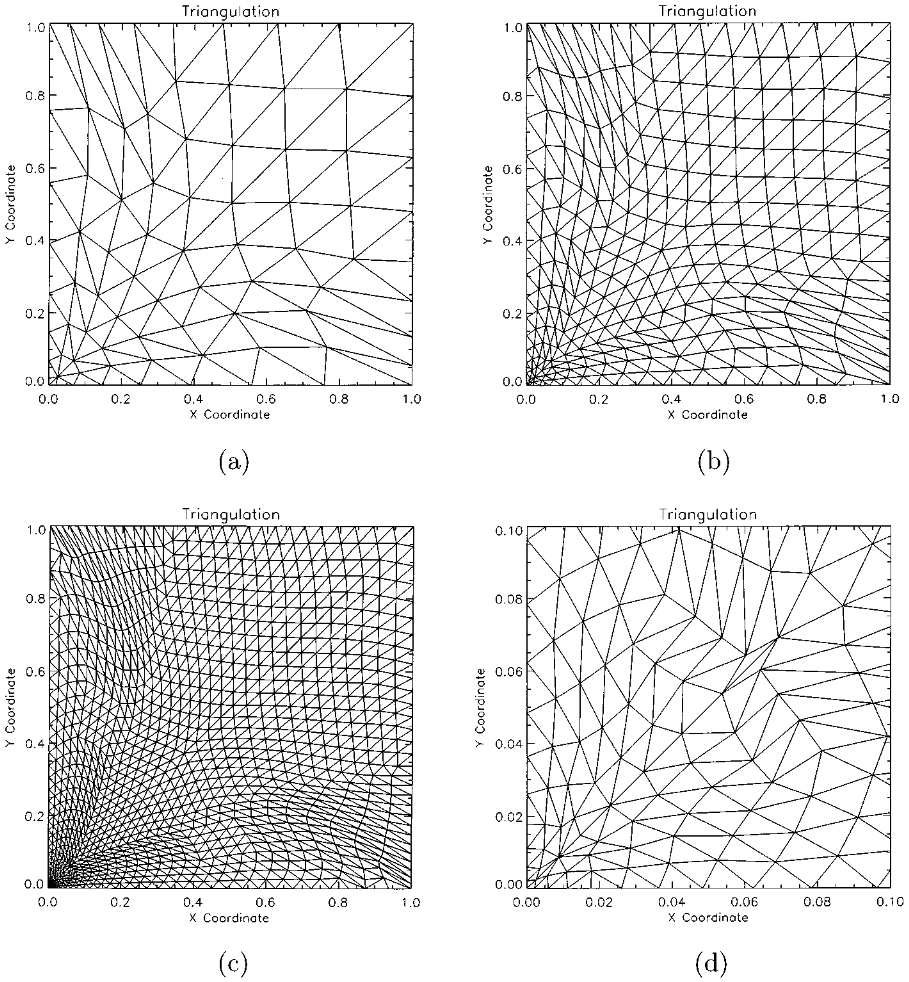
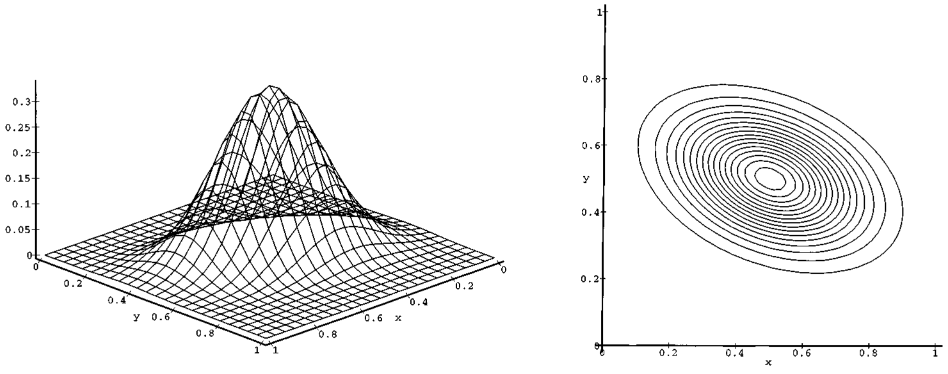


FIGURE 4. Some locally optimal triangulations for the approximation of the function $f(x) = \sqrt{|x|}$ by piecewise linears: (a) $N = 128$, (b) $N = 510$, (c) $N = 2034$ and (d) enlarged view of that triangulation near the origin



(a) (b)

FIGURE 5. (a) Graph and (b) contour lines of the function $f(x) = \frac{1}{3}e^{-\frac{81}{4}\xi}$, where $\xi = (x_1 - \frac{1}{2})^2 + (x_1 - \frac{1}{2})(x_2 - \frac{1}{2}) + 2(x_2 - \frac{1}{2})^2$

Example 7.2.

$$f(x) = \frac{1}{3}e^{-\frac{81}{4}\left[(x_1-\frac{1}{2})^2+(x_1-\frac{1}{2})(x_2-\frac{1}{2})+2(x_2-\frac{1}{2})^2\right]}.$$

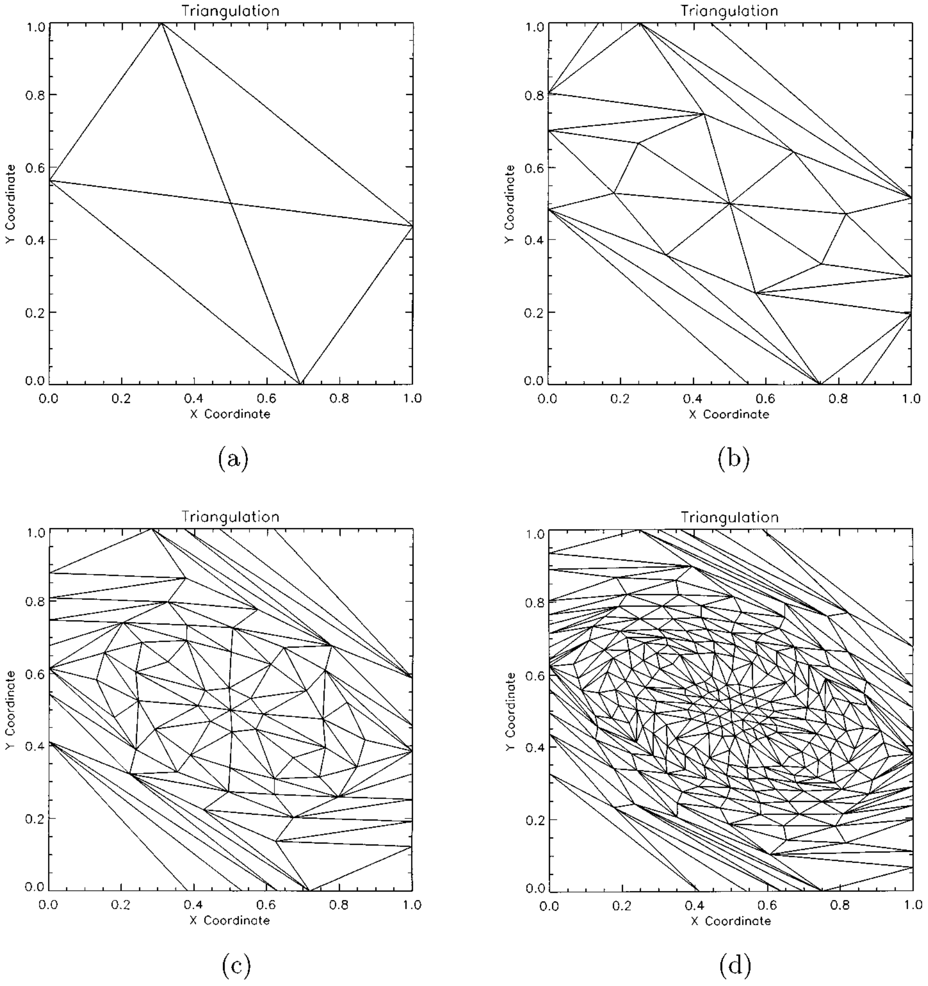
The graph and contour plot are shown in Figure 5. Locally optimal triangulations for the approximation of this function by piecewise linear polynomials are displayed in Figure 6.

Example 7.3.

$$f(x) = \tanh(20(|x|^2 - \frac{1}{2})).$$

As the graph and contour plot in Figure 7 show, this function has a sharp gradient along the curve $|x|^2 = 1/2$. Figure 8 (a) shows the locally optimal mesh computed by the algorithm, using a uniform mesh to start the iteration. The error for the uniform mesh (with 128 triangles) is $\sqrt{E}(\Delta^{uni}) = 0.40517872\text{D}{-}01$ while, for the computed mesh (with 118 triangles), we have $\sqrt{E}(\Delta^*) = 0.42442848\text{D}{-}02$. Using the same uniform starting mesh, but with a value of $\delta = 10^{-8}$ in Algorithm 3, we found that the algorithm converges to a slightly different triangulation (with 116 triangles), shown in Figure 8 (b), with an error $\sqrt{E}(\Delta^*) = 0.41575984\text{D}{-}02$. This dependence of the final mesh on the particular choice of parameters suggests the presence of many nearby local minima of the error functional.

The last two examples constitute severe tests for our algorithm, requiring in some cases over a thousand iterations.



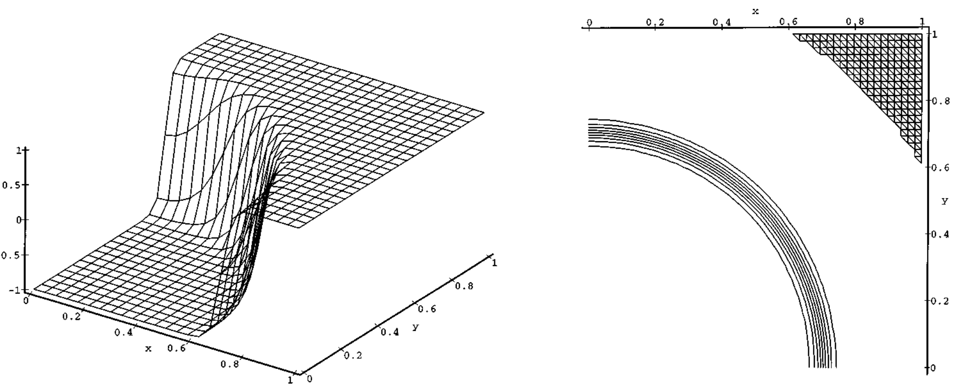
(a)

(b)

(c)

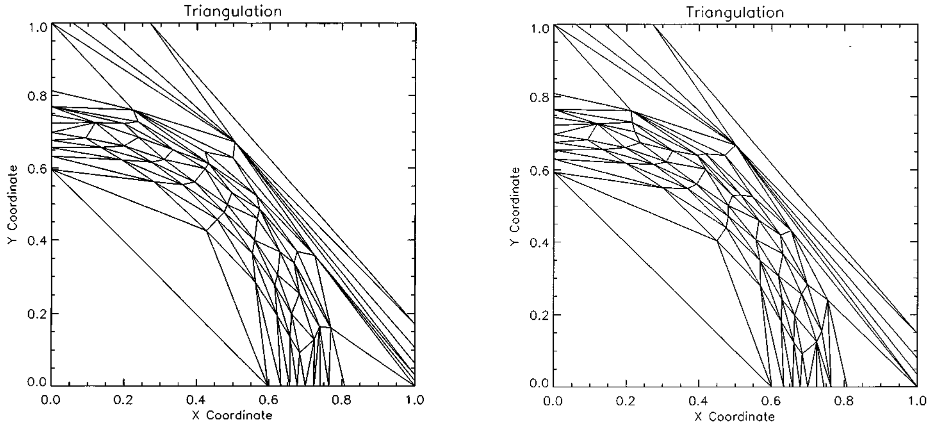
(d)

FIGURE 6. Locally optimal meshes for the approximation of the function $f(x) = \frac{1}{3}e^{-\frac{81}{4}\xi}$, where $\xi = (x_1 - \frac{1}{2})^2 + (x_1 - \frac{1}{2})(x_2 - \frac{1}{2}) + 2(x_2 - \frac{1}{2})^2$, by piecewise linear polynomials. (a) 8, (b) 32, (c) 122 and (d) 468 triangles



(a) (b)

FIGURE 7. (a) Graph and (b) contour lines of the function $f(x) = \tanh(20(|x|^2 - \frac{1}{2})$



(a) (b)

FIGURE 8. Two locally optimal meshes for the approximation of the function $f(x) = \tanh(20(|x|^2 - \frac{1}{2}))$ by piecewise linear polynomials. The same uniform mesh (with 128 triangles) was used as a starting mesh with, however, different values of δ in Algorithm 3. (a) $\delta = 10^{-10}$. (b) $\delta = 10^{-8}$

8. Conclusion

In this work, we have addressed the problem of generating locally optimal meshes for L_2 approximation by discontinuous piecewise polynomials. The salient feature of the iterative procedure which we considered is that the cost of each iteration is roughly that of projecting the function onto the finite-dimensional space induced by the current mesh. We have shown that the L_2 error decreases monotonically.

In the one-dimensional case, we were able to use this property to obtain convergence results.

In the two-dimensional case, it is necessary to adjust both the nodal positions and the connectivity between the nodes. Numerical experiments indicated that the qualitative convergence properties of the algorithm are much the same as in the one-dimensional case. In particular, we found that, although the first few iterations are very effective, the algorithm exhibits a poor asymptotic rate of convergence when the number of triangles is large. It might be possible to speed up the convergence by formulating a "multigrid" approach. This conjecture deserves further attention.

The numerical techniques used in this paper should generalize to higher dimension. The extension of the algorithm for moving the nodes is straightforward. The generalization of Lawson's algorithm and of the recipe for element removal does, however, require more careful consideration.

References

1. T. Apel and B. Heinrich, *Mesh refinement and windowing near edges for some elliptic problem*, SIAM J. Numer. Anal. **31** (1994), 695–709. MR **95j:**65132
2. M. J. Baines, *Algorithms for optimal discontinuous piecewise linear and constant L_2 fits to continuous functions with adjustable nodes in one and two dimensions*, Math. Comput. **62** (1994), 645–669. MR **94g:**65015
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. Comput. **32** (1978), 1131–1143. MR **58:**1853
4. A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comput. **31** (1977), 333–390. MR **55:**4714
5. H. Burchard, *On the degree of convergence of piecewise polynomial approximation on optimal meshes*, Trans. Am. Math. Soc. **234** (1977), 531–559. MR **58:**1857
6. D. Catherall, *The adaption of structured grids to numerical solutions for transonic flow*, Internat. J. Numer. Meths. Engrg. **32** (1991), 921–937.
7. K. Chen, *Error equidistribution and mesh adaption*, SIAM J. Sci. Comput. **15** (1994), 798–818. MR **95a:**65156
8. P. G. Ciarlet, *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation*, Masson, Paris, 1982. MR **84c:**65002a
9. E. F. D'Azevedo, *Optimal triangular mesh generation by coordinate transformation*, SIAM J. Sci. Stat. Comput. **12** (1991), 755–786. MR **92a:**65041
10. E. F. D'Azevedo and R. B. Simpson, *On optimal triangular meshes for minimizing the gradient error*, Numer. Math. **59** (1991), 321–348. MR **92c:**65095
11. C. de Boor, *A Practical Guide to Splines*, Springer–Verlag, New York, 1978. MR **80a:**65027
12. M. Delfour, G. Payre and J. P. Zolésio, *An optimal triangulation for second-order elliptic problems*, Comput. Meths. Appl. Mech. Engrg. **50** (1985), 231–261. MR **86j:**65151
13. R. A. DeVore and B. J. Lucier, *High order regularity for solutions of the inviscid Burgers equation*, in Lecture Notes in Mathematics, vol. 1402, Springer–Verlag, New York, 1989, 147–154. MR **90k:**35164
14. R. A. DeVore and V. A. Popov, *Interpolation spaces and non-linear approximation*, in Function Spaces and Applications, M. Cwikel, J. Peetre, Y. Sagher and H. Wallin, eds., Springer Lecture Notes in Mathematics, vol. 1302, Springer–Verlag, New York, 1988, 191–205. MR **89d:**41035

15. N. Dyn, D. Levin and S. Rippa, *Data dependent triangulations for piecewise linear interpolation*, IMA J. Numer. Anal. **10** (1990), 137–154. MR **91a:**65022
16. N. Dyn, D. Levin and S. Rippa, *Algorithms for the construction of data dependent triangulations*, in Algorithms for Approximation, J. C. Mason and M. G. Cox, eds., Chapman and Hall, London, 1990, 185–192. CMP 91:01
17. W. Huang, Y. Ren and R. D. Russell, *Moving mesh PDEs based on the equidistribution principle*, SIAM J. Numer. Anal. **31** (1994), 709–730. MR **94m:**65149
18. W. Huang and D. M. Sloan, *A simple adaptive grid method in two dimensions*, SIAM J. Sci. Comput. **15** (1994), 776–797. MR **95a:**65155
19. D. McGlure, *Nonlinear segmented function approximation and analysis of line patterns*, Quart. Appl. Math. **33** (1975), 1–37. MR **57:**3709
20. E. Nadler, *Piecewise linear best L_2 approximation on triangulations*, in Approximation Theory V, C. K. Chui, L. L. Schumaker, J. D. Ward, eds., Academic Press, Boston, 1986.
21. R. B. Simpson, *Anisotropic mesh transformations and optimal error control*, Appl. Numer. Math. **14** (1994), 183–198. CMP 94:11
22. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965. MR **32:**1894

School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom
E-mail address: y.tourigny@bristol.ac.uk

Department of Mathematics, University of Reading, P.O. Box 220, Reading RG6 6AF, United Kingdom
E-mail address: m.baines@reading.ac.uk