CS-28-19

Printing Requisition/Graphic Services 34

Dept. No. 46676

1. Please complete unshaded areas on form as applicable. (4 part no carbon required).
2. Distribute copies as follows: White, Canary and Pink—Printing, Arts Library or applicable Copy Centre Goldenrod—**Retain.**
3. On completion of order, pink copy will be returned with printed material. Canary copy will be costed and returned to requisitioner, **Retain as a record of your charges.**
4. Please direct enquiries, quoting requisition number, to Printing/Graphic Services, Extension 3451.

Title or Description Automatic Local Refinement for Irregular Rectangular Meshes			
Date Dec. 16/81	**Date Required** Dec. 16/81	**Account** 126-4100-01	
Signature	**Signing Authority** Freda Bruner		
Department Computer Science	**Room** 5179	**Phone** 3143	**Delivery** [ ] Mail [✓ Pick-up] [ ] Via Stores [ ] Other
Reproduction Requirements [ ] Offset [ ] Signs/Repro's [✓ Xerox]	**Number of Pages** 14	**Number of Copies** 1	
Type of Paper Stock [✓ Bond] [ ] Book [ ] Cover [ ] Bristol [ ] Supplied			
Paper Size [✓ 8½ x 11] [ ] 8½ x 14 [ ] 11 x 17			
Paper Colour [✓ White] [ ] Other	**Ink** [ ] Black		
Printing [✓ 1 Side] [ ] 2 Sides	**Numbering** to		
Binding/Finishing Operations [✓ Collating] [✓ Corner Stitching] [ ] 3 Ring [ ] Tape [ ] Plastic Ring [ ] Perforating			
Folding Finished Size	**Cutting** Finished Size		
Special Instructions			

Cost: Time/Materials		Run	Prod Util	Prod Cor	Mat	Total
Signs/Repro's	1					
Camera	2					
Correcting & Masking Negatives	3					
Platemaking	4					
Printing	5					
Bindery	6					
Sub. Total Time						
Sub. Total Materials						
Prov. Tax						
Total						

Film Qty Size	**Plates** Qty Size & Type
Paper Qty Size	**Plastic Rings** Qty Size
Outside Services	

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 14, 1665–1678 (1979)

AUTOMATIC LOCAL REFINEMENT FOR IRREGULAR RECTANGULAR MESHES

R. BRUCE SIMPSON
Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada

SUMMARY

The use of local mesh refinements for the generation of meshes for the finite element or finite difference methods is studied. A class of rectangular meshes which admit restricted local refinements, referred to as irregular rectangular meshes, is introduced and its representation discussed. Properties of algorithms for mesh refinements are discussed from the viewpoints of termination with a mesh in the specified class, memory utilization, symmetry and fragmentation of the mesh.

INTRODUCTION

The generation of a suitable mesh of elements is well recognized as an important part of the finite element method for continuum problems, as well as for the other numerical techniques using piecewise defined functions. One aspect of this topic is the production of meshes for irregularly shaped regions.[1,4,5,12] However, in this paper we deal with another side of the question, i.e. the generation of meshes which require different gradings in different parts of the region to represent functions whose scales of local behaviour vary significantly. Functions with singularities or boundary layers are obvious examples. Our interest has been oriented towards a related class of examples—boundary value problems posed for infinite domains which for numerical purposes are replaced by approximating truncated domains and boundary conditions.[13] The significant solution behaviour is localized on a scale that puts the truncated boundaries far away, and so one wants 'small elements' in this region of significant solution behaviour and large ones near the truncated boundary.

A simple but flexible and, for many problems, convenient mesh generation technique uses as input a crude mesh of rectangles or triangles and a subdivision factor f, a positive integer. The algorithms of the method then subdivide each element into f^2 congruent sub-elements to generate the desired mesh.[10,21] Another approach, which produces graded meshes of triangles, is based on splitting the process into two stages. In the first stage, a distribution of mesh vertices is generated and in the second stage these are connected up into triangles.[9,15] The mesh grading can be controlled by specifying the distribution of vertices through a density function.

The technique discussed in this paper for rectangular meshes uses as input an initial (presumably crude) mesh and a mesh concentration or density function, $C(x, y)$. The algorithms of the method then attempt, through local refinements, to produce a mesh of rectangles whose areas are inversely proportional to the mesh concentration function (i.e. where the concentration is high, each rectangle's area is small). Our discussion deals with algorithms for implementing this technique independently of the particular choice of mesh concentration function used. However, an obvious application is the case in which $C(x, y)$ is derived from a (presumably crude) approximation to the solution of the problem being solved, based on the initial mesh. So as not to preclude this approach, we assume that the concentration function is a

0029–5981/79/1114–1665$01.00
Received 22 March 1978
Revised 20 January 1979

mesh function, i.e. is only defined for the rectangles of the mesh by a list of values in a piecewise constant manner. Hence, if a mesh refinement splits a rectangle into two sub-rectangles, new function values must be obtained by interpolation. Clearly, if the concentration function were provided explicitly and globally (as an arithmetic expression, say), such interpolation would be unnecessary. In the case of the finite element method, smoother piecewise defined concentration functions may be available with the help of the element shape functions.

On the other hand, if the concentration function is defined using higher derivatives of the shape function, as in Reference 19, it may be only piecewise constant, as we have assumed. Several discussions of the relation of mesh refinement to problem solution for finite element and finite difference methods appear in the literature, e.g. References 2, 14 and 18. The ideas of this paper were used adaptively in a finite difference solution of mixed diffusion–convection transport problems, some results of which have been presented in Reference 13. A more general approach has been described by Rheinboldt and Mesztenyi, in Reference 17 and in connection with it an extensive analysis of relating the mesh concentration function, $C(x, y)$, to error control has been combined with experimental computations by Babuska (see references in Reference 2). Alternative choices for $C(x, y)$ based on energy considerations are given in References 16 and 20.

We discuss, in the next section, the introduction of a class of rectangular meshes that allow restricted local refinements and which form the basis of the technique. Then, we examine, in the following section, algorithms for performing mesh refinements to achieve a desired contribution of rectangle sizes, and in the final section we describe some experimental computations with these algorithms.

IRREGULAR RECTANGULAR MESHES

For simplicity, it will be assumed that the basic region on which a mesh is wanted is a rectangle

$$R = \{(x, y) | a \leqslant x \leqslant b, c \leqslant y \leqslant d\}$$

Extensions to regions which are unions of rectangles are obvious. The term 'a regular rectangular mesh' on R will denote the set of rectangles formed from a set of intervals of the x-axis defined by lines $x = x_i$, for $i = 1, 2, \ldots, K$, with those defined on the y-axis by $y = y_j$, $j = 1, 2, \ldots, N$, where $\{x_i\}$ and $\{y_j\}$ are monotone increasing with $x_1 = a$, $x_K = b$, $y_1 = c$, $y_N = d$. To eliminate some unwanted special cases, we assume $K \geqslant 4$, $N \geqslant 4$, and we will refer to the rectangles of the mesh as mesh cells. The term 'bisecting a cell in the x-direction' will be used for splitting a rectangle into two congruent sub-rectangles by a vertical line through its centre (and similarly for the y-direction). Such bisections form a very simple local mesh refinement and we could consider the class of all meshes obtained from a regular mesh by an arbitrary sequence of bisections. However, such a general class of meshes seems considerably more flexible than necessary for the discretization of continuously varying functions, and the flexibility is gained at the price of complexity in its representation. Hence we postulate some restrictions on the bisected mesh.

When two cells have a side, or part side, in common they will be referred to as neighbours. A cell with four neighbours each having commensurate common sides will be called a regular cell since locally, at that cell, the mesh appears regular. Other cells will be called irregular. Such irregular cells usually involve some form of 'overhead' costs to the finite difference or finite element methods. They may require special treatment in the forming or solving of linear equations (as discussed in References 6, 7 and 8 for the finite element method) or they may introduce anomalous errors.[13] Hence it seems desirable to keep the proportion of such cells to a minimum. This proportion, the ratio of irregular cells to total number of cells in a mesh, will be referred to as the fragmentation of the mesh.

The first restriction to be placed on the meshes is that no cell should have more than two neighbours on any side. This restriction results in simplifying the representation of the mesh considerably. The second restriction is aimed at reducing the mesh fragmentation. It is that no cell should have two neighbours on each of a pair of parallel sides—see Figure 1(a). This restriction is extended to the case of boundary cells (i.e. cells with at least one side on the boundary of R) to require that no cell shall have two neighbours on a side opposite a boundary—see Figure 1(b).

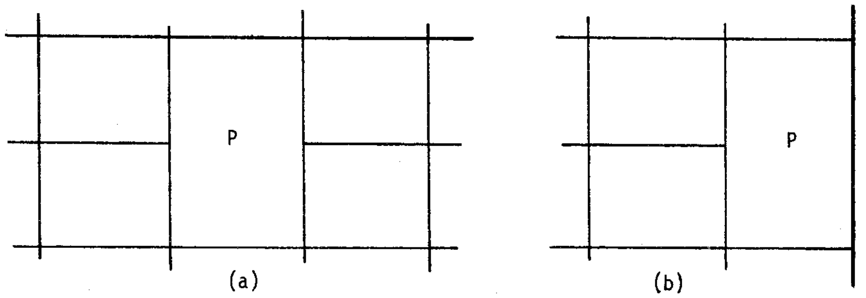

Figure 1. Inadmissible neighbour's configuration for P

Bisecting the cell P of Figure 1(a) or 1(b), in either the x- or y-direction, will result in configurations of cells meeting both restrictions. However, bisecting in the x-direction increases the fragmentation of the mesh, whereas bisecting in the y-direction decreases it and is hence the preferred way to meet the mesh requirements from these configurations.

An irregular rectangular mesh will be defined as a mesh obtained from a regular rectangular mesh (called the basic mesh) by a series of cell bisections, and such that each cell satisfies the two restrictions stated above. It is perhaps surprising that for such an elementary class of meshes there are 192 configurations of neighbours which a cell may have (including the possibility of no neighbours on one, or two, adjacent sides). In the following we will abbreviate 'irregular rectangular mesh' to IRM.

In our implementation, an IRM is represented as an array of cell descriptions. Each cell is numbered with its array index, and its cell description consists of the following fields: co-ordinates of the cell centre; x- and y-dimensions of the cell; indices of the neighbouring cells (encoded in four integers); application data for the cell.

Storage space for the array is set up to handle a maximum of M cells. The neighbour's data for each side is encoded into one integer word in the form

$$(\text{sign})(j \times (M+1) + k)$$

where $j = k = 0$ implies no neighbours on this side; sign = +implies $j = 0$ and k is the index of the single neighbour on this side; and sign = −implies there are two neighbours on this side, with indices j and k.

As most operations involving mesh cells require knowledge of each cell's neighbour's configuration, it has proved convenient to introduce a sub-program which, for a given cell, makes this information readily available by returning four groups of three integers—one for each side of the cell. The first integer is a configuration code which can be used to drive CASE or FORTRAN computed GO TO statements:

= 1 if no neighbours are present on that side
= 2 if one neighbour is present with larger side length
= 3 if one neighbour is present of the same side length
= 4 if two neighbours are present.

The other two integers are the neighbour's indices, as relevant. If no further encoding is done, the IRM representation uses $4M$ floating point words, plus $4M$ integer words, plus space for application data (in the case of our examples, one function value for the concentration function per cell for an additional M floating point words). For the finite element method, the element nodes would have to be generated and numbered and the element incidence list added to the mesh representation.

Perhaps a comment on the simplicity of this representation is in order. The finite difference or finite element methods require some knowledge of the neighbour's configuraton for each cell. For a regular mesh, i.e. one in which every cell has the same configuration of neighbours, this information is either used implicitly (element compatibility) or can be summarized in a few global parameters, e.g. the cell shape is triangular or rectangular. If local refinements introduce irregular cells, then the neighbour's configuration data varies from cell to cell. If the variation is allowed to be quite large, then the information can be appropriately stored in a list with variable length records. A data structure of wide flexibility for this purpose has been devised by Rheinboldt and Mesztenyi.[17] The restrictions that we have placed on the mesh limit the variability of the neighbours, so that the neighbour's data can be accommodated in a record of modest 'maximal' size. Our viewpoint is, then, that the simplicity gained by using fixed length records of this size probably justifies the inefficiency in the use of memory space that results even without the encoding described above.

THE MESH REFINEMENT PROCESS

Mesh refinement is accomplished by repeated scans through the cell list, inspecting cells according to a set of bisection criteria, and modifying the list when a cell bisection is made. The mesh is initialized to a regular rectangular mesh. The process terminates when a scan of the list is made which results in no bisections. The phrase 'bisecting a cell' refers to modifying the cell description list; the index of the cell will pass to one of the two newly created cells (the children) and the other will be added to the bottom of the list. It is straightforward to update the fields of list items for the child cells (co-ordinates, neighbours, etc.), except possibly for the application data when it is only known as discrete cell values. In general, this poses the non-trivial question of the economical approximation of a function based on values known at a variable (albeit small) number of data points irregularly spaced in the plane. For the examples in the next section, we have used applications data given at the centres of cells, and have generated new cell data by an *ad hoc* interpolation scheme, which is second order for regular cells and first order for irregular cells. It is worth noting that this same approximation requirement occurs whenever the applications data is wanted at points other than provided by the irregular mesh; in particular, at output time when a table of values on some regular mesh is wanted either for printing or graphing. Hence it is worthwhile not to restrict the interpolation method to apply only to cells which allow imminent bisection in one direction or another.

Before discussing the mechanics of the mesh refinement process further, some comments on the objectives for the process should be made. In order to form the basis of a mesh generation module which can interface reliably in a program package with discrete equation generation modules, the mesh refinement process should be guaranteed to terminate with an admissible mesh. A second objective is concerned with the fact that the number of cells in a mesh determine the cost or even feasibility in terms of time and memory requirements of a finite element or finite difference computation; hence it is important to be able to put an upper limit on this number of cells. Two further objectives, of a more qualitative nature, are that the mesh should be fragmented as little as possible and that it should reflect the symmetries of the concentration function as much as possible. The fact that the irregular cells typically involve special overhead to the discrete equation processes was remarked on above and is the motivation for the low fragmentation objective. The motivation for the symmetry objective is more subjective. Initial experiments were made with processes that generated meshes with noticeable lack of symmetry from symmetrical concentration functions. Such a mesh is disconcerting to view. One could argue that convergent discretization methods yield results which are independent of the discretization mechanism. Hence, as long as the cell data distribution after refinement maintains its symmetry, it is irrelevant whether the mesh itself does. However, the following counter-argument seems more forceful in that it does not apply only in the convergence limit, and it supports our sense of aesthetics in mesh generation. Consider two mesh refinement processes, A and B, with A consistently generating IRM's with more of the symmetries of the generating data than B. Then we take this as evidence that A is more sensitive to data than is B, and hence we prefer A.

The mesh refinement process can be conveniently divided into two sub-topics: (a) the cell bisection decision algorithm, and (b) the process of scanning the mesh. These will be discussed in the two following subsections.

The cell bisection decision

One source of criteria for bisecting a cell is the restrictions placed on the cell neighbour's configuration mentioned in the last section and repeated here for reference:

No cell can have more than two neighbours on one side (1)

No interior cell can have a pair of neighbours on each of two parallel sides (2)

No cell adjacent to the region's boundary (i.e. a boundary cell) may have two neighbours on the side opposite the boundary (3)

The other source of criteria for cell bisection comes from the mesh concentration function. Using the area, $A(P)$ of a cell P, and its concentration, $C(P)>0$, a cell content can be defined as

$$\operatorname{cont}(P) \equiv C(P)A(P) \tag{4}$$

A maximum permissible cell content is specified for the refinement process, ε, and the criterion for accommodating the mesh to the function can be written

$$\operatorname{cont}(P) \leqslant \varepsilon \tag{5}$$

It will be seen that there is the possibility of contention between criteria (5) and (1), since (5) may require a cell to be bisected whereas (1) does not permit it; a contention that the bisection decision algorithm must resolve. If (5) does require a cell to be bisected, there is the question of in which direction. For the direction of bisection, there are (at least) three possible contending criteria: the configuration of mesh neighbours, the accommodation of the mesh to the function, and the shape (i.e. height to width ratio) of the resulting cells. A short discussion of the algorithm's approach to the latter two criteria follows.

If a cell P is to be bisected into P_1 and P_2, then there may be no constraint from the neighbour's configuration on the direction of bisection, as for example with a regular cell. To make the choice, the algorithm performs a computation to see which direction will produce the bigger concentration difference, and hence content difference, between the two new cells. If, for example, bisection in the x-direction would produce a concentration difference between P_1 and P_2 that is significantly greater than bisection in the y-direction, then the x-direction would be preferred on the basis of trying to isolate regions of high concentration. If the choice of direction does not affect the concentration difference between P_1 and P_2 significantly, then the direction (if any) which produces the lesser elongation of the cells is chosen. The measure of significance is a gradient tolerance preset in the decision algorithm. Specifically, the process can be described as follows:

1. Compute concentration gradients through differencing in the x- and y-directions, D_xC, D_yC.
2. Compare their difference to the gradient tolerance, g_{tol}; and if not significantly different, replace by 1, i.e.

$$\begin{aligned} \text{if} \quad & (||D_xC|-|D_yC||<g_{\text{tol}}) \\ \text{then} \quad & (D_xC=1, D_yC=1) \end{aligned}$$

3. Compute concentration differentials

$$\Delta_xC=|D_xC|h_x; \qquad \Delta_yC=|D_yC|h_y$$

4. If $(\Delta_xC>\Delta_yC)$, then concentration differential is bigger in x-direction, otherwise concentration differential is bigger in the y-direction.

Using this, then, the cell bisection decision algorithm may be described as follows:

```
   comment—check restrictions (2) and (3)
1. if (cell has two neighbours on both north and south boundaries) then bisect in x-direction,
   end
2. if (cell has two neighbours on both east and west boundaries) then bisect in y-direction, end
3. if (cell is a boundary cell, with two neighbours on side opposite region boundary)
   then bisect in direction consistent with neighbours, end
   comment—check criterion (5)
4. if (content of cell < $\varepsilon$)
     then do not bisect, end
     else
       determine $i$ as one of the following four cases
       case 1—cell has oversized neighbours on vertical boundary only
       case 2—cell has oversized neighbours on horizontal boundary only
       case 3—cell has oversized neighbours on both horizontal and vertical boundaries
       case 4—cell has no oversized neighbours
       perform case ($i$) of
       1. bisect in x-direction, end
       2. bisection in y-direction, end
       3. do not bisect, end
       4. if (concentration differential is bigger in x-direction)
            then bisect in x-direction, end
            else bisect in y-direction, end
```

Scanning the mesh

Two basic approaches to scanning the mesh are apparent. The first involves inspecting a cell: if the inspection results in a decision to bisect, then bisecting it, and then selecting a next cell for inspection. The second approach involves separating the scan into an inspection phase, followed by a bisection phase. In the inspection phase, the current mesh is scanned passively, and a list of cells to be bisected is built up (the bisection list). Subsequently, the cells in the bisection list are subdivided to form the new mesh. Using an analogy to pointwise iterative methods for solving linear equations, we shall refer to the first approach as a successive scanning process and the second approach as a simultaneous scanning process.

As with pointwise iterative methods, the successive processes seem more natural from a programming point of view, since no storage of the bisection list is necessary. However, simultaneous scanning offers two significant advantages—one in the area of symmetry and the other in the case of mesh cell list overflow. Successive scanning processes require an order for selecting cells for inspection and bisection and the resulting mesh refinements depend on the order used. On the other hand, simultaneous scanning can be organized (see below) so that the resulting mesh refinement is independent of the order in which cells are inspected and subdivided. This latter order only influences the order in which cells appear in the bisection list. Hence, simultaneous processes can be expected to produce highly symmetric meshes from symmetric data. Despite a considerable effort in trying to devise effective cell orderings for successive processes by stacking or queueing neighbours of cells to be subdivided, we were unable to produce a strategy which gave results comparable in their symmetry properties to the corresponding simultaneous scans.

It may happen that the cell content tolerance has been set so low that the mesh scan will produce more cells than the maximum allowed by the cell list storage space. A successive scanning process will discover this in the middle of a scan, where there is little it can do to gracefully continue. However, after the inspection phase, a simultaneous process can determine how many cells will be produced if the bisections are carried out, while the current mesh is still intact. If this number will result in a list overflow, then remedial action can be taken. For example, in the algorithm given below, the tolerance level is raised by a factor, and the inspection phase repeated. The mesh refinement process has been described as consisting of repeated scans of the mesh, until one is completed in which no bisections were required. If care is taken to see that the mesh resulting from the bisections of each scan is an irregular rectangular mesh in the sense of the last section, then one can ensure that the mesh resulting from a simultaneous scanning process will be an irregular rectangular mesh despite the need to modify the strategy of mesh refinement to avoid list overflow.

In the next section, an example of results obtained with both successive and simultaneous scanning processes is given. High level descriptions of the processes are given here, both of which refer to the cell bisection decision algorithm of the previous subsection. The successive scan can be simply described as follows:

```
while (the current scan is not complete and the cell list is not full)
  1. get the next cell in the scanning order
  2. submit it to the cell bisection decision algorithm
  3. if (bisection is called for)
       then
       3.1. interpolate application data to get values for the child cells
       3.2. perform the bisection (i.e. update the cell list)
```

The inspection phase of our simultaneous scan maintains a count of the number of cells to be bisected—to be referred to as the bisection count. As long as the number of cells in the current mesh plus the bisection count is less than the cell list capacity, the inspection phase maintains the bisection list of cells to be subdivided during the bisection phase. This list consists of the indices of such cells, and of the values of data for the cells to be created by bisection, obtained by interpolation of the data in the current mesh. A convenient place to store this list is in the unused part of the array of cell descriptions that forms the IRM representation. The bisection list will fit there if and only if the subsequent bisection phase can be carried out without cell list overflow. If the bisection count does exceed the space available, building of the bisection list is suspended, but the inspection scan continues, and maintains the bisection count, so as to be able to print out a useful monitoring message.

It is significant, in helping to maintain the symmetries of the refined meshes from scan to scan, that the interpolation of application data be done during the inspection phase, since during the bisection phase the current mesh may lose some of its symmetry 'temporarily' while in transition from one level of local refinement to the next. Hence, this interpolation data must be stored in the bisection list, along with the indices of cells to be bisected. The inspection scan can be described as:

```
for each cell of the current mesh
  1. submit it to the cell bisection decision algorithm
  2. if bisection is called for and the cell is not already in the bisection list
       then
       2.1. set bisection count increment to 1
       2.2. if neighbours at ends of proposed bisection line must
            be bisected to meet restrictions (2) or (3) of the last subsection
            then
            2.2.1. if neighbours are not already in bisection list then increment bisection count
                   increment and
                     temporarily store neighbour's indices
       2.3. if bisection count + increment $\leqslant M$
            then
            2.3.1. interpolate applications data to get values for child cells
            2.3.2. enter indices and data into bisection list
```

The purpose of 2.2 is to ensure that; after each scan, the mesh qualifies as an irregular rectangular mesh. The bisection phase can then be described as:

```
  if the bisection count plus the number of cells in the current mesh does not exceed the cell list
    capacity
    then
    1. for each cell in the bisection list perform the bisection
       (i.e. update the cell list)
    else
    2. multiply the cell content tolerance by 4
    3. repeat the inspection phase
```

The use of the factor 4 in 2. is a simple, if rather arbitrary, form of remedial action to avoid cell list overflow. It seems effective in the tests we have performed.

SOME EXAMPLES

In the examples of this section, the mesh refinement process has been used to produce a zone of refined regular cells in the part of the domain where the function to which the mesh is being tailored is 'interesting', separated by a transition zone of irregular cells from the remaining part of the domain where the coarse mesh is regarded as adequate. The functions used, $f(x, y)$, take on positive values, vary significantly in magnitude over the domain, and are interesting in their regions of high values. To obtain a zone of regular cells in these regions, we define f_c to be the lower limit of what is considered as a high value, and define the concentration for generating the mesh to be

$$C(x, y) = \text{minimum}\,(f(x, y), f_c)$$

Figure 2 shows the Gaussian lump

$$f(x, y) = 100 \exp\left(-(x-4{\cdot}5)^2/7{\cdot}2-(y-3)^2/5{\cdot}8\right) \tag{6}$$

in the rectangle $0 \leqslant x \leqslant 9$, $0 \leqslant y \leqslant 6$, and Figure 3 shows the concentration obtained by cutting it off at $f_c = 20$. Two meshes were generated from this concentration, starting with the regular

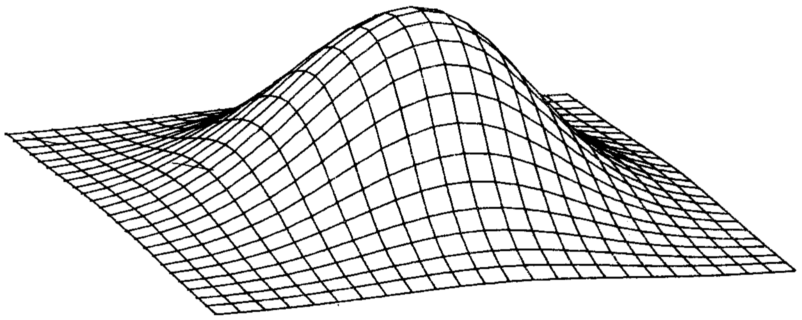
Figure 2. Gaussian lump of height 100 (equation (6))

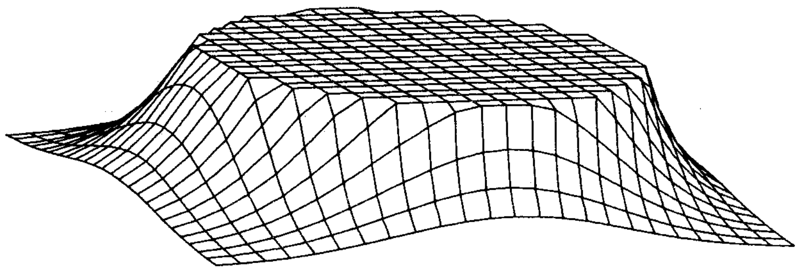
Figure 3. Gaussian lump of Figure 2 cut off at height 20

rectangular mesh of 25 cells shown in Figure 4, and using the cell bisection algorithm of the first part of the last section with maximum cell content of $\varepsilon = 10$. Figure 5 shows the mesh resulting from the simultaneous scanning algorithm of the second part of the last section and Figure 6 shows the result of using the successive scanning algorithm. All the interior cells have been bisected at least once in each direction (as a result, they are commensurate with the boundary strip of cells). However, in the centre a core of cells has been bisected once more in each direction resulting in 124 cells in Figure 5 and 120 in Figure 6.

The lack of symmetry in Figure 6 is quite obvious. Although the mesh has only a small number of cells in this example, it is typical of the difference between the two scanning techniques. The

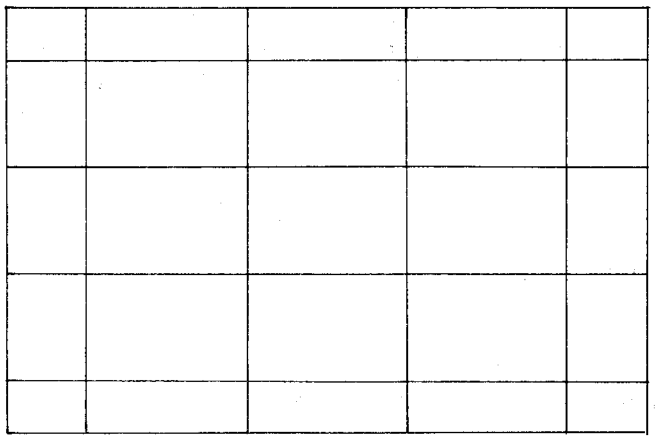
Figure 4. Basic regular rectangular mesh of 25 cells

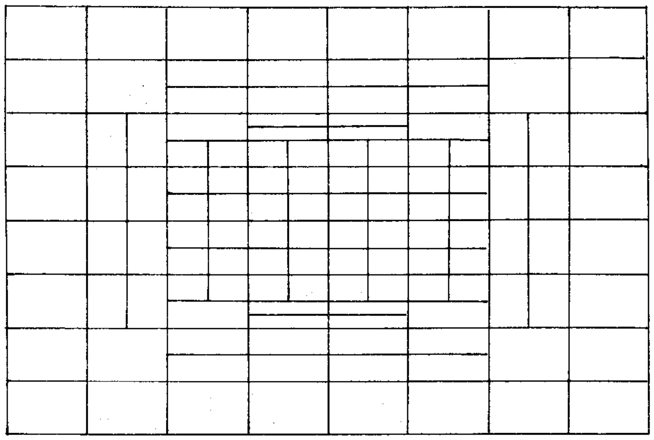
Figure 5. Irregular rectangular mesh generated by simultaneous scan and concentration of Figure 3

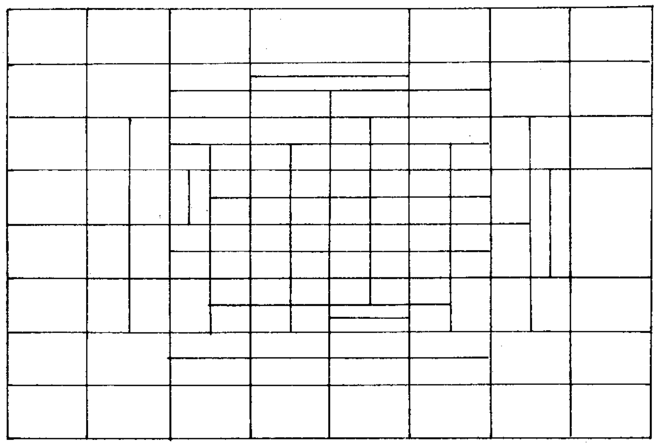
Figure 6. Irregular rectangular mesh generated by successive scan and concentration of Figure 3

number of bisections undergone by a cell of the basic mesh which lies in the refined core is determined by its original area A, the cut-off value, f_c, and the cell content maximum ε as

$$L = \lceil \log_2 (Af_c/\varepsilon) \rceil \qquad (7)†$$

This formula is helpful in determining ε to obtain a desired value of L, typically 2 or 4. Since, for these examples, the interior cells of the basic mesh all have the same area, L is a property of the mesh refinement process parameters and it will be referred to as the bisection level of the mesh.

As a second example, we look at meshes generated from the solution of the constant coefficient diffusion convection equation

$$f(x, y) = 200 \,.\, \exp(x/80) K_0(5\sqrt{(x^2/1600 + y^2/400)}) \qquad (8)$$

that represents a two-dimensional steady-state plume of effluent spreading into an infinite half-plane; $K_0(z)$ is the K-type Bessel function of order zero. The surface representation of this function is shown in Figures 7 and 8 for the rectangle $50 \leqslant x \leqslant 2050$; $-80 \leqslant y \leqslant 80$. We have

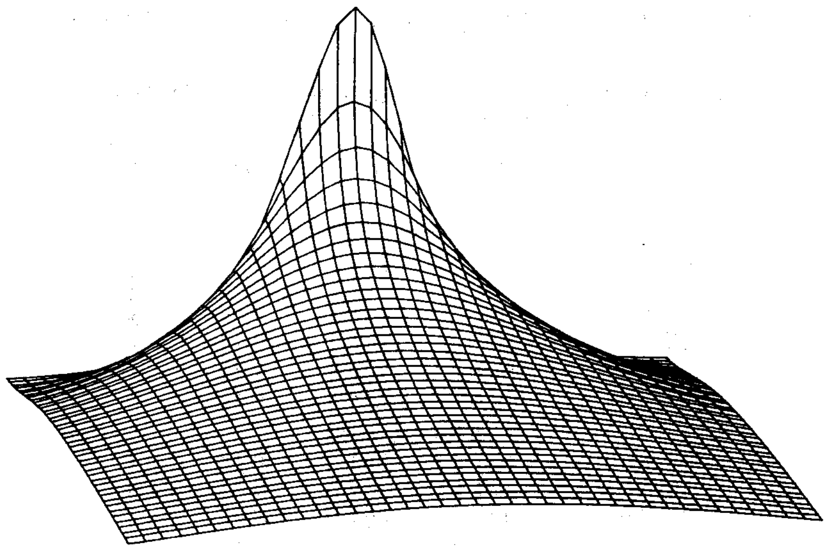

Figure 7. Effluent plume profile used for generating meshes of Table I (end view)

generated a series of increasingly refined meshes from this function by decreasing the cell content maximum, and tabulated some summary data on them in Table I. The basic mesh used for this example had 10 cells along the horizontal side of the region and 8 cells along the vertical. The boundary cells had half the size of the interior cells in the dimension perpendicular to the boundary as in Figure 4; hence the dimensions of the interior cells of this basic mesh are 222×30. The function cut-off level was chosen to be $f_c = 30$, and at this value the content of cells in the basic mesh is 200,000. Columns A and B of Table I contain the cell content maximum and the maximum length of the cell list; these are parameters to the mesh refinement process. For the entries in the lower three rows of the table, they had values for which the mesh refinement process attempted to generate more cells than the cell list could accommodate, i.e. cell list

† $\lceil x \rceil$ denotes the smallest integer $\geqslant x$.

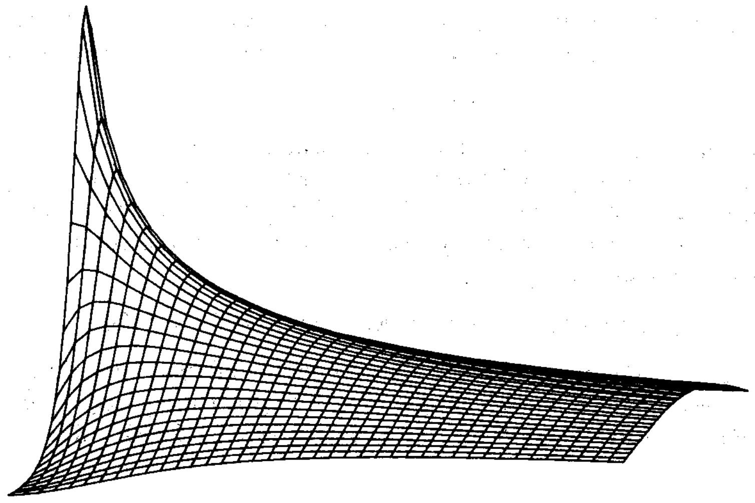

Figure 8. Effluent plume profile used for generating meshes of Table I (side view)

Table I. Summary of mesh refinement statistics for the second example (plume profile)

A	B	C	D	E
	No cell list overflow			
60,000	400	136	0·71	2
30,000	400	248	0·69	3
15,000	800	447	0·69	4
12,000	800	567	0·73	5
10,000	800	659	0·72	5
8,000	800	787	0·71	5
	Cell list overflow†			
15,000	400	242	0·68	†
12,000	400	244	0·68	†
10,000	600	546	0·75	†

† ε was raised to 4ε during mesh refinement.
A, cell content maximum (ε).
B, maximum number of cells allowed.
C, number of cells generated.
D, ratio of numbers of regular cells to total.
E, bisection level.

overflow. In this case, when the cell content maximum was replaced by four times its original value the resulting mesh could fit into the cell list and is reported in the table. It is of interest to note that the proportion of regular cells in the resulting meshes, virtually all of which lie in the refined zone, is fairly constant at about 70 per cent, with a transition zone comprised of about 30 per cent of the cells. These figures are quite typical of other tests done with similar strategies, although concentrations can be defined which give rise to sharper transition zones. In particular, for a bisection level of two, meshes with transition zone of 10–15 per cent of the cells are not difficult to generate from concentrations with sharp fronts or drops between a region of high concentration and a region of low concentration.

CONCLUSIONS

In this paper a well-defined class of meshes (the irregular rectangular meshes) admitting local refinements has been posed and algorithms given which refine the mesh according to the features of prescribed mesh functions. The algorithms have the firm properties of terminating with an allowable mesh, and requiring no more memory than needs to be allocated to store the list of cells for the mesh of maximum size. They have the less firm property of producing meshes with a high degree of the symmetries of the generating function. Our tests suggest that while it is possible to generate refined zones of regular cells, a significant proportion of the cells of an irregular mesh will be irregular cells.

The irregular rectangular meshes are particularly simple, in that any cell is determined by four parameters; we have used the cell centre co-ordinates and its dimensions. However, there does not appear to be any fundamental reason why these ideas cannot be directly extended to meshes of triangles in the plane or bricks in three dimensions, and other basic mesh cell shapes. In such cases, restrictions on the configuration of neighbours which may be used to define the class of meshes will pose the same difficulties for the properties of mesh refinement algorithms as discussed here.

ACKNOWLEDGEMENT

The support of the Natural Sciences and Engineering Research Council Canada and helpful discussions with D. C. L. Lam of the Canada Centre for Inland Waters are gratefully acknowledged.

REFERENCES

1. A. A. Amsden and C. W. Hirt, 'A simple scheme for generating general curvilinear grids', *J. Comp. Phys.* **11**, 348–359 (1973).
2. I. Babuska, 'The self adaptive approach in the finite element method', in *The Mathematics of Finite Elements and Applications II, MAFELAP 1975* (Ed. J. R. Whiteman), Academic Press, New York, 1976, pp. 125–142.
3. A. Brandt, 'Multi-level adaptive solutions to boundary-value problems', *M. Comp.* **31**, 333–390 (1977).
4. W. R. Buell and B. A. Bush, 'Mesh generation—a survey', *Trans. ASME J. Eng. Ind.* **95**, 332–338 (1973).
5. A. Bykat, 'Automatic generation of triangular grid: I—Subdivision of a general polygon into convex subregions, II—Triangulation of convex polygons', *Int. J. num. Meth. Engng*, **10**, 1329–1341 (1976).
6. G. F. Carey, 'A mesh refinement scheme for finite element computations', *Comp. Meth. Appl. Mech. Eng.* **7**, 93–105 (1976).
7. J. C. Cavendish, 'Local mesh refinement using rectangular blended finite elements', *J. Comp. Phys.* **19**, 211–228 (1975).
8. J. C. Cavendish, W. J. Gordon and C. A. Hall, 'Substructured macro elements based on locally blended interpolation', *Int. J. num. Meth. Engng*, **11**, 1405–1421 (1977).
9. J. C. Cavendish, 'Automatic triangulation of arbitrary planar domains for the finite element method', *Int. J. num. Meth. Engng.*, **8**, 679–696 (1975).
10. J. A. George, 'Computer implementation of the finite element method', *Thesis*, Stanford University (1971).
11. W. J. Gordon and C. A. Hall, 'Construction of curvilinear co-ordinate systems and applications to mesh generation', *Int. J. num. Meth. Engng*, **7**, 461–477 (1973).
12. R. E. Jones, 'A self organizing mesh generation program', *Trans. ASME J. Press. Vessel Tech.*, sect. J, **96**, 193–199 (1974).
13. D. C. L. Lam and R. B. Simpson, 'Modelling coastal effluent transport using a variable finite difference grid', in *Advances in Computer Methods for Partial Differential Equations, II, Proc. IMACS* (Ed. R. Vichnevetsky), 1977.
14. M. Lentini and V. Pereyra, 'An adaptive finite difference solver for nonlinear two pont boundary value problems with mild boundary layers', *SIAM J. Num. Anal.* **14**, 91–111 (1977).
15. B. A. Lewis and J. S. Robinson, 'Triangulation of planar regions with applications', *Comp. J.* **11**, 324–330 (1978).
16. R. J. Melosh and P. V. Marcal, 'An energy basis for mesh refinement of structured continua', *Int. J. num. Meth. Engng*, **11**, 1083–1091 (1977).
17. W. C. Rheinboldt and C. K. Mesztenyi, 'On a data structure for adaptive finite element mesh refinements', *Technical Report TR-660*, University of Maryland (1978).

18. R. D. Russell and J. Christiansen, 'Adaptive mesh selection strategies for solving boundary value problems', *SIAM J. Num. Anal.* **15**, 59–80 (1978).
19. E. G. Sewell, 'An adaptive computer program for the solution of Div $(p(x, y)$ grad $u) = f(x, y, u)$ on a polygonal region', in *The Mathematics of Finite Elements and Applications II, MAFELAP 1975* (Ed. J. R. Whiteman), Academic Press, New York, 1976, pp. 343–353.
20. D. J. Turcke and G. M. McNeice, 'Guidelines for selecting finite element grids based on optimization study', *Comp. Struct.* **4**, 499–519 (1974).
21. O. C. Zienkiewicz and D. V. Phillips, 'An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' coordinates', *Int. J. num. Meth. Engng.*, **3**, 519–528 (1971).

AUTOMATIC LOCAL REFINEMENT FOR IRREGULAR RECTANGULAR MESHES

by

R. Bruce Simpson

Research Report CS-78-19

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

March 1978

Abstract

A class of rectangular meshes which admit local refinement of mesh cells is defined. Some objectives for algorithms to perform local refinements in this class to resolve features of a given mesh function are discussed, i.e. properties of termination with a mesh in the specified class, memory utilization, symmetry and fragmentation of the meshes generated. Some algorithms and examples are given.

Printing Requisition/Graphic Services

Dept. No. 01531

Title or Description
AUTOMATIC LOCAL REFINEMENT FOR IRREGULAR RECTANGULAR MESHES

Date 13 March 1978
Date Required
Account ~~[illegible]~~ 126-6020

Signature Virginia
Signing Authority R.B. Simpson ~~XXXAdams~~
[illegible] -4

Department Computer Science
Room 5100B
Phone 3402

Delivery
☐ Mail
☐ Pick-up
[✓ Via Stores]
☐ Other

1. Please complete unshaded areas on form as applicable. (4 part no carbon required).
2. Distribute copies as follows: White, Canary and Pink—Printing, Arts Library or applicable Copy Centre Goldenrod—**Retain.**
3. On completion of order, pink copy will be returned with printed material. Canary copy will be costed and returned to requisitioner, **Retain as a record of your charges.**
4. Please direct enquiries, quoting requisition number, to Printing/Graphic Services, Extension 3451.

Reproduction Requirements
[✓ Offset] ☐ Signs/Repro's ☐ Xerox
Number of Pages 33
Number of Copies 50

Type of Paper Stock
[✓ Bond] ☐ Book ☐ Cover ☐ Bristol ☐ Supplied

Paper Size
[✓ 8½ x 11] ☐ 8½ x 14 ☐ 11 x 17

Paper Colour
[✓ White] ☐ Other

Ink
☐ Black

Printing
[✓ 1 Side] ☐ 2 Sides

Numbering to

Binding/Finishing Operations
[✓ Collating] [✓ Corner Stitching] ☐ 3 Ring ☐ Tape ☐ Plastic Ring ☐ Perforating

Folding
Finished Size 3 staples

Cutting
Finished Size

Special Instructions
To be put into Faculty of Mathematics Research Report Covers. (ENCLOSED)

Cost: Time/Materials

	Fun.	Prod.Un.	Prod. Cl.	Opr. No.	Mins.	Total	
Signs/Repro's	1						
Camera	2						
Correcting & Masking Negatives	3						
Platemaking	4						
Printing	5						
Bindery	6		21	30			
			22	15			
Sub. Total Time							
Sub. Total Materials							
Prov. Tax							
Total							

Film Qty Size	**Plates** Qty Size & Type
Paper Qty Size	**Plastic Rings** Qty Size
Outside Services	

Graphic Services May 68 482-2