Data-Dependent Grids

P.K. Sweby

Numerical Analysis Report 7/87

This work forms part of the research programme of the Institute for Computational Fluid Dynamics at the Universities of Reading and Oxford.

Abstract

Extensions of the one-dimensional algebraic grid generation technique of equidistribution are investigated in two-dimensions. Firstly extensions producing quadralateral grids are considered before proceeding to an approximate equidistribution technique for obtaining unstructured triangular grids.

Keywords:

Grid Generation, Equidistribution, Two-Dimensional, Triangulation.

1. Introduction

The literature is full of numerical grid generation techniques which produce quadralateral grids, see [7] for an excellent compendium. One such technique is the equidistribution of some density function, which when a function of the data to be represented on the grid gives rise to an adaptive or data-dependent mesh. Equidistribution is an essentially one-dimensional technique, usually carried into two-dimensions by first generating a non-adaptive grid (say using body fitted coordinates) and then adapting this grid to equidistribute the density function along its coordinate lines.

In Section 2 we first look at one-dimensional equidistribuition and the choice of density function, one proposed by Carey and Dinh [2] to minimise the error of interpolation, in particular the L_2 error of piecewise-linear interpolation - the representation commonly used in finite element methods and some finite difference methods. In Section 3 some extensions to two dimensions which do not require the generation of an initial non-adaptive grid are presented. Section 4 then extends the technique to the generation of unstructured triangular grids, together with a post-processing regularisation. Finally comments on the implementation of techniques presented are given in Section 5.

All the techniques described here are illustrated with example grids, and although the initial motivation for the work was the generation of an initial grid for the Moving Finite Element method [5] , [8], their applications are much more widespread.

2. Equidistribution in one-dimension and choice of monitor function

One algebraic technique of one-dimensional grid generation is to distribute the nodal points, x_i , according to some (positive) density function $\rho(x)$. This is often referred to as equidistribtuion with respect to $\rho(x)$ (see e.g. [1],[2],[7]) since the nodes thus obtained are such that

$$\int_{x_i}^{x_{i+1}} \rho(x)\, dx = \text{constant} \quad \forall i . \tag{2.1}$$

If the distribution of points is considered as a mapping $x(\xi)$ where integer values of the parameter ξ represent the nodal positions (see Figure 1), we have

$$\xi k = \int_{x(o)}^{x(\xi)} \rho(x)\, dx$$

where k is a constant. This constant can easily be determined in terms of the integral of ρ over the whole line giving

$$\xi = N \frac{\int_{x(o)}^{x(\xi)} \rho(x)\, dx}{\int_{x_L}^{x_R} \rho(x)\, dx} \tag{2.2}$$

where x_L , x_R are the end points of the region, and N+1 the number of nodes (we assume subscripting starts at 0). Nodal positions x_i can then be obtained by solving (2.2) for x with $\xi = i$; Figure 2 shows this pictorially.

There are various methods of solution of (2.2) ([2]) , most of which involve numerical quadrature to evaluate the integrals and an iteration to solve $\xi - i = 0$. For example Gauss quadrature or the trapezodal rule can be used for the integration, and Newton or bisection for the iteration. See Section 5.

It remains now to choose the density function $\rho(x)$. If it depends in some way on the data, $u(x)$, then it becomes a monitor function for that data. The grid is then distributed according to the function/data that is represented on it. A common form of monitor function (see e.g. [1], [7]) is

$$\rho(x) = 1 + A|u_x| \tag{2.3}$$

for some constant A , however Carey and Dinh [2] obtained an expression for the optimal monitor function which minimises the H^m seminorm

$$|e|_m^2 = \int_a^b (e^{(m)})^2 \, dx$$

of the error when using kth order interpolation for $u(x)$ on the grid, namely

$$\rho(x) = [u^{(k+1)}]^{2/(2(k+1-m)+1)} \tag{2.4}$$

For the case of linear interpolation in the L_2 norm (k=1 , m=0) which we are interested in, this becomes

$$\rho(x) = [u''(x)]^{2/5} \qquad (2.5)$$

Some examples of grids produced using this monitor function are given in Figure 3.

3. Extensions to two-dimensions - Quadraleteral grids

In two dimensions, without further constraints, an "area" equidistribution, i.e. one of the form

$$\iint_{\Omega} \rho(x,y)\ d\Omega = \text{const} , \tag{3.1}$$

will not be unique unlike its one-dimensional counterpart. Also the analysis of Carey and Dinh [2] is not readily extendable to two dimensions, and so we seek to extend the equidistribution technique by applying it along "lines" in two dimensions. We shall use a monitor funtion of the form (2.5) from now on.

3.1 "Dimensional Reduction"

Probably the simplest extension of the equidistribution technique to two dimensions is to equidistribute "slabs" in the two coordinate directions [4]. That is, define slabs in the coordinate directions such that

$$\int_{x_i}^{x_{i+1}} \int_{y_o}^{y_M} u_{xx}^{2/5}(x,y)\ dy\ dx = \text{constant} \tag{3.2a}$$

and

$$\int_{y_j}^{y_{j+1}} \int_{x_o}^{x_N} u_{yy}^{2/5}(x,y)\, dx\, dy = \text{constant} \tag{3.2b}$$

See Figure 4.

The x and y coordinates of the slab boundaries are then obtained by solving the independent equations

$$\begin{aligned} \xi(x_i) &= i \\ \eta(y_j) &= j \end{aligned} \tag{3.3}$$

where

$$\xi(x) = N \frac{\int_{x_o}^{x} \int_{y_o}^{y_M} u_{xx}^{2/5}\, dy\, dx}{\int_{x_o}^{x_N} \int_{y_o}^{y_M} u_{xx}^{2/5}\, dy\, dx}$$

and

$$\eta(y) = M \frac{\int_{y_o}^{y} \int_{x_o}^{x_N} u_{yy}^{2/5}\, dx\, dy}{\int_{y_o}^{y_M} \int_{x_o}^{x_N} u_{yy}^{2/5}\, dx\, dy} \tag{3.4}$$

As can be seen from Figure 5 this technique, being highly one dimensional will distribute clustered grid lines across the entire region due to a feature of the data isolated to a small part of the region. Also, due to the averaging in the orthogonal coordinate direction, this method can be insensitive to some features of the data, giving sparsely spaced grid lines, as observed by Johnson [4].

3.2 "Dimensional Splitting"

Instead of equidistributing an averaged monitor function over a slab, thus producing a rectangular grid, we could perform a one-dimensional equidistribution along a line and track the loci of the nodes as the line moves across the region - see Figure 6. If this is done in both coordinates directions a grid is formed on the region - Figure 7.

It would obviously be impractical to generate a grid in this manner, but a simple iteration can be used to obtain the nodes x_{ij} and y_{ij} as follows

$$\xi(x,y) = N \frac{\int_{x_o}^{x} u_{xx}^{2/5}(x,y)dx}{\int_{x_o}^{x_N} u_{xx}^{2/5}(x,y)\,dx}$$

$$\eta(x,y) = M \frac{\int_{y_o}^{y} u_{yy}^{2/5}(x,y)dy}{\int_{y_o}^{y_M} u_{yy}^{2/5}(x,y)\,dy} \qquad (3.5)$$

then for each node i,j

1. Estimate $x_{ij}^{(o)}$, $y_{ij}^{(o)}$ e.g. by transfinite interpolation [7] between equidistributed boundary points.
2. Obtain $x_{ij}^{(k+1)}$ by solving $\xi(x_{ij}^{(k+1)}, y_{ij}^{(k)}) = i$

3. Obtain $y_{ij}^{(k+1)}$ by solving $\eta(x_{ij}^{(k+1)}, y_{ij}^{(k+1)}) = j$

4. Repeat from 2. until convergence.

Figure 8 shows some grids thus produced.

Grids generated in this manner do not equidistribute the monitor function along the grid lines themselves, rather along lines in the coordinate directions as shown in Figure 9. A technique which can be used to equidistribute along grid lines is the elliptic generator of Thompson [6]. Anderson [1] showed that by suitable choice of control functions (i.e. source terms) Thompson's scheme acts as an equidistribution scheme along grid lines.

The equations generating the grid are

$$\nabla^2\xi = P \qquad\qquad \nabla^2\eta = Q \tag{3.6}$$

where P and Q are the grid control functions. Interchanging dependent and independent variables gives

$$\alpha(\underset{\sim}{r}_{\xi\xi} - \Phi\underset{\sim}{r}_{\xi}) - 2\beta\underset{\sim}{r}_{\xi\eta} + \gamma(\underset{\sim}{r}_{\eta\eta} - \Psi\underset{\sim}{r}_{\eta}) = 0 \tag{3.7}$$

where $\underset{\sim}{r} = (x,y)^T$,

$$\alpha = x_\eta^2 + y_\eta^2 \;, \quad \beta = x_\xi x_\eta + y_\xi y_\eta \qquad \gamma = x_\xi^2 - y_\xi^2 \;,$$

$$P = \frac{\alpha}{J^2}\Phi , \qquad Q = \frac{\gamma}{J^2}\Psi$$

and

$$J = x_\xi y_\eta - x_\eta y_\xi$$

By choosing $\Phi = \frac{1}{w_1} \frac{\partial w}{\partial \xi}$

$$\Psi = \frac{1}{w_2} \frac{\partial w}{\partial \eta} \qquad (3.8)$$

w_1 and w_2 are equidistributed along grid lines, and so we choose

$$w_1 = u_{\xi\xi}^{2/5}$$

$$w_2 = u_{\eta\eta}^{2/5} \qquad (3.9)$$

which are easily calculated in the process of solving (3.7).

Figure 10 shows a grid generated using this technique, however for complicated u(x,y) solution of (3.7) seems proned to divergence - and under relaxation is needed.

All of these techniques generate quadralateral grids, triangular grids can be produced by either joining diagonals, or by applying a triangulation routine, e.g. Delauney [3], to the nodes of the quadralateral grid. Figure 11 gives some examples. We can, however, use equidistribution techniques to generate a triangular grid directly, without starting with quadralaterals, as described in the next section.

4. Triangular grids - Approximate Equidistribution

For triangular grids, there are no (unique) paths crossing the region along which to equidistribute, recalling that the equidistribution uses the total integral along a line as well as the local integral between nodes. This is a consequence of the unstructured nature of triangular grids.

Therefore, instead of an exact equidistribution, we approximately equidistribute along element (the triangles) edges, i.e. if node i is connected to node j, then

$$\int_{s_i}^{s_j} \rho(s)\, ds \leq \delta \qquad (4.1)$$

where s is a parameterisation of the line connecting the two nodes and $\delta > 0$ is a "tolerance".

The density function used is the directional analogue of (2.5), namely

$$\rho(x,y) = (\cos^2\theta\, u_{xx} + 2\cos\theta\, \sin\theta\, u_{xy} + \sin^2\theta\, u_{yy})^{2/5} \qquad (4.2)$$

where

$$\tan\theta = \frac{(y_j - y_i)}{(x_j - x_i)} \qquad (4.3)$$

There are various ways to construct a triangular grid pocessing such a property, we outline two such here.

4.1 Subdivision of Elements

One approach to approximate equidistribtuion over element edges is to start with a coarse mesh of a few elements and then to subdivide recursively until (4.1) is satisfied on all edges. If Delauney triangulation is used, the grid can be easily and cheaply retriangulated after each subdivision since the Delauney algorithm constructs the grid by adding one node at a time to an existing grid. Also, since Delauney can reconnect nodes during the course of triangulation, one element edge (not its circumference) at a time should be subdivided. Figure 12 illustrates the process, and Figure 13 shows grids produced this way. Note that a major drawback of this method is the tendency of excessive clustering of elements, and the lack of reflection of any one-dimensional behaviour of $u(x,y)$.

4.2 Contouring

An alternative approach is to generate a set of nodes and then to triangulate. The choice of nodes is not usually obvious, nor unique! The method we use here is to approximately follow the contours of the data function $u(x,y)$. This is achieved in the following manner.

1. Equidistribute nodes along the boundary of the region
2. For each boundary node
   (a) Determine the direction of the contour passing through the node using $\Psi = \tan^{-1}\left\{\frac{-u_x}{u_y}\right\}$ (4.4)
   (b) If the contour is directed out of region delete the node if it does not help describe the region else
   Approximate contour as straighline with parameter s , of length such that

$$\int_{node}^{s} \rho(s)ds = \delta \qquad (4.5)$$

   where $\rho(s)$ is given by (4.2). If the line lies entirely within the region place a node at the endpoint and follow new contour direction, otherwise place node at intersection with boundary and move onto next boundary node.
3. Triangulate.

Figure 14 illustrates the process. Improved results are obtained if in step one, (4.2) is multiplied by

$$|\hat{\underset{\sim}{\theta}} \cdot \hat{\underset{\sim}{\phi}}| \qquad (4.6)$$

where $\hat{\underset{\sim}{\theta}}$ is a unit vector in the direction of the boundary, and $\hat{\underset{\sim}{\phi}}$ is a unit vector in the direction normal to the contour. This helps to space the contours followed so that (4.1) is satisfied (although not guaranteed) in the interior of the region. Figure 15 shows grids generated by this technique.

4.3 Regularisation

Finally, although the triangulation will produce a grid with certain properties determined by the triangulation e.g. with Delauney the no obtuse angle property - it may not give the best representation of the data.

For example, consider a quadralateral as in Figure 16. A triangulation may connect the diagonal normal to the contours, which is clearly a worse representation than the connection of the other diagonal. The characterisation of these possible cases is that with the first connection, the diagonal has the larger value of $\int \rho(s)ds$. This suggests an iterative regularisation in which pairs of elements are considered as a quadralateral, and if appropriate and geometrically possible the diagonals are swapped.

The results of such a regularisation are contrasted in Figure 17. Although this process can produce elements with a large aspect ratio and/or obtuse angles, the improvement of data representation can be significant when using methods such as moving finite elements, where subsequent adaption of the grid relies heavily on the initial grid.

5. Comments on implementation

For equidistributing points along a line the most robust method of solution of (2.2), and similar, is to use a trapezoidal rule for the integration, "marching" from x_o until $\xi > i$ and then interpolating back to find x_i such that $\xi = i$. For rapidly changing u" a large number of evaluation points may be needed to accurately obtain the distribution. For less pathelogical u" Gauss quadrature coupled with Newton iteration is fast and adequate.

All the techniques presented here involve the second, and sometimes first, derivatives of the data. If that data has a simple analytic form then calculation of these quantites is straightforward, otherwise finite differences may be used as an approximation. If the data is not available as an analytic function then point values are needed on a grid which is fine enough to characterise the underlying function and finite differences are again used. This approach has been very successfully used in practice.

For the contouring algorithm, problems arise if there are internal extrema whose contours do not intersect the boundary. In such cases additional boundaries in the form of branch cuts should be introduced which do intersect these features of the data (see Figure 18).

The tolerance δ of (4.1) can be chosen semi-automatically by specifying the ideal minimum number of nodes per side, N_{min} . Then we can use

$$\delta = \min_{sides} \{ \int \rho(s)ds \,/ (N_{min}+1) \} \tag{5.1}$$

Again this technique works well in practice, and is easily modified to cope with situations where a fixed number of nodes (e.g. zero) is required on one or more individual sides.

Finally, the Delauney algorithm triangulates a convex region but this restriction is easily overcome to deal with non convex regions. The algorithm is applied as usual, ensuring that boundary nodes are correctly connected within the triangulation, elements external to the region are then deleted. A simple test for an external element is to connect its centroid to a known interior point and count the number of boundary intersections - odd if element is outside region, even if inside. See Figure 19.

References

1. D.A. Anderson "Constructing Adaptive Grids with Poisson Grid Generators", Proc. Numerical Grid Generation in Computational Fluid Dynamics, (1986) Eds. J. Häuser & C.Taylor, Pineridge Press pp 125-136.

2. G.F. Carey & H.T. Dinh "Grading Functions and Mesh Redistribution" SIAM J.Num. Anal. 22 (1985) pp 1028-1040.

3. P.J. Green & R. Sibson, "Computing Direchlet Tessellation in the Plane", Comp. J. 21 (1978), pp 168-173.

4. I.W. Johanson "The Application of an Equidistributing Principle to the Generation of Initial Meshes for the Solution of Non-Linear Partial Differential Equations in One and Two Space Dimensions". AWRE Report (1984).

5. K. Miller & R. Miller "Moving Finite Elements Part I", SIAM J. Num. Anal. 18 (1981) pp 1019-1032.

6. J.F. Thompson, F.C. Thames & C. W. Mastin, "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Fields Containing any Number of Arbitrary Two-Dimensional Bodies", JCP 15 (1974) pp 299-319.

7. J.F. Thompson, Z.U.A. Warsi & C.W. Mastin, "Numerical Grid Generation Foundations and Applications" (1985) North-Holland.

8. A.J. Wathen & M.J. Baines "On the Structure of the Moving Finite Element Equations", IMA J. Num. Anal. 5 (1985) pp 161-182.

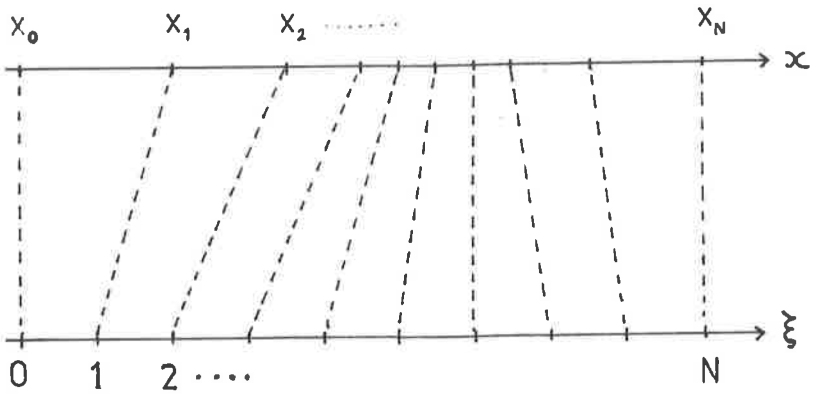


Figure 1. The mapping x(ξ).

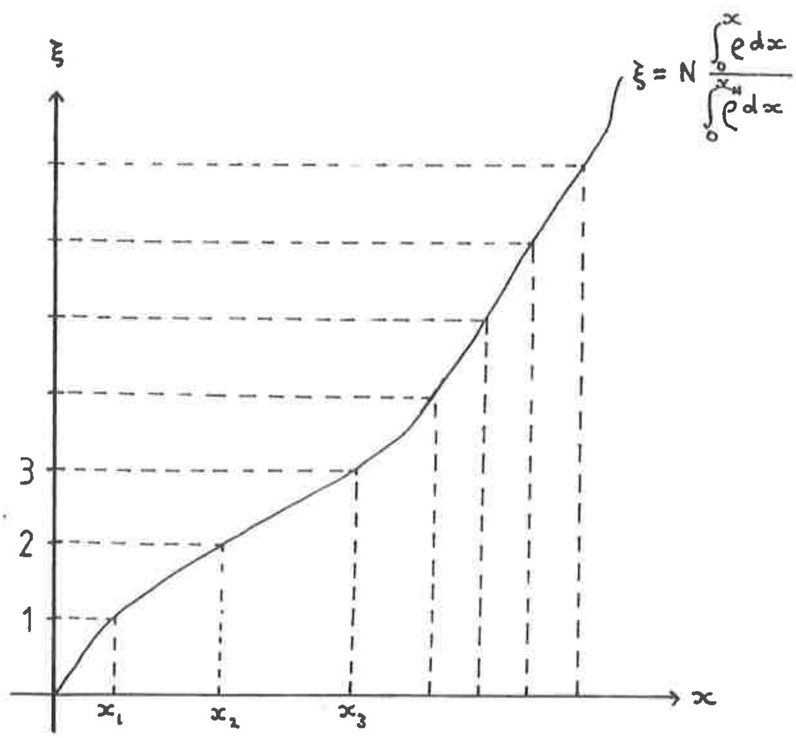
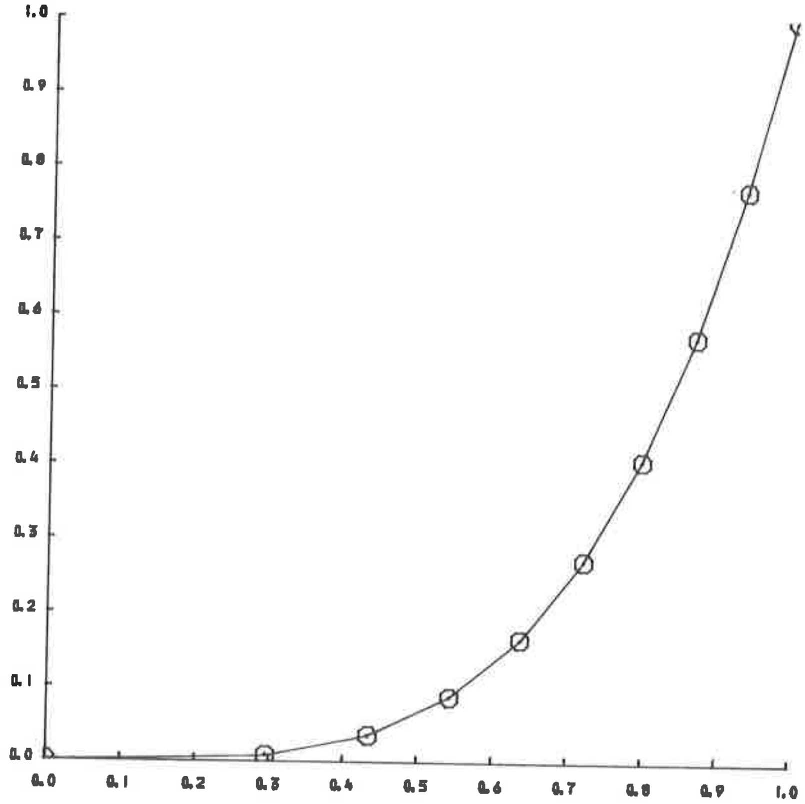
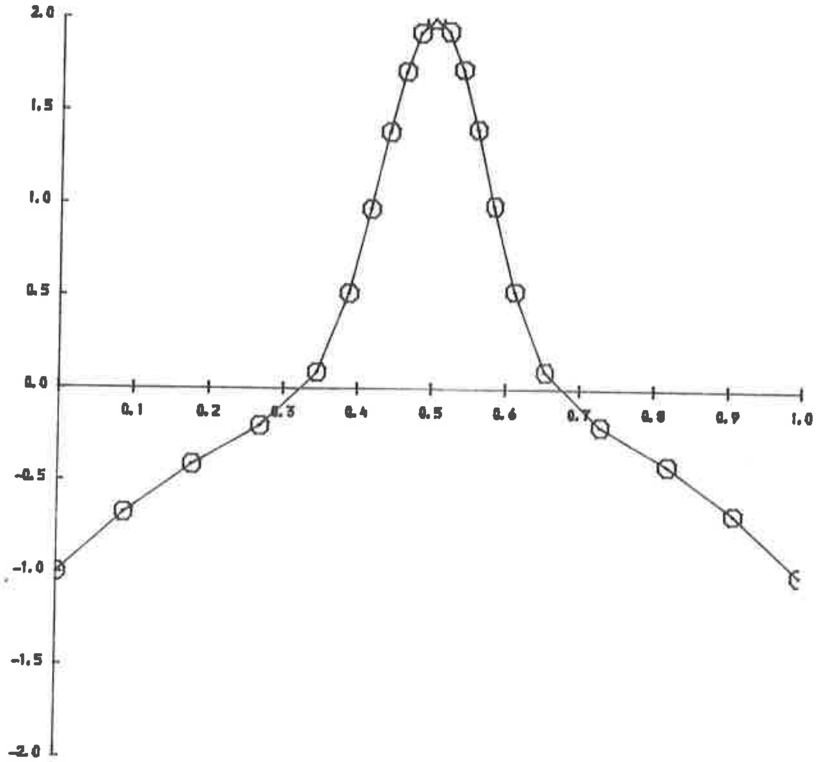


Figure 2. Nodal positions obtained by solving (2.2) with ξ=i



(a) $u = x^4$



(b) $u = 2e^{-100(y-\frac{1}{2})^2} - 4(y-\frac{1}{2})^2$

Figure 3. One-dimensional equidistribution of nodes with $\rho = (u'')^{2/5}$

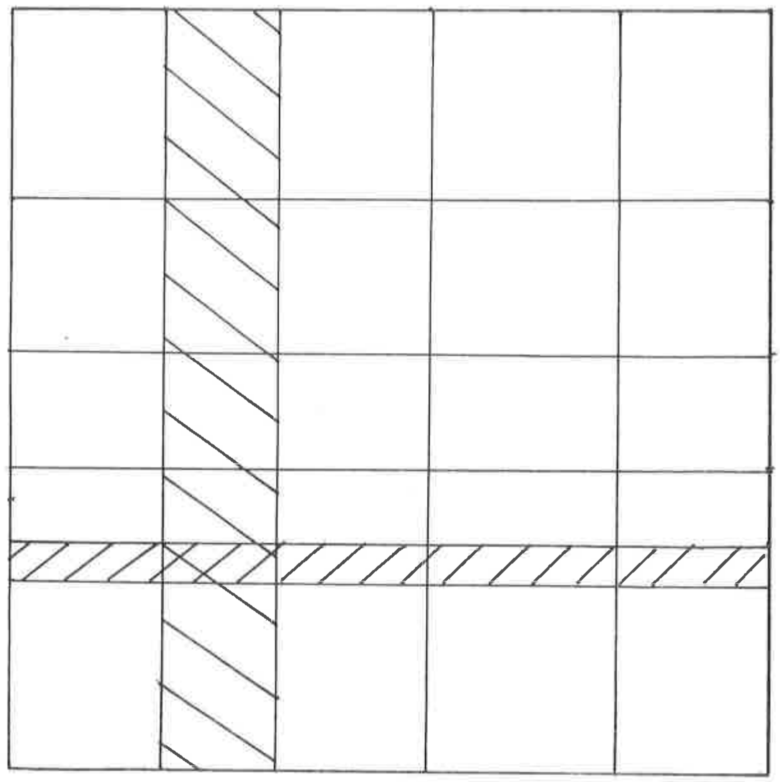
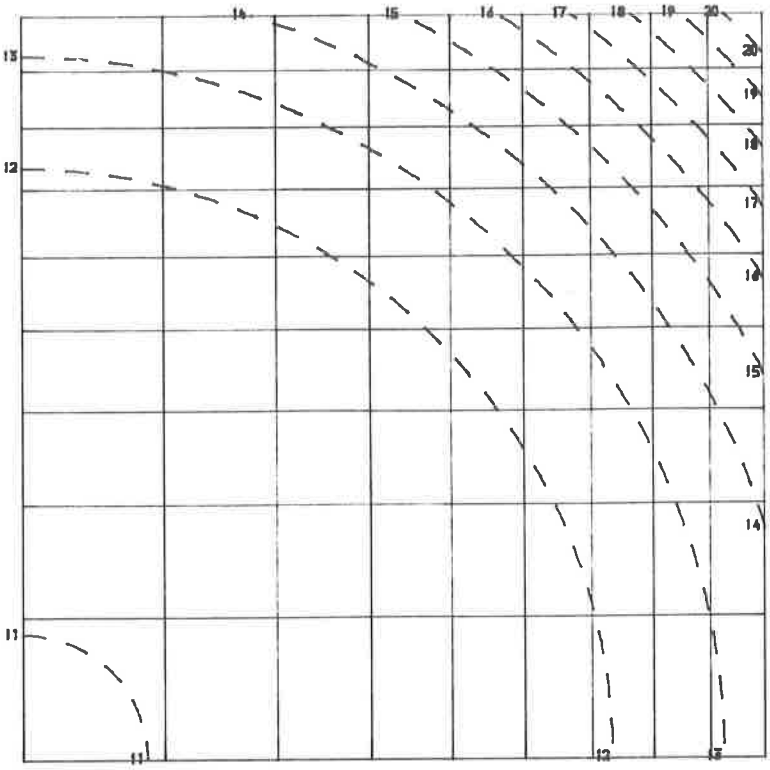
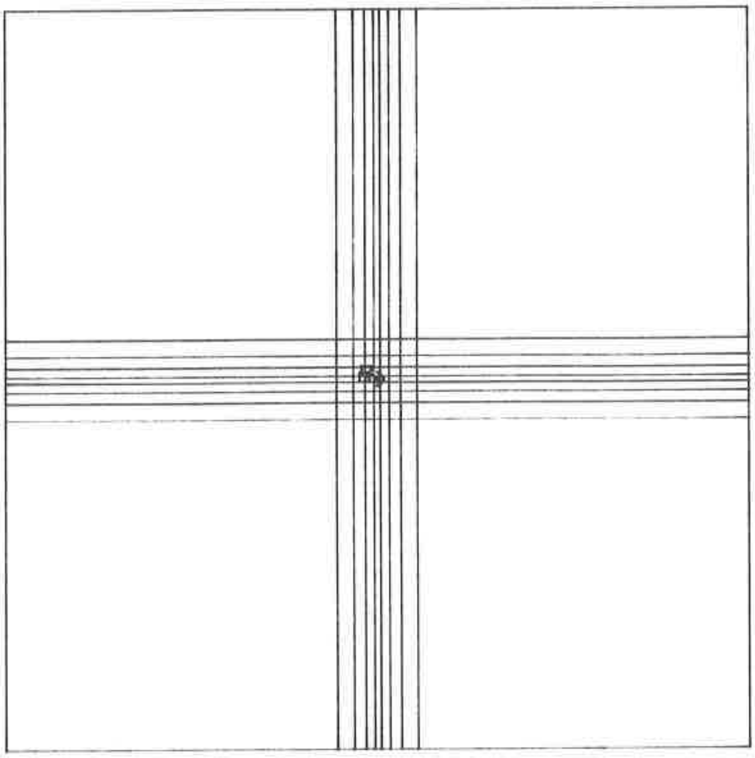


Figure 4. Equidistribution of slabs in the two coordinate directions.



(a) $u = (x^2 + y^2)^2$ (contours shown dotted).



(b) $u = e^{-50\sqrt{(x-½)^2 + (y-½)^2}}$ - a solitary peak

Figure 5. Grids produced using slab equidistribution

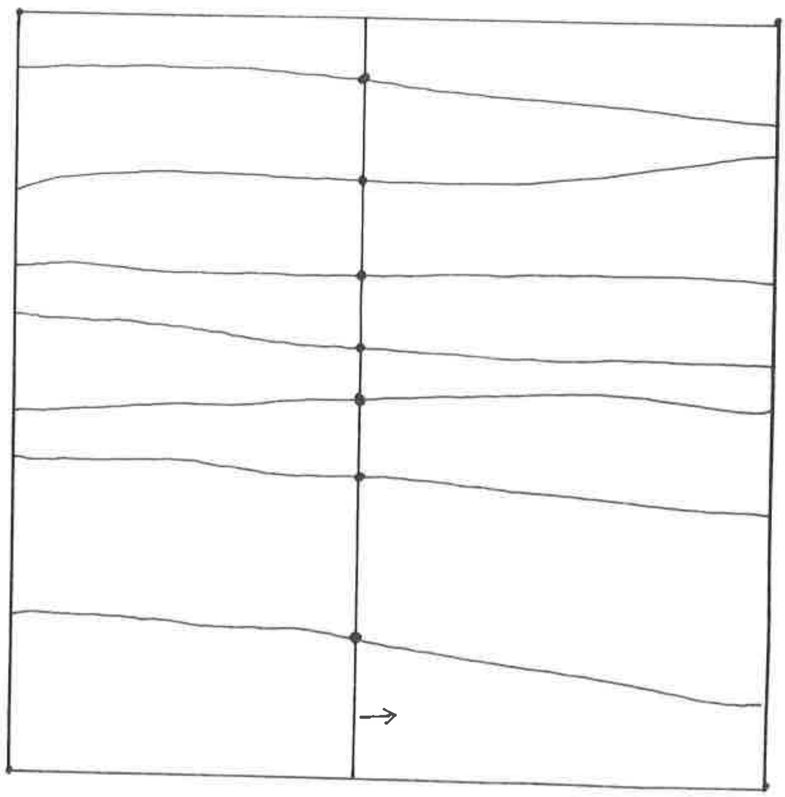
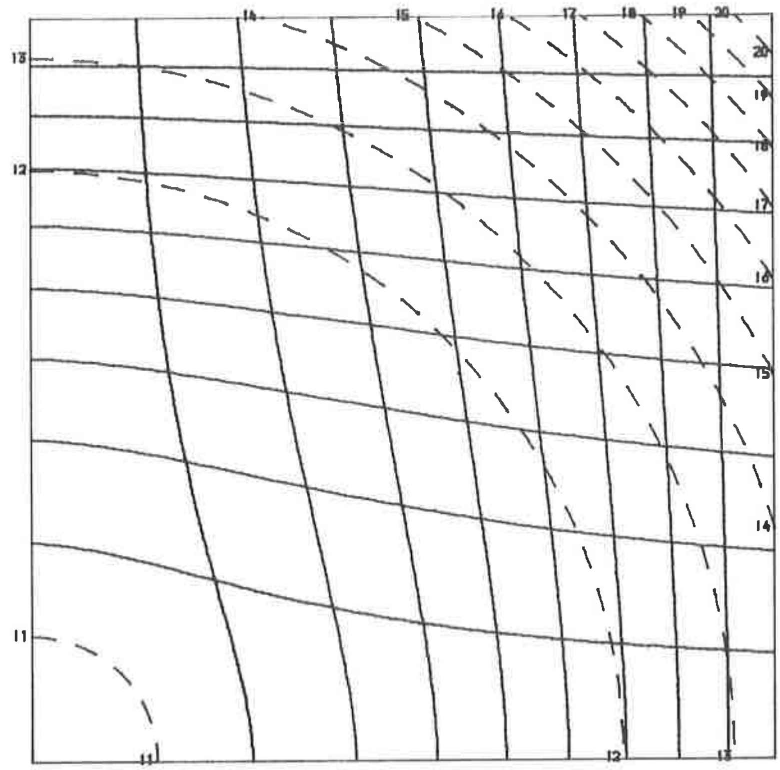
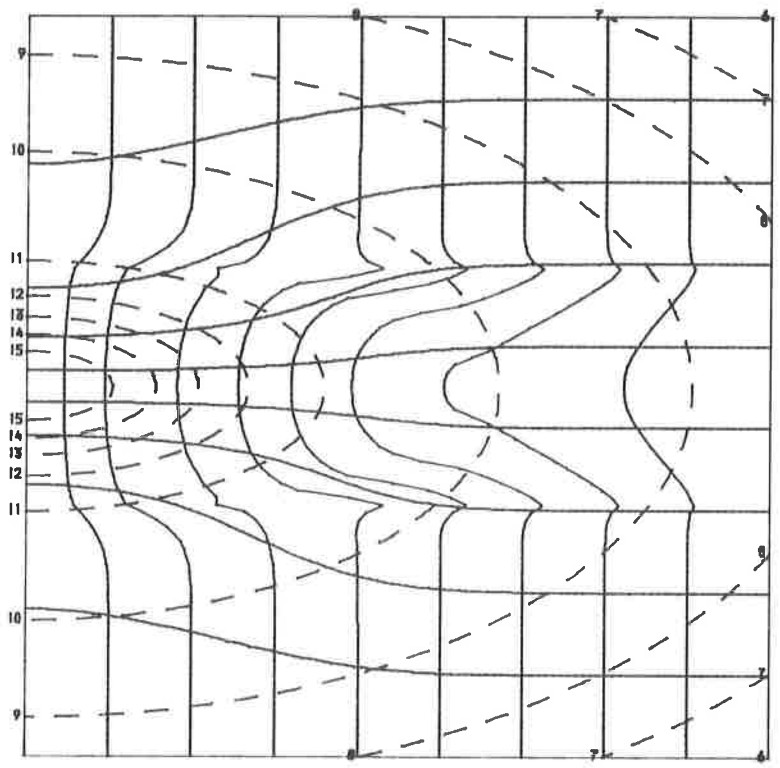


Figure 6. Tracking the loci of a one-dimensional equidistribution.

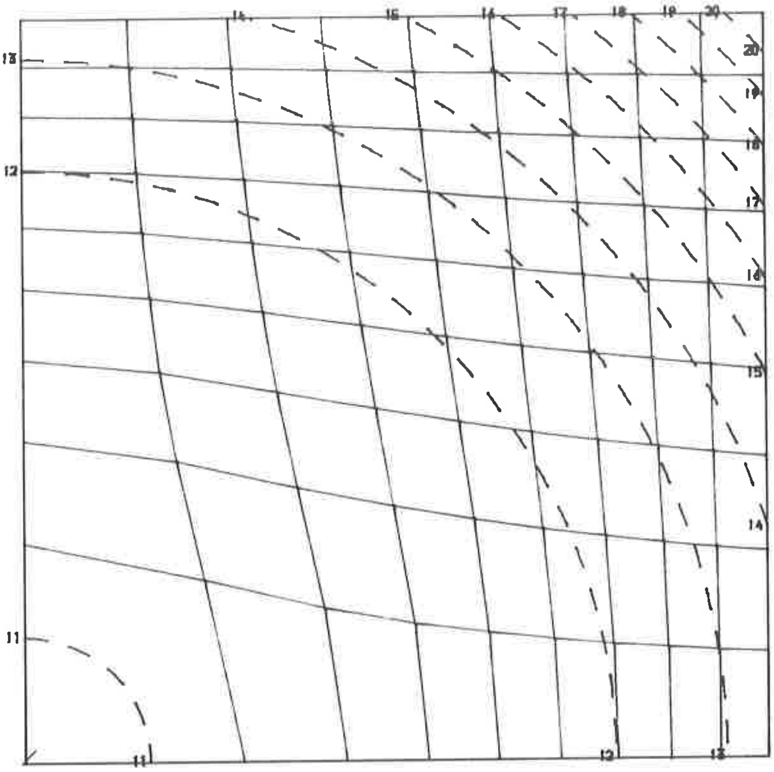


(a) $u = (x^2 + y^2)^2$

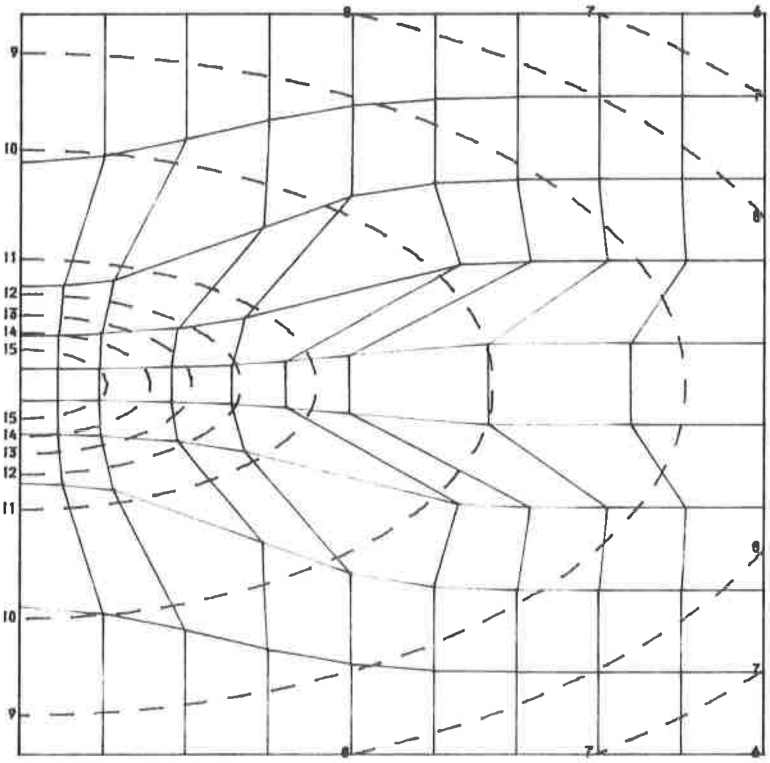


(b) $u = 2e^{-(16x^2+100(y-\frac{1}{2})^2)} - x^2 - 4(y-\frac{1}{2})^2$

Figure 7. Sample loci; data contours shown dotted.



(a) $u = (x^2 + y^2)^2$



(b) $u = 2e^{-(16x^2+100(y-\frac{1}{2})^2)} - x^2 - 4(y-\frac{1}{2})^2$

Figure 8. Grids produced by iteration on loci intersections.

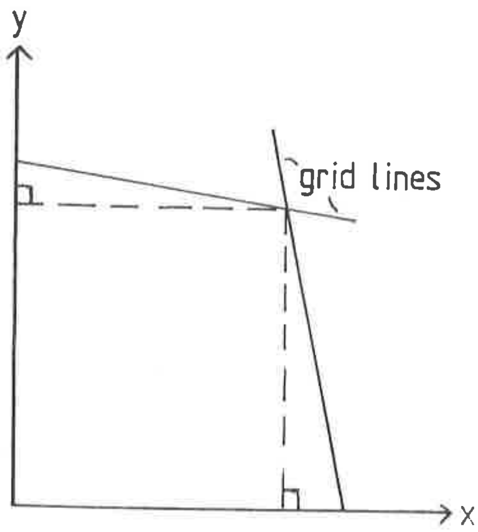


Figure 9. Equidistribution is along lines in coordinate directions (dotted) and not the grid lines.

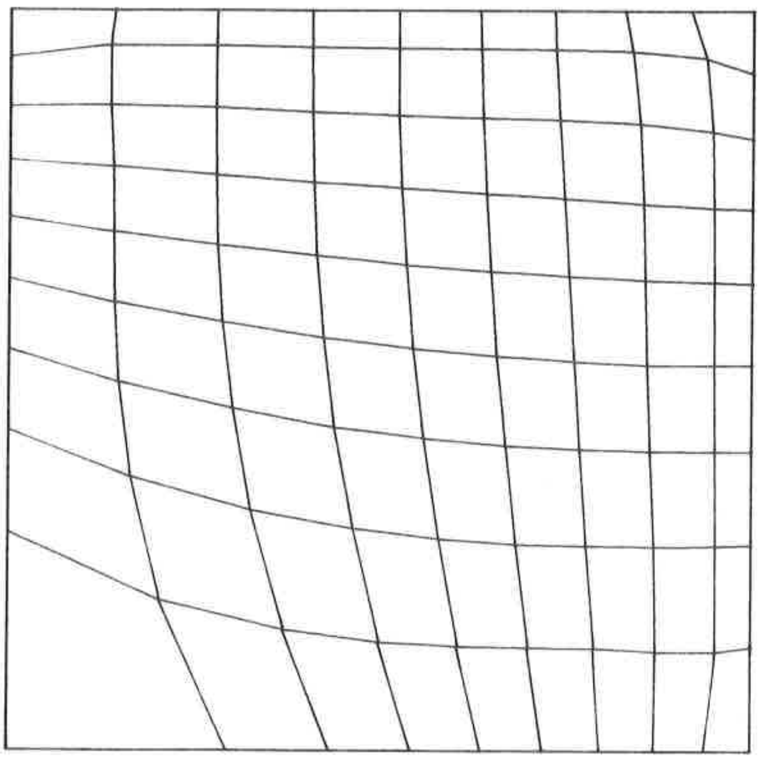
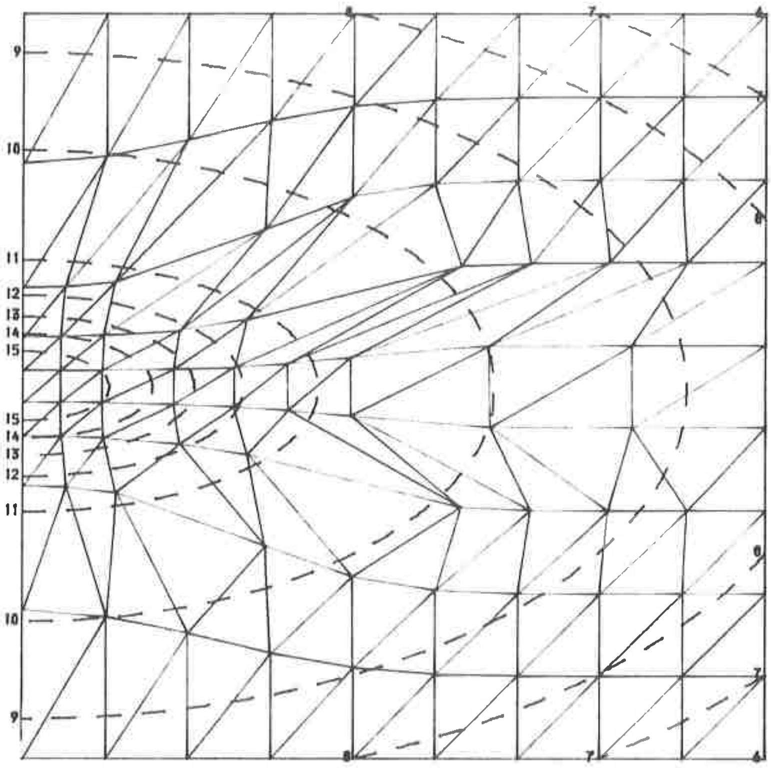
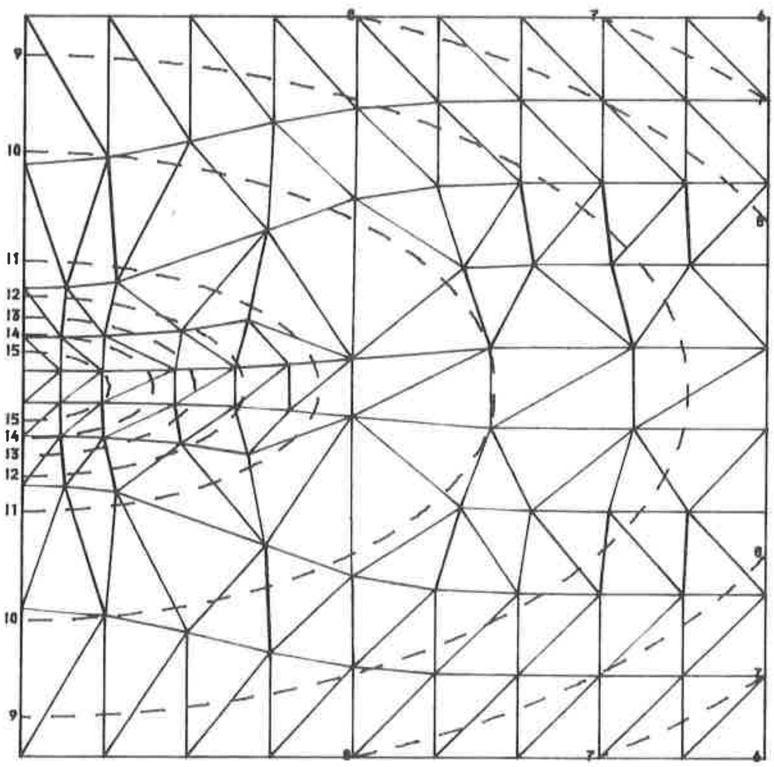


Figure 10. Grid produced using Anderson's adaptation of Thompson's grid generator. Data is $(x^2+y^2)^2$.



(a) Connection of diagonals.



(b) Delauney triangulation.

Figure 11. Triangulation of a quadralateral grid.

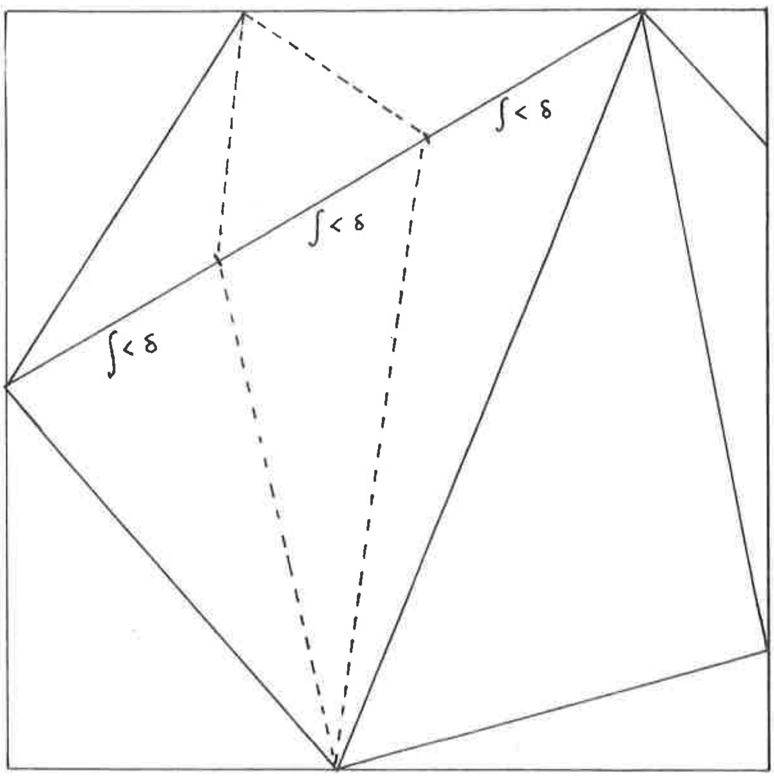
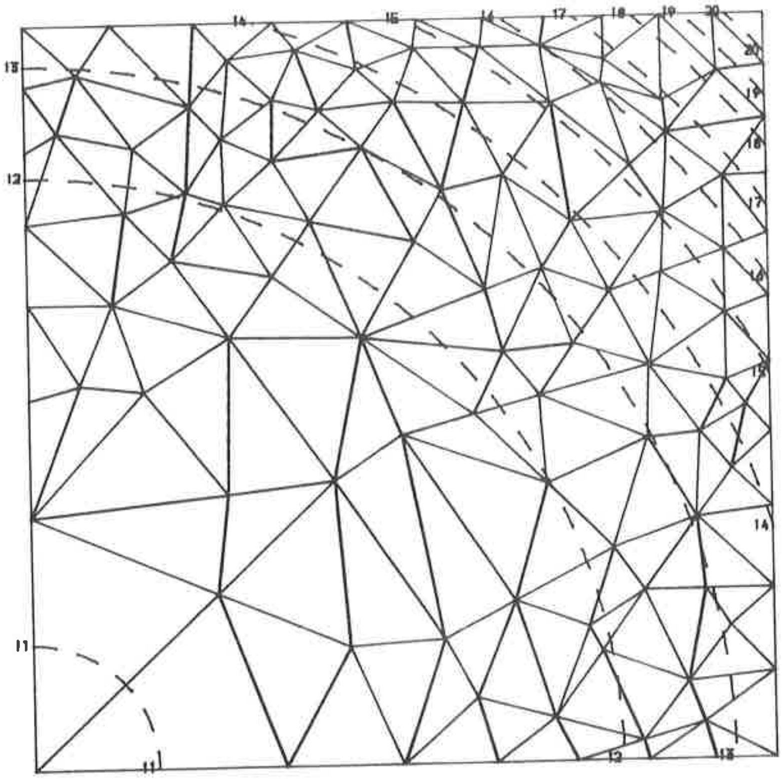
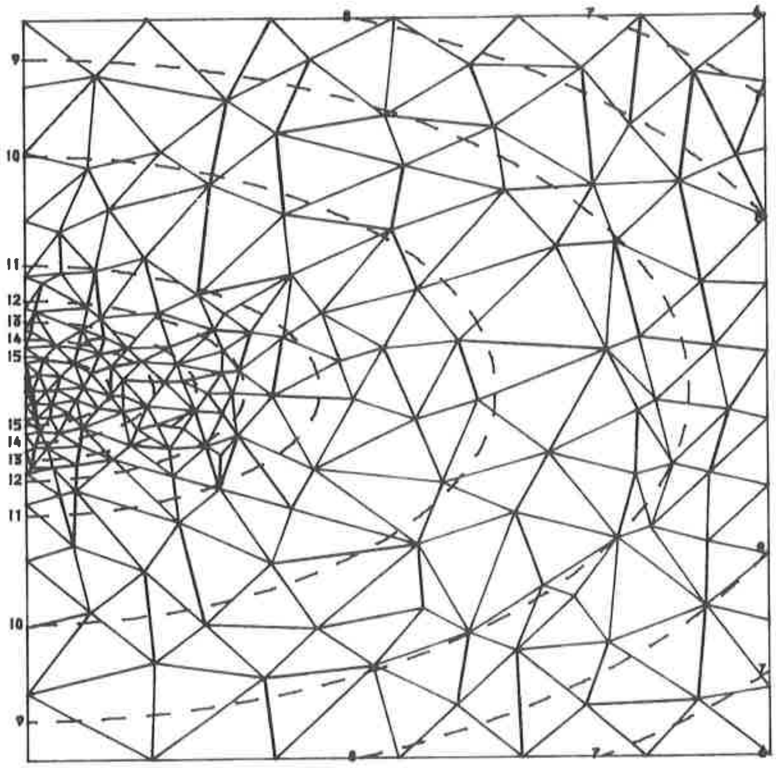


Figure 12. Subdivision of existing grid.



(a) $u = (x^2 + y^2)^2$



(b) $u = 2e^{-(16x^2+100(y-\frac{1}{2})^2)} - x^2 - 4(y-\frac{1}{2})^2$

Figure 13. Grids produced by subdivision.

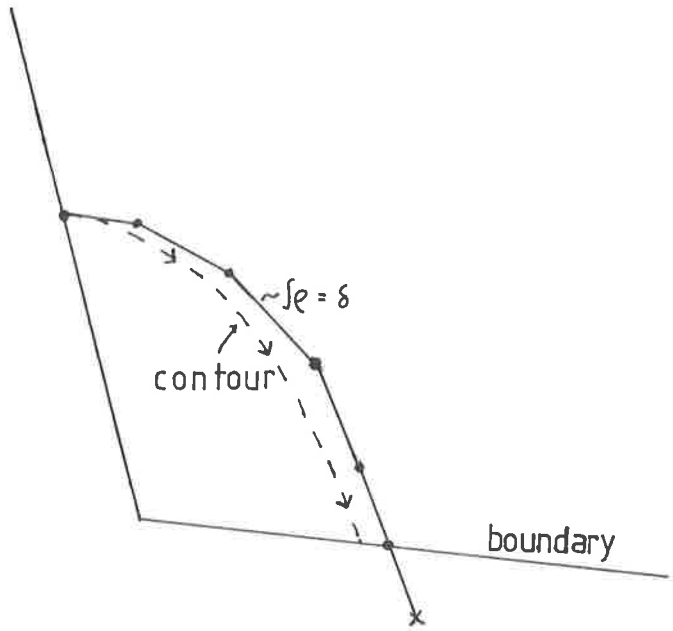
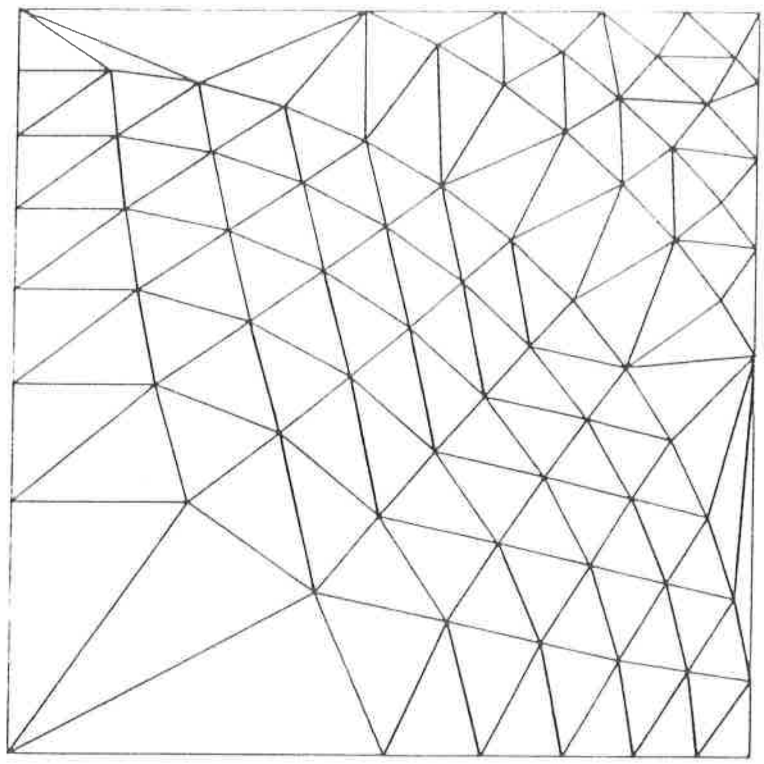
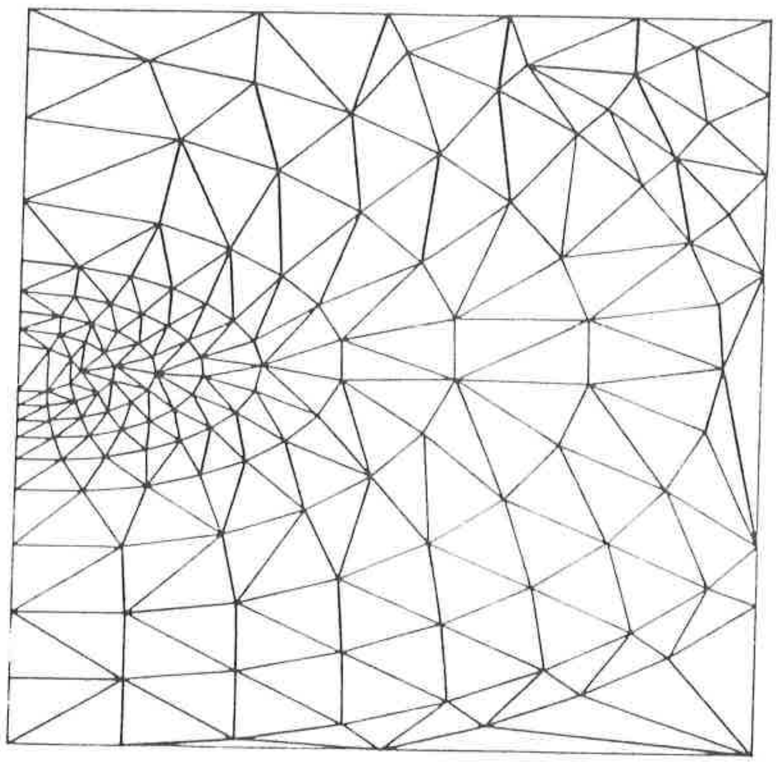


Figure 14. Approximate tracking of contours.



(a) $u = (x^2 + y^2)^2$



(b) $u = 2e^{-(16x^2+100(y-\frac{1}{2})^2)} - x^2 - 4(y-\frac{1}{2})^2$

Figure 15. Grids produced by contouring.

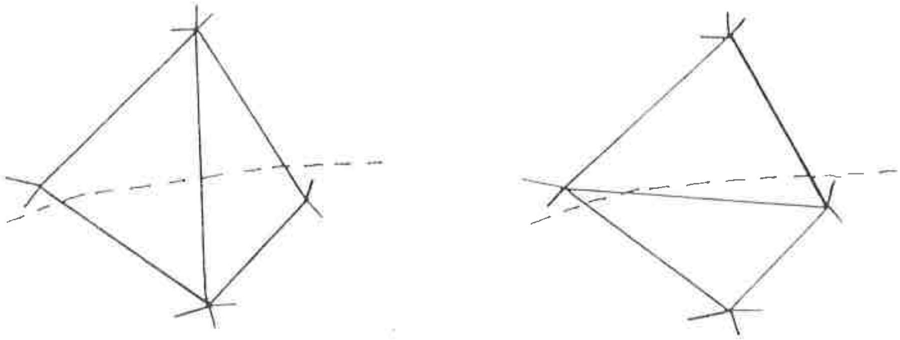
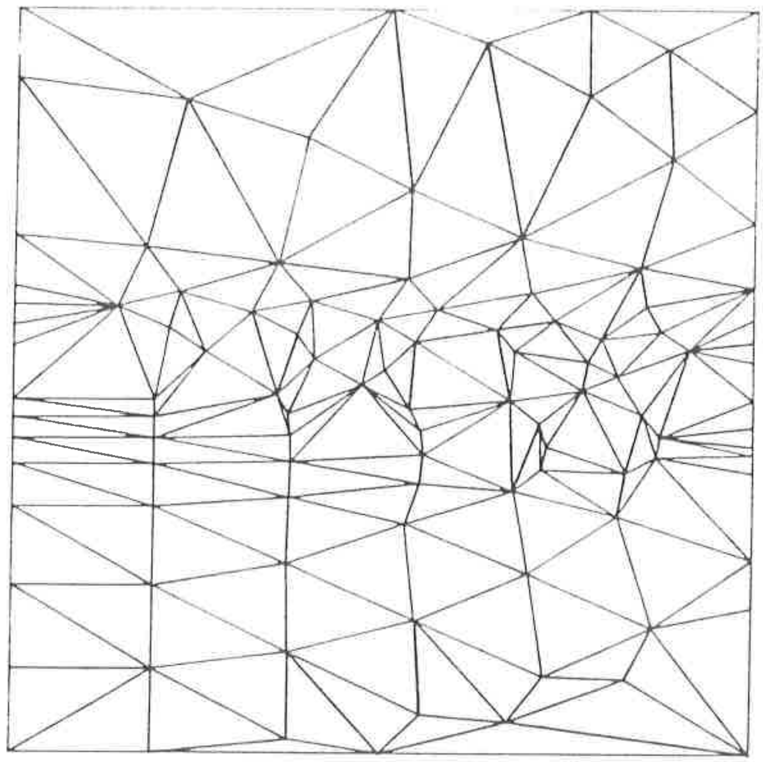
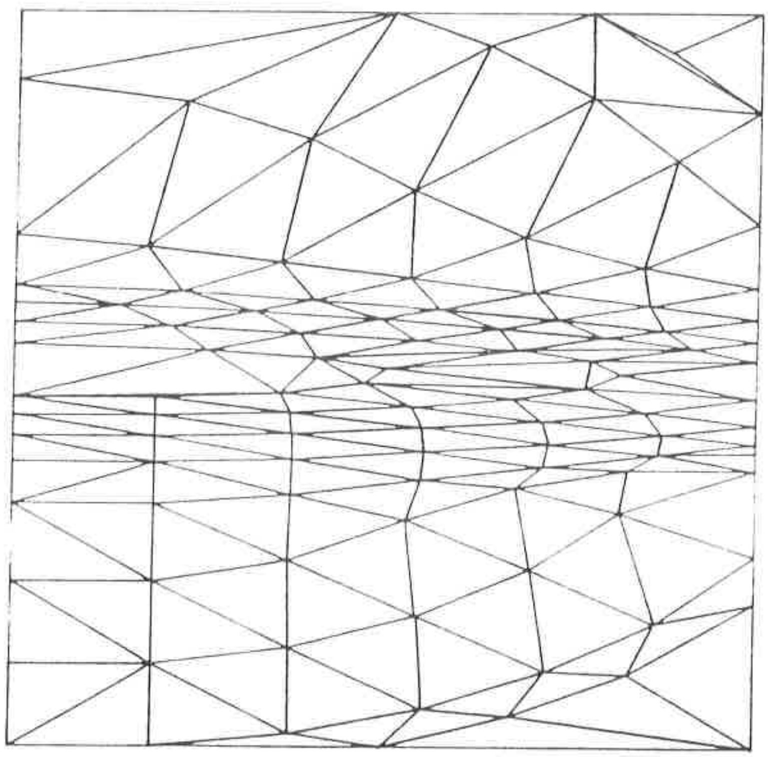


Figure 16. Possible connections of diagonals.
Data contour shown dotted.



(a) Raw grid.



(b) Regularised grid.

Figure 17. The effect of regularisation.

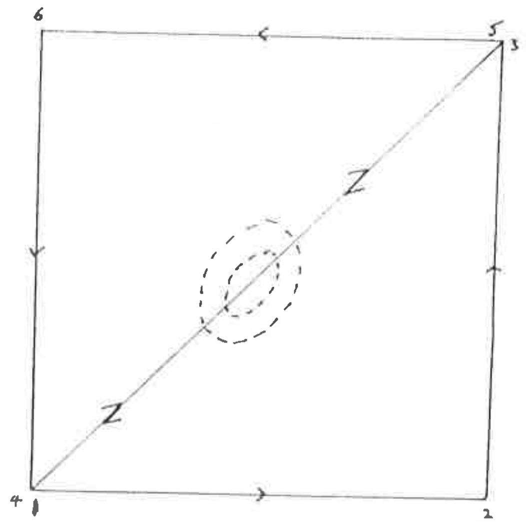


Figure 18. The use of branch cuts to intersect features of the data. Numbers denote ordering of boundary points.

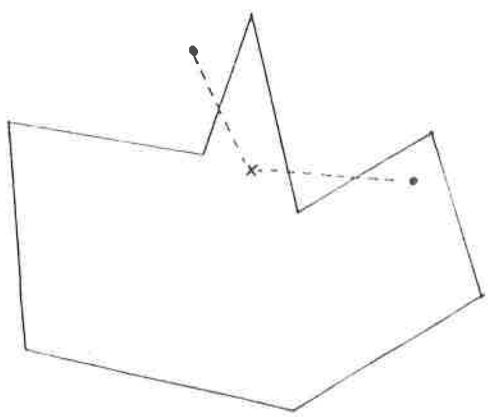


Figure 19. The counting of boundary intersections to determine external (odd number of intersections) points and internal (even number of intersections) points.