37
4-19-77
25 copy DNTS

UCID- 17415

Lawrence Livermore Laboratory

AREAL MAPPING OF ARBITRARY PLANE REGIONS ONTO RECTANGULAR MESHES

Stephen I. Warshaw

March 1977

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the laboratory.

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

AREAL MAPPING OF ARBITRARY PLANE REGIONS
*
ONTO RECTANGULAR MESHES

Stephen I. Warshaw

ABSTRACT

A simple and powerful method is presented for calculating all areas of intersection that occur when a plane region of arbitrary shape and connectivity overlaps the cells of a rectangular mesh. This procedure has been successfully used in an algorithm for a digital computer, and details are given for a unique procedure which effectively sidesteps floating point problems associated with machine noise. A simplifying procedure is also outlined for areal mapping of a collection of complex planar configurations onto a rectangular Eulerian mesh.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

*Work was performed under the auspices of the U. S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Table of Contents

Abstract.....	i
I. Introduction.....	1
II. Mapping Single Regions	
A. Eulerian Grid Preliminaries.....	2
B. Mapping Scheme.....	2
III. Algorithmic Details	
A. Definitions and Limits.....	5
B. Segment Processing.....	6
C. Calculation of Volume or Area.....	8
D. Treatment of Machine Noise.....	10
IV. Overlay Treatment of Multiple Regions.....	15
References.....	20
Distribution.....	21

AREAL MAPPING OF ARBITRARY PLANE REGIONS ONTO RECTANGULAR MESHES

Stephen I. Warshaw

I. Introduction

When the initial conditions for a finite two-dimensional Eulerian hydrodynamic calculation are to be set up, it is often necessary to be able to accurately map well-defined geometrical regions onto the static Eulerian mesh. This consists of assigning to each cell of the mesh the area or volume of intersection between the cell and the region. Physical attributes associated with the region (density, energy, etc.) can then be carried along with these volume fragments in their assembly to the initial Eulerian configuration. Such a mapping is absolutely essential if, for example, one wants to copy the results of a two-dimensional Lagrangean hydrodynamic calculation onto the Eulerian mesh; in this case the "regions" are the polygonal Lagrangean zones themselves.

We describe in this paper a working recipe for mapping arbitrarily shaped regions onto a rectangular Eulerian mesh on a digital computer, with accuracy limited only by the word size of the machine. This recipe is the basis for the generator and link codes of the BBC code system^[1]. The recipe could in principle apply to mappings onto other curvilinear orthogonal meshes, but this has not yet been explicitly done.

The only other closely related mapping program to our knowledge is R. M. Bentley's routine ZOVOLI,^[2] which is restricted to the mapping of triangles and quadrilaterals onto rectangular meshes.

II. Mapping Single Regions

A. Eulerian Grid Preliminaries

We define a rectangular Eulerian grid by the horizontal and vertical lines

$$x = X_k, k = 1, 2, \dots, k_{\max}$$

$$y = Y_\ell, \ell = 1, 2, \dots, \ell_{\max}$$

which are monotonically increasing in the indices k and ℓ , viz.

$$X_{k+1} > X_k$$

$$Y_{\ell+1} > Y_\ell$$

This divides the Cartesian plane into $(k_{\max} - 1)(\ell_{\max} - 1)$ rectangular cells, each of which is labeled by convention by the (k, ℓ) index pair of the intersection at the upper right hand corner. Depending on the type of symmetry required by the Eulerian calculation, the cells can have unit thickness in the z -direction (planar slab), or they can be rotated about the axis $y = Y_1 = 0$. (Toroids of rectangular cross-section with radii Y_ℓ , and width $\Delta x = X_k - X_{k-1}$.)

B. Mapping Scheme

Let a geometrical region in the plane be defined by a piecewise continuous and closed boundary curve $y = f(x)$.

The calculation of the areas common to the region and each rectangular cell is based on the principle that the area of a region is given by

$$A = \oint y \, dx$$

where the closed line integral is over the entire boundary of the region. It is assumed that this boundary is traversed in the usual sense, that is: the region lies on the left of the boundary as one proceeds along it. Since the boundary curve $y = f(x)$ is necessarily multivalued, it must be broken up into single-valued segments so that the integration reduces to a sum of signed areas, one under each segment.

This is illustrated in Figure 1, where, for the region shown,

$$\oint y \, dx = + \int_2^1 - \int_2^3 + \int_4^3 - \int_4^1 f(x) \, dx$$

The point to note here is that the areas lie between the segments of $f(x)$ and the x-axis (or other convenient reference line).

Figure 1.

These concepts are put on a cell-wise basis in constructing the mapping scheme. For each cell in the Eulerian mesh a signed area (or volume) of

intersection is calculated, corresponding to the intersection between the cell and the areas under each single-valued segment. The resultant cell assignment is the sum of all these non-zero segment contributions. For example, if a cell lies below the region, the contributions sum to zero. If a cell lies at the region boundary, all contributions cancel except those resulting in the actual intersection area or volume.

As long as the region boundary $y = f(x)$ can be represented by an unbroken sequence of single-valued segments, and $f(x)$ can be integrated over the segment limits, the region is a good candidate for the mapping. Non-simply connected regions (i.e. in many parts or with holes) can obviously be handled just as well.

III. Algorithmic Details

Translating this prescription into a computer program is not *a priori* as simple as it may appear, because the finite word size of digital computers creates subtle problems. In particular, one might know analytically of a case where the chosen boundary function $f(x)$ will pass exactly through the intersection of two grid lines, but the machine will not be able to determine that, within its inherent binary bit accuracy. Rather than basing such a determination on a threshold value of accuracy (e.g. $f(x_k)$ lie within a specified ϵ of the grid point), we have developed a logical scheme which avoids this problem, and which is therefore dependent on neither the word size of the machine nor the precision to which the grid lines are assigned. The details which follow were influenced to some extent by these considerations.

A. Definitions and Limits

Given the region boundary curve $y = f(x)$; we divide it into single-valued monotonic segments defined by

$$y = f_i(x) \quad , \quad i = 1, 2, \dots \quad , \quad \frac{\partial f_i}{\partial x} \text{ one sign,}$$

where the endpoints of each segment are labeled as x_1^i and x_2^i , where $x_1^i < x_2^i$. This requirement of monotonicity is necessary for the following logic chosen for the mapping.

B. Segment Processing

We sweep through the grid to determine how and where the segment $y = f_i(x)$ crosses the grid lines. Every instance of crossing a $y = Y_\ell$ line is recorded as the intersection x_ℓ^I where $f_i(x_\ell^I) = Y_\ell$. (The I is a label.)

In addition, the manner in which the contour crosses any given cell is recorded by assigning a logical index number, I_{kl} to the cell as follows: assign the numbers 1, 2, 4 and 7 to the four sides of the cell as shown in Figure 2, and add the numbers of the sides intersected by the curve. This is I_{kl} . Table I lists all possible intersection combinations, their I_{kl} values, and the limits for the integration over the cell. (Note that the relative simplicity of this table would have been severely compromised had the monotonicity requirement been relaxed.)

Figure 2.

The configurations for which $I_{kl} = 1, 2, 4$ and 7 are those where the segment endpoint lies within the cell; there is only one intersection. The case where both endpoints lie within the cell is conveniently assigned the value $I_{kl} = 10$. Where two possibilities exist for $I_{kl} = 1, 4$, and 5 , the choice is easily made on the basis of the slope of the segment. And, of course, $I_{kl} = 0$ for no intersection at all.

Table I

Configuration	I_{kl}	Integration Limits
1	x_1^i	x_ℓ^I
1	x_ℓ^I	x_2^i
2	x_1^i	x_k
3	x_ℓ^I	x_k
4	x_1^i	$x_{\ell-1}^I$
4	$x_{\ell-1}^I$	x_2^i
5	x_ℓ^I	$x_{\ell-1}^I$
5	$x_{\ell-1}^I$	x_ℓ^I
6	$x_{\ell-1}^I$	x_k
7	x_{k-1}	x_2^i
8	x_{k-1}	x_ℓ^I
9	x_{k-1}	x_k
10*	x_1^i	x_2^i
11	x_{k-1}	$x_{\ell-1}^I$

* Assigned by fiat

C. Calculation of Volume or Area

For each cell where $I_{kl} \neq 0$, the volume or area within is calculated by integration (numerically, if need be) between the limits appropriate to that I_{kl} . If the segment is a straight line, the well-known quadrilateral formulas for area or volume can be used directly. For each cell below the one where $I_{kl} \neq 0$, the volume or area is also calculated with the same integration limits, and the calculation is much simpler, since the area is a rectangle.

If the limits of integration are, say x_a to x_b (as taken from Table I), we have the following:

For the cell where $I_{kl} \neq 0$,

$$\begin{aligned} \text{planar area } A_{kl} &= \int_{x_a}^{x_b} (f_i(x) - Y_{l-1}) dx \\ \text{cylindrical volume } V_{kl} &= 2\pi \int_{x_a}^{x_b} ([f_i(x)]^2 - Y_{l-1}^2) dx \end{aligned}$$

and for every cell below that one (l decreasing from $l-1$, k same),

$$A_{kl} = (Y_l - Y_{l-1})(x_b - x_a) = A_{\text{cell}} \left(\frac{x_b - x_a}{X_k - X_{k-1}} \right)$$

$$V_{kl} = 2\pi(Y_l^2 - Y_{l-1}^2)(x_b - x_a) = V_{\text{cell}} \left(\frac{x_b - x_a}{X_k - X_{k-1}} \right)$$

We observe that the ratios in the parentheses are equal and constant for the lower strip of zones.

The totality of these cell volumes or areas is the contribution made by the segment $f_i(x)$ to the mesh. This is shown schematically in Figure 3, where a typical segment is illustrated. The integrals for the cells where $I_{kl} \neq 0$ are shown by cross-hatching, those for the lower cells by shading, and the integers in the cells are the non-zero I_{kl} .

As each segment $f_i(x)$ is treated its cellular area or volume contributions have to be appropriately signed, depending on where the region lies in regard to the segment. If, in traveling along the segment from the higher endpoint (x_2^i) to the lower (x_1^i) the region lies to the left, then the positive sign is chosen (counter-clockwise convention). Otherwise the sign is negative. For example, if the segment of Figure 3 were to represent that part of the contour of Figure 1 between points 2 and 3, the cellular areas or volumes for it would be negative.

When all segments $f_i(x)$ of a given boundary are processed, all the positive and negative volumes in each cell are summed. This assigns to the mesh the cell volumes or areas for that region. If done correctly, cells

outside the region should have no area or volume contribution, while those covered by it will.

Finally, we observe that with this prescription as outlined above, parts of a region that lie below or to one side of the Eulerian mesh will contribute nothing and hence can be ignored. However, if part of a region lies above the mesh, this part cannot be discarded, since the cells in the mesh below it still have to be filled. One way of automatically taking this into consideration is to set up a row of dummy cells corresponding to $l = l_{\max} + 1$ atop the mesh, high enough to cover the region, and then do the mapping onto this augmented mesh. Afterwards, the dummy cells and their contents may be discarded.

D. Treatment of Machine Noise

If the boundary curve $y = f_i(x)$ should happen to pass through an intersection of two grid lines, then it is possible for the computer to logically "make a mistake" in deciding where the intersection on one line lies with respect to the location of the other line. As a consequence, "wrong" I_{kl} values can be assigned to the four cells about this point. The four possible pairs of decisions for one example are shown magnified schematically in Figure 4, with the I_{kl} for each cell also given.

Figure 4

Cases (a) and (b) are legitimate and therefore proper, but cases (c) and (d) are not. In (c) we have an illegal $I_{kl} = 13$, and in both (c) and (d) the wrong integration limits will be used (Table I).

This problem arises because it is not generally known *a priori* how, eg., X_k compares with x_l^I or how Y_l compares with $f_1(X_k)$, as computed. Because of roundoff and truncation in a finite machine, this comparison is in practice not generally predictable.^[3] We solve this problem by fully accepting the unpredictability of the situation, and working around it.

It turns out that bad situations typified by (c) and (d) in Figure 4 always provide characteristic diagonal I_{kl} pair signatures, which depend on the slope of the segment. There are four bad intersections in all, two for each slope sign. These are shown in Figure 5, with the I_{kl} and the intersection points.

Figure 5

The signatures are the diagonal I_{kl} combinations 4-2, 1-7, 2-1 and 4-7, respectively.

The easiest way to repair these bad situations when they occur is to simply reassign the four I_{kl} values about the offending intersection so that they "look right". The values for the intersections $f_1(X_k)$ and x_l^I

are not touched, as there is no need to. In Figure 6 we show one of a set of possible consistent reassignments of the I_{kl} from the corresponding ones given in Figure 5, symbolically showing the "reassigned" intersections also. These changes constitute the desired logical repair, and depend in no way on any considerations of computational precision.

Figure 6

The reassignments constitute, in effect, a "moving" of one of the intersection points from one side of a grid line to the other. The other possible sets of reassignments arise when the other point is "moved".

A slightly more subtle approach to this machine noise problem is needed when very short contour segments are processed. These contours are covered by a 2×2 array of adjacent cells, and the eight distinctly different possibilities that can occur are shown with their I_{kl} values and slopes in Figures 7 and 8 on the next page. The "good" cases are given in Figure 7, and the "bad" ones in Figure 8.

Figure 7

Figure 8

Inspection of these figures reveals two complications not considered before. First, the "good" cases have the diagonal I_{kl} signatures that were deemed to be bad from the preceding discussion. Second, the I_{kl} values shown are the same for good and bad cases which have opposite slope. Hence an additional, special test is required for these 2 x 2 zone-contour configurations. For positive slope, we check to see if I_{kl} at the upper left is 6 or I_{kl} at the lower right is 8. If either condition occurs, then the

zones have been correctly tagged. Similarly, for negative slope, the labels are correct when I_{kl} at the upper right is 11 or I_{kl} at the lower left is 3. If these I_{kl} assignment conditions are not met, then the same logical I_{kl} repairs outlined previously are clearly indicated. This, as we have shown, is easy to do.

IV. Overlay Treatment of Multiple Regions

If the mapping scheme is used to set up an initial Eulerian configuration, many geometrical regions will be mapped. Ordinarily this would require almost every region boundary to be calculated at least twice, since that curve would be common to the regions on either side. This can be needlessly expensive in terms of machine time and labor required to set up input data. We detail below a simple method to eliminate the need for calculating a boundary segment more than once, in cases where such application is obvious.

Suppose for concreteness we specify three regions to be mapped onto the Eulerian mesh, which overlap among themselves as shown in Figure 9. We

Figure 9

wish to reassign the cell volumes which are covered by all these areas in such a way that the configuration shown in Figure 10 is the result.

Figure 10

This reassignment is like laying down first R_1 , then R_2 , then R_3 , one at a time, in that order.

To avoid complicating the mapping scheme by trying to include the case of mapping onto a partly filled mesh, and also to conserve storage, the following algorithmic procedure was devised. The regions are processed in reverse order of "laying down" -- first R_3 , then R_2 , then R_1 .

R_3 is mapped onto the mesh by itself, and an array of assigned volumes $V_3(k, l)$ is obtained. For convenience, these should have been converted to cell volume fractions $f_3(k, l)$, given by

$$f_3(k, l) = \frac{V_3(k, l)}{V_{\text{cell}}(k, l)}$$

Thus, $f_3(k, l) = 1$ if the cell is completely filled.

This array is added to an initially zero array, $f_T(k, l)$, that is, we set

$$f_T(k, l) = f_3(k, l) + f_T(k, l) = f_3(k, l)$$

where the prime distinguishes the change in the array $f_T(k, \ell)$.

Next, R_2 is mapped onto the mesh by itself, and the array $f_2(k, \ell)$ is obtained. These volume fractions are reassigned by subtracting $f'_T(k, \ell)$ from it cell by cell, and discarding negative results. Thus,

$$f'_2(k, \ell) = \max \begin{cases} f_2(k, \ell) - f'_T(k, \ell) \\ 0 \end{cases}$$

is the properly assigned fractional volume array for region 2. This array is added to $f'_T(k, \ell)$ to form $f''_T(k, \ell)$:

$$f''_T(k, \ell) = f'_T(k, \ell) + f'_2(k, \ell)$$

$f''_T(k, \ell)$ represents the volume fractions for the union of R_3 and R_2 .

The same procedure is now carried out for R_1 (and all subsequent regions if present):

$$f'_1(k, \ell) = \max \begin{cases} f_1(k, \ell) - f''_T(k, \ell) \\ 0 \end{cases}$$

$$f'''_T(k, \ell) = f''_T(k, \ell) + f'_1(k, \ell)$$

and so on.

The procedure outlined above is clearly of advantage if the Eulerian configuration can be represented as a successive overlay of many regions, one atop another. Otherwise recourse must be had to the more straightforward procedure of mapping the regions just as they are set up, and doing the boundaries twice.

The overlay algorithm is summarized as follows for an N-region case.

1. Set $f_T(k, \ell) = 0$.
2. Loop over j from $j = N$ to $j = 1$, i.e., backwards.
 - a) Map R_j and obtain $f_j(k, \ell)$
 - b) Reassign volumes: $f_j(k, \ell) = \max \begin{cases} f_j(k, \ell) - f_T(k, \ell) \\ 0 \end{cases}$
 - c) Reset f_T : $f_T(k, \ell) = f_T(k, \ell) + f_j(k, \ell)$
3. After loop is done, the contents of $f_T(k, \ell)$ should be all 1's.

After each pass through the loop, $f_T(k, \ell)$ will be the array of volume fractions for each successive union of the regions, that is: at the k^{th} pass, for $R_N \cup R_{N-1} \cup R_{N-2} \dots \cup R_{N-k+1}$.

Unfortunately, this simple procedure has a drawback that depends on the particular region configuration being mapped, and so should not be used indiscriminately. Suppose in the reassignment of the volume fractions for region j there arose the situation where, in one cell (k, ℓ) , both $f_j(k, \ell)$ and $f_T(k, \ell)$ sum to less than 1, and $f_T(k, \ell) > f_j(k, \ell)$. Then the reassignment of $f_j(k, \ell)$ would be to set it to zero, since the algorithm would think $f_T(k, \ell)$ "covered" $f_j(k, \ell)$. Most of the time this would be correct; however, if the region boundaries within the cell were as shown in Figure 11, it would not!

Figure 11

The same argument also holds if the situation were reversed, i.e. $f_j(k,l) > f_T(k,l)$. Then $f_j(k,l)$ would be reset to the difference $f_j(k,l) - f_T(k,l)$, but if the boundaries were like those in Figure 11, it would not be right either.

This problem arises because the overlay algorithm does not take into account where a region is within a cell. For most situations this distinction is not important; but where it is important the overlay technique should not be used.

References

1. W. G. Sutcliffe, S. I. Warshaw and R. F. Litterst, Preliminary User's Manual for BBC, LLL Report UCID-14124, May 1976 and references cited therein.
2. R. M. Bentley, Subroutine ZOVOLI, LRL Internal Memorandum, March 1962.
3. For example, B. Wendroff, First Principles of Numerical Analysis, Addison-Wesley, 1969, especially Section 1.4.

:ps

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

NOTICE

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Energy Research & Development Administration to the exclusion of others that may be suitable.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy \$; Microfiche \$3.00

Page Range	Domestic Price	Page Range	Domestic Price
001-025	\$ 3.50	326-350	10.00
026-050	4.00	351-375	10.50
051-075	4.50	376-400	10.75
076-100	5.00	401-425	11.00
101-125	5.50	426-450	11.75
126-150	6.00	451-475	12.00
151-175	6.75	476-500	12.50
176-200	7.50	501-525	12.75
201-225	7.75	526-550	13.00
226-250	8.00	551-575	13.50
251-275	9.00	576-600	13.75
276-300	9.25	601-up	*
301-325	9.75		

*Add \$2.50 for each additional 100 page increment from 601 to 1,000 pages;
add \$4.50 for each additional 100 page increment over 1,000 pages.

Technical Information Department

LAWRENCE LIVERMORE LABORATORY

University of California | Livermore, California | 94550