JOURNAL OF COMPUTATIONAL PHYSICS 37, 371–387 (1980)

A Method for Automating the Construction of Irregular Computational Grids for Storm Surge Forecast Models

W. C. THACKER, A. GONZALEZ, AND G. E. PUTLAND

Sea–Air Interaction Laboratory, Atlantic Oceanographic and Meteorological Laboratories National Oceanic and Atmospheric Administration, Miami, Florida 33149

Received February 1, 1979; revised July 30, 1979

Irregular grids for computing storm surges can be generated from digital data defining the bathymetry and boundaries of the basins in which the surges occur. The method presented here for constructing these grids proceeds in two stages. First, a curvilinear grid is constructed to have boundaries which coincide with the curving coastlines of bays and barrier islands. In the second stage, a fully irregular grid is constructed by removing points from the curvilinear grid. Because the separation of neighboring points is proportional to the square root of the basin depth, the irregular grid provides for economical computations with a minimum of numerical dispersion. This method can be generalized for other purposes. It is possible to construct grids comprised of quadrilateral or triangular elements, corresponding to both curved and planar domains, to be used for a wide variety of calculations using either finite-difference or finite-element methods.

I. INTRODUCTION

Irregular computational grids have two attractive features which conventional uniform grids lack. They provide for a natural representation of irregular boundaries and also for a desirable distribution of interior grid points. Although irregular grids have been used for a wide variety of calculations, here attention is to be focused on grids for computing storm surges and near-shore water circulation. In this case, the irregular boundary corresponds to the curving coastlines of bays and barrier islands, and the desired density of grid points is inversely proportional to the basin depth. These features can be seen in the computational grid for Mobile Bay [1] shown in Fig. 1. Construction of this grid by hand was a tedious task. Many man-hours were required to position each of the 5566 points and to identify the neighboring points. The purpose of this paper is to present a more efficient way to construct irregular grids.

These irregular grids can be used for both finite-difference and finite-element calculations [2]. The distribution of grid points does not depend on which method is used. Grid spacing is dictated by the equations to be solved, which in this case are the shallow-water wave equations. Because the phase velocity is inversely proportional to the square root of the depth, the length of a wave changes as it propagates through water of varying depth. In order to reduce numerical dispersion, the same

0021-9991/80/120371–17$02.00/0

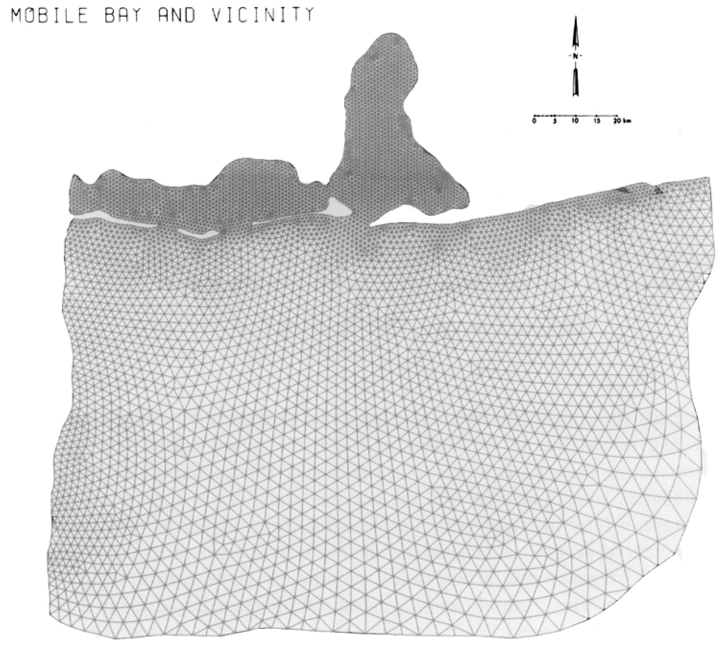

FIG. 1. This computational grid for Mobile Bay was constructed by hand.

number of points per wavelength should be used throughout the grid. This requires that the areas of the grid elements should be roughly proportional to the corresponding depths. Even though these grids are suitable for either computational method, transient phenomena such as storm surges can be simulated much more economically using the finite-difference method.

A fully automatic method is presented here for constructing irregular computational grids. The entire construction process, from start to finish, can be accomplished by a digital computer without the need for human intervention at any intermediate stage. One alternative would be a semi-automatic approach with the user interacting with the computer and directing each step of the construction. This might offer an advantage of more user control over the construction process. However, a fully automatic approach is more suitable to the task of providing grids for storm surge forecast models for each of the many vulnerable bays and estuaries in the United States and other parts of the world.

The automatic construction method presented here consists of two phases. The first is concerned with the requirement that the boundaries of the grid conform to the coastlines, and the second, with the requirement that the distances between neighboring points reflect the bathymetry of the basin. In the first phase of construction, a uniform grid with zigzag boundaries is transformed into a curvilinear grid (see Figs. 2–4), and in the second phase, the curvilinear grid is transformed into the desired irregular grid by removing some of the points from the deeper regions and

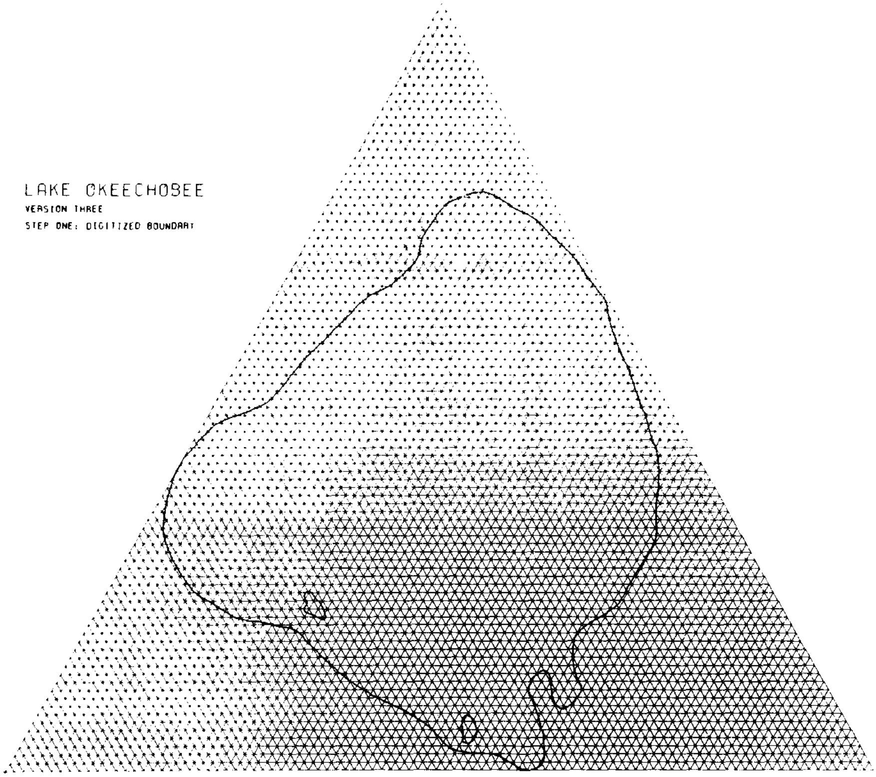

FIG. 2. The data which determine the boundary of the lake are shifted so that they touch the lower and left sides of the triangular grid. Note that the islands are rather small for this grid spacing.

then repositioning the remaining points so that the grid elements are as equilateral as possible (see Figs. 7 and 9–11).

Construction is based on data defining the boundaries and bathymetry of the basin. The boundary data consist of sequences of connected line segments on which the boundary points of the grid are constrained to lie, and the bathymetric data consist of depth values for a set of points distributed arbitrarily throughout the basin, later to be interpolated to the grid points. The zigzag boundaries of the initial uniform grid are determined automatically from these input data. In fact, at no stage in the construction process is it necessary for the user to specify the position of any grid point or to supply any information about the grid configuration. The construction proceeds automatically by processing these input data, which might be taken from smooth sheets used in the preparation of navigation charts. If computations are to be noise-free, the details of the coastal and bathymetric features represented by these data should be consistent with the resolution of the grid. Ideally, the radius of curvature of these features should be no less than the distance between the grid points which will represent them. This restriction is incorporated into the construction

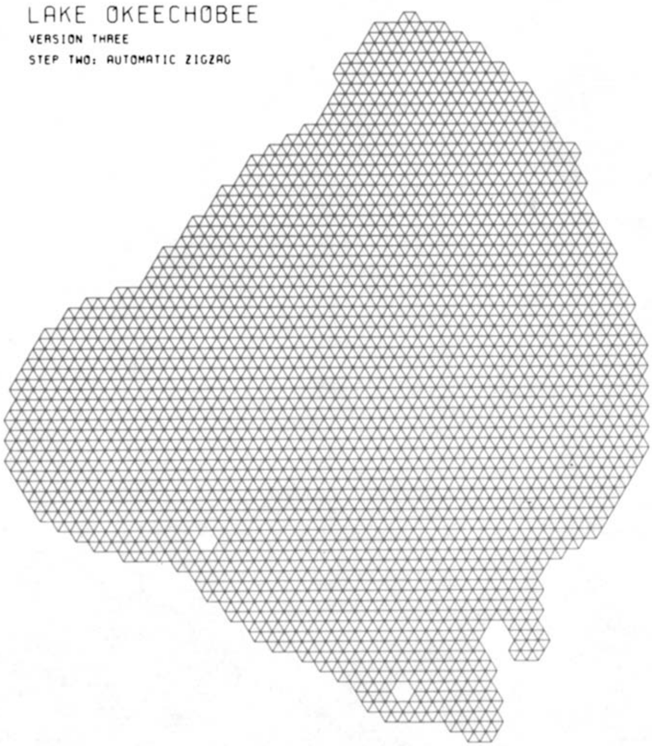

FIG. 3. This zigzag grid was obtained by eliminating unneeded points from the triangular grid (Fig. 2).

procedure through the choice of a nominal value for the distance between neighboring grid points. Attempts to include smaller details into the data can lead to poorly constructed grids. If additional resolution is economically infeasible and small-scale features which are too small to be adequately resolved *must* be represented by the grid, then these features should be added by hand to a grid constructed from smooth data.

II. EQUILATERAL TRIANGULAR GRID AND ZIGZAG GRID

An irregular grid can be constructed automatically by modifying a triangular grid composed of equilateral elements. This approach has two advantages over attempting to imitate the manual procedure of positioning points one at a time. The constraint of nearly equilateral elements can be satisfied because construction begins with exactly equilateral elements, and neighboring points can be systematically identified because the initial grid is regular. The first modification is the elimination of exterior points to obtain a grid with a zigzag boundary. Next, the zigzag grid is transformed into a curvilinear grid with a better representation of the boundary. If depths are nearly the same throughout the basin, then the curvilinear grid is the final product of the automatic construction process. However, if some parts of the basin are deeper than others, then it is necessary to remove points from the corresponding parts of the curvilinear grid and to reposition the remaining points so that the grid spacing is roughly proportional to the square root of the depth.

The first step in constructing the initial triangular grid is the choice of grid spacing. Because the spacing will be approximately the same in the shallow portions of the resulting irregular grid, it should be small enough to resolve important coastal and bathymetric features but not so small that computations will be prohibitively expensive. The construction procedure is sufficiently economical that it is possible to generate several irregular grids, corresponding to different values for initial grid spacing, and to choose the one that is most appropriate.

The triangular grid must be large enough for its boundaries to encompass the outer boundaries of the basin. This is accomplished by taking the coordinate origin at the lower left corner of the triangular grid and by shifting the data so that the outer boundary of the basin is justified to the lower and left edges of the triangle. The right edge is chosen to give the smallest triangular grid having the specified mesh size that can encompass the entire basin. Figure 2 shows the triangular grid and the input boundary data for Lake Okeechobee.

A zigzag grid (Fig. 3) can be obtained by eliminating points from the corresponding triangular grid (Fig. 2). The zigzag boundaries are determined by the points on the triangular grid which are closest to the boundaries of the basin. Because exterior points are excluded, it is necessary to recognize which points are exterior and which are interior. The interior and boundary points can be sequentially reindexed, and their coordinates and neighboring points can be tabulated to define the zigzag grid.

The data defining the boundaries of the basin consist of sequences of coordinates of boundary points. One sequence corresponds to the outer boundary and the others to boundaries of islands within the basin. The data are ordered to that line segments between consecutive points within a sequence are boundary segments. The procedure for determining the points on the zigzag boundaries is to examine consecutive boundary segments to determine where they cross line segments (grid segments) connecting adjacent grid points (Fig. 2). The grid points closest to the intersections are taken to be the boundary points of the zigzag grid.

If the boundary data are not sufficiently smooth or if the choice of grid spacing is too large, then the zigzag boundaries might have undesirable features. For example, a narrow pass might lack interior points for calculating the flow, or narrow peninsulas, inlets, or islands might have sharp corners. Even if the data are smooth and the grid spacing is small, the zigzag boundaries might have unwanted spikes because the boundary data meander about a line midway between two adjacent rows of grid points. The spikes can be recognized as three consecutive points which are vertices of a common triangular element, and they can be eliminated by deleting the second of the three points from the boundary sequence. After the spikes have been eliminated, the boundary should be examined to determine if there are further undesirable features associated with the choice of grid spacing or with insufficiently smooth data. These features can be spotted by searching for two nonconsecutive zigzag boundary points which are separated by only one grid space. If any are found, the construction process should stop and a decision should be made whether to correct the boundary data to improve the grid resolution. Note that the resolution of the islands in Fig. 2 is just barely acceptable.

Next, it is necessary to classify the points on the triangular grid that are not on the zigzag boundary as interior or exterior. This can be done by examining each of the points in turn, proceeding from left to right and from bottom to top, starting with the lower left corner of the triangular grid. If a point is on one of the three edges, then it must be an exterior point if it is not known to be a boundary point. If the point is not on an edge and is not a boundary point, then its classification is determined by the previously classified neighboring points, as long as the undesirable features discussed in the previous paragraph are absent form the boundaries. Finally, the points on the zigzag grid can be reindexed sequentially and the values of their coordinates and the indices of their neighbors can be tabulated.

III. Curvilinear Grid

The procedure used to transform the zigzag grid (Fig. 3) into the corresponding curvilinear grid (Fig. 4) is straightforward. It involves the solution of a set of simultaneous equations for the coordinates of the points on the curvilinear grid. If the line segments between points on the zigzag grid are thought of as being springs under tension and if the boundary points are constrained to lie on the piecewise linear boundary curves defined by the boundary data, then the equilibrium configuration of the spring system coincides with the curvilinear grid.

The springs are assumed to have negligible length when unstretched, and all interior springs are assumed to have the same strength. It is convenient to assume that the boundary springs are stronger than the interior springs by a factor r; otherwise the configuration of neighboring points may cause some boundary elements to be undesirably small. The coordinates of the interior grid points, which correspond to the equilibrium state of such a system of springs, satisfy the equations.

$$
\begin{aligned}
x_k &= \frac{1}{N_k} \sum_{i=1}^{N_k} x_{n_i(k)}, \\
y_k &= \frac{1}{N_k} \sum_{i=1}^{N_k} y_{n_i(k)},
\end{aligned}
\tag{1}
$$

where k is the index of the interior point and $n_i(k)$ are the indices of its N_k neighbors. (For the zigzag grid, each interior point has exactly six neighbors, so $N_k = 6$ for each interior point. However, these equations will also be used to position points on the

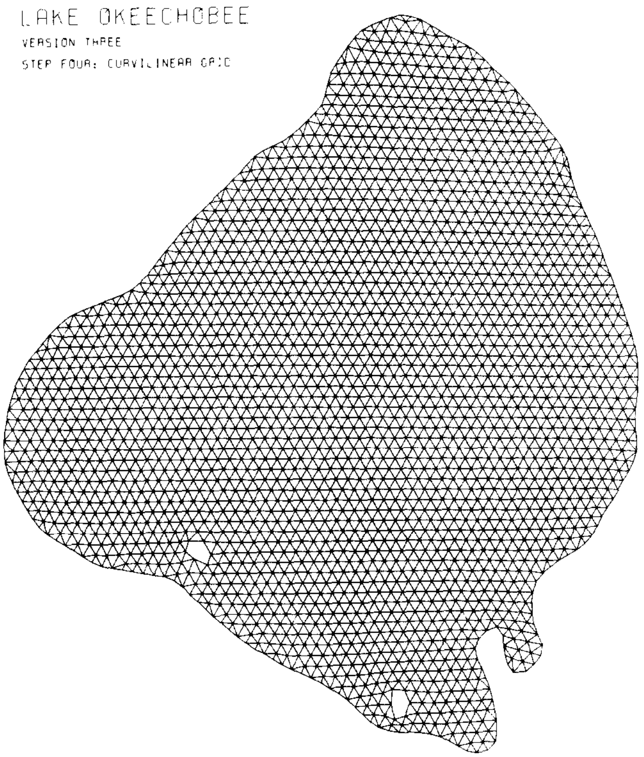

FIG. 4. This curvilinear grid was obtained by assuming that the grid segments are springs which have relaxed to their equilibrium configuration.

irregular grid (Fig. 9) where N_k may be 5, 6, or 7.) These equations state that each interior point is positioned so that it is at the center of the polygon determined by its neighbors. The equations for the boundary points are somewhat more complicated:

$$\begin{aligned} x_k &= \sum_{i=1}^{N_k} \phi_{n_i(k)} \left[x_{n_i(k)} \cos^2 \theta_{b(k)} + y_{n_i(k)} \cos \theta_{b(k)} \sin \theta_{b(k)} \right] \\ &\quad + x_{b(k)} \sin^2 \theta_{b(k)} - y_{b(k)} \cos \theta_{b(k)} \sin \theta_{b(k)}, \\ y_k &= \sum_{i=1}^{N_k} \phi_{n_i(k)} \left[x_{n_i(k)} \cos \theta_{b(k)} \sin \theta_{b(k)} + y_{n_i(k)} \sin^2 \theta_{b(k)} \right] \\ &\quad - x_{b(k)} \cos \theta_{b(k)} \sin \theta_{b(k)} + y_{b(k)} \cos^2 \theta_{b(k)}. \end{aligned} \tag{2}$$

The boundary point, k, is constrained to lie on the line through the adjacent boundary data points, $b(k)$ and $b(k)+1$, as shown in Fig. 5. The slope of this line is $\tan \theta_{b(k)} = [y_{b(k)+1} - y_{b(k)}]/[x_{b(k)+1} - x_{b(k)}]$. The factors $\phi_{n_i(k)}$ account for the different strengths of boundary and interior springs; $\phi_{n_i(k)} = 1/(2r + N_k - 2)$ if $n_i(k)$ is an interior point and $\phi_{n_i(k)} = r/(2r + N_k - 2)$ if $n_i(k)$ is a boundary point. For the grid shown in Fig. 4,

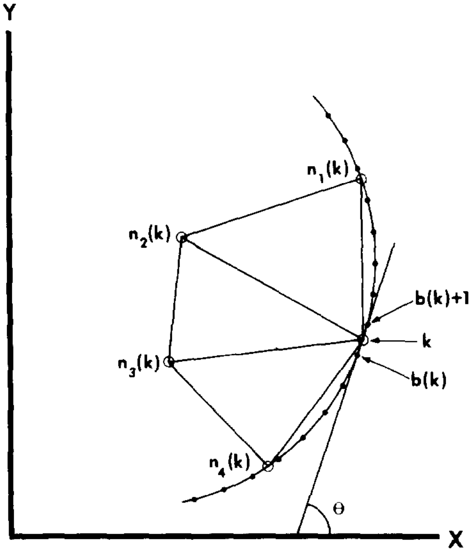

FIG. 5. The boundary segment, between points $b(k)$ and $b(k)+1$, must be determined simultaneously with the coordinates of the grid points.

$r = 10$ for springs corresponding to the line segments along the outer boundary and $r = 1$ for those corresponding to the island boundaries. The special treatment of the islands was necessary because of the fact that they are only minimally resolved by the grid; stronger springs would cause the islands to be more nearly circular.

Because, in solving these equations, for the coordinates of the points on the curvilinear grid, it is also necessary to determine on which interval of the boundary curve each boundary point of the grid should lie, an iterative method of solution is appropriate. The positions of the interior points on the zigzag grid provide an initial estimate of the positions of the corresponding points on the curvilinear grid. The boundary points can be placed initially at the points of intersection of grid segments and boundary segments nearest to the corresponding points on the zigzag boundary and these boundary segments provide initial estimates for $\theta_{b(k)}$, $x_{b(k)}$, and $y_{b(k)}$ in Eqs. (2). The iterative procedure is to fix the postions of all but one grid point and to adjust the position of that one so that the springs connecting it to its neighbors are in equilibrium. After point k has been repositioned, point $k+1$ can be repositioned in the same manner. Each point is repositioned as many times as necessary, until the new positions coincide with the old positions to the accuracy of the computer. This is the familiar Gauss–Seidel method, except for the special consideration that must be given to the boundary intervals. Equations (2) constrain the boundary point to be repositioned along a line through the points $b(k)$ and $b(k)+1$ on the boundary curve, and it is possible that it might be pulled outside of the interval corresponding to those two points. In that case, the new position for the point can be taken to be the end point of that segment in the direction in which the point was moved, and the adjacent segment should be used for the next iteration.

Several other coordinate transformations have been used to construct curvilinear grids [3–18]. However, Eqs. (1) and (2) which define this transformation can also be used in the second stage of construction to guarantee that the elements of the irregular grid (Figs. 9 and 10) are as equilateral as possible.

IV. Depth Interpolation

The next step in the construction process is the determination of values for the depth for each grid point (Fig. 6) from data specifying the depths at points distributed at random throughout the basin. These data can be obtained from navigation charts, such as those published by the National Ocean Survey of the United States Department of Commerce, or from corresponding magnetic computer tapes containing the hydrographic data from which the charts were made. In either case, it is important

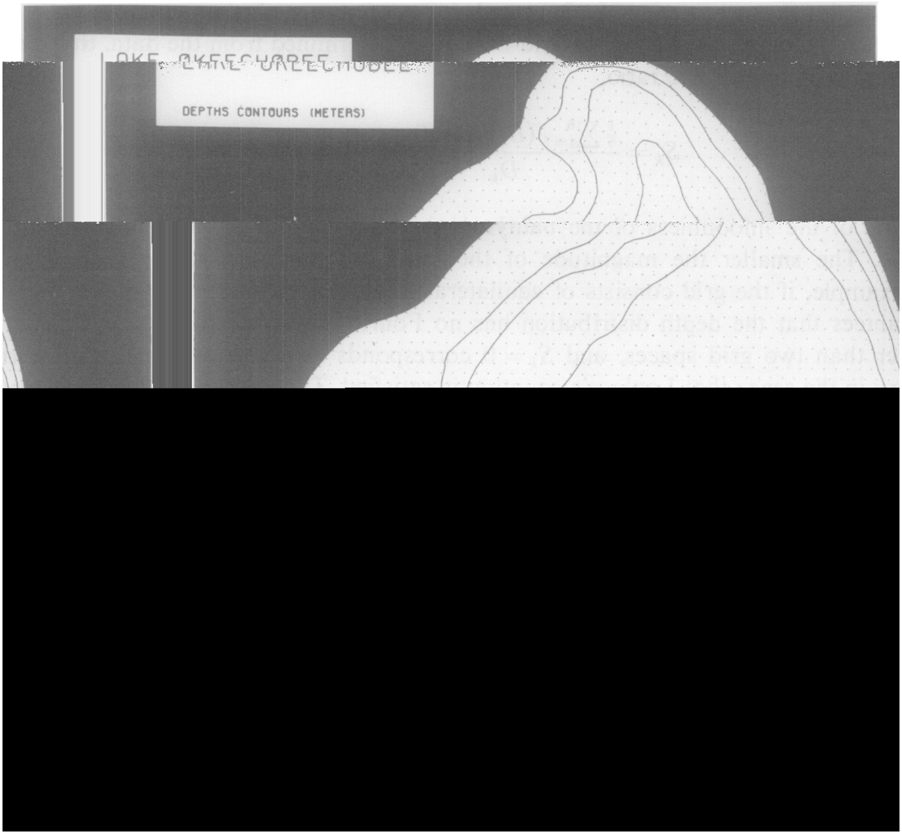

Fig. 6. These depth contours represent data from a navigational chart which were interpolated to the grid points and then smoothed.

that the data represent the depths smoothly on the scale of the grid spacing. Because the data from the magnetic tapes correspond to large numbers of closely spaced soundings, nearby values can be averaged to smooth the data and to reduce the quantity of data.

The value for the depth at a grid point with coordinates (x, y) can be obtained by linearly interpolating the depths, D_a, D_b, and D_c from the three closest data points,

$$D = \Phi_a D_a + \Phi_b D_b + \Phi_c D_c. \tag{3}$$

The interpolating coefficients depend on the coordinates of the points. For example,

$$\Phi_a = \frac{(x - x_b)(y - y_c) - (x - x_c)(y - y_b)}{(x_a - x_b)(y_a - y_c) - (x_a - x_c)(y_a - y_b)} \tag{4}$$

and similar expressions for Φ_b and Φ_c can be obtained from Eq. (4) by cyclically permuting the indices (a, b, c). The denominator in Eq. (4) corresponds to twice the area of the triangle with vertices (a, b, c) and does not vary with the permutations.

After the depth values for the grid have been determined from the data, they can be examined for smoothness. The ratio,

$$S_k = \frac{\frac{1}{6}\sum_{i=1}^{6} (D_{n_i(k)} - D_k)}{D_k}, \tag{5}$$

provides an estimate of the smoothness of the bathymetry in the vicinity of the interior grid point k. The smaller the magnitude of the ratio, the smoother is the bathymetry. For example, if the grid consists of equilateral triangular elements, then $|S_k| < \pi^2/12$ guarantees that the depth distribution has no Fourier components with wavelengths shorter than two grid spaces, and $S_k = 0$ corresponds to the smoothest possible bathymetry in the sence that Laplace's equation is satisfied. The requirement, $S_k = 0$, is too strict because, in that case, interior depths are determined by depths along the boundaries. An arbitrary choice can be made, say $S_{\max} = 1/2$, so that if

$$|S_k| < S_{\max} \tag{6}$$

for every interior point k, then the bathymetry is considered to be sufficiently smooth. Otherwise, the depths can be redefined by replacing D_k with $D_k(1 + pS_k)$, where $0 < p \leqslant 1$ to achieve a smoother distribution. If the new values are still not smooth enough, the procedure can be repeated until Eq. (6) is satisfied everywhere.

If the depth of the basin is approximately constant, then the curvilinear grid is appropriate for storm surge computation, and the construction process can be terminated. The curvilinear grid can also be used for basins with variable depth. For example, normal mode oscillations of circular basins with parabolic depths were successfully simulated using a curvilinear grid [2]. However, the computational expense of using a grid with elements in deep water that are the same size as those in shallow water for practical calculations (see Fig. 1) can be prohibitive. In that case, it is necessary to continue the construction process by removing points from the deep-water regions of the grid and by adjusting the positions of the remaining points so that the elements are as nearly equilateral as possible.

V. Irregular Grid

Before transforming the curvilinear grid (Fig. 4) into a fully irregular grid (Fig. 11), it is necessary to establish a criterion for judging whether the grid spacing is suited to the bathymetry (Fig. 6). Because the grid elements are constrained to be almost equilateral, the criterion can be cast in terms of polygonal areas associated with each interior grid point. For the curvilinear grid, these polygons are hexagons whose vertices are the six neighboring grid points. However, after points have been removed to form the irregular grid, some of the remaining grid points will have five or seven neighbors, while most will still have six. If the convention is adopted that the order of the neighboring points is counterclockwise and that the ith neighbor of the kth point, $n_i(k)$, is the same as the $(i \pm N_k)$th neighbor, then the polygonal area associated with the kth point is given by

$$A_k = \frac{1}{2} \sum_{i=1}^{N_k} x_{n_i(k)} \left[y_{n_{i+1}(k)} - y_{n_{i-1}(k)} \right], \tag{7}$$

where N_k is the number of neighboring points (5, 6, or 7).

The criterion for the grid spacing to be suited to the bathymetry is that the ratio of the polygonal area to the local depth,

$$R_k = A_k/D_k, \tag{8}$$

be approximately the same for each interior grid point k. When this is true, the distance between neighboring points is approximately proportional to the square root of the depth and the time required for a wave to propagate between neighboring points is approximately the same throughout the basin.

For the curvilinear grid, the ratios, R_k, can be expected to be largest for points in shallow water near shore and to be smallest for points in the deepest regions of the basin. Since the curvilinear grid is to be modified by removing points from the deep regions, the ratios for the irregular grid should be approximately the same as the maximum value for the curvilinear grid. To be more precise, the grid spacing can be considered to be compatible with the depth if

$$\alpha R_{\max} < R_k < R_{\max}, \tag{9}$$

for all interior points k, where $R_{\max}$ is the maximum value of the ratios on the curvilinear grid and α is a fraction, $0 < \alpha < 1$, which must be small enough to allow for the variability in the ratios due to the irregularities of the grid but large enough to force the grid spacing to reflect the depth distribution. Because the ratio of the area of

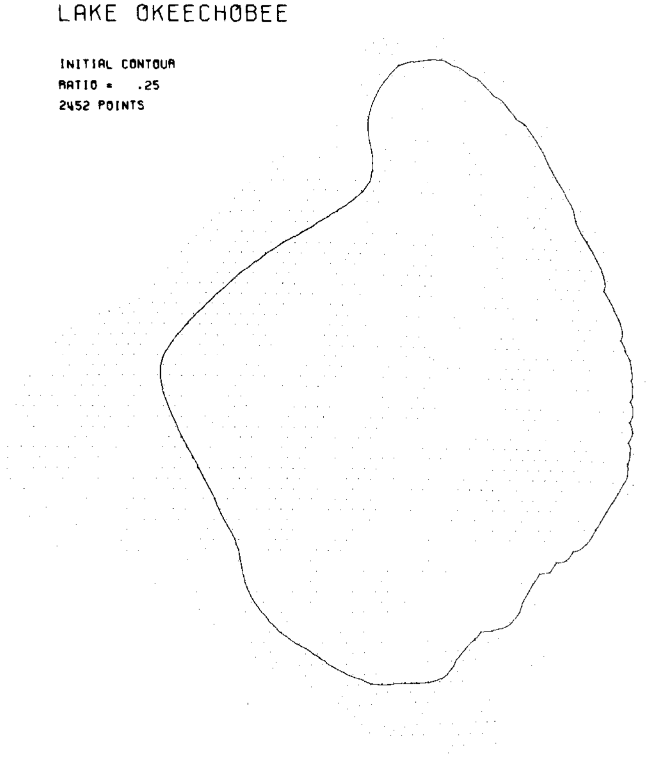

FIG. 7. Points should be removed from within this contour so the grid spacing will be better suited to the depth.

a regular pentagon to a regular septagon, whose sides are the same length, is 0.47, a reasonable value for α is 0.25.

It is useful to imagine contour lines (Fig. 7) separating grid points for which $R_k > \alpha R_{\max}$ from those for which $R_k < \alpha R_{\max}$. The points outside of these contours are appropriately spaced, but those within the contours are too close together. Thus, points should be removed from the interior, unacceptable regions. After some of the points have been removed and the remaining points have been repositioned, then new values can be computed for the ratios (8) which will correspond to new contour lines around smaller unacceptable regions. More points can be removed until Eq. (9) is satisfied for every remaining point. At that stage, the construction of the irregular grid will be finished.

A sequence of adjacent points can be removed form the curvilinear grid as illustrated in Fig. 8. Although points with five and seven neighbors are introduced at the ends of the cut, the other points all have six neighbors. This suggests a systematic procedure for altering the curvilinear grid. The first cut can be made starting at the point closest to the critical contour bounding the region with unacceptable grid spac-

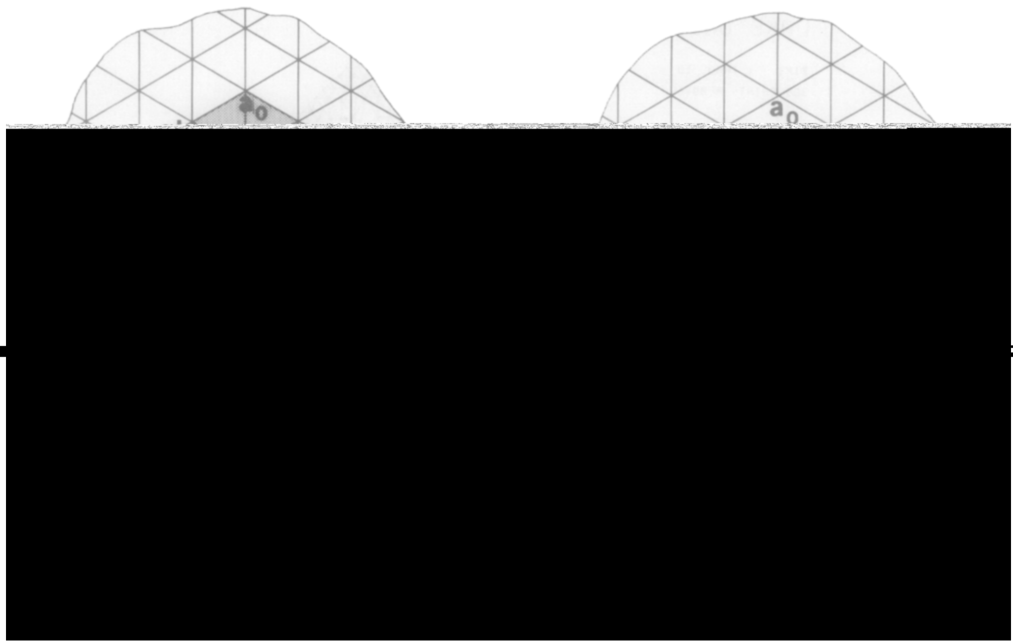

FIG. 8. The polygon formed by removing N grid points can be divided into triangular elements as shown. The same equations used to position points on the curvilinear grid can be used to reposition points on the irregular grid so that each point is at the center of the polygon formed by its neighbors.

ing through that region and terminating at a point near the critical contour on the opposite side of the basin. If a boundary is reached before the critical contour, the cut can terminate at the boundary. After this sequence of points is removed, the critical contour can be redefined, surrounding a smaller unacceptable area. Now, another cut, similar to the first, can be made. The only restriction is that it must remove only points with six neighbors, so that respecification of neighbors does not become complicated. Since the points with five and seven neighbors are situated on the periphery on the unacceptable area, they do not pose a problem. Cuts can be made repetitively until the spacing throughout the grid is acceptable.

Because of the regularity of the curvilinear grid, the processes of removing a sequence of adjacent points is quite straightforward. The starting point of the cut a_0, in Fig. 8, is chosen because of its proximity to the critical contour, and the first point to be removed, a_1, is the neighbor of a_0 with the smallest ratio of area to depth. The next point to be removed a_2, is the third neighbor of a_1 following a_0. Each successive point can be determined in the same way; a_{k+1} is the third neighbor of a_k following a_{k-1}. The terminal point of the cut, a_{N+1}, is the first in the sequence with a ratio of area to depth greater than $\alpha R_{\max}$ unless the sequence terminates at a boundary. If, in the process of determining which points to remove, a point is encountered with five or seven neighbors, corresponding to the end of a previous cut, then the sequence can be abandoned and another can be determined starting with a new choice for a_0.

581/37/3-7

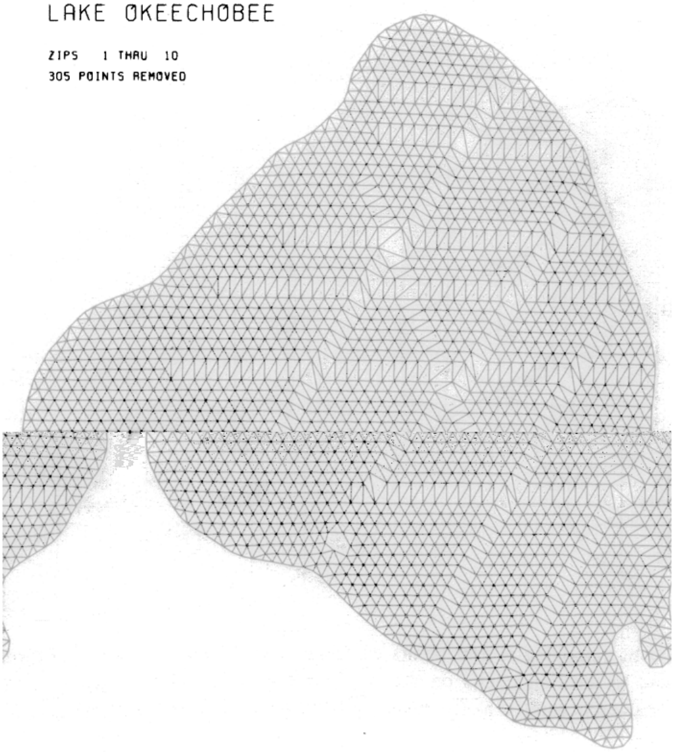

FIG. 9. Ten rows of points have been removed from the region indicated by the contour in Fig. 7, and neighboring points in the vicinity of the cuts have been respecified.

The points bordering the cut, $b_1,..., b_{N+1}$ and $c_0,..., c_N$, for which neighbors must be reassigned can be identified in a similar manner; b_k and c_{k-1} are the neighbors of a_{k-1} preceding and following a_k. Except for the points c_0, b_1, c_N, and b_{N+1}, the respecification of the neighbors follows a simple pattern. For b_k, the neighbors a_k and a_{k-1} are replaced with b_k and b_{k-1}; for c_k, a_k and a_{k+1} are replaced with b_k and b_{k+1}. Only one neighbor must be replaced for each of c_0 and b_{N+1}; for c_0, a_1 is replaced with b_1, and for b_{N+1}, a_N with c_N. The points b_1 and c_N each have seven neighbors; for b_1, a_1 is replaced with the pair, c_1 and c_0, and for c_N, a_N with b_N and b_{N+1}. The points a_0 and a_{N+1} each have only five neighbors after the points a_1 and a_N are removed. If the cut terminates at a boundary, the reassignment of neighbors can be made in a similar systematic way.

After removing a group of points (Fig. 9) the remaining points can be reindexed sequentially and repositioned in a manner similar to the relocation of points from the zigzag grid to the curvilinear grid with each interior point located at the center of the polygon determined by its neighbors (Fig. 10). The values for the depths at the new locations can be obtained by linearly interpolating the values from the old locations, according to Eqs. (3) and (4), just as the bathymetric data were interpolated to the

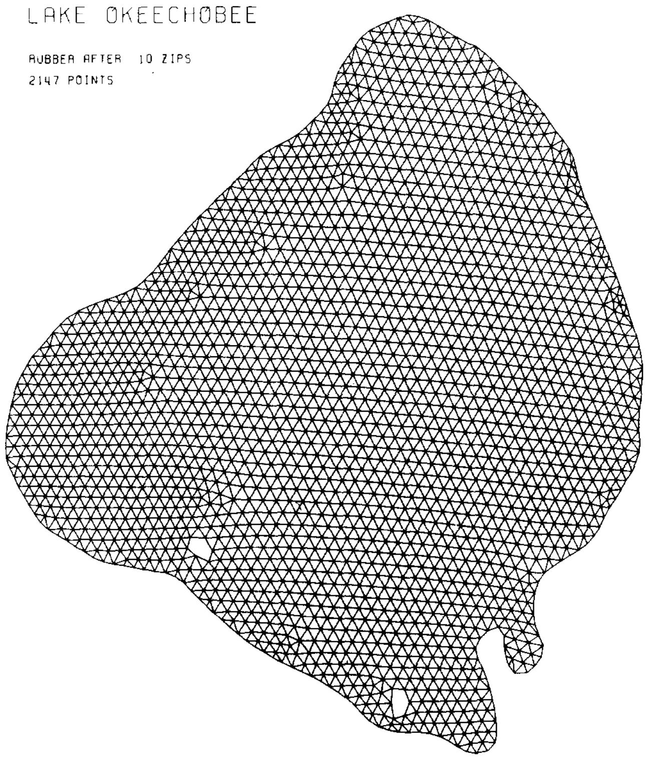

FIG. 10. Points on the grid in Fig. 9 have been repositioned to coincide with the equilibrium configuration of the equivalent network of springs.

curvilinear grid points. New values for the ratios of area to depth can be calculated, and if the grid spacing is still not satisfactory, more points can be cut from the grid. When no more points can be used, the grid is obtained which has the desired distribution of points (Fig. 11).

VI. CONCLUSION

Although this grid construction scheme has been presented within the context of storm surge computations, it is, in fact, quite general. The only aspect of this scheme that is particular to storm surge models is the requirement that the grid spacing be proportional to the square root of the basin depth. In this case, the distribution of computational points is dictated by the shallow-water wave equations, but, for other problems, the distribution should be based on the nature of the governing equations. Some ratio, similar to that of polygonal area to depth but appropriate to that problem, should then be made as constant as possible throughout the grid, while the triangular elements are as equilateral as possible.

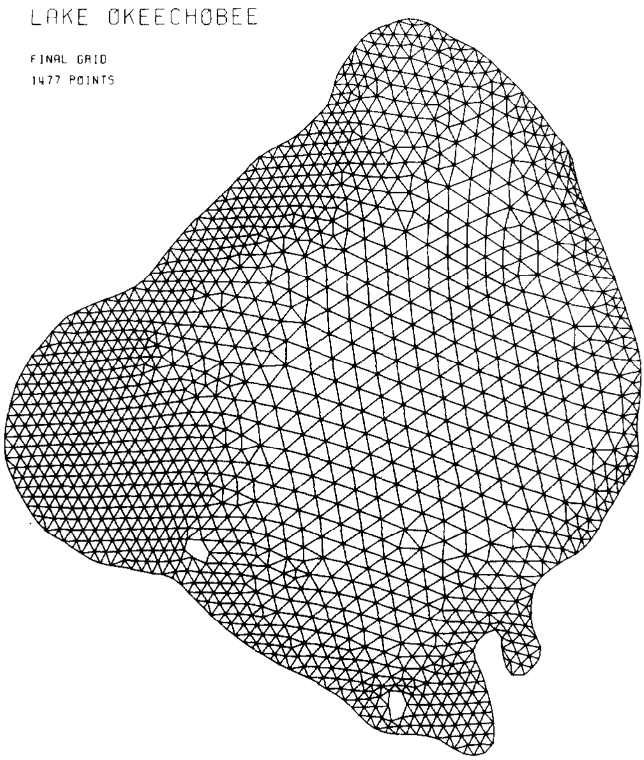

FIG. 11. The process of removing points terminates with a grid having points spaced according to the depth and elements which are nearly equilateral.

It is possible to extend this method so that it can be used to construct grids on curved surfaces. For example, an irregular grid for the interconnected ocean basins could be quite useful for computing tidal oscillations or wind-drive oceanic circulation. Since the surface of the earth is, to a good approximation, spherical, it [illegible] surface numerically. The data could consist of the coordinates of points distributed over the surface, which can be connected to form triangular facets that provide a piecewise linear approximation to the surface that is analogous to the piecewise linear representation of the coastlines described above. The grid points can be constrained to lie on that surface just as boundary points are constrained to lie on the coastlines. For each surface on which a grid is to be constructed, it is necessary to provide an initial grid, analogous to the triangular grid in Fig. 3. For the example of a sphere, this could be constructed by joining edge to edge, 20 identical triangular grids to form the faces of an icosahedron and projecting these grids onto the surface of the sphere [19, 20]. From that point on, the method is identical to that presented here for constructing planar grids.

It is just as easy to construct grids that have quadrilateral elements as those having triangular elements. Even when it is necessary to remove a row of points to adjust the density of grid points, the resulting grid can be specified so that it has only quadrilateral elements. At each end of the cut there will be one point with five neighbors and another with three neighbors, whereas every other interior point will have four neighbors.

Finally, it should be emphasized that, whatever the use of the grid might be, features with dimensions on the order of the grid spacing will be poorly resolved and might result in computational inaccuracy. With this scheme, these small-scale features might also result in a poorly constructed grid. The best solution to this problem is to increase the density of grid points, but if this is not economically feasible, care should be taken to edit these features by hand.

References

1. W. C. Thacker, "Marine Forecasting" (Jacques C. J. Nihoul, Ed.), Elsevier, Amsterdam/New York, 1978.
2. W. C. Thacker, *J. Phys. Oceanogr.* **8** (1978), 680.
3. A. M. Winslow, *J. Comput. Phys.* **2** (1967), 149.
4. S. K. Godunov and G. P. Prokopov, *USSR Comput. Math. Math. Phys.* **7** (1967), 89.
5. W. D. Barfield, *J. Comput. Phys.* **5** (1970), 23.
6. W. D. Barfield, *J. Comput. Phys.* **6** (1970), 417.
7. W.-H. Chu, *J. Comput. Phys.* **8** (1971), 392.
8. O. C. Zienkiewicz and D. V. Phillips, *Int. J. Numer. Meth. Engng.* **3** (1971), 519.
9. A. A. Amsden and C. W. Hirt, *J. Comput. Phys.* **11** (1973), 348.
10. S. K. Godunov and G. P. Prokopov, *USSR Comput. Math. Math. Phys.* **12** (1972), 182.
11. W. J. Gordon and C. A. Hall, *Int. J. Numer. Meth. Engng.* **7** (1973), 461.
12. J. F. Thompson, F. C. Thomas, and C. W. Mastin, *J. Comput. Phys.* **15** (1974), 299.
13. R. Medyer, *J. Comput. Phys.* **17** (1975), 53.
14. C. A. Hall, R. W. Luczak, and A. G. Serdy, *ACM Trans. Math. Software* **2** (1976), 257.
15. T. K. Hung and T. D. Brown, *J. Comput. Phys.* **23** (1979), 343.
16. J. F. Thompson, F. C. Thomas, and C. W. Mastin, *J. Comput. Phys.* **24** (1977), 274.
17. S. B. Pope, *J. Comput. Phys.* **26** (1978), 197.
18. A. Denayer, *Comput. Structures* **9** (1978), 359.
19. G. W. Platzman, *J. Phys. Oceanogr.* **8** (1978), 323.
20. W. C. Thacker, *J. Comput. Phys.* **37** (1980), 355.