l Report NOS OES 007

)RTHOGONAL CURVILINEAR GRID GENERATION

Silver Spring, Maryland
April 1995

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT OF COMMERCE
National Ocean Service
Office of Ocean and Earth Sciences
Marine Analysis and Interpretation Division
Coastal and Estuarine Oceanography Branch

Office of Ocean and Earth Science
National Ocean Service
National Oceanic and Atmospheric Administration
U.S. Department of Commerce

The Office of Ocean and Earth Sciences provides for the understanding of the coastal and ocean environment through the conduct of applied research and development in geophysics; the measurement, analyses, and product development of ocean and lake water levels; the collection, analyses, product development, and dissemination of coastal and global marine data; and the synthesis and interpretation with numerical and mechanistic modeling of global marine data sets. The Office cooperates with the U.S. Navy in conducting oceanographic activities for defense and mixed defense-civil sector purposes and applications.

It plans, develops and coordinates NOAA participation in Federally conducted oceanographic programs and activities, and facilitates cooperative programs, projects, and activities with the oceanographic research community. It monitors and analyzes oceanographic activities between NOAA and other organizations and agencies; identifies potential conflicts, overlaps, and opportunities for joint or cooperative efforts; and develops and maintains cooperative agreements, Memoranda of Understanding and other arrangements as appropriate to resolve issues and to ensure maximum benefits from programs of mutual interest. It develops and maintains inventories of oceanographic programs, projects, systems, and activities of other organizations and agencies to provide a basis for integrating current and future programs, systems and activities to ensure maximum efficiency, and effectiveness in meeting national goals and requirements. The Office conducts research and development; carries out theoretical studies, data analyses, and engineering development; and formulates and executes programs encompassing technological development and application to oceanography, geophysics, geodesy, and related fields.

For the Great Lakes, coastal estuaries, sea coast, and oceans, the Office plans, develops, and applies numerical and mechanistic models and produces predictions, forecasts, and analysis guidance materials of oceanographic and related marine meteorological phenomena; collects, analyzes, and disseminates tide and water-level observations and associated information; and computes water-level datums for hydrographic, marine boundary, and other special surveys. It evaluates and improves methods of data analysis; compares and integrates existing and new classes of data and products; provides and quality controls data sets and an array of output products; and assures science and technology transfer to and from the Office's programs and projects. The Office produces and disseminates operational marine environmental forecast and analysis guidance materials; manages and supports ocean climate studies; installs and operates real-time marine data collection systems; and formulates requirements for marine data sets and for data processing and communications systems; and designs and manages computer-based systems in support of these requirements.

NOAA Technical Report NOS OES 007

ORTHOGONAL CURVILINEAR GRID GENERATION

Kurt W. Hess

April 1995

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

National Oceanic and Atmospheric Administration
D. James Baker, Under Secretary

National Ocean Service
W. Stanley Wilson
Assistant Administrator

Office of Ocean and Earth Sciences
Melbourne G. Briscoe

Marine Analysis and Interpretation Division

Coastal and Estuarine Oceanography Branch
Bruce Parker

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Numerical methods of generating curvilinear orthogonal grids to be used by the Princeton Ocean Model are described. The concept of orthogonality is discussed and two methods for generating curvilinear orthogonal grids are explained in detail. Other approaches to generating an orthogonal grid are briefly mentioned. Properties along the parametric line are determined using the cubic spline interpolation technique. Orthogonality is assumed to hold either (1) at the center of a line segment intersecting two other lines (i.e., at two opposite sides of a cell), or (2) at the center of a cell. The key feature of the present methodology of generating curvilinear orthogonal grids is that by using three or more Given Lines, the orientation of cells deep within the domain can be controlled. Therefore, grids can be constructed that closely follow coastlines, channels, or barriers. Several applications of each method are discussed.

GLOSSARY

Boundary Lines	The two user-specified, non-intersecting lines (defined by a series of points) that bound the region of interest. See Figure 4.1.
Cell Height	The spacing of cells in the J-direction (along the direction of the Intermediate Lines) and denoted by DY. See Figure 4.5.
Cell Width	The spacing of cells in the I-direction (normal to the Intermediate Lines) and denoted by DX. See Figure 4.5.
Generator Line	One of the Given Lines (selected by the user) along which cell spacing (height) is specified. Grid generation begins with the cells along this line. See Figure 4.3.
Given Lines	Collectively, the Boundary Lines and any additional non-intersecting lines which are placed between the Bounda ry Lines to force the grid to follow terrain features such as channels, islands, or causeways. See Figure 4.1.
Intermediate Lines	The set of both Given Lines and the Interpolated Lines. See Figure 4.2.
Interpolated Lines	Program-generated set of lines located between the Given Lines and which are defined by points computed with either constant or variable cell spacing between the points that define the Given Lines. See Figure 3.2.

1. INTRODUCTION

The National Ocean Service (NOS), as part of its Tampa Bay Oceanography Project (NOS, 1990; Zervas, 1993) used the Princeton three-dimensional, orthogonal curvilinear grid model (Blumberg and Mellor, 1987) to simulate the circulation. Because Tampa Bay has irregular land-water boundaries (as do most estuaries), the use of numerically-generated curvilinear coordinates is desirable for following the shoreline and bathymetry more closely than rectangular or polar coordinates can.

NOS has developed a Fortran computer program, *CGEN*, to create generalized orthogonal, curvilinear coordinate system grids. Although other grid generation programs exist, many of them are limited to creating grids in a single, four-sided region. *CGEN*, however, allows for controlling cell size and orientation in different ways in many sub-regions so that grid cells can fit estuarine geometries more closely.

1.1. OUTLINE OF THE COMPUTER PROGRAM

Construction of the grid depends on the existence of sets of user-defined lines that determine the location and orientation of cells. These lines are defined in the Glossary.

The numerical construction of the grid proceeds in several steps:

- First, *CGEN* constructs a user-specified number of Interpolated Lines between the Given Lines at any given cell spacing to provide finer resolution.
- Next, the cell spacing is marked off at uniform lengths along the Generator Line.
- Starting at one particular point on the Generator Line, *CGEN* constructs all cells on the entire grid so that orthogonality is satisfied at each cell.
- Finally, the program creates an output file containing the coordinates of the cell corners.

The grid generated for Tampa Bay provides an example of the method (Figure 1.1).

To apply the computer program, the user must create inputs that define the region of interest:

- First, the user specifies the location of two non-intersecting Boundary Lines (defined by a series of latitude-longitude points) that bound the region of interest. Then additional non-intersecting lines may be placed between the first two to force the grid to follow terrain features such as channels, islands, or causeways. Collectively, these lines are called the Given Lines.
- The Generator Line is selected and the spacing of cells along it are specified.
- Finally, the number and spacing of cells to be created between the Given Lines are defined.

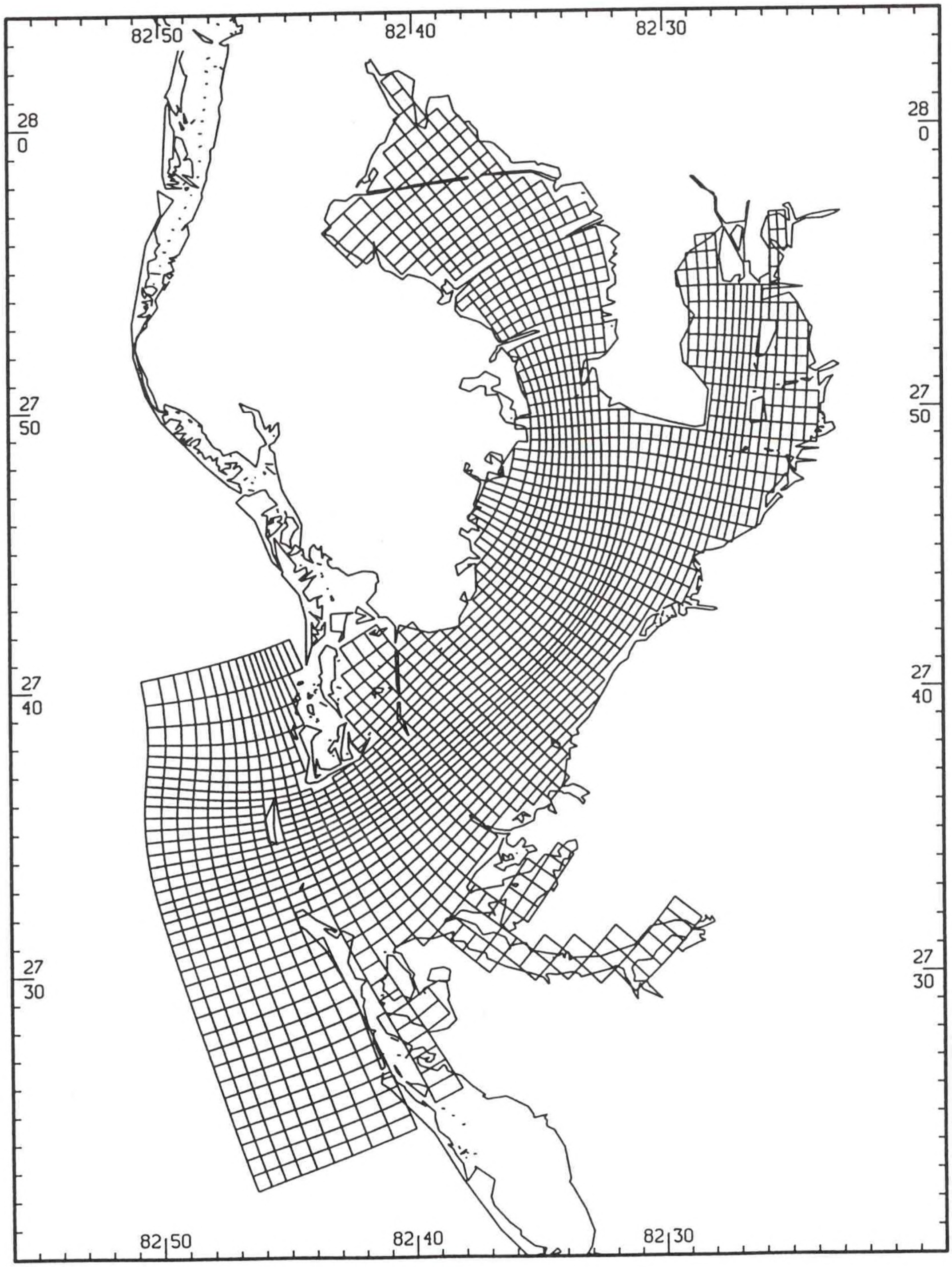


Figure 1.1. Orthogonal curvilinear grid for Tampa Bay produced by program *CGEN*. Grid cells are approximately 1000 meters on a side in the along-bay direction.

The input to the program consists of a data file containing the above information; examples are given in Appendices B through D.

1.2. ORGANIZATION OF THIS REPORT

The explanation of the gridding program begins in Section 2 with the introduction of the elementary principles about parametric lines and how they are generated from a set of points using the cubic spline interpolation. Next, the principle of orthogonality is introduced. Then, two methods of constructing an orthogonal line segment between two parametric lines is discussed.

The next Section explores ways of generating large numbers of non-intersecting parametric lines by interpolating between a few, initial parametric lines, using a variety of methods to control the spacing between the lines. There is also a discussion of the spacing along the parametric lines.

Section 4 describes the grid generation process using the principle developed in the previous two Sections but restating them in the array format that is used in the program.

Section 5 shows some applications of variable spacing and explains how the grid for Tampa Bay was generated.

Next, Section 6 describes the execution of the computer program and the input and output files. Section 7 summarizes the report.

2. ORTHOGONAL LINES AND CELLS

It is important to begin by establishing the concepts underlying the grid generation, including parametric lines and orthogonality. The following sections rely heavily on ideas discussed in Blumberg and Herring (1987), Mellor (1990), and Thompson et al. (1985).

2.1. PRINCIPLES

Parametric Lines

The properties of parametric lines are first established. Consider the three points A, B, and C (Figure 2.1) in (x,y) space (e.g. Mercator space). A continuous line can be constructed by any of several suitable interpolation schemes through the points. Here we construct a parametric line {x(r),y(r)}, defined along the coordinate direction r, with the cubic spline interpolation scheme (Appendix A).

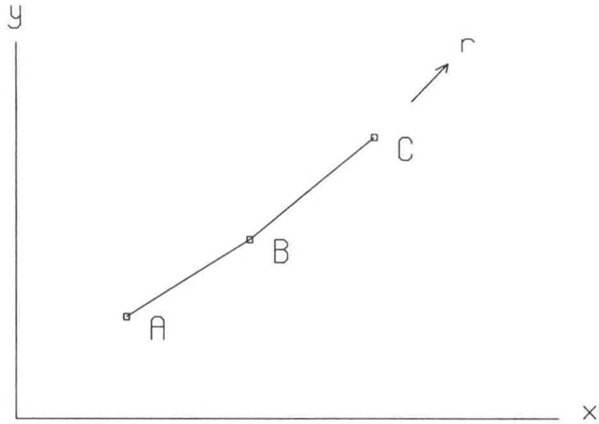


Figure 2.1. A parametric line in (x, y) space.

Consider the line in Figure 2.1. Point A has coordinates (x_a, y_a), where

$$x_a = x(r_a) \qquad and \qquad y_a = y(r_a). \tag{2.1}$$

Similarly, Point B has coordinates

$$x_b = x(r_b) \qquad and \qquad y_b = y(r_b), \tag{2.2}$$

and so on. The value of the coordinate r is defined in terms of the scaled distance along the interpolated line from Point A (the origin) to any other point, i, as follows:

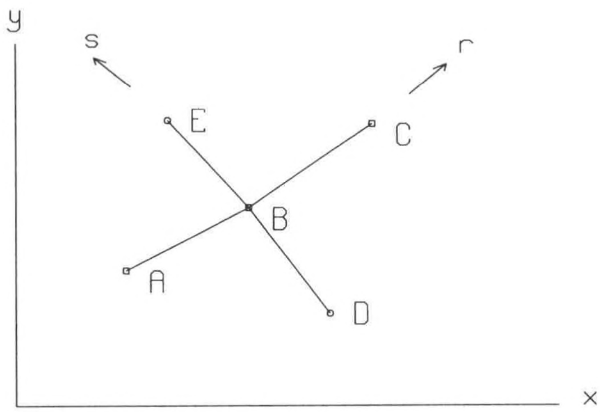
$$r_i = S_f \int_{point\ A}^{point\ i} (dx^2 + dy^2)^{1/2}, \tag{2.3}$$

where S_f is an arbitrary scale factor.

Orthogonality

The condition of orthogonality is now defined. Consider a set of two intersecting lines, one defined by the coordinate direction r, the other defined by the coordinate direction s and consisting of the points {x(s),y(s)}, as shown in Figure 2.2. At the point of intersection, B, the conditions for

Figure 2.2. Intersecting lines.



orthogonality are expressed in terms of the slopes of the lines

$$\frac{\partial x}{\partial r} = \frac{\partial y}{\partial s} \quad and \quad \frac{\partial x}{\partial s} = -\frac{\partial y}{\partial r}. \tag{2.4a,b}$$

Multiplying the left side of Eq. 2.4a by ∂x/∂s and the right side of Eq. 2.4a by -∂y/∂r (which equals ∂x/∂s by Eq. 2.5b) gives

$$\frac{\partial x}{\partial r}\frac{\partial x}{\partial s} + \frac{\partial y}{\partial r}\frac{\partial y}{\partial s} = 0, \tag{2.5}$$

as the condition of orthogonality. The numerical equivalent of Eq. 2.5 is then applied at specific locations within the grid to enforce orthogonality.

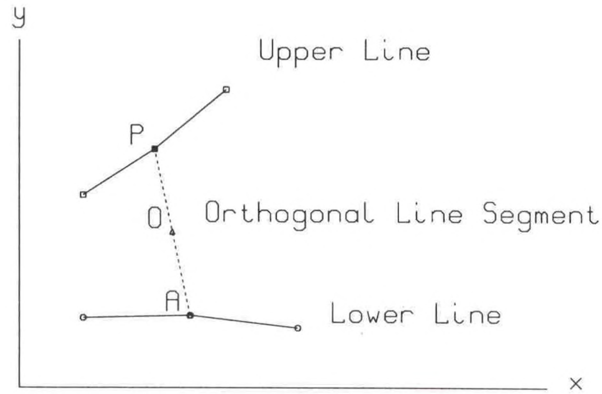
For the analytic case, two sets of lines can be constructed so that the condition of orthogonality applies at all points of intersection of any line from one set with all lines from the other set. Orthogonality also holds everywhere else, since additional orthogonal lines could be created with increasingly smaller spacing. By contrast, for orthogonal lines defined by sets of points in a grid (i.e., the cell corners) connected by straight line segments, orthogonality applies only in an approximate sense and only at specific locations that depend on the numerical procedure used to construct the grid. It is assumed here that these locations will be so closely spaced that orthogonality holds in a general sense throughout the entire grid.

2.2. METHOD A: NUMERICAL CONSTRUCTION OF AN ORTHOGONAL LINE SEGMENT

The first basic problem in grid generation, given two non-intersecting lines, is to construct an orthogonal line segment from a given point on one line to the other line. The location of the point on the second line is initially unknown and must be precisely selected so that orthogonality is satisfied. A numerical procedure for this construction, which uses the orthogonality condition (Eq. 2.5) and the spline interpolation technique, is described below.

Consider the two non-intersecting lines, called here the Lower and Upper Lines, as shown in Figure 2.3. The relative position of the Upper Line in x,y-space is arbitrary; it can be below (i.e., have smaller y values than) the Lower Line. The object is to construct a line segment from Point A on the Lower Line to the Upper Line. The point of intersection on the Upper Line, P, is found by iteration so that the orthogonality condition applies numerically at Point O, halfway between Points

Figure 2.3. Definition sketch for construction of an orthogonal straight line segment A-O-P. Point A is known and the location of Point P must be calculated. The orthogonality condition applies at Point O, which is halfway between Points A and P.



A and P. Methods for selecting the initial estimate of the location of Point P are discussed in Section 2.5. At all points on the Upper and Lower Lines, the values of the tangential (r-direction) gradients (i.e., ∂x/∂r and ∂y/∂r) can be determined by the cubic spline interpolation scheme. The values of the r-direction gradients at Point O are then approximated by averaging the computed gradients at A and P, i.e.,

$$\frac{Dx}{Dr} = \frac{1}{2}\left[\left(\frac{\partial x}{\partial r}\right)_A + \left(\frac{\partial x}{\partial r}\right)_P\right] \quad and \tag{2.6}$$

$$\frac{Dy}{Dr} = \frac{1}{2}\left[\left(\frac{\partial y}{\partial r}\right)_A + \left(\frac{\partial y}{\partial r}\right)_P\right]. \tag{2.7}$$

The values of the s-direction gradients at Point O are approximated by the following finite differences

$$\frac{Dx}{Ds} = \frac{x_P - x_A}{\Delta s} \quad and \tag{2.8}$$

$$\frac{Dy}{Ds} = \frac{y_P - y_A}{\Delta s}, \tag{2.9}$$

where value of Δs is approximated by computing the distance between A and P by

$$\Delta s = \pm S_f \left[(x_P - x_A)^2 + (y_P - y_A)^2 \right]^{1/2}. \tag{2.10}$$

The sign is positive when P is in the +s direction from A, and negative otherwise. The numerical equivalent to Eq. 2.5 at Point O is therefore

$$\frac{Dx}{Dr}\frac{Dx}{Ds} + \frac{Dy}{Dr}\frac{Dy}{Ds} = R, \tag{2.11}$$

where R represents a small residual and, for orthogonality, R = 0.

Based on the above relationships, the algorithm for constructing a single orthogonal line segment this way (Method A) is as follows:

1. Obtain an estimate of the location of Point P (x_p, y_p) on the Upper Line (see Section 2.5).

2. Using the spline interpolation technique, compute the tangential gradients $\partial x/\partial r$ and $\partial y/\partial r$ at Point A.

3. Using the spline interpolation technique, compute the tangential gradients $\partial x/\partial r$ and $\partial y/\partial r$ at Point P, and then using Eqs. 2.6 and 2.7 average them with the gradients at Point A to obtain the mean gradients, Dx/Dr and Dy/Dr, at Point O.

4. Compute Δs from Eq. 2.10.

5. Compute the s-direction gradients Dx/Ds and Dy/Ds at Point O from the finite-difference approximation to Eq. 2.4:

$$\frac{Dx}{Ds} = -\frac{Dy}{Dr} \quad \text{and} \quad \frac{Dy}{Ds} = \frac{Dx}{Dr}. \tag{2.12}$$

6. Obtain an intermediate estimate of x_p from Eq. 2.8 and y_p from Eq. 2.9:

$$x_P' = x_A + \frac{Dx}{Ds}\Delta s \quad \text{and} \quad y_P' = y_A + \frac{Dy}{Ds}\Delta s. \tag{2.13a,b}$$

7. If the absolute magnitude of Dx/Dr is larger than the absolute magnitude of Dy/Dr, obtain another value of y_p' from spline interpolation using x_p'. If the absolute magnitude of Dy/Dr is larger than the absolute magnitude of Dx/Dr, obtain another value of x_p' from spline interpolation using y_p'.

8. Average estimates with the old values to obtain the new values for x_p and y_p.

$$x_P^{new} = \frac{x_P' + x_P}{2} \quad and \quad y_P^{new} = \frac{y_P' + y_P}{2}. \qquad (2.14a,b)$$

With the new values, compute the gradient Dx/Ds from Eq. 2.8 and Dy/Ds from Eq. 2.9, then compute the residual at this iteration, R_i, from Eq. 2.11.

9. Apply the convergence criteria. These are (a) if R_i is sufficiently small,

$$R_i < 10^{-3} \left| \left(\frac{\partial x}{\partial r}\right)_A \left(\frac{\partial y}{\partial r}\right)_A \right| + \epsilon, \qquad (2.15a)$$

where ϵ is a small number (e.g., 10^{-10}, but which may be hardware-dependent), or (b) R_i has not changed significantly from the previous iteration,

$$|R_i - R_{i-1}| < 10^{-2} |R_{i-1}|, \qquad (2.15b)$$

or (c) the updated values of x_p^{new} and y_p^{new} have changed by a small amount,

$$|x_p^{new} - x_p| < 5\times10^{-3}(|x_p| + |y_p|) \quad and$$
$$|y_p^{new} - y_p| < 5\times10^{-3}(|x_p| + |y_p|), \qquad (2.15c)$$

or (d) the number of iterations is over 30, exit; otherwise, go to Step 3.

The criterion (2.15a) is based on the following assumptions: orthogonality (Eq. 2.5) is approximately true at Point A; each product in Eq. 2.5 is therefore about equal to the other; and Eq. 2.4b applies so that only derivatives with respect to r (which do not change during the iteration) appear. The term ϵ allows the criteria to be met when the product in Eq. 2.15a is zero.

This algorithm was found to converge rapidly and will construct an orthogonal line segment from A to P. Although this method is usually reliable, the results should be plotted and inspected. A possible problem is that some orthogonals may cross others; a check for this condition is discussed in Section 2.4.

Note that Mercator coordinates are used when the input data are in latitude-longitude coordinates to preserve angles. In any coordinate system, higher accuracy can obtained by subtracting the minimum value from each coordinate to reduce truncation error.

In the above method, orthogonality holds in the center of the line segment. Another method, described in the next Section, can be used to construct a grid for which orthogonality holds at the center of each cell.

2.3. METHOD B: NUMERICAL CONSTRUCTION OF AN ORTHOGONAL CELL

The following method is used to construct a grid cell when the locations of three of the four corners are known; the condition of orthogonality applies at the center of the cell. For example, if the location of the three points A, B, and C are known (Figure 2.4), it remains to determine the location of Point P that will satisfy orthogonality, given only that Point P must lie somewhere on the Upper Line.

Consider the cell in Figure 2.4. The gradients at Point O are approximated by averaging the numerical equivalents using the four corner points:

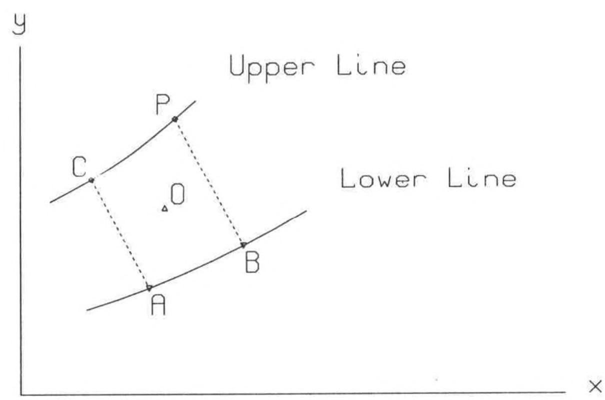


Figure 2.4. Definition sketch for construction of an orthogonal cell. The locations of Points A, B, and C are known and the location of Point P is to be determined. Orthogonality applies at Point O at the center of the cell.

$$\frac{Dx}{Dr} = \frac{1}{2}\left[\frac{x_P - x_C}{\Delta q_{PC}} + \frac{x_B - x_A}{\Delta q_{BA}}\right] \qquad \frac{Dx}{Ds} = \frac{1}{2}\left[\frac{x_P - x_B}{\Delta q_{PB}} + \frac{x_C - x_A}{\Delta q_{CA}}\right] \tag{2.16a,b}$$

$$\frac{Dy}{Dr} = \frac{1}{2}\left[\frac{y_P - y_C}{\Delta q_{PC}} + \frac{y_B - y_A}{\Delta q_{BA}}\right] \qquad \frac{Dy}{Ds} = \frac{1}{2}\left[\frac{y_P - y_B}{\Delta q_{PB}} + \frac{y_C - y_A}{\Delta q_{CA}}\right], \tag{2.17a,b}$$

where

$$\Delta q_{ij} = \pm S_f\left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{1/2} \tag{2.18}$$

and the sign is positive when Point i has a larger r or s value (whichever is applicable) than does Point j, and negative otherwise. The numerical equivalent of the orthogonality condition that applies at the center of this cell is Eq. 2.11.

The method for constructing an orthogonal cell this way (Method B) is as follows

1. Obtain an estimate of the location of Point P (x_p, y_p) on the upper line (see Section 2.5).

2. Estimate the gradients at Point O using the gradient approximations in Eqs. 2.16a,b and 2.17a,b and Δq_{ij} determined from Eq. 2.18.

3. If the number of iterations is two or more, apply the convergence criteria listed for Method A (see Eqs. 2.15a, b, c). Because numerical derivatives are used, the limit on the residual R_i used here is

$$R_i < 10^{-3}\left|\left(\frac{x_B - x_A}{\Delta q_{BA}}\right)\left(\frac{y_B - y_A}{\Delta q_{BA}}\right)\right| + \epsilon . \tag{2.19}$$

Or, if there have been a sufficient number of iterations (30), go to Step 5; otherwise continue.

4a. If |Dx/Dr| is smaller than |Dy/Dr|, got to Step 4b. Otherwise, update the estimate of Dx/Ds at O by

$$\frac{Dx}{Ds} = \frac{\frac{R_i}{2} - \frac{Dy}{Dr}\frac{Dy}{Ds}}{\frac{Dx}{Dr}} \tag{2.20}$$

(i.e., find Dx/Ds that reduces the residual by half), then find the next estimate of x_p from Eq. 2.16b, average x_p with the previous estimate, then find$_p$y from spline interpolation. Go to Step 2.

4b. Update the estimate of Dy/Ds at O by

$$\frac{Dy}{Ds} = \frac{\frac{R_i}{2} - \frac{Dx}{Dr}\frac{Dx}{Ds}}{\frac{Dy}{Dr}} , \tag{2.21}$$

then find the next estimate of y_p from Eq. 2.17b, average y_p with the previous estimate, then find x_p from spline interpolation. Go to Step 2.

5. Check the spline coordinate value along the r direction. If $r_p > r_c$, exit. If not, there are crossed orthogonals, so stop the grid generation.

This method was found to converge rapidly and can used throughout the grid, provided the locations of three of the four cell corners are known and the solution is started by using Method A to determine Point C at the left-most cell. Step 5 is a consistency check, although at the end of the sweep the entire grid is rechecked for crossed orthogonals as described in the next Section.

2.4. THE PROBLEM OF CROSSED ORTHOGONALS

A possible problem in the above two methods is the finding of a solution that has crossed orthogonals. Consider the cell depicted in Figure 2.4 and let the angle from line AB counterclockwise to line AC be denoted as ϕ_{BAC}. Then, if the cell is a quadrilateral,

$$\phi_{BAC} + \phi_{ACP} + \phi_{CPA} + \phi_{PBA} = 2\pi . \tag{2.22}$$

If this condition does not apply (as, for example, it would not if Point P in Method B has a smaller r value than Point C), the orthogonals are crossed. The program checks for this condition at all cells, and prints out the number at which Eq. 2.22 is not met. If there are crossed orthogonals, the simplest remedies are to reduce the curvature in the Given Lines in the region of the problem, or, if Method B were used, to use Method A.

2.5. ESTIMATION OF THE POINT OF INTERSECTION

There are two methods for obtaining the first estimate for the location of the intersection of the orthogonal segment and the Upper Line (Point P). Given the location of Point B on the Lower Line, the first method is to make Point P the point on the Upper Line with the same r-coordinate value. The second method is to make Point P the point on the Upper Line closest to Point B.

For an example of the first method, suppose that given Point B on the Lower Line (A-B-C) is known (Figure 2.5) and the starting estimate for Point P on the Upper Line (D-E) is required. If the two lines are similar in length and orientation, then an estimate of P is the point with the same fractional distance along the line. To construct it, first compute

$$r_p^{/} = \left(\frac{r_B}{r_C} \right) r_E \tag{2.23}$$

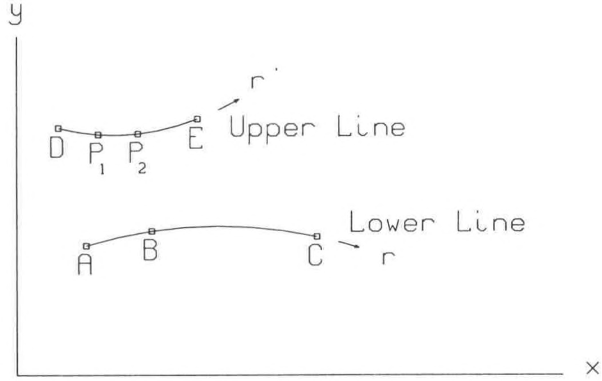


Figure 2.5. Definition sketch for selection of the first estimate of Point P on the Upper Line. Given the location of Point B on the Lower Line (A-B-C), Point P on the Upper Line (D-E) can be (1) P_1, the point with the same relative r-value as B, or (2) P_2, the closest point on the Upper Line to Point B.

and find x_p and y_p from the spline interpolation using r'_p as input. This point is denoted as P_1.

However, if line D-E is not similar to A-B-C, then the point on D-E closest to Point B may be a better first estimate. The closest point (P_2) can be found iteratively by selecting points along the line, computing the distance to Point B, and saving the location with the smallest distance.

The program *CGEN* uses the second method (in subroutine PFIRST), but the first method is also available in the code.

3. CELL SPACING, INTERPOLATED LINES, AND INTERMEDIATE LINES

The application of the previous two algorithms for generating an orthogonal grid can proceed only when a set of non-intersecting Interpolated Lines have been generated between each pair of Given Lines. The spacing of these lines determines the cell widths. Cell heights are determined by the spacing along the Generator Line. The details of the spacing and the interpolation process are explained below.

3.1. CELL SPACING BETWEEN GIVEN LINES

CGEN constructs a temporary line segment connecting pairs of corresponding points on adjacent Given Lines (corresponding points are, for example, the first point on one Given Line and the first point on the adjacent Given Line). This temporary line is divided into a number of intervals whose end points will (a) determine cell width and (b) define the Interpolated Lines. The set of points is found by interpolation across the space between pairs of points on adjacent Given Lines. For example, if A is a point on one Given Line and B is a corresponding point on the adjacent Given Line, there can be (for example) two intermediate Points: C and D. If Point A has coordinates (X_a, Y_a) and Point B has coordinate (X_b, Y_b), then the coordinates of Points C and D are determined by interpolation along the straight line connecting A and B by

$$X_n = X_A + f_n(X_B - X_A) \quad \text{and} \quad Y_n = Y_A + f_n(Y_B - Y_A), \tag{3.1}$$

where Point C corresponds to n = 1 and Point D to n = 2. The spacing function f_n is determined by

$$f_n = \frac{1}{w_T} \sum_{j=1}^{n} w_j \quad \text{and} \quad w_T = \sum_{j=1}^{N} w_j \,, \tag{3.2}$$

where w_j is the relative cell width and N is the number of cells spanning the distance from A to B. N equals 1 plus the number of Interpolated Lines (in the above example, N equals 3).

Three patterns of cell widths are described below. They are uniform widths, linearly increasing/decreasing widths, and geometrically increasing/decreasing widths.

Uniform Cell Widths

For uniform cell spacing, the cell width is

$$w_j = 1 \quad \text{and} \quad w_T = N. \tag{3.3}$$

The cell widths need not be uniform, however; they can increase or decrease across the line from A to B.

Linearly Increasing Cell Widths

Here cell widths increase in a linear manner according to

$$w_j = 1 + (j - 1)\delta . \tag{3.4}$$

However, since

$$R = \frac{w_1}{w_N} = \frac{1}{1 + (N - 1)\delta} , \tag{3.5}$$

then

$$\delta = \frac{R^{-1} - 1}{N - 1} . \tag{3.6}$$

Therefore

$$w_j = 1 + (j-1)\left(\frac{R^{-1} - 1}{N - 1}\right) \quad \textit{and} \quad w_T = N + \tfrac{1}{2}N(N-1)\left(\frac{R^{-1} - 1}{N - 1}\right) . \tag{3.7}$$

Cell widths decrease when R is less than 1.

Geometrically Increasing Cell Widths

If the ratio of the widths of adjacent cells is a constant, Q, so that

$$\frac{w_{j+1}}{w_j} = Q \quad \textit{or} \quad w_{j+1} = w_j Q, \tag{3.8}$$

then the width increases geometrically. If the ratio of the width of the first cell (i.e., the one closest to Point A) to the last cell is R, then letting

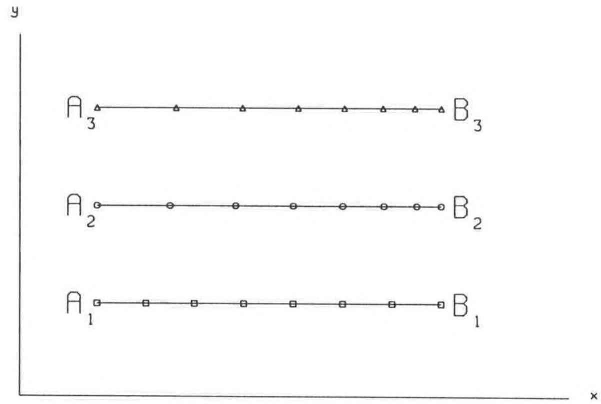
$$Q = R^{\frac{1}{1-N}} \tag{3.9}$$

gives, for $w_1 = 1$,

$$w_j = Q^{j-1} \quad \textit{and} \quad w_T = \sum_{k=1}^{N} Q^{k-1} = \frac{1 - Q^N}{1 - Q} , \tag{3.10}$$

where the summation of the terms in the geometrical progression has been replaced by its algebraic equivalent. Examples of the locations of interpolated points with three cells for R equal to unity, linearly increasing, and geometrically increasing are shown in Figure 3.1.

Figure 3.1. Interpolation between corresponding points on Give Lines A and B. The first temporary line (A_1, B_1) is divided into 7 segments with uniform spacing, the second line (A_2, B_2) with linearly-increasing spacing (R = 3.0), and the third (A_3, B_3) with geometrically-increasing spacing (R = 3.0). R is the ratio of the width of the first segment (on the left) to the width of the last segment (on the right). Segments will determine cell widths and the points that separate the segments will determine Interpolated Lines.



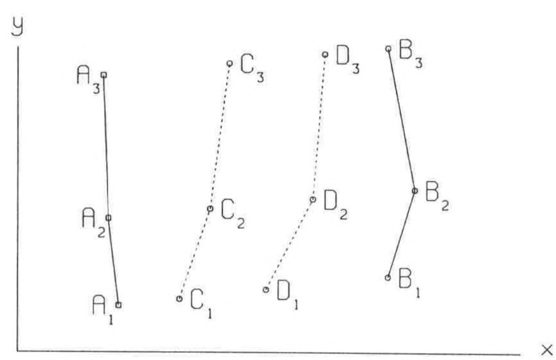
3.2. THE INTERPOLATED LINES

Using the above methods and cell spacing, *CGEN* constructs a set of Intermediate Lines between each pair of Given Lines by interpolation between all pairs of points defining the two Given Lines. For example, suppose that three cells are to span the space from the Given Line A (defined by the Points A_i) to Given Line B (defined by the Points B_i) in Figure 3.2. This means that two Interpolated Lines need to be constructed.

The procedure is as follows. First, two points are defined by interpolation between the Points A_1 and B_1: these are denoted C_1 and D_1. Then, another pair of Points (C_2, D_2) is interpolated between A_2 and B_2, and the value for R may differ from that used in the first interpolation. This process is continued for all pairs of points on the pair of Given Lines. The two Interpolated Lines are then generated by spline fit, first through the Points C_i and then through the Points D_i. This process is then repeated as needed for the remaining pairs of Given Lines.

The Given Lines A and B, along with the interpolated lines C and D, are known collectively as the Intermediate Lines. These lines form half of the final set of lines of the grid; the orthogonals form the other set.

Figure 3.2. Four Intermediate Lines after interpolation between the lines {A_i} and {B_i}. Note that for the interpolation between A_1 and B_1, R = 0.5 and geometrically-increasing spacing is used; for the interpolation between A_2 and B_2, R = 1 and uniform spacing is used; and for the interpolation between A_3 and B_3, R = 2 and linearly-decreasing spacing is used.



3.3. SPACING ALONG THE GENERATOR LINE

Spacing of cells along an Interpolated Line (cell height) is determined from spacing along the adjacent line. To start the solution, however, one of the Given Lines is user-selected to be the Generator Line. Spacing along the Generator Line is uniform (and selected by the user) and is determined by the method described below. Along other lines, the numerical scheme computes cell spacing to satisfy orthogonality.

Suppose that the cell spacing along the Generator Line is to be ΔL (meters). For the Mercator transformation for north latitude ϕ (degrees) and east longitude λ (degrees),

$$X_m(\lambda, x_o) = \frac{\pi}{180}\lambda - x_o \qquad \textit{and} \tag{3.11a}$$

$$Y_m(\phi, y_o) = \log_n\left(\frac{1 + \cos(90 - \phi)}{\sin(90 - \phi)}\right) - y_o, \tag{3.11b}$$

where x_o and y_o are arbitrary reference values. The distance (in Mercator space) corresponding to ΔL is approximately

$$D = Y(\phi_{ref} + \frac{1}{2}\frac{\Delta L_{nmi}}{60}) - Y(\phi_{ref} - \frac{1}{2}\frac{\Delta L_{nmi}}{60}), \tag{3.12}$$

where ΔL_{nmi} is the cell size in nautical miles and ϕ_{ref} is a reference latitude for the region.

Consider some Point J on the Generator Line. The location of the next cell crossing point along the line, J+1, is found by iteration so that

$$\left((x_{J+1} - x_J)^2 + (y_{J+1} - y_J)^2\right)^{1/2} = D, \tag{3.13}$$

provided the parametric values satisfy

$$r_{J+1} > r_J. \tag{3.14}$$

The methods of generating the Interpolated Lines and the Generator Line have now been explained. The process of generating the entire grid will be discussed in the next Section.

4. THE GRID GENERATION PROCESS

The previous sections have described the principles of grid generation. Here the relationships are presented in array format (in preparation for coding in Fortran) and put together in an overview. Application to the Princeton model grid (Mellor, 1990) is also discussed.

4.1. THE GIVEN LINES

The Given Lines are chosen by the user as inputs to the problem, and they delineate the bounds of the area of interest and define the grid orientation (Figure 4.1). Each line is defined by a set of points in latitude-longitude space. More formally, the locations of the points along the lines are saved in the arrays $XG_{L,M}$ and $YG_{L,M}$, where L is the index for the number of the line and M is the index for the number of the point along the line. The total number of Given Lines is LMAX. *CGEN* allows the points to be defined in either non-dimensional {x,y} space (FORM=1) or in {latitude,longitude} space (FORM=2). Also, to allow the spline interpolation to operate properly, the number of points in each line must be at least four.

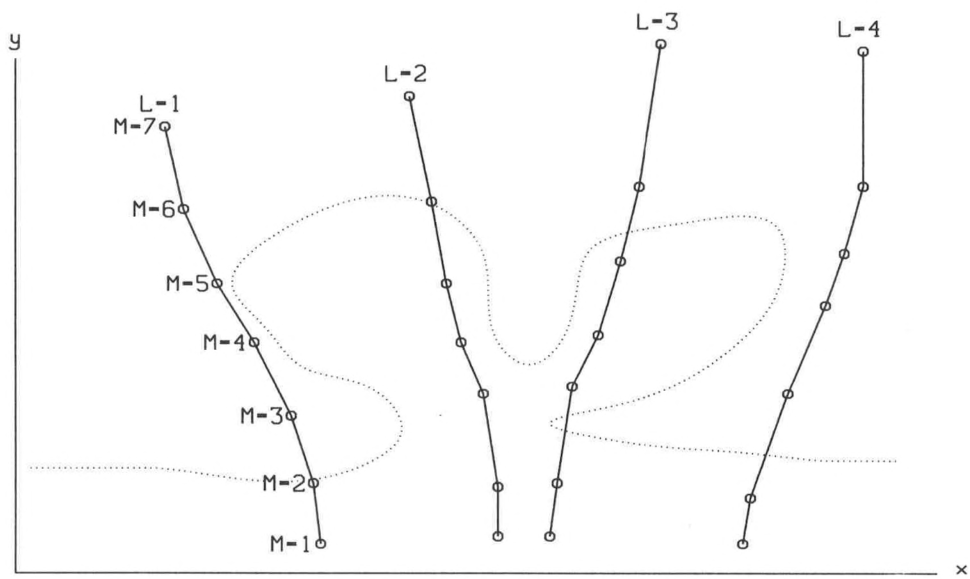


Figure 4.1. Sample location of four Given Lines (L=1, 2, 3, and 4). Each Given Line here is defined by seven points, indexed M = 1, 2, .., 7. The dotted line represents the coastline. Lines L=1 and L=4 are the Boundary Lines.

The outer lines (L=1 and L=4) are the Boundary Lines and they enclose the embayment of interest. One of the Given Lines is designated as the Generator Line (L = LGEN). The number of points on the Generator Line is MMAX. The number of points in the other Given Lines need not be equal to MMAX; if they are not, the program *CGEN* will define MMAX points by spline fit when creating the Interpolated Lines (see Section 4.2).

4.2. THE INTERPOLATED LINES

After the Given Lines have been read in, the Interpolated Lines are constructed. Between each pair of Given Lines there will be a constant number of cells, $NCELL_L$. The set of Given Lines plus Interpolated Lines is called the Intermediate Lines. The index denoting the number of the Intermediate Line is I. The points that define the Intermediate Lines are saved in the intermediate grid arrays $XI_{I,M}$ and $YI_{I,M}$, where $1 \le I \le IMAX$ and $1 \le M \le MMAX$. IMAX is computed as

$$IMAX = 1 + \sum_{L=1}^{LMAX-1} NCELL_L. \tag{4.1}$$

An example of the Intermediate Lines is shown in Figure 4.2.

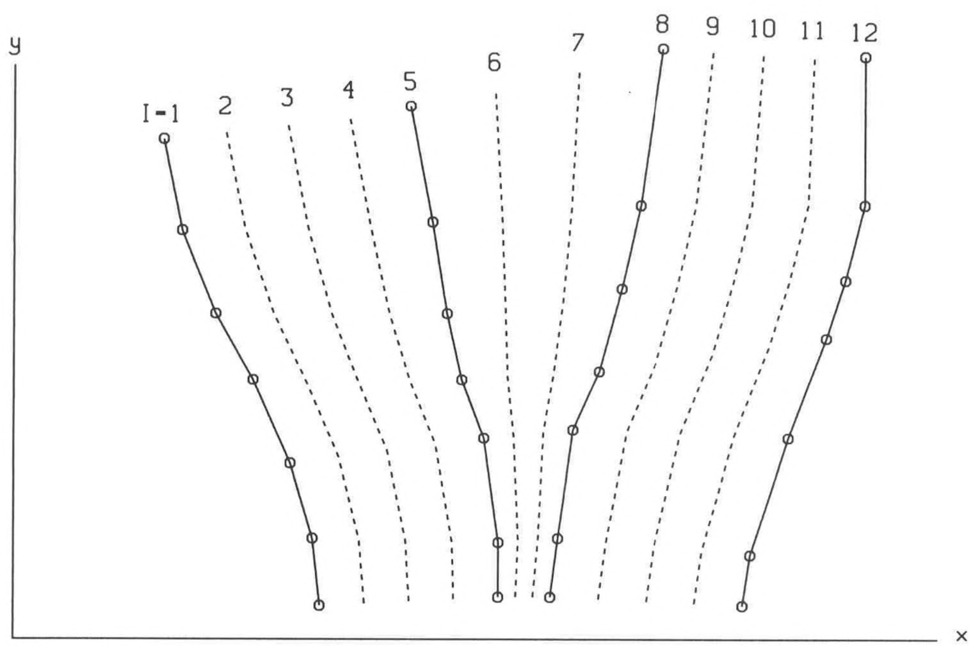


Figure 4.2. Location of the Intermediate Lines, indexed from I=1 to I=12. Here, $NCELL_1$=4, $NCELL_2$=3, and $NCELL_3$=4. The Given Lines (see Figure 4.1) are solid and the Interpolated Lines are dashed.

Each pair of points along a pair of lines will have the value of the ratio of the first cell width to the last cell width, $R_{L,M}$, and the coordinates of the Intermediate Lines are computed as follows:

$$XI_{I,M} = XG_{L,M} + f_{I,M}(XG_{L+1,M} - XG_{L,M}) \tag{4.2a}$$

$$YI_{I,M} = YG_{L,M} + f_{I,M}(YG_{L+1,M} - YG_{L,M}), \tag{4.2b}$$

where $f_{I,M}$ is determined from Eq. 3.2, given the type of spacing, $NCELL_L$, and $R_{L,M}$. To find L from I, first define the function

$$K(L) = 1 + \sum_{k=0}^{L-1} NCELL_k, \tag{4.3}$$

where $NCELL_o = 0$. Then, L is determined so that

$$K(L) \leq I < K(L+1). \tag{4.4}$$

4.3. THE GENERATOR LINE

Cell corners are marked off along the Generator Line by the principles discussed in Section 3.3. Program *CGEN* limits the subdivision to the segment of the Generator Line between points M = 1 and M = MMAX. These points are indexed from J = 1 to J = JMAX, and become points in the final grid denoted as $XF_{IGEN,J}$, $YF_{IGEN,J}$, where

$$IGEN = 1 + K(LGEN). \tag{4.5}$$

Cell spacing created along the Generator Line are shown in Figure 4.3.

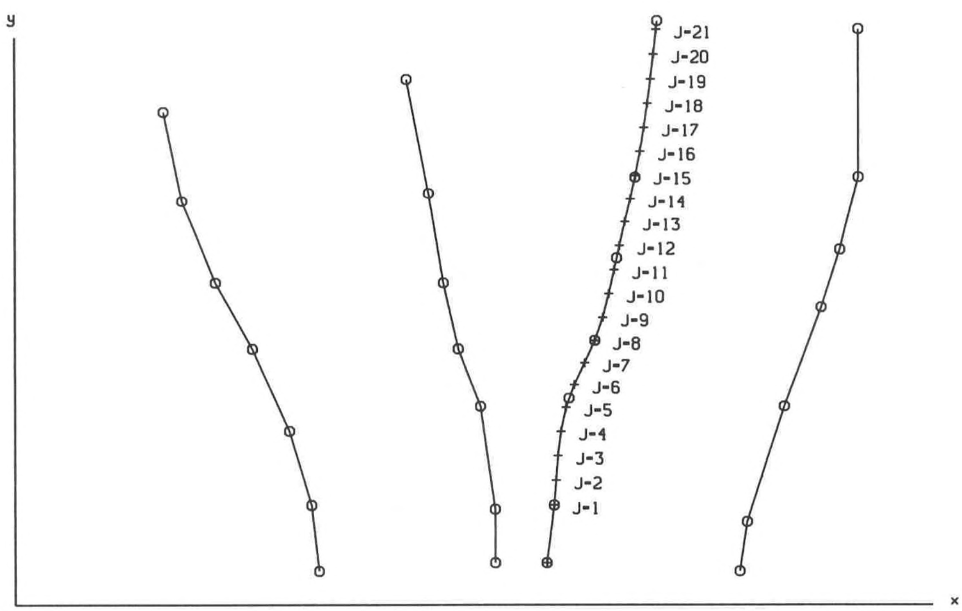


Figure 4.3. Grid Generator Line at L=3. The location of cell corners (denoted by ticks and indexed J=1 to J=21) has been determined for constant spacing, starting from M=2 and continuing to just before M=7.

The first cell corner point is set equal to the JFIRST point on the Generator Line, i.e.,

$$XF_{IGEN,1} = XG_{LGEN,JFIRST} \qquad and \qquad YF_{IGEN,1} = YG_{LGEN,JFIR.} \qquad (4.6)$$

In Figure 4.3, JFIRST = 2.

4.4. THE GRID GENERATION PROCESS

When the cell corners along the Generator Line are constructed, *CGEN* will begin creating the orthogonal cells, starting at the point (I = IGEN, J = JFIRST), and proceeding first on the side I < IGEN (if it exists), then on the other side I > IGEN (if that exists). Since the construction begins on this point on the Generator Line, the location of this point in the region (especially when modeling an estuary) is crucial to fitting the final grid to coastline features. The results are shown in Figure 4.4.

Construction proceeds as follows. First, a point of the adjacent Intermediate Line (e.g., at I=IGEN-1, J=JFIRST) is found by Method A. Then, the fourth cell corner (at I=IGEN-1, J=JFIRST+1) is determined either by Method A or Method B, depending on the user's option. The remaining cell corners along the entire line I=IGEN-1 are then determined by the same method.

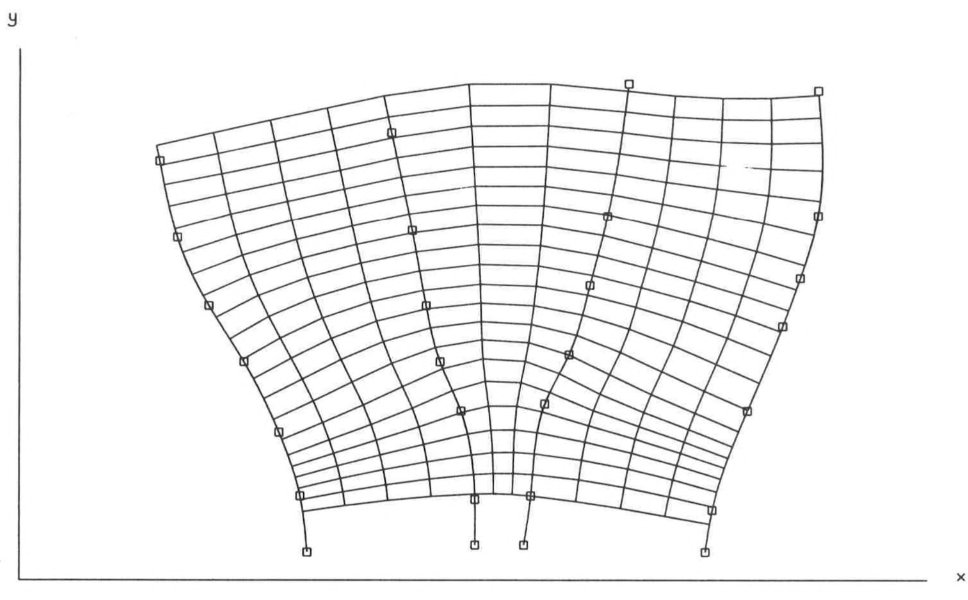


Figure 4.4. Given Lines and the grid generated from them.

The process is repeated at line I=IGEN-2, and so on to line I=1. *CGEN* then repeats the preceding steps at Line I=IGEN+1, IGEN+2, etc., until the grid is complete. The final results are stored in arrays $XF_{I,J}$, $YF_{I,J}$, which are the lower corners (closest to the origin in x,y space) of each cell.

4.5. THE PRINCETON MODEL GRID

The grid generated this way can be easily adapted for use in the Princeton model, which has a grid indexed I in one coordinate direction, J is indexed in the other direction, and $1 \leq I \leq IM$, $1 \leq J \leq JM$. To adapt the grid to the Princeton model code, the user selects the subset of the generated grid that most economically fits the region of interest.

In order to be used in the Princeton model, each cell's length, width, area, and orientation must be computed (cell depths can be obtained from a gridded bathymetry file, or by plotting the grid on semi-transparent paper and placing the paper over a nautical chart). For a rectangular grid, calculation of distances and areas is straightforward because the cell width, DX, and cell length, DY, are constants. To compute lengths and areas for the curvilinear grid defined by latitude-longitude points, grid coordinates must be converted to distance-conserving coordinates. This is accomplished with the equirectangular map projection as follows:

$$X = \alpha(\lambda - \lambda_o)\cos(\phi_{ref}) \tag{4.7a}$$

$$Y = \alpha(\phi - \phi_o), \tag{4.7b}$$

where X and Y are in meters, west longitude is negative, λ_o and ϕ_o are the minimum latitude and longitude respectively, ϕ_{ref} is a user-selected reference latitude chosen to be at approximately the central latitude of the gridded region, and $\alpha = 1.1112 \times 10^5$ m/deg converts degrees to meters.

For computational purposes, the shape of a single cell in the curvilinear grid is assumed to be a simple quadrilateral. Consider the cell (1, 2, 3, and 4) in Figure 4.5, where Points 1 and 2 are on the same row, J, and Points 3 and 4 are on an adjacent row, J+1; Points 1 and 3 are on the same column, I, and Points 2 and 4 are on the next column, I+1. Point A is halfway between Points 1 and 3, etc. DX is then the distance from A to B, and DY is the distance from C to D. Using a subscript to denote,
the point, the grid sizes are

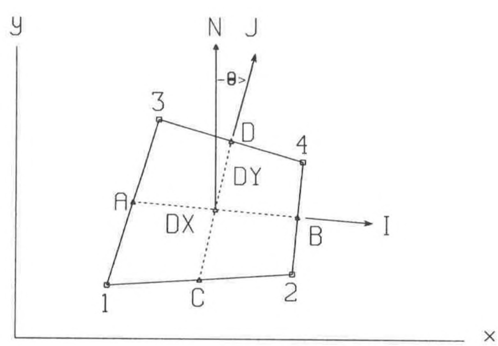


Figure 4.5. Cell (I,J) in x,y-space.

$$DX = \left(\frac{1}{4}(X_4 + X_2 - X_1 - X_3)^2 + \frac{1}{4}(Y_4 + Y_2 - Y_1 - Y_3)^2\right)^{1/2} \tag{4.8}$$

$$DY = \left(\frac{1}{4}(X_4 + X_3 - X_1 - X_2)^2 + \frac{1}{4}(Y_4 + Y_3 - Y_1 - Y_2)^2\right)^{1/2}. \tag{4.9}$$

Cell area is approximated by the simple product

$$ART = (DX)(DY), \tag{4.10}$$

to be consistent with model computations. The area associated with the locations of the U and V components of velocity are

$$ARU_{I,J} = \frac{1}{2}(ART_{I,J} + ART_{I-1,J}) \tag{4.11}$$

$$ARV_{I,J} = \frac{1}{2}(ART_{I,J} + ART_{I,J-1}). \tag{4.12}$$

Also needed is the rotation angle, θ, which is defined here as the angle (positive clockwise) from north to the cell's y-axis. Theoretically, Mercator coordinates should be used since they are angle-conserving, but since the distances here are small, the equirectangular coordinates will be used. With reference to Figure 4.5, the angle is computed by

$$\theta = \frac{\pi}{4} - \frac{1}{4}\left[\tan^{-1}\left(\frac{Y_3 - Y_1}{X_3 - X_1}\right) + \tan^{-1}\left(\frac{Y_4 - Y_2}{X_4 - X_2}\right) + \tan^{-1}\left(\frac{Y_2 - Y_1}{X_2 - X_1}\right) + \tan^{-1}\left(\frac{Y_4 - Y_3}{X_4 - X_3}\right)\right]. \tag{4.13}$$

Therefore, for wind components to the east, W_e, and to the north, W_n, conversion to local model directions in each cell are

$$W_x = W_e \cos\theta - W_n \sin\theta \quad and \quad W_y = W_e \sin\theta + W_n \cos\theta, \tag{4.14}$$

and to reorient currents in each cell into eastward and northward components,

$$U_e = U_y \sin\theta + U_x \cos\theta \quad and \quad U_n = U_y \cos\theta - U_x \sin\theta. \tag{4.15}$$

With the above relationships for using the grid in the Princeton model, the analytical development is now complete.

5. APPLICATIONS

The above methods have been successfully applied to several cases, but for a general set of input conditions care must be taken to ensure that a useful grid is generated. The following section explains certain details of the application, limits of application, and ways to improve on results.

5.1. GENERAL APPLICATION IN X-Y SPACE

Adjustments to Cell Spacing

Consider first the simple case of two vertical, parallel Given Lines enclosing a third, sinusoidal Given Line in x,y space (Figure 5.1a). Each line is determined by 13 points, and there are to be 10 cells in the region to the left of the Generator Line (the central Given Line) and 13 cells to the right. The input data file is given in Appendix B.

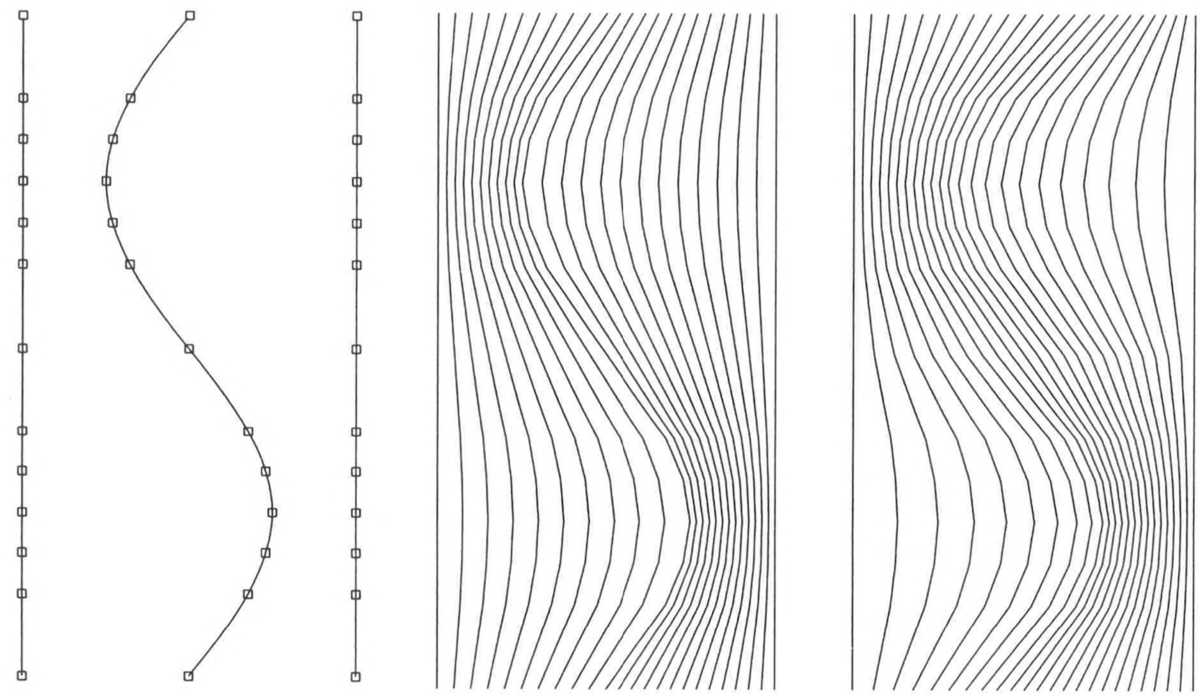


Figure 5.1. (a) Given Lines, (b) Intermediate Lines with constant cell spacing, and (c) Intermediate Lines with variable cell spacing, designed to better match widths across the central Given Line.

CGEN was then run to produce Intermediate Lines. The ratio R is initially set to 1 everywhere and Method A was used. The Intermediate Lines run generally in the vertical direction (Figure 5.1b). Even though the number of cells on either side is nearly identical, the line spacing on either side of the Generator Line can be quite different. This is especially true near the points of maximum curvature of the Generator Line. One way to remedy this problem is to alter R, the ratio of the widths of the first and last cells.

For example, consider the region between two Given Lines with uniform cell widths, W_U. Let the region to the left be denoted by A and the right-most cell to the left, W_A, and the region to the right by B and the left-most cell width on the right, W_B. The first cell width can be made equal to W_A as follows:

$$w_{first} = W_U - \Delta = W_A \qquad so\ that \qquad \Delta = W_U - W_A. \tag{5.1}$$

If the cell widths in the central region are linearly increasing, then any amount subtracted from the first cell must be added to the last cell (and proportionally for cells in between) to maintain the same total width. Therefore, in the central region the ratio of the width of the first cell to the last cell required to match the last cell width in region A is

$$R_A = \frac{w_{first}}{w_{last}} = \frac{W_U - \Delta}{W_U + \Delta} = \frac{W_A}{2W_U - W_A}, \tag{5.2}$$

provided that

$$2W_U - W_A > 0. \tag{5.3}$$

Suppose the object is to match the cell width to the right. The last cell width can be made equal to W_B as follows:

$$w_{last} = W_B = W_U - \Delta \qquad so\ that \qquad \Delta = W_U - W_B. \tag{5.4}$$

If the widths in the central region are linearly increasing, then for the ratio of the width of the first cell to the last cell required to match the first cell in region B is

$$R_B = \frac{w_{first}}{w_{last}} = \frac{W_U + \Delta}{W_U - \Delta} = \frac{2W_U - W_B}{W_B}, \tag{5.5}$$

provided

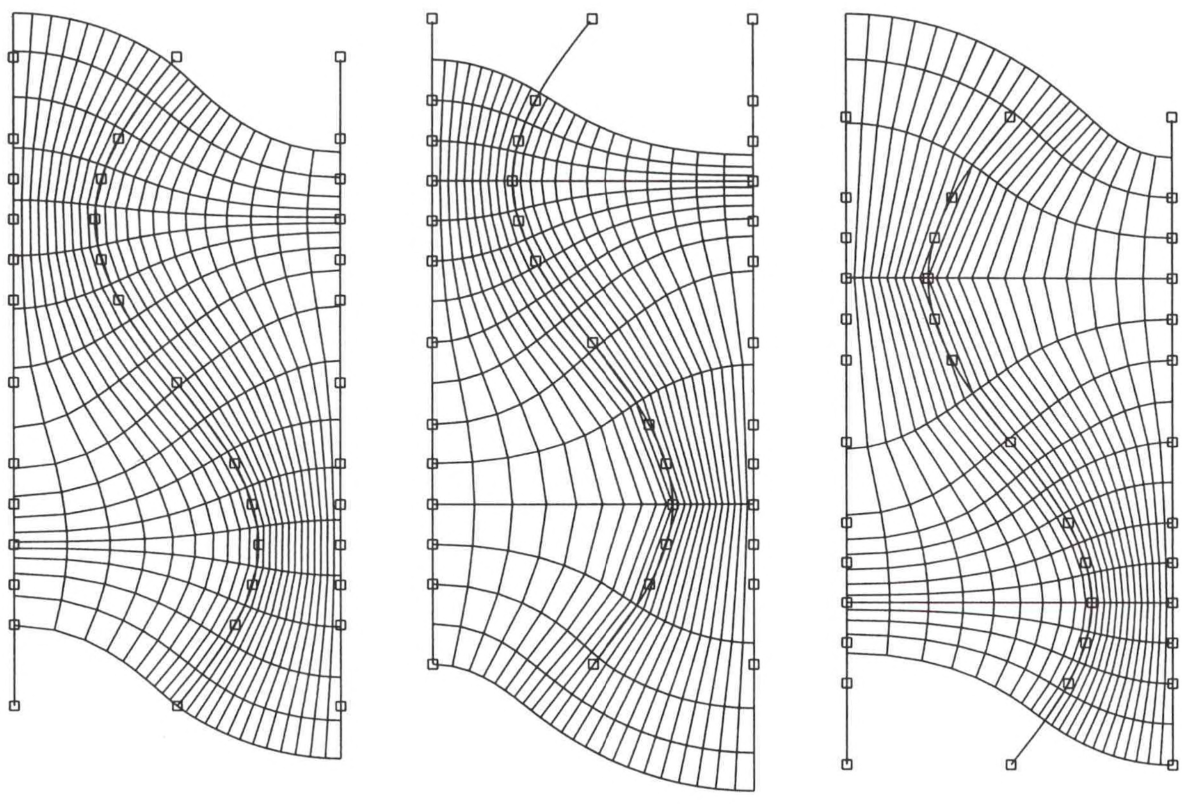
$$2W_U > W_B. \tag{5.6}$$

Program *CGEN* prints out cell widths and matching ratios (for both linear and geometric width changes) for all points along each Given Line pair. These data were used to adjust the R values to the left of the Generator Line for points in the lower half of the figure, and the R values to the right of the Generator Line for points in the upper half. The results of cell matching (see the file in Appendix C) are shown in Figure 5.1c.

Selection of the Generator Line

For this case, the middle line (L = 2) is the Generator Line and Method A was used to generate the grid (Figure 5.2a). The orthogonals, which are constructed perpendicular to the centerline, run generally in the lateral direction.

Note that the lowest curved line of the grid passes through the first point on the Generator line, but the precise crossing point of the uppermost curved line cannot be predetermined; it crosses the Generator Line somewhere between the last and next-to-last points. Also note the focusing effect near both the lower left corner and the upper right corner of the grid. This is due to concavity of the closest portion of the Generator Line. Conversely, the widely-spaced cells along the Boundary lines on the upper left and lower right are due to the convexity of the Generator Line.



(a) Grid with LGEN=2. (b) Grid with LGEN=1. (c) Grid with LGEN=3.

Figure 5.2. Given Lines, Interpolated Lines, and grids generated with various Generator Lines (Method A).

Selection of Method A or B

For this case, the middle line is the Generator Line and both Methods A and B were used to generate the grid (Figure 5.3). The orthogonals, which are constructed perpendicular to the centerline, run generally in the lateral direction.

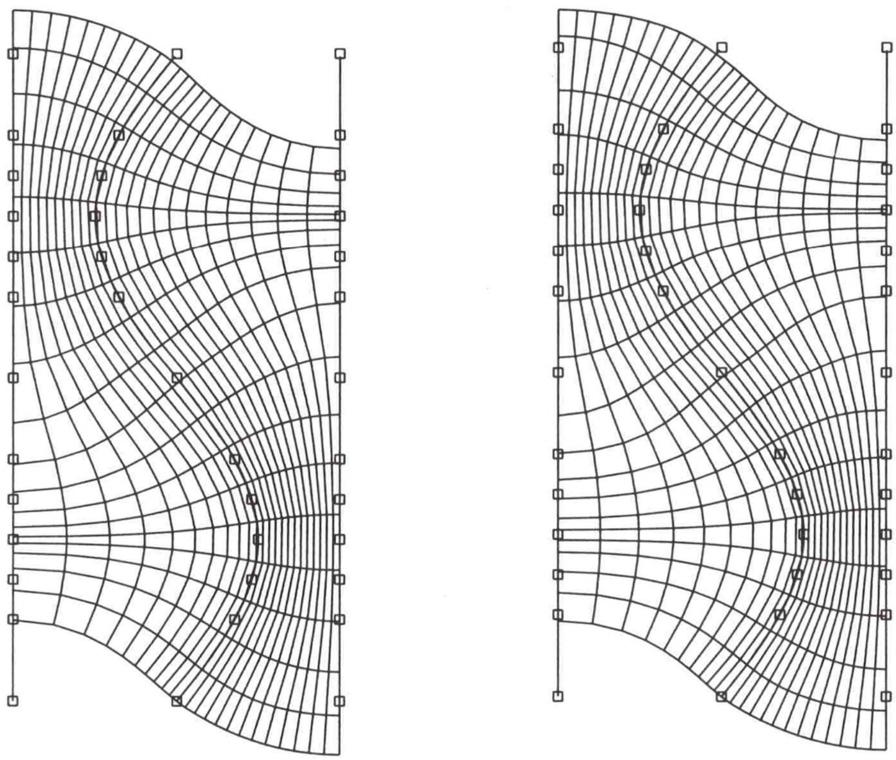


Figure 5.3. Grid drawn when Generator Line is 2 and (a) Method A and (b) Method B.

When the Generator Line was set to 1 or 3, the method did not converge because some of the orthogonals crossed.

5.2. APPLICATION TO TAMPA BAY

The above methods for grid generation have been successfully applied to Tampa Bay. The following steps were taken to create the input data file (the use of a NOAA navigation chart or other bathymetric chart is necessary).

Categorize the region of interest as either (a) coastal or (b) estuarine. For purposes of generating a grid, a coastal region will consist of a land-water shoreline boundary and a deep-water open boundary running approximately parallel to the shoreline. An estuarine region will consist of a long, narrow embayment bordered by two roughly parallel shoreline boundaries.

For a coastal region, create a Boundary Line (in either general or earth coordinates) that represents the location of the offshore boundary [this line will be designated as the Generator Line]. The Boundary Line is defined by a series of points (start with about 10) in earth coordinates. Note that the final orthogonal grid will begin at the second point on this line and end at or near the next-to-last point. Next select a Boundary Line landward of the shoreline, using (if possible) about the same number of points. If a shore-following grid line is desired, define an Interior Line with the about the same number of points that follows the coast.

For an estuarine region, create two Boundary Lines (in either general or earth coordinates) that bound all land on two sides of the estuary. These Boundary Lines are defined by a series of points (start with about 10). Next, select one or more Interior Lines to represents the location of the main axis of the embayment or the location of causeways, islands, rivers, or channels; use about the same number of points. Select one line, such as that along the main axis, as the Generator Line. Note that the final orthogonal grid will begin at the second point on this line and end at or near the next-to-last point.

Finally, select the grid spacing (meters) along the Generator Line (cells will be equally-spaced) and select the number of cells to be constructed between each pair of Boundary Lines and/or Interior Lines.

The data file for Tampa Bay created by the above method is shown in Appendix D. Figure 5.4 shows the Given Lines plotted against the local geography. The program *CGEN* is now run to create the grid. Figure 5.5 shows the final grid. Details of the program are given in the next Section.

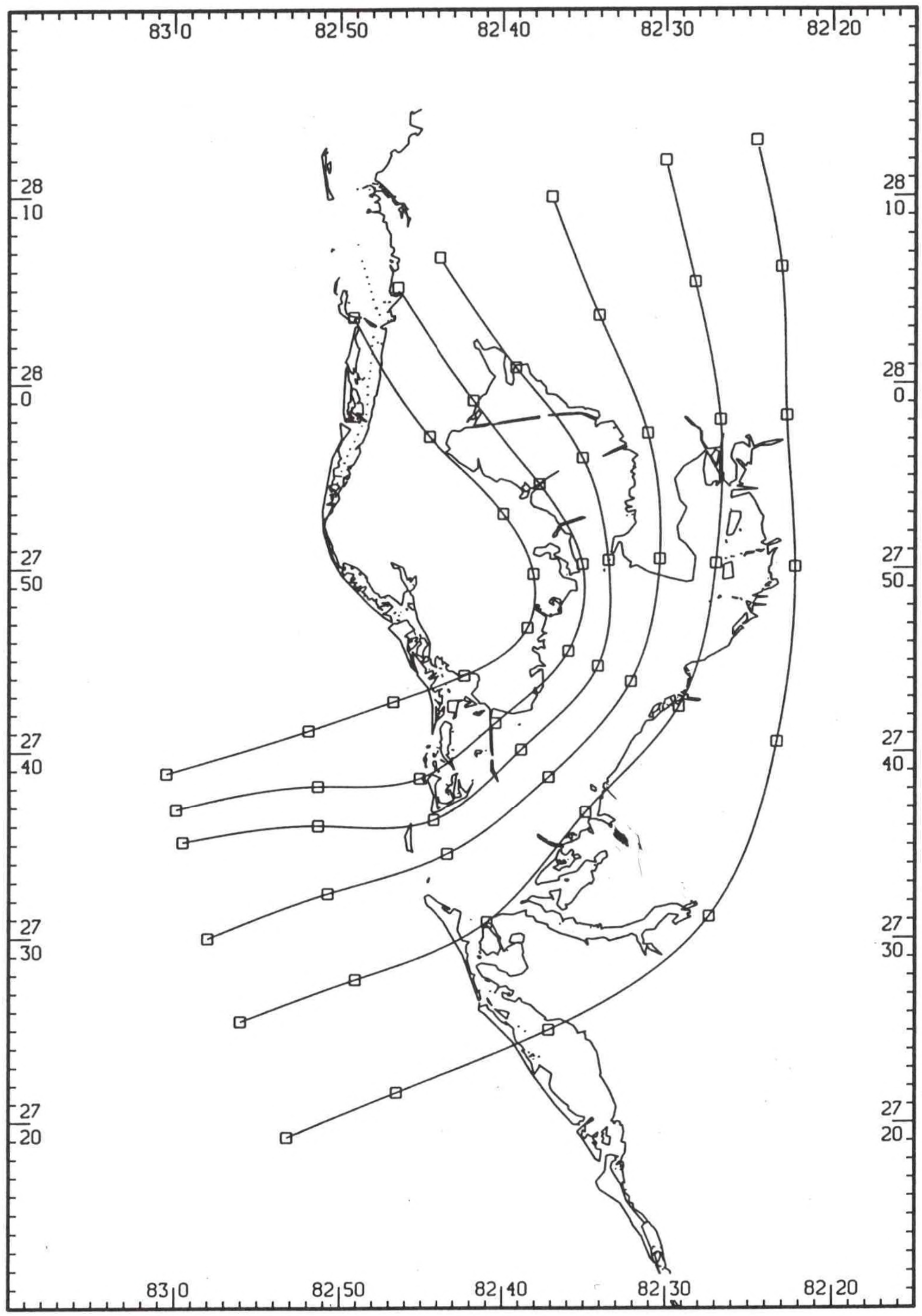


Figure 5.4. Given Lines for the Tampa Bay grid. Here, the fourth line from the upper left is the Generator Line.

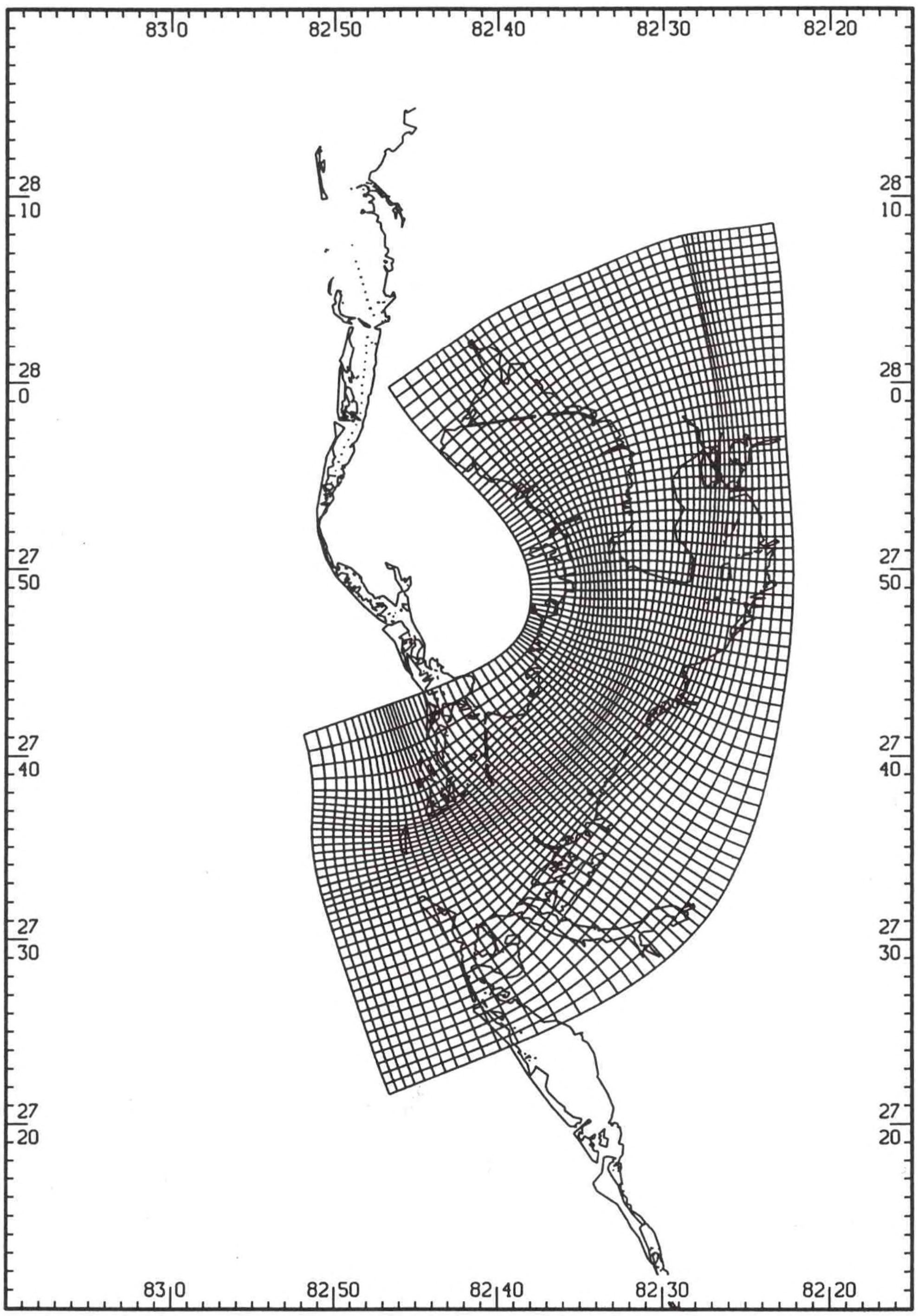


Figure 5.5. The final Tampa Bay grid plotted over the coastline.

6. COMPUTER PROGRAM *CGEN*

A Fortran computer program, *CGEN*, has been developed in NOS to generate an orthogonal curvilinear coordinate system grid using methods described in the previous sections. The Fortran 77 program runs on a Hewlett-Packard 9000/825 under Unix. This Section describes the program structure, data input and output files, and a guide for running the program.

6.1. PROGRAM STRUCTURE

The program *CGEN* has the following structure:

```
MAIN
 |
 |------READIN------FFOPEN
 |                  RDLINE
 |                  UMERC
 |                  VMERC
 |
 |------INTERP------MATCH
 |
 |------CHECK
 |
 |------DSAVE-------UINV
 |                  VINV
 |
 |------CENLIN------SPLINE
 |
 |------SWEEP-------PFIRST
 |                  ORTH_A------SPLINE
 |                  ORTH_B------SPLINE
 |      CHECK
 |
 |------DSAVE-------UINV
 |                  VINV
 |
(end)
```

Each program element in *CGEN* performs part of the gridding task.

Subroutine READIN opens the input data file with FFOPEN, reads the file with RDLINE, and if necessary converts latitude-longitude coordinates to Mercator coordinates (Eq. 3.11) in functions UMERC and VMERC.

Subroutine INTERP generates the Interpolated Lines from the Given lines. MATCH determines the ratios needed to match first and last cell widths with widths in adjacent regions across Given Lines.

Subroutine CHECK looks for crossed orthogonals using Eq. 2.22.

Subroutine CENLIN marks off the locations of cell corners along the Generator Line using the cubic spline routine SPLINE. Subroutine DSAVE converts (if necessary) the Intermediate Lines back to latitude-longitude coordinates using the inverse Mercator functions UINV and VINV, where

$$\lambda = (X_m + x_o)\frac{180}{\pi} \tag{6.1}$$

$$\phi = 90 - 2\arctan(e^{-Y_m - y_o})\frac{180}{\pi}. \tag{6.2}$$

DSAVE then writes the output to an intermediate file.

Subroutine SWEEP loops through the grid, starting at the Generator Line. At each cell, the estimated point on the adjacent line is found by PFIRST, then the final location is determined by constructing the orthogonal line segment by Method A in ORTH_A, or an orthogonal cell by Method B in ORTH_B. Subroutine SPLINE is called to interpolate points and slopes along the lines by cubic spline. CHECK looks for crossed orthogonals.

Subroutine DSAVE is again called to write the output (XF, YF) to a result file.

6.2. INPUT AND OUTPUT FILES

The program is run with sets of input and output files. The input file consists of the Given Line and other data such as shown in Appendices B, C and D. Output consists of files containing the intermediate lines and the final grid.

Input Data Files

The input file contains the Given Lines, reference angle, and cell spacing information. A typical input file has the structure shown in Table 6.1. Data are free formatted. West longitudes are negative. Variables in the second record are as follows:

LMAX	= total number of Given Lines,
LGEN	= number of the Given Line which is to be the Generator Line,
IFORM	= format index for points defining the lines (1=arbitrary units; 2=latitude,longitude in degrees and minutes),
REFLAT	= reference latitude (degrees) used to convert DEL to degrees (if needed), and
DEL	= cell spacing along the Generator Line (non-dimensional if IFORM is 1, meters if IFORM is 2).

Table 6.1. Structure of a typical input file.

Number of Records	Contents
1	Header Text
1	LMAX, LGEN, IFORM, DEL, REFLAT
1	MPOINT(1), NCELL(1), ITYPE(1)
	XG(1,1), YG(1,1), R(1,1)
MPOINT(1)	XG(2,1), YG(2,1), R(2,1)
	XG(3,1), YG(3,1), R(3,1)
	: : :
	: : :
1	MPOINT(2), NCELL(2), ITYPE(2)
	XG(1,2), YG(2,2), R(1,2)
MPOINT(2)	XG(2,2), YG(2,2), R(2,2)
	XG(3,2), YG(3,2), R(3,2)
	: : :
	: : :
1	MPOINT(3), NCELL(3), ITYPE(3)
	XG(1,3), YG(1,3), R(1,3)
MPOINT(3)	XG(2,3), YG(2,3), R(2,3)
	XG(3,3), YG(3,3), R(3,3)
	: : :
	: : :

For each set of Given Lines,

MPOINT	= number of points in the line,
NCELL	= number of cells to be created between the present line and the next Given Line, and
ITYPE	= type of spacing (1=uniform, 2=linearly increasing, 3=geometrically increasing).

For points defining a Given Line,

XG,YG	= coordinates of the point, and
R	= ratio of width of the first cell (i.e., the cell closest and to the right of the Given Line) to last cell (i.e., the cell just to the left of the next Given Line).

Sample input files appear in Appendices B, C, and D.

Output Data Files

The output files generated are shown in Table 6.2.

Table 6.2. Output files generated by *CGEN*.

Output File	Contents
cgen.out1	Intermediate cells (XI, YI)
cgen.out2	Final grid (XF, YF)
cgen.out1r	Intermediate cells (XI, YI) in latitude, longitude coordinates
cgen.out2r	Final grid (XF, YF) in latitude, longitude coordinates
cgen.out3	Cell sides of grid as it is being generated
cgen.out4	Widths and ratios for cell width matching
cgen.out5	Input file based on re-proportioned points on original input lines

The output files cgen.out1 and cgen.out1r contain the intermediate grid, while cgen.out2 and cgen.out2r contain the final grid. These files have the following structure.

Table 6.3. Structure of output files cgen.out1, cgen.out1r, cgen.out2, and cgen.out2r.

Number of Records	Contents
1	Header Text
1	IMAX, JMAX, IFORM, REFLAT
	X(I,J), Y(I,J), {I = 1, IMAX; J = 1}
JMAX	X(I,J), Y(I,J), {I = 1, IMAX; J = 2}
	: : :
	X(I,J), Y(I,J), {I = 1, IMAX; J = JMAX}

The output file cgen.out3 contains cell sides which are printed out as they are being created during the generation process. If for any reason the generation is halted, the data in this file can be plotted to show the grid generated up to that point. Each record has values for two points (X_1, Y_1, X_2, and Y_2); each pair of points defines adjacent corners of a cell.

The output file cgen.out4 contains cell widths at all points on each side of each Given Line and ratios for each side needed to match the widths. These files have the following structure.

Table 6.4. Structure of the output file cgen.out4.

Number of Records	Contents
1	L, L+1 (Cell Matching in Region between Line L and L+1)
MMAX	M, W_{left}, W_{right}, $W_{uniform}$, R_{left}, R_{right}, R_{left}, R_{right} : : : :
1	L, L+1 (Cell Matching in Region between Line L and L+1)
MMAX	M, W_{left}, W_{right}, $W_{uniform}$, R_{left}, R_{right} : : :

Here W_{left} is the width of the first cell, W_{right} is the width of the last cell, and $W_{uniform}$ is cell width if cells were uniform. The first value of R_{left} is the ratio for linearly-increasing widths that would match the first cell with the cell width to the region to the left, and the first value R_{right} is the ratio for linearly-increasing widths that would match the last cell with the cell width to the region to the right. The second set of ratios is for geometrically-increasing widths. For these last two, if there is no adjacent region, or R is negative, then all asterisks are printed.

6.3. EXECUTING THE PROGRAM

CGEN is set up to run interactively. The prompts are shown with typical responses.

```
ENTER INPUT FILE NAME :                                                  test12.dat
BEGIN INTERPOLATED LINES. ENTER 1 TO PRINT :                                      0
ENTER 1 TO EQUALLY REDISTRIBUTE POINTS ON GENERATOR LINE :                        0
ENTER 1 TO EQUALLY REDISTRIBUTE POINTS ON OTHER GIVEN LINES :                     0
ENTER JFIRST :                                                                    2
  JMAX= 13
ENTER JLAST (OR -1 to = JMAX) :                                                  -1
BEGIN CENTERLINE. ENTER 1 TO PRINT :                                              0
 TO PRINT CALCULATIONS AT KNOWN POINT I,J,
ENTER I,J FOR PRINT; 0,0 FOR NO PRINT;' -1,-1 TO PRINT ALL :                    0 0
ENTER I,J FOR SPLINE PRINT (OR 0,0 FOR NONE) :                                  0 0
ENTER 1 FOR METHOD A , 2 FOR METHOD B :
```

At the end of the run, the following lines appear:

```
RUN COMPLETE. OUTPUT IN FILE cgen.out2
TOTAL CELLS = 374
NUMBER OF NON-CONVERGENT CELLS = 0
```

At the end of the run, the total number of grid cells created, and the number at which orthogonality could not be achieved, is printed. If the number is greater than 0, further investigation must be made. Use cell print options (seventh input line) to get additional information.

7. SUMMARY AND CONCLUSIONS

The previous discussion has developed the principles of parametric lines and orthogonality and explained two methods for generating curvilinear orthogonal grids. Properties along the parametric line are determined using the cubic spline interpolation technique. Orthogonality was taken to hold at either the center of a line segment intersecting two other lines, or at the center of a cell. The first application, Method A, was found to be very reliable and always was able to generate a grid. The second, Method B, was less reliable (it sometimes constructed crossed orthogonals), but when it was successful the final grid was similar to that constructed by Method A.

The key feature of the present methodology of generating curvilinear orthogonal grids is that by using three or more Given Lines the orientation of cells deep within the domain can be controlled. Therefore, grids can be constructed that closely follow coastlines, channels, or barriers. Other generation methods usually allow only cell spacing and the locations of the four corners of the domain to be input; this results in very little control of cell orientation or spacing in the interior of the domain.

Although several methods of construction have been explored, by no means have all approaches to generating an orthogonal grid been explored here. The following are just a few of the possible modifications that can be examined in the future:

- Using finite differences for the terms involving the r-derivatives in Eqs. 2.6 and 2.7 (see Mellor, 1990).
- Allowing variable spacing of cells along the Generator Line.
- Including other interpolation schemes for spacing between Given Line pairs, not just uniform, linearly increasing/decreasing, or geometrically increasing/decreasing.
- Use of smoothing of second derivatives of XG, YG (based on local curvature) to obtain a more regular grid.
- Using another interpolation scheme such as the B-spline, which would allow more localized control of fit to the input points.
- Specifying not just the non-intersecting Given Lines, but specifying one or more of the orthogonal lines.
- Basing a scheme on orthogonality at cell corners (i.e., line intersections). For example, in Figure 2.2, if

$$\Delta_{ab} = [(x_a - x_b)^2 + (y_a - y_b)^2]^{1/2}, \tag{7.1}$$

then orthogonality(Eq. 2.5) at Point B can be approximated numerically as

$$\left(\frac{x_b - x_a}{\Delta_{ba}} + \frac{x_c - x_b}{\Delta_{cb}}\right)\left(\frac{x_e - x_b}{\Delta_{eb}} + \frac{x_b - x_d}{\Delta_{bd}}\right) + \left(\frac{y_b - y_a}{\Delta_{ba}} + \frac{y_c - y_b}{\Delta_{cb}}\right)\left(\frac{y_e - y_b}{\Delta_{eb}} + \frac{y_b - y_d}{\Delta_{bd}}\right) = 0 \qquad (7.2)$$

- Developing a composite grid by combining a number of smaller, orthogonal grids generated within four-sided regions by attaching them along common sides.

REFERENCES

Blumberg, A. F., and H. J. Herring, 1987: Circulation modeling using orthogonal curvilinear coordinates. in **Three-Dimensional Models of Marine and Estuarine Dynamics** (J. C. J. Nihoul and B. M. Jamart, eds), Elsevier Oceanography Series, 45, 55 - 88.

Blumberg, A. F., and G. L. Mellor, 1987: A description of a three-dimensional coastal ocean circulation model. **Three-Dimensional Coastal Ocean Models**, (ed. Heaps), American Geophysical Union, Washington, DC. 1 - 16.

Mellor, G. L., 1990: User's Guide for a three-dimensional, primitive equation, numerical ocean model. Princeton University Report. 51 pp.

National Ocean Service, 1990: Tampa Bay Oceanography Project: Plan for FY 1990 to FY 1992. NOAA, National Ocean Service, Office of Oceanography and Marine Assessment, Rockville, MD. 68 pp.

Thompson, J. F, Z. U. A. Warsi, and C. W. Mastin, 1985: **Numerical Grid Generation: Foundations and Applications**. North-Holland, New York, NY. 483 pp.

Zervas, C. Z., 1993: Tampa Bay Oceanography Project: Physical Oceanographic Synthesis. NOS Technical Report NOS OES 002. NOAA, National Ocean Service, Office of Ocean and Earth Sciences, Silver Spring, MD. 184 pp.

APPENDIX A. INTERPOLATION BY CUBIC SPLINE

Given the four data Points A, B, C and D (Figure A.1), a line in (x,y) space can be defined along the coordinate direction r, through the points. The line, x(r),y(r), is defined by the N points and a cubic spline interpolation scheme (Thompson et al., 1985).

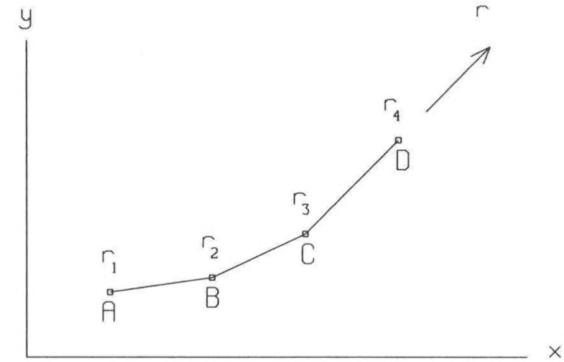


Figure A.1. A line segment in x,y space along direction r and passing through Points A, B, C, and D.

A.1. INTERPOLATING RELATIONSHIPS

First, the values of r_i are somewhat arbitrary and are calculated as follows. Setting

$$r_1 = 0 \tag{A.1}$$

allows the value of r to be defined as the scaled distance along the line; i.e.,

$$r_i = S_f \sum_{j=2}^{i} \left((x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 \right)^{1/2}, \tag{A.2}$$

where S_f is an arbitrary scale factor (here $S_f = 1$) and the integration is carried out numerically. For the cubic spline interpolation, the second derivative of x or y with respect to r is assumed to be unknown and to vary linearly between defining data points. Thus for $r_i \le r \le r_{i+1}$,

$$\frac{d^2x}{dr^2} = \frac{(r_{i+1} - r)x_i^{//} + (r - r_i)x_{i+1}^{//}}{r_{i+1} - r_i}, \tag{A.3}$$

where x''_i and x''_{i+1} are the second derivatives of x with respect to r at points i and i+1, respectively. After integrating Eq. A.3 twice and using the end conditions $x(r_i) = x_i$ and $x(r_{i+1}) = x_{i+1}$, the process yields

$$x(r) = \frac{(r_{i+1} - r)^3 x_i^{//} + (r - r_i)^3 x_{i+1}^{//}}{6(r_{i+1} - r_i)} + \left(\frac{x_i}{r_{i+1} - r_i} - \frac{r_{i+1} - r_i}{6} x_i^{//} \right)(r_{i+1} - r)$$
$$+ \left(\frac{x_{i+1}}{r_{i+1} - r_i} - \frac{r_{i+1} - r_i}{6} x_{i+1}^{//} \right)(r - r_i). \tag{A.4}$$

The tangential derivative of x at any point is then found by differentiating Eq. A.4 to give

$$\frac{dx}{dr} = \frac{(r_{i+1} - r)^2 x_i^{\prime\prime} + (r - r_i)^2 x_{i+1}^{\prime\prime}}{2(r_{i+1} - r_i)} - \left(\frac{x_i}{r_{i+1} - r_i} - \frac{r_{i+1} - r_i}{6} x_i^{\prime\prime} \right)$$
$$+ \left(\frac{x_{i+1}}{r_{i+1} - r_i} - \frac{r_{i+1} - r_i}{6} x_{i+1}^{\prime\prime} \right) . \tag{A.5}$$

By equating the tangential derivative at the right end of interval (i to i+1) to the derivative at the left end of the adjacent interval (i+1 to i+2), it can be shown that

$$(r_i - r_{i-1}) x_{i-1}^{\prime\prime} + 2(r_{i+1} - r_{i-1}) x_i^{\prime\prime} + (r_{i+1} - r_i) x_{i+1}^{\prime\prime} = 6 \left(\frac{x_{i+1} - x_i}{r_{i+1} - r_i} - \frac{x_i - x_{i-1}}{r_i - r_{i-1}} \right) , \tag{A.6}$$

which is the recursive relationship for computing the unknown second derivatives x''_i. At the end points (i = 1 and i = N), it is assumed that x''_i is zero:

$$x_i^{\prime\prime} = x_N^{\prime\prime} = 0 . \tag{A.7}$$

The development is similar for y(r).

A.2. PRACTICAL CONSIDERATIONS

Input to the spline function includes the arrays {x} and {y}; the number of points in each array, N; test values of X, Y, I, and R; and an index to select one of the previous four inputs, IGIV. The output consists of revised values of X, Y, I, and/or R, depending on IGIV, and values of dx/dr and dy/dr at the point. The program begins by establishing values for r_i using Eq. A.2 and then computes the second derivatives from Eq. A.6 and an equivalent expression for y.

The most frequent use of the spline function is the determination of one coordinate (e.g., Y) when the other (e.g., X) is known. Generally speaking, there will be four situations encountered. A discussion of each follows.

Normal Case

In the normal case, given the coordinate X, the spline program loops through each pair of points $\{x_i, y_i\}$ to find a value of j such that:

$$x_j \leq X < x_{j+1} . \tag{A.8}$$

For example, in Figure A.2, suppose that X_1 is given. Then Eq. A.4 is solved iteratively to find the precise value of R which yields X_1, then Y_1 is found by inserting this value of R into the equivalent parametric expression for y(r). The determination of Y_1 in this case is straightforward because there

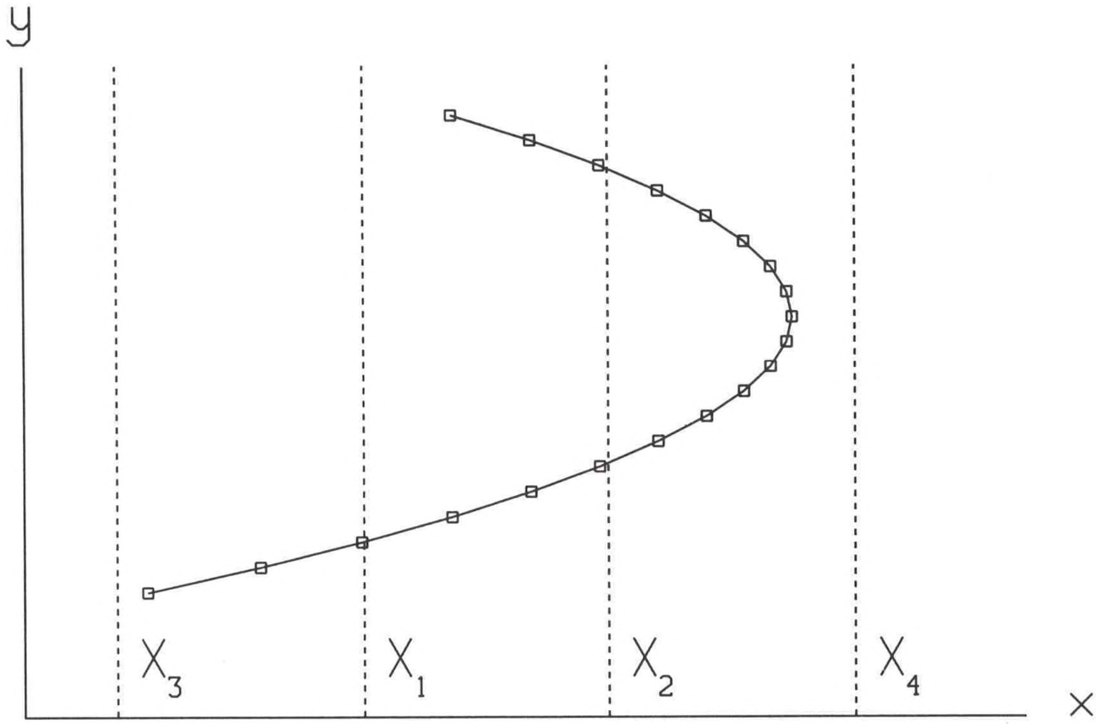


Figure A.2. A line in x,y-space determined by several points, and four conditions: the normal case, X_1; the case of multiple values, X_2; the case of line extension beyond endpoints, X_3; and the case of no interval, X_4.

is only one value of y along the line corresponding to X_1 (also, since $|dx/dr|$ is greater than $|dy/dr|$ here, the determination of Y from X should be more accurate than the determination of X from Y).

Multiple Values

The first kind of problem can arise when, for example, given X, there are multiple values of Y (i.e., two intervals containing X). For example, in Figure A.2, given X_2, there are two possible values of y. To avoid this error, an estimate of Y is required as input to the spline routine; then the x which has a y-value closest to the estimate is chosen as the correct output value.

Line Extension

A second problem can arise when there are no intervals in which X lies, but X is close to either end point. For example, X_3 (Figure A.2) has no corresponding Y value. Two estimates of y are determined by line extension. The first is made by assuming that the line determined by the points extends beyond the last point and with a slope equal to that in the last interval and having r values proportionate to the scaled distance from the last point (R will be positive and will be greater than r_N). The second is made by assuming that the line extends beyond the first point with a slope equal to that in the first interval and with r values proportionate to the scaled distance from the first point (R will be negative). The correct point is taken to be the one closer to the given values of X and Y.

No Interval

The third problem arises when there is no interval containing X (e.g., X_4 in Figure A.2). The spline routine will compute the distances to each end point (D1 and D2) and compare those with the distance to the middle point, D3. If D3 is smaller than either D1 or D2, the routine changes the given input from X to Y and continues. Subsequently, a solution can usually be found.

APPENDIX B. SAMPLE INPUT DATA FILE WITH UNIFORM CELL WIDTHS

The following is the data file used to create Figures 5.1a and 5.1b. The first line contains only text.

```
TEST DATA SET FOR EVALUATION
 3   2   1  0.5    27.6     LMAX, LGEN, FORM(2=DEG,MIN), DEL(METERS), RLAT
13      10     1          mpoint,  ncell, itype(1=uniform, 2=linear, 3=geometric)
 1.000  1.0  1.00
 1.000  2.0  1.00
 1.000  2.5  1.00
 1.000  3.0  1.00
 1.000  3.5  1.00
 1.000  4.0  1.00
 1.000  5.0  1.00
 1.000  6.0  1.00
 1.000  6.5  1.00
 1.000  7.0  1.00
 1.000  7.5  1.00
 1.000  8.0  1.00
 1.000  9.0  1.00
13      13     1                 mpoint,  ncell, itype
 3.0  1.0    1.00
 3.71 2.0    1.00
 3.92 2.5    1.00
 4.0  3.0    1.00
 3.92 3.5    1.00
 3.71 4.0    1.00
 3.0  5.0    1.00
 2.29 6.0    1.00
 2.08 6.5    1.00
 2.0  7.0    1.00
 2.08 7.5    1.00
 2.29 8.0    1.00
 3.0  9.0    1.00
 13      0     1                 mpoint, ncell (not used), itype (not used)
 5.0  1.0    1.00
 5.0  2.0    1.00
 5.0  2.5    1.00
 5.0  3.0    1.00
 5.0  3.5    1.00
 5.0  4.0    1.00
 5.0  5.0    1.00
 5.0  6.0    1.00
 5.0  6.5    1.00
 5.0  7.0    1.00
 5.0  7.5    1.00
 5.0  8.0    1.00
 5.0  9.0    1.00
```

APPENDIX C. SAMPLE INPUT DATA FILE WITH VARIABLE CELL WIDTHS

The following is the data file used to create Figures 5.1c and 5.2a. The first line contains only text.

```
 TEST DATA SET FOR EVALUATION
  3   2   1  0.5    27.6     LMAX, LGEN, FORM(2=DEG,MIN), DEL(METERS), RLAT
 13      10     2           mpoint,  ncell, itype(1=uniform, 2=linear, 3=geometric)
  1.000  1.0  1.60
  1.000  2.0  4.46
  1.000  2.5  6.03
  1.000  3.0  6.8
  1.000  3.5  6.03
  1.000  4.0  4.46
  1.000  5.0  1.60
  1.000  6.0  1.
  1.000  6.5  1.
  1.000  7.0  1.
  1.000  7.5  1.
  1.000  8.0  1.
  1.000  9.0  1.
 13      13     2                 mpoint,  ncell, itype
  3.0  1.0     1.00
  3.71 2.0     1.00
  3.92 2.5     1.00
  4.0  3.0     1.00
  3.92 3.5     1.00
  3.71 4.0     1.00
  3.0  5.0     1.00
  2.29 6.0     0.45
  2.08 6.5     0.32
  2.0  7.0     0.28
  2.08 7.5     0.32
  2.29 8.0     0.45
  3.0  9.0     1.86
  13      0     1            mpoint,  ncell (not used), itype (not used)
  5.0  1.0     1.00
  5.0  2.0     1.00
  5.0  2.5     1.00
  5.0  3.0     1.00
  5.0  3.5     1.00
  5.0  4.0     1.00
  5.0  5.0     1.00
  5.0  6.0     1.00
  5.0  6.5     1.00
  5.0  7.0     1.00
  5.0  7.5     1.00
  5.0  8.0     1.00
  5.0  9.0     1.00
```

APPENDIX D. INPUT DATA FILE FOR TAMPA BAY GRID

The following is the input data file for the Tampa Bay grid. Here the number of Boundary Lines (LMAX) is 3, line No. 2 is the Generator line (LGEN), the format of the data points is 2 (IFORM) {i.e., latitude and longitude (west is positive) in degrees and minutes}, the grid interval along the Generator Line (DEL) is 1000 meters, and the reference latitude (RLAT) is 27.6 degrees (used to convert DEL into degrees).

For each Boundary Line there is the number of points (mpoint), the ratio of the first cell to the last (ratio), the number of cells between it and the next Boundary Line (ncell), and the points.

```
         FILE=top09.dat
   6   4   2 1000.00    27.60 LMAX, LGEN, IFORM, DEL, RLAT
         9        4        2          mpoint, ncell, itype
  -83.   0.4985  27. 38.8403     1.0000
  -82. 51.9164  27. 41.1497     1.5000
  -82. 46.7253  27. 42.7249     1.5000
  -82. 42.3441  27. 44.1719     2.0000
  -82. 38.4631  27. 46.7759     2.0000
  -82. 38.1065  27. 49.7035     4.0000
  -82. 39.9646  27. 52.9571     3.0000
  -82. 44.5161  27. 57.1557     2.3000
  -82. 49.2000  28.   3.5392     1.0000
         9        6        2          mpoint, ncell, itype
  -82. 59.9149  27. 36.8909     1.0000
  -82. 51.3354  27. 38.1413     1.0000
  -82. 45.0943  27. 38.5638     3.0000
  -82. 40.4320  27. 41.5647     1.5000
  -82. 35.9628  27. 45.5129     1.0000
  -82. 35.0949  27. 50.2409     1.0000
  -82. 37.7399  27. 54.5325     1.0000
  -82. 41.8117  27. 59.0854     1.0000
  -82. 46.4877  28.   5.1313     1.0000
         9        8        2          mpoint, ncell, itype
  -82. 59.5125  27. 35.1569     1.0000
  -82. 51.3194  27. 36.0283     1.0000
  -82. 44.2094  27. 36.3316     1.0000
  -82. 38.8431  27. 40.1239     1.5000
  -82. 34.1377  27. 44.7036     1.5500
  -82. 33.4991  27. 50.4593      .5000
  -82. 35.0912  27. 56.0049      .6600
  -82. 39.1937  28.    .8904      .5000
  -82. 43.8840  28.   6.7280     1.0000
         9       10        2          mpoint, ncell, itype
  -82. 58.0000  27. 30.0000      .5000
  -82. 50.7390  27. 32.4042      .5000
  -82. 43.3676  27. 34.5266      .5057
  -82. 37.1562  27. 38.6346      .6980
  -82. 32.1199  27. 43.8519      .7991
  -82. 30.3589  27. 50.5293     1.7928
  -82. 31.1092  27. 57.3470     1.7995
  -82. 34.0851  28.   3.6655     1.0017
  -82. 37.0001  28.   9.9998      .5000
         9       11        2          mpoint, ncell, itype
  -82. 56.0312  27. 25.5011     1.0000
  -82. 49.0535  27. 27.7652     1.0000
  -82. 40.9428  27. 30.8659     1.0000
  -82. 34.8862  27. 36.7307      .2000
```

```
-82. 29.1994  27. 42.4693      .3500
-82. 26.9531  27. 50.2924      .5000
-82. 26.6904  27. 58.0465      .5000
-82. 28.2307  28.  5.4282      .2500
-82. 30.0380  28. 11.9744     1.0000
     9          0       2           mpoint, ncell, itype
-82. 53.2169  27. 19.1936     1.0000
-82. 46.5161  27. 21.6447     1.0000
-82. 37.1388  27. 25.0440     1.0000
-82. 27.3482  27. 31.1698     1.0000
-82. 23.3199  27. 40.5418     1.0000
-82. 22.2185  27. 50.0933     1.0000
-82. 22.7504  27. 58.2605     1.0000
-82. 23.0695  28.  6.2109     1.0000
-82. 24.5700  28. 13.0277     1.0000
```