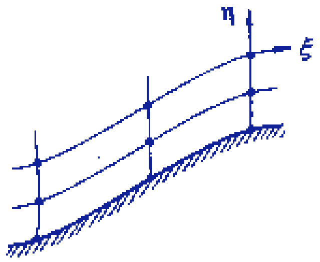
I. INTRODUCTION

The numerical solution of partial differential equations requires some discretization of the field into a collection of points or elemental volumes (cells). The differential equations are approximated by a set of algebraic equations on this collection, and this system of algebraic equations is then solved to produce a set of discrete values which approximates the solution of the partial differential system over the field. The discretization of the field requires some organization for the solution thereon to be efficient, i.e., it must be possible to readily identify the points or cells neighboring the computation site. Furthermore, the discretization must conform to the boundaries of the region in such a way that boundary conditions can be accurately represented. This organization is provided by a coordinate system, and the need for alignment with the boundary is reflected in the routine choice of cartesian coordinates for rectangular regions, cylindrical coordinates for circular regions, etc., to the extent of the handbook's resources.



The current interest in numerically-generated, boundary-conforming coordinate systems arises from this need for organization of the discretization of the field for general regions, i.e., to provide computationally for arbitrary regions what is available in the handbook for simple regions. The curvilinear coordinate system covers the field and has coordinate lines (surfaces) coincident with all boundaries. The distribution of lines should be smooth, with concentration in regions of strong solution variation, and the system should ultimately be capable of sensing these variations and dynamically adjusting itself to resolve them.

A numerically-generated grid is understood here to be the organized set of points formed by the intersections of the lines of a boundary-conforming curvilinear coordinate system. The cardinal feature of such a system is that some coordinate line (surface in 3D) is coincident with each segment of the boundary of the physical region. The use of coordinate line intersections to define the grid points provides an organizational structure which allows all computation to be done on a fixed square grid when the partial differential equations of interest have been transformed so that the curvilinear coordinates replace the cartesian coordinates as the independent variables.

This grid frees the computational simulation from restriction to certain boundary shapes and allows general codes to be written in which the boundary shape is specified simply by input. The boundaries may also be in motion, either as specified externally or in response to the developing physical solution. Similarly, the coordinate system may adjust to follow variations developing in the evolving physical solution. In any case, the numerically-generated grid allows all computation to be done on a fixed square grid in the computational field which is always rectangular by construction.

In the sections which follow, various configurations for the curvilinear coordinate system are discussed in Chapter II. In general, the computational field will be made rectangular, or composed of rectangular sub-regions, and a wide variety of configurations is possible. Coordinate systems may also be generated separately for sub-regions in the physical plane and patched together to form a complete system for complex configurations. The basic transformation relations applicable to the use of general curvilinear coordinate systems are developed in Chapter III; the construction of numerical solutions of partial differential equations on those systems is discussed in Chapter IV; and consideration is given in Chapter V to the evaluation and control of truncation error in the numerical representations.

Basically, the procedures for the generation of curvilinear coordinate systems are of two general types: (1) numerical solution of partial differential equations and (2) construction by algebraic interpolation. In the former, the partial differential system may be elliptic (Chapter VI), parabolic or hyperbolic (Chapter VII). Included in the elliptic systems are both the conformal (Chapter X), and the quasi-conformal mappings, the former being orthogonal. Orthogonal systems (Chapter IX) do not have to be conformal, and may be generated from hyperbolic systems as well as from elliptic systems. Some procedures designed to produce coordinates that are nearly orthogonal are also discussed. The algebraic procedures, discussed in Chapter VIII, include simple normalization of boundary curves, transfinite interpolation from boundary surfaces, the use of intermediate interpolating surfaces, and various other related techniques.

Coordinate systems that are orthogonal, or at least nearly orthogonal near the boundary, make the application of boundary conditions more straightforward. Although strict orthogonality is not necessary, and conditions involving normal derivatives can certainly be represented by difference expressions that combine one-sided differences along the line emerging from the boundary with central expressions along the boundary, the accuracy deteriorates if the departure from orthogonality is too large. It may also be more desirable in some cases not to involve adjacent boundary points strongly in the representation, e.g., on extrapolation boundaries. The implementation of algebraic turbulence models is more reliable with near-orthogonality at the boundary, since information on local boundary normals is usually required in such models. The formulation of boundary-layer equations is also much more straightforward and unambiguous in such systems. Similarly, algorithms based on the parabolic Navier-Stokes equations require that coordinate lines approximate the flow streamlines, and the lines normal thereto, especially near solid boundaries. It is thus better in general, other considerations being equal, for coordinate lines to be nearly normal to boundaries.

Finally, dynamically-adaptive grids are discussed in Chapter XI. These grids continually adapt during the course of the solution in order to follow developing gradients in the physical solution. This topic is at the frontier of numerical grid generation and may well prove to be one of its most important aspects.

The emphasis throughout is on grids formed by the intersections of coordinate lines of a curvilinear coordinate system, as opposed to the covering of a field with triangular elements or a random distribution of points. Neither of these latter collections of points is suitable for really efficient numerical solutions (although numerical representations can be constructed on each, of course) because of the cumbersome process of identification of neighbors of a point and the lack of banded structure in the matrices. Thus the subject of triangular mesh generators, per se, is not addressed here. (Obviously a triangular mesh can be produced by construction rectangular mesh diagonals.)

Considerable progress is being made toward the development of the techniques of numerical grid generation and toward casting them in forms that can be readily applied. A comprehensive survey of numerical grid generation procedures and applications thereof through 1981 was given by Thompson, Warsi, and Mastin in Ref. [1], and the conference proceedings published as Ref. [2] contains a number of expository papers on the area, as well as current results. Other collections of papers on the area have also appeared (Ref. [3] and [4]), and a later review through 1983 has been given by Thompson in Ref. [5]. Some other earlier surveys are noted in Ref. [1]. A later survey by Eiseman is given in Ref. [37]. The present text is meant to be a developmental treatment of the techniques of grid generation and its applications, not a survey of results, and therefore no attempt is made here to cite all related references, rather only those needed to illustrate particular points are noted. The surveys mentioned above should be consulted directly for references to examples of various applications and related contributions. (Ref [l] gives a short historical development of the ideas of grid generation.) Other surveys of particular areas of grid generation are cited later as topics are introduced.

Finally, in regard to implementation, a configuration for the transformed (computational) field is first established as discussed in Chapter II. The grid is generated from a generation system constructed as discussed in Chapters VI -- X. (If the grid is to be adaptive, i.e., coupled with the physical solution done thereon, then the gr1d must be continually updated as discussed in Chapter XI.) In the construction of the grid, due account must be taken of the truncation error induced by the grid discussed in Chapter V. The partial differential equations of the physical problem of interest are transformed according to the relations given in Chapter III. These transformed equations are then discretized, cf. Chapter IV, and the resulting set of algebraic equations is solved on the fixed square grid in the rectangular transformed field.